International Journal of Informatics Society

06/22 Vol. 14 No.1 ISSN 1883-4566

Informatics Society

Editor-in-Chief: Hiroshi Inamura, Future University Hakodate
Associate Editors: Katsuhiko Kaji, Aichi Institute of Technology
Yoshia Saito, Iwate Prefectural University
Takuya Yoshihiro, Wakayama University
Tomoki Yoshihisa, Osaka University

Editorial Board

Hitoshi Aida, The University of Tokyo (Japan)
Huifang Chen, Zhejiang University (P.R.China)
Christian Damsgaard Jensen, Technical University of Denmark (Denmark)
Teruo Higashino, Kyoto Tachibana University (Japan)
Tadanori Mizuno, Aichi Institute of Technology (Japan)
Jun Munemori, The Open University of Japan (Japan)
Yuko Murayama, Tsuda University (Japan)
Ken-ichi Okada, Keio University (Japan)
Norio Shiratori, Chuo University / Tohoku University (Japan)
Ian Wakeman, University of Sussex (UK)
Ismail Guvenc, North Carolina State University (USA)
Qing-An Zeng, North Carolina A&T State University (USA)
Tim Ziemer, University of Bremen (Germany)
Justin Zhan, University of Cincinnaty Computer Science Faculty (USA)
Xuyun Zhang, Macquarie University (Australia)

Aims and Scope

The purpose of this journal is to provide an open forum to publish high quality research papers in the areas of informatics and related fields to promote the exchange of research ideas, experiences and results.

Informatics is the systematic study of Information and the application of research methods to study Information systems and services. It deals primarily with human aspects of information, such as its qu ality and value as a resource. Informatics also referred to as Information science, studies t he structure, algorithms, behavior, and interactions of natural and a rtificial systems that store, process, access and communicate information. It also develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields. The advent of computers, its ubiquity and ease to use has led to th e study of info rmatics that has computational, cognitive and social aspects, including study of the social impact of information technologies.

The characteristic of informatics' context is amalgamation of technologies. For creating an informatics product, it is necessary to integrate many technologies, such as mathematics, linguistics, engineering and other emerging new fields.

Guest Editor's Message

Michiko Oba

Guest Editor of the Fortieth Issue of the International Journal of Informatics Society

We are delighted to have the Fortieth issue of the International Journal of Informatics Society (IJIS) published. This issue includes selected papers from the Fifteenth International Workshop on Informatics (IWIN2021), which was held online, Sept. 12-13, 2021. The workshop was the fifteenth event for the Informatics Society, and was intended to bring together researchers and practitioners to share and exchange their experiences, discuss challenges and present original ideas in all aspects of informatics and computer networks. In the workshop, 29 papers were presented in seven technical sessions. The workshop was successfully finished with precious experiences provided to the participants. It highlighted the latest research results in the area of informatics and its applications that include networking, mobile ubiquitous systems, data analytics, business systems, education systems, design methodology, intelligent systems, groupware, and social systems.

Each paper submitted to IWIN2021 was reviewed in terms of technical content, scientific rigor, novelty, originality, and quality of presentation by at least two reviewers. Through those reviews, 19 papers were selected for publication candidates of IJIS Journal, and they were further reviewed as a Journal paper. We have three categories of IJIS papers, Regular papers, Industrial papers, and Invited papers, each of which was reviewed from different points of view. This volume includes papers among those accepted papers, which have been improved through the workshop discussion and the reviewers' comments.

We publish the journal in print as well as in an electronic form over the Internet. We hope that the issue would be of interest to many researchers as well as engineers and practitioners over the world.

Michiko Oba received her B.S. in physics from Japan Women's University in 1982, and Ph.D. in engineering from Osaka University, Japan in 2001. She worked in the Systems Development Laboratory and the Software Division of Hitachi Ltd from 1982. Presently, she is a professor in the Department of Media Architecture, Future University Hakodate, Japan. She is a Council Member of the Science Council of Japan and a fellow of the Information Processing Society of Japan (IPSJ). She is a member of IEEE Computer Society, IPSJ, and the Institute of Electrical Engineers of Japan (IEEJ).

ISSN1883-4566 © 2022 - Informatics Society and the authors. All rights reserved.

Regular Paper

Contour Detection Method by CycleGAN Using CG Images as Ground Truth

Tsukasa Kudo†

†Faculty of Informatics, Shizuoka Institute of Science and Technology, Japan
kudo.tsukasa@sist.ac.jp

Abstract - Recently, various studies on object recognition have been conducted, with remarkable development, especially using deep learning. Here when the target objects are relatively small and arranged in high density in the image, its contour detection is often required as a preprocessing. Several methods have also been proposed for contour detection using deep learning, including generative adversarial networks (GANs) to perform pixel-by-pixel contour detection. However, it is necessary to prepare ground truth images that have a one-to-one correspondence with the target images for these methods. Thus, their application is often challenging. Cycle-consistent adversarial networks (CycleGAN) translates images between two domains and can be trained without preparing one-to-one ground truth images corresponding to the target images. Therefore, this study proposes a contour detection method through CycleGAN with efficiently generated ground truth images using computer graphics. Furthermore, through the comparative evaluations with the case of using ground truth images with a one-to-one correspondence with the target images, it is shown that the accuracy of the proposed method can be achieved close to this case.

Keywords: CycleGAN, Contour detection, Image recognition, Computer graphics, CG, Image-to-image translation

1 INTRODUCTION

Recently, object recognition in images has been used in various fields, and necessary information can be automatically extracted. When the target object is relatively small in the image, e.g., as in the face recognition, the object is first detected as a bounding box; then, the object recognition is performed for this area [26]. However, when objects are arranged in high density, a more precise method, such as contour detection, is often required.

As a research field, the author has been dealing with inventory management of parts in machine assembly factories stored in bulk containers (hereinafter, container). Since a factory often has thousands or more containers, efficient data collection and automatic information extraction are required. Thus, the author has been studying how workers wear wearable cameras and automatically collect images of containers. However, since the containers are densely arranged, contour detection is required to determine the target container.

Several methods using deep learning (DL) are proposed for contour detection [15], [27]. It has been shown that the target area or contour can be detected in a pixel-by-pixel manner using these methods. However, since these methods require a one-to-one correspondence with the original and its ground truth images as training data, it is often challenging to actual application.

Meanwhile, Cycle-consistent adversarial networks (CycleGAN) have been proposed as a kind of generative adversarial networks (GAN) [5], [31]. Moreover, it has been shown that mutual image translation between two types of images can be performed using CycleGAN, such as between horses and zebras. Although both images must be prepared for the training data, it is not necessary to make the above-mentioned correspondence of each data in both images. This suggests that the training data of CycleGAN for translating an image, including target objects into one with detected contours of the objects, can be prepared efficiently.

Therefore, this study demonstrates the feasibility of the method using the ground truth images of the contour efficiently created by computer graphics (CG) for training the CycleGAN model. For this purpose, two types of ground truth images are prepared by targeting photos of the books arranged on the bookshelf. One is the above CG images for the proposed method, and another is the contour images being a one-to-one correspondence with the targeting photos. Furthermore, the comparative evaluations of accuracy between both cases are performed. Consequently, It is shown that the accuracy of the former using the proposed method, can be close to the latter case.

The remainder of this paper is organized as follows. Section 2 describes this study's motivation and related works. Section 3 presents the proposed contour detection method using CG images. Section 4 shows the implementation of the experimental system. Section 5 presents the evaluations of the proposed method. Section 6 discusses the evaluation results. Finally, Sec. 7 presents the conclution.

2 MIOTIVATION AND RELATED WORKS

2.1 Motivation of This Study

Currently, a huge amount of data is input as big data from various sensors with the progress of the Internet of things (IoT). Additionally, several attempts have been made to automatically extract useful information from these data by using DL to achieve high discrimination accuracy [16], [25]. For example, to monitor targets by images, a large number of cameras, such as surveillance, in-vehicle, river, and wearable cameras, have been used, and necessary information is automatically extracted from several videos and images.

Regarding such IoT applications, the author has been study-

ISSN1883-4566 © 2022 - Informatics Society and the authors. All rights reserved.

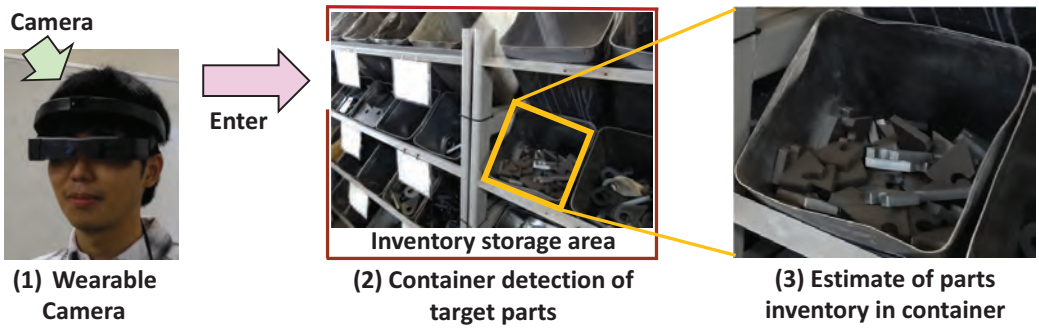


Figure 1: Automatic inventory management using videos

ing automated inventory management in machine assembly factories. Various parts are stored in containers (Fig. 1 (3)), and they cannot be counted visually from the outside. The number of these containers often reaches several thousand. For this issue, this study focused on the fact that inventory fluctuates only with inventory operations, such as the worker's replenishing or shipping of parts. Additionally, this study extracts the necessary inventory information automatically from the videos shot through the wearable camera worn by the worker (Fig. 1 (1)). Workers can take the video of the target objects without extra load by shooting video continuously while working.

Furthermore, this study showed that some necessary information for inventory management could be extracted automatically by applying DL to these videos in the previous study, such as determining workers' positions and estimating inventory of parts in containers [9], [10]. When the target is relatively small in the image, object detection is required before object recognition. Additionally, when the target is moving against the background, such as the worker picking up parts during the work, it could be detectd using the optical flow [11]. However, each container does not move against its background, also containers are densely arranged, as shown in Fig. 1 (2). Thus, container detection remains an issue.

The motivation for this study is to develop a method to efficiently detect each object's area from such dense objects including the preparation of training data for DL.

2.2 Related Works

To detect objects, as shown in Fig. 1 (2), some methods have been proposed in the conventional image processing fields. In contour tracking, the target's contour is detected to identify its area. In edge detection, the boundary of the target is specified using a filter or the like. In segmentation, mean-shift clustering, pixel value, or texture gradient determines the target area [28]. However, these contour detection methods using image processing are challenging when contours are incomplete or not closed [4].

With the progress of DL, several studies on object detection have been actively conducted, and various methods have been proposed. One is based on the region proposal that detects the bounding boxes where the objects exist. You Only Look Once (YOLO) have achieved high efficiency by detecting images using Convolutional Neural Network (CNN) processing only once; Single Shot Detector (SSD) has been used to detect objects of various sizes with one processing, and RatinaNet improved its efficiency [13]–[15], [17], [18]. Furthermore, some methods have been demonstrated for dense objects to detect the individual object's region, such as detecting new objects repeatedly (IterDet) and separating duplicate regions by post-processing [1], [19]. However, the exact area of targets cannot be specified since these region proposals use bounding boxes, namely rectangles.

Several methods have been proposed to detect object regions or contours in a pixel-by-pixel manner. A basic method has been proposed where each pixel examines an object's contour using CNN [22]. Semantic and instance segmentation methods have been proposed based on the region proposal, and they are segmentation for each object's class and object itself [7], [30]. Furthermore, after the method of image-to-image translation using conditional GANs (cGAN) between original and feature images was shown as pix2pix [8], methods using GAN have been investigated actively [12], [23], [27], [29]. However, since these methods require a one-to-one correspondence with the original and ground truth images as training data, their practical applications are often challenging.

CycleGAN has been proposed as a GAN for image-to-image translation. It can be used to translate images in a domain into the ones in another domain mutually. The previous study showed examples of mutual translation such as between photos and painter's drawings, and summer and winter photos [31]. Regarding the above challenges for practical use, CycleGAN has an essential feature that it is not necessary to make one-to-one correspondence with images between two domains as training data, i.e., it is easier to prepare training data.

Various applications of CycleGAN have been proposed. The first is the augmentation of training data used in the fields where it is difficult to generate sufficient training data for DL, such as medical treatment and detection of plant lesions [21], [24]. The second is to translate the original image into an image that is easier to detect objects as a preprocessing; methods that combine YOLO and RetinaNet have been proposed [3], [20]. However, the study targeting contour detection of densely arranged objects could not be found.

This study aims to develop a method to efficiently detect the contours of densely arranged objects, as shown in Fig. 1

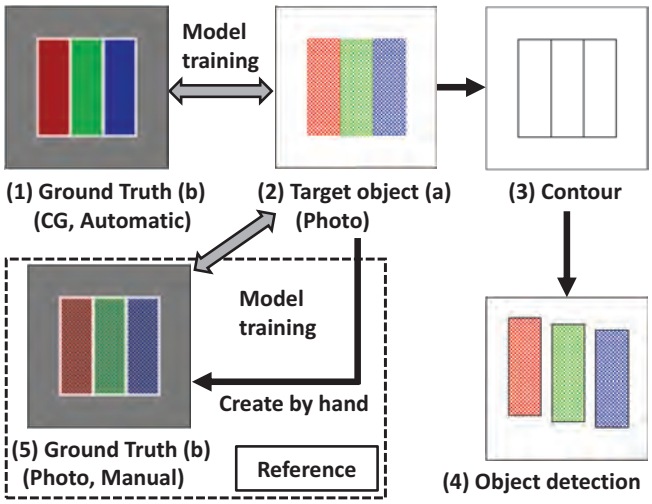


Figure 2: Contour detection method by CycleGAN using CG images as ground truth

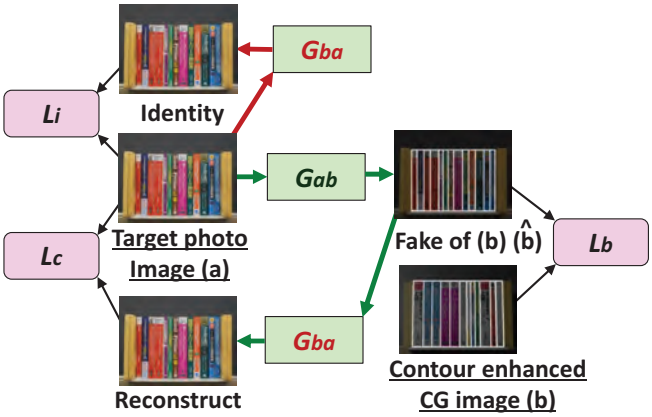


Figure 3: Configuration of CycleGAN model in this study

(2), by utilizing the above advantage of CycleGAN. Here, it is assumed that these contours are used for object recognition. Thus, this study employs ground truth images generated automatically using CG and not paired with the target images. The comparative evaluations of the following CycleGAN models are performed. One model is trained with the above CG images. Another model is trained with ground truth images that have a one-to-one correspondence with the target images. In other words, the goal of this study is to show that the former model can achieve accuracy close to the latter model.

3 CONTOUR DETECTION METHOD USING CG IMAGES

Figure 2 shows the proposed contour detection method from an image, including plural dense objects using CycleGAN with CG images as the ground truth contours. As shown in this figure, (1) shows the ground truth contour CG image, indicated by b below; (2) is the target photo, indicated by a below. The CycleGAN model is trained by images of the domains a and b. As mentioned in Sec. 2, since there is no need for a one-to-one correspondence between the ground truth and target images in CycleGAN training, b can be generated automatically using the CG tool.

Figure 3 shows the configuration of the CycleGAN model used in this study. To translate between a and b mutually, the CycleGAN model has two generators shown below.

$$\hat{a} = G_{ba}(b) \tag{1}$$
$$\hat{b} = G_{ab}(a) \tag{2}$$

Here, $\hat{a}$ and $\hat{b}$ are the fake images of the image a and b, respectively. For image a, the model is trained using discriminators that monitor the following three losses.

$$L_b = \| G_{ab}(a) - b \| \tag{3}$$
$$L_c = \| G_{ba}(G_{ab}(a)) - a \| \tag{4}$$
$$L_i = \| G_{ba}(a) - a \| \tag{5}$$

Here, $\|\|$ indicates an error, and similar losses are monitored for b. L_b evaluates the error between the fake image $\hat{b}$ and b; L_c evaluates the reconstruction image generated by applying these two generators sequentially, namely, between the fake image $\hat{a}$ and a; L_i evaluates the identity image generated from itself through the generator that generating its fake image. In training, these losses are added with a specified weight to make the total loss.

In other words, in the CycleGAN model, the image a is regenerated via the fake image of b, and the regenerated image is distinguished from original a by discriminator L_c. By using this configuration, it is not necessary to prepare the ground truth image that has a one-to-one correspondence with a, namely the contour enhanced image of a itself. For reference, Fig. 2 (5) shows a case of creating a contour-enhanced image from the photo shown in (2) instead of generating it using CG tool. Here, it is necessary to specify the contour of each object manually.

It is necessary that a and b are translated to each other by G_{ab} and G_{ba} in the training. However, since a contour image shown in (3) of Fig. 2 cannot be translated to a, a contour-enhanced image is used for b. That is, the contours are brighter than the threshold, and the rest are darker in b, so b can be translated to a. In addition, this also applies to (5) of Fig. 2.

Using the trained CycleGAN model, a is translated to a corresponding fake contour-enhanced image. Then, the contour image shown in Fig. 2 (3) is obtained by extracting the region where the brightness is above the threshold value from this image. And, using this contour, each target object can be detected, as shown in (4), for their recognition.

4 IMPLEMENTATION OF EXPERIMENTAL SYSTEM

4.1 Contour-enhanced Image Generation Using CG

Figure 4 shows the photo and contour-enhanced CG image of books arranged on the bookshelf, which are the target of this study. They correspond to (2) and (1) of Fig. 2, respectively.

(a) Target photo image (b) Contour enhanced CG image

Figure 4: Target image of experiment: books on bookshelf

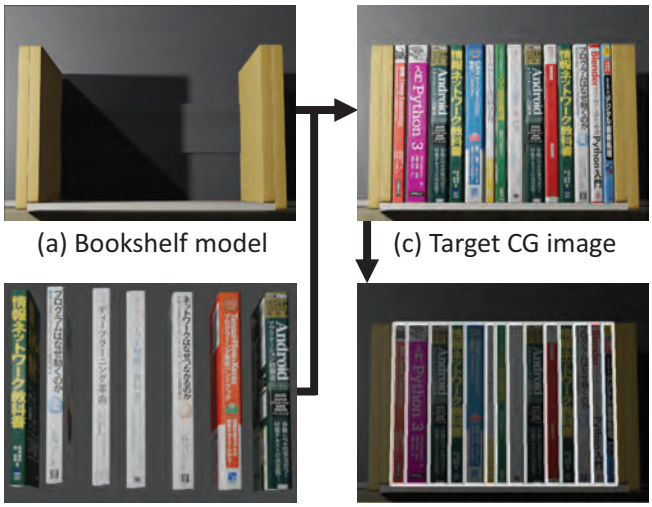


Figure 5: Contour-enhanced CG image generation procedure

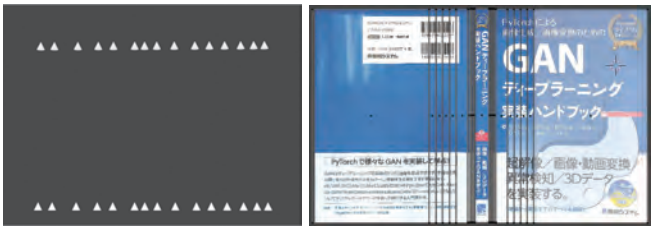
Figure 5 shows the contour-enhanced image generation procedure using a CG tool. First, a bookshelf model (a) and book models (b) are created. Next, books are randomly selected from (b) and arranged on (a) from the left. At the end of this process, books are selected according to the remaining space so that the width of the bookshelf and arranged books match. Finally, the CG image (c) of books on the bookshelf (hereinafter, bookshelf) is generated. Since the placement position and width of each book are grasped when the book is arranged, the contour information of all books can be obtained.

And, using this contour information and the image in (c), a contour-enhanced CG image (d) is generated by converting the brightness as follows.

$$f(x) = \begin{cases} cx & x \notin contour \\ cx + t & x \in contour \end{cases} \tag{6}$$

Here, t is the threshold value mentioned in Sec. 3, and it was set to 156 in anticipation of errors; c is the coefficient of conversion, and it was set to $100/255$. Consequently, since the brightness in the range of $[0, 255]$ is converted to $[0, 100]$ and $[156, 255]$ respectively, the contour area is divided from the other areas.

The CG images were generated using Blender Ver. 2.93.0, and OpenCV Ver. 3.4.2 with Python Ver. 3.7.10 for the image



(a) Vertex models of books' contour (b) Cover image and bevels of book

Figure 6: Contour vertices and book modeling

processing required after generation. First, the bookshelf and books models (hereinafter, bookshelf models) were created (Figs. 5 (a) and (b)). Then, using Python script in Blender's Scripting workspace, the target CG images and their contour-enhanced CG images shown in (c) and (d) were generated automatically and repeatedly. The CG images of (c) and (d) were rendered and saved after each time generation.

The position of each contour's vertex of the rendered image is accompanied by camera-induced distortion. As shown in Fig. 5 (c), the center of the camera's field of view, which is set at the bottom of the central book, is large, and the periphery is small. Therefore, as shown in Fig. 6 (a), the rendering is performed on a model where cones are placed at the position of the vertices at first. Then, the position information of vertices is obtained from this rendered image to create a contour-enhanced CG image.

It was presumed that the information on the boundaries of books is important for contour detection using CycleGAN shown in Sec. 3. As shown in Fig. 4 (a), there are black boundary lines generated from the gap between the books. In the CG book model (b), the left and right corners of the back cover are rounded, and shadow is generated by illuminating it from the side. Consequently, the book boundaries become clear when books are arranged (Fig. 4 (c)). Its concrete method is shown as follows.

In book modeling, the cover image of the book was created using a scanned image of its book jacket. First, a rectangular model to fit the size of the book is created. Then, made its cover image corresponds to the scanned jacket image using the UV Editing workspace of Blender. After that, the above corners were rounded using the bevel function of Blender. These results are shown in Fig. 6 (b). The vertical lines near the center are the bevel positions converted to smooth roundness at rendering. For the bevel function, the width was set to 0.5; the number of segments was set to 10; the shape was set to 0.7.

Figure 7 shows the rendering environment. A point light was placed on the left side to create shadows between books, and a spotlight was placed to prevent the right side from being dark. The position of the camera and black plane, standing behind the bookshelf, was adjusted to match the shooting environment of the photo shown in Fig. 4 (a). The positions of the point light, spotlight, and camera are $(10.4, -10.4, 4.0)$, $(9.9, -10.1, 10.9)$, $(20.0, 0.0, 0.0)$ when the origin is at the bottom of the back of the central book.

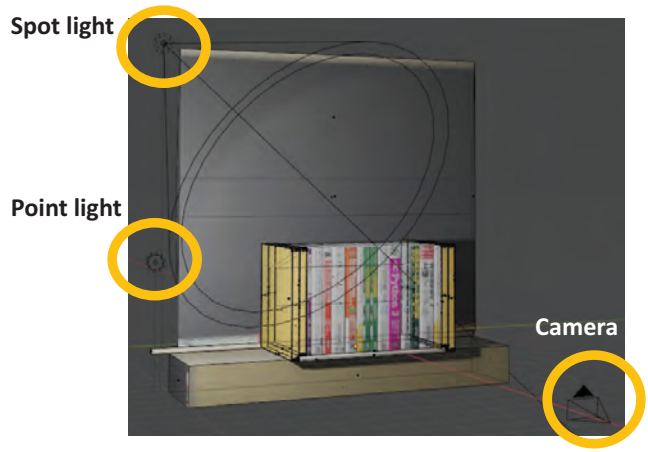


Figure 7: Rendering environment for CG images

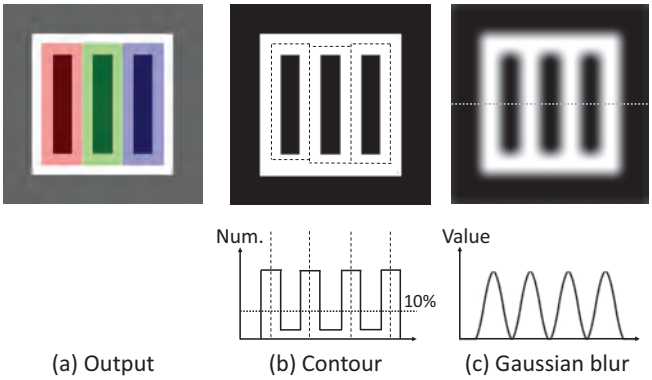


Figure 8: Contour detection and blur for metrics

4.2 Implementation of CycleGAN Model

To evaluate the effectiveness of the proposed method, an experimental system with the necessary function was implemented. This system was constructed on a personal computer, a CPU with i9-10850K (3.6 GHz), and 64 GB memory. GPU was GeForce RTX 3090 with 24 GB memory, and OS was Windows 10. Tools and programming languages were Keras Ver. 2.4.3, Tensorflow-GPU Ver. 2.4.1, and the above Python. The code of the CycleGAN model was created based on the published program code [2], along with the necessary functions for the experiment, such as the early stop control, and saving and displaying results.

For the CycleGAN model training, the above bookshelf images were resized to 128×128, and the batch size was 32. For the configuration of this model, residual networks (ResNet) were used for the shortcut connections skipping one or more layers to increase the depth of the network [6]. Additionally, the weights of the losses in Eqs. (3) to (5) were set to 4, 10, and 2. Their weighted summation was reflected for the training as the total loss.

Figure 8 shows the implementation of contour detection. Figure 8 (a) shows the output image of the CycleGAN model, the fake contour-enhanced image. It is converted to grayscale, then the contour and another area are separated based on the threshold, which was set to 111 considering the grayscale error. Consequently, the contour area is white, and another area is black as shown in Fig. 8 (b). The line segment of the contour area shows the detected contour in the next step.

Contour detection was performed on the image in Fig. 8 (b), and the vertical contour was first detected. Firstly, the number of pixels in the contour area in the vertical direction for each horizontal position was calculated as shown in the lower graph of Fig. 8 (b). Then, the horizontal area with 10% or more contour pixels was selected as a contour candidate area. For the case of Fig. 8 (b), there were four candidate areas. The position of the largest number of pixels was selected as the vertical part of the contour for each area, which is indicated by the broken line.

Next, the number of pixels in the contour area of the horizontal direction was counted between each adjacent vertical part of the above contours. The vertical area with 50% or more numbers was selected as the horizontal part candidates of the contour. Similar to the vertical part, the horizontal part of the contour was selected. Then, the contour of each object was detected, as shown by the broken line in Fig. 8 (b). They were rectangles, in this experiment.

In the training of the CycleGAN model, the following contour differences were used as metrics to determine the completion of the training. One is the contour created from b shown in Fig. 2 (1), which is the ground truth contour. The other is created from $\hat{b}$ shown in Eq. (2) called fake b. To illustrate the distance between the contour lines, Gaussian blur with a kernel size 5×5 and a standard deviation of 0.3 is performed, as shown in Fig. 8 (c). The mean absolute error (MAE) between both is used as the metric.

When the brightness of the pixel of the coordinate (i, j) is shown by g_{ij} for b and f_{eij} for $\hat{b}$ at epoch e respectively, the metric is shown as follows.

$$m_e = \sum_{i=1}^{n} \sum_{j=1}^{n} |f_{eij} - g_{ij}| / n^2 \tag{7}$$

Here, n is the number of pixels in each coordinate axis. In the training, m_e was calculated for each epoch. Every time m_e became the minimum compared before, the model weight was saved as the best weight. Then, when m_e did not improve the specified number of times, the training was completed. In this experiment, this number was set to 15 epochs.

5 EVALUATIONS

5.1 Evaluations of Contour Detection

Contour detection was conducted for both cases: in one case, the contour-enhanced images generated by CG were used; in another case, the contour-enhanced images manually created from the target photo images were used.

For the training and test data, 128 bookshelf photo images shown in Fig. 4 (a) were prepared and inflated to 256 pairs by horizontal flip. Contour-enhanced photo images were manually created from each image through the procedure shown in Eq. (6) and Fig. 8. Images of each group were randomly divided into 204 training data and 52 test data. Since CycleGAN was used, the division of the training and test data in each group did not correspond to each other. On the other

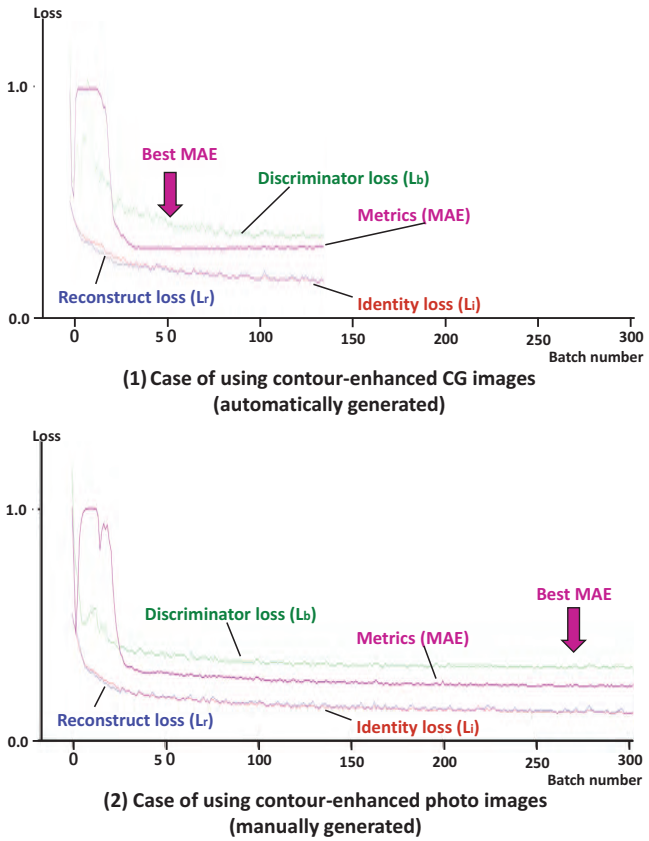


Figure 9: Transition of losses and metrics with training

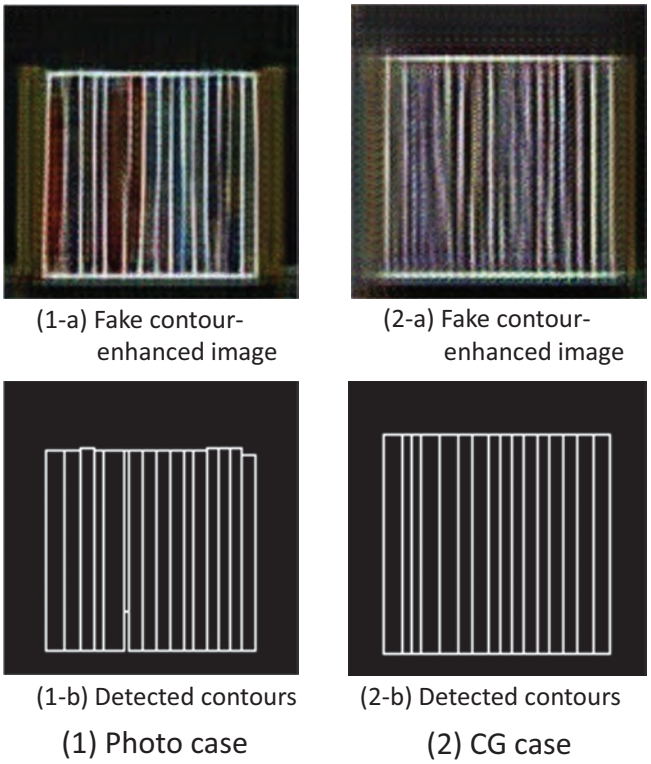


Figure 10: Fake contour-enhanced image and detected contours

hand, 204 contour-enhanced CG images were generated for the training data by the procedure shown in Sec. 4.1.

For the case of the contour-enhanced CG images (hereinafter, CG case), the CycleGAN model was trained with these CG images and the above training data of the photo without contours. The metrics were monitored using the test data simultaneously. Then, the contours were detected from the test data. For the case of the contour-enhanced photo images (hereinafter, photo case), the above contour-enhanced photo images and training data of photo were used. Here, the metrics monitoring and contour detection were performed by the same data with the CG case.

Figure 9 shows the transition of losses and metrics with training shown in Eqs. (3) to (5) and (7). The horizontal axis of the figure shows the batch number, and one epoch corresponds to five batches. During the period immediately after the start of training, each loss and metrics showed individual tendencies. Then, they showed the same tendency. The arrow of "best MAE" indicates the epoch number when the metrics become the best. In the CG case, the best MAE 0.303 was obtained at epoch number 10. Similarly, in the photo case, 0.2278 was obtained at epoch 54.

Figures 10 (1-a) and (2-a) show fake contour-enhanced images generated from the bookshelf image shown in Fig. 4 (a) using the trained model for the above two cases, respectively. Their detected contours using the procedure shown in Fig. 8 (b) are shown in Figs. 10 (1-b) and (2-b), respectively. As shown in Fig. 10 (2), the affine transformation of the above bookshelf image is performed so that the book area of the photo matches the area of the CG image shown in Fig. 5 (b). So, there is a difference in image size between Figs. 10 (1) and (2).

As shown in the upper row of Figs. 10 ((1-a) and (2-a)), the fake contour-enhanced images of the target images were generated in each case, although there are some distorted and blurred parts. Additionally, contours were detected from these images, as shown in the lower row of Figs. 10 ((1-b) and (2-b)). However, some contours are too narrow in width or inconsistent in height compared to books.

5.2 Evaluations of Object Recognition

To evaluate the accuracy of the contour detection, object recognition accuracy was evaluated by template matching of the back cover images between the target photo books, as shown in Fig. 4 (a), and the extracted books using the detected contours. The former back cover images were created using the contours set manually, as mentioned in Sec. 3. The latter books were extracted from the target photo books using the detected contours of each case. Additionally, test data excluded inflated images by horizontal flip were used for target photo books.

Figure 11 shows the template matching results of the CG case. Template matching was performed using matchTemplete function of OpenCV, where the normalized correlation coefficient matching was used. Fig. 11, (1) shows the back covers of target photo books; (3) shows one created using detected contours. (2) shows the template matching results between the two, and the back cover images are the same as (1), and the image of (3) is shown by a white rectangle. Similarly,

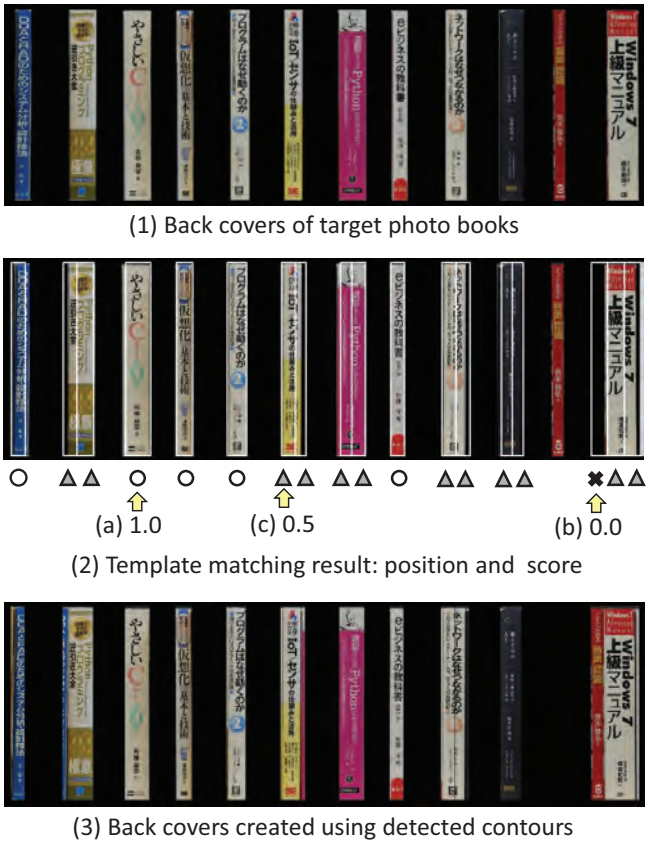


Figure 11: Book recognition results using detected contours of CG case

Fig. 12 shows the results of the photo case.

The following three scores were introduced according to the matching level to evaluate the accuracy of object recognition, as shown in Fig. 11 (2). (a) shows the case of almost the same indicated by ◯, which score is 1.0; (b) shows the case of not almost overlap indicated by ×, which score is 0.0; (c) shows the case in between both of the above indicated by △, that is, there is an extra area or only a part of both is overlap. Here, (c) is the case when 50% or more of the detected contour area is included in one of the target photo books; (b) is the case of less than 50%.

Figure 11 (2) and Fig. 12 show that there are differences in the results of each book recognition. For example, the second back cover from the left is detected by dividing into two in Fig. 11 (2), but it is detected as one in Fig. 12. Additionally, the third back cover from the right is recognized as score 0.5, and the second is not recognized in Fig. 11 (2). For these back covers, the recognition result was reversed in Fig. 12.

Figure 13 shows the percentage of back covers recognized with scores of 1.0 and 0.5 out of a total of 237 back covers. Here, even if the back cover is recognized twice with a score of 0.5, such as the book on the right end of Fig. 12, it is counted as one book. When only the score of 1.0 was targeted, the recognition rate was 55.7% for the CG case, lower than 59.9% for the photo case. However, including the score of 0.5, it was 84.5% and 81.5%, and the CG case was higher than the photo case.

To analyze the status of the score, the average score in each

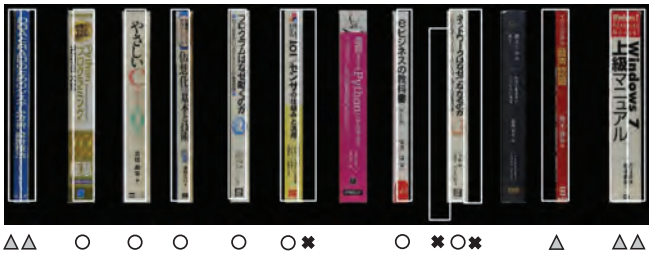


Figure 12: Book recognition results using detected contours of photo case

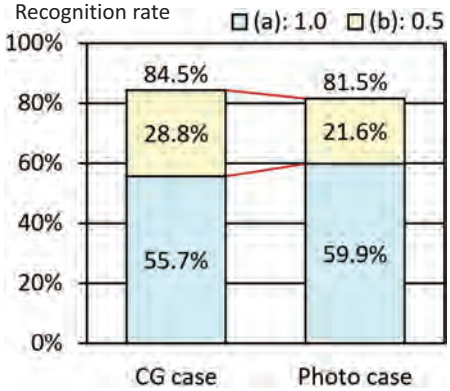


Figure 13: Book recognition rate using detected contour

target book image is shown in Fig. 14, and the total number of each score is shown in Fig. 15. As shown in Fig. 14, the line graph shows the result arranged in descending order to grasp the score's tendency between each case. Note that even if the image number in this figure is the same, both results are not for the same target book image. The score was from 0.87 to 0.39 in the CG case and 1.0 to 0.43 in the photo case, and the averages were 0.63 and 0.69, respectively. That is, the average of the CG case was 9.5% lower than the photo case.

Figure 15 shows the total number of the target photo books is 237, and the number of detected contours exceeds this in both cases. There are 379 detected contours, which is more than the 328 in the photo case. The number of the score of 1.0 is 162. Similarly, the number of the score of 0.5 is 150.

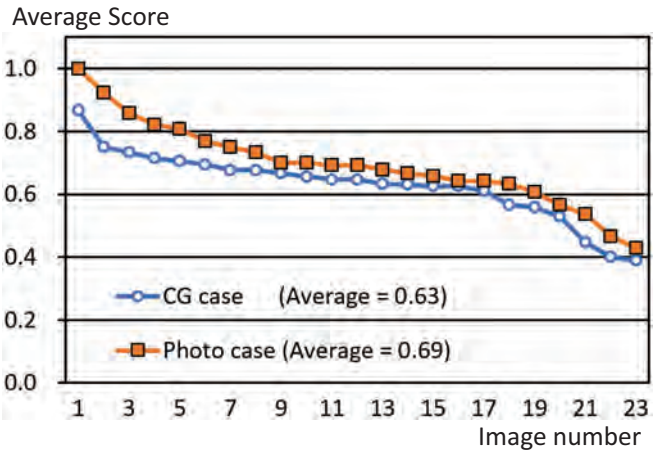


Figure 14: Average score of book recognition for each image

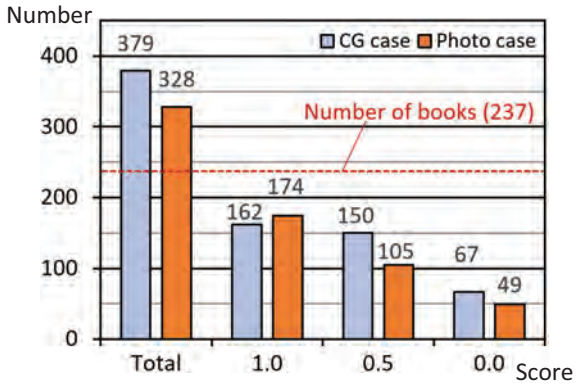


Figure 15: Number of each score in book recognition

Figure 16: Evaluation of contour-enhanced image using target images of different sizes

Comparing the photo case, though the former is less than 174 of the photo case, the latter is more than 105.

5.3 Evaluations of Influence with Contour Size Error

The influence of the error in the size of the photo image of the book and the contour-enhanced CG image was evaluated. There was also an error in their positions. Thus, this experiment was conducted without closely matching their sizes by the affine transformation mentioned in Sec. 5.1. Figure 16 shows the images used in this experiment. Figure 16 (a) shows a book photo image, and Fig. 16 (b) shows a book CG image from which a contour-enhanced image was created. The latter is a little larger, especially positions of the upper sides of the contours are high. Figure 16 (d) shows the fake contour-enhanced image generated using the model trained with these images. Here, Fig. 16 (c) is a photo image

Figure 17: Book recognition results using target images of different sise

of the book converted to the size of 128×128 pixels to input the model. In Fig. 16 (d), the position of the upper side is higher than in (c), similar to the CG image of the book shown in (b).

Contour detection was performed using this fake contour-enhanced image shown in Fig. 16 (d). Then, each book image was extracted by these contours from the bookshelf photo image shown in Fig. 16 (a). Figure 17 shows the results of book recognition using detected contours similar to Sec. 5.2.

As shown in Fig. 17, the contour is detected as large size, and most of them have an empty area at the top. In the matching result of the second book from the left, the bottom of the rectangle is omitted that extends beyond the figure.

6 DISCUSSIONS

This study shows that it is possible to perform contour detection using CycleGAN without preparing ground truth images with a one-to-one correspondence to the target images. For the books arranged on the bookshelf, the comparative evaluation was performed on the accuracy of the target book's back cover recognitions using the detected contour by the CG and photo cases (Figs. 11 and 12).

Consequently, the almost same accuracy was obtained in these two cases (Fig. 13). At a score of 1.0, the accuracies of the CG and photo cases were 55.7% and 59.9%, respectively, meaning that the photo case was higher than the CG case. However, at scores of 1.0 and 0.5, the accuracies of the CG and Photo cases were 84.5% and 81.5%, meaning that the CG case was higher than the photo case.

The score 1.0 ratio was lower in the CG case is considered because the ratio of divided contour for a back cover is more than in the photo case (Fig. 15). For example, the contour of one back cover was detected as two contours of score 0.5 as shown in the second back cover from the left in Fig. 11 (2). The score of 1.0 was less in the CG case than in the photo case, but that of 0.5 was more. This is a general tendency, as shown in Fig. 14. The average score of the CG case is lower than that of the photo case.

To improve the score 1.0 ratio in the CG case, it is conceivable to adjust the threshold for detecting vertical contours shown in Fig. 8, which is currently set to 10%, to an appropriate value. As shown in Fig. 10 (2-a), since the fake contour-enhanced image has distortion and blurring of the contour area, it is expected that these parts can be excluded by setting the threshold higher.

Furthermore, to improve the contour detection accuracy,

it is considered that the method to incorporate the metrics, which are used for only the early stop of training the model in this study, as a discriminator of the CycleGAN model to feedback error between the following. One is the detected contour created by a fake contour-enhanced CG image, as shown in Fig. 10 (2-b). Another is the ground truth contour created from the CG vertex model, as shown in Fig. 6 (a). These improvements are feature challenges.

The experiment showed that the ground truth images could be automatically generated by the CG tools using the prepared bookshelf and book models in advance, instead of manually creating each ground truth image. This method using CG images is considered effective in the field, where many objects with relatively few types are randomly arranged. It means that various application fields of contour detection are expected such as objects placed on desks and cars on roads.

As shown in Fig. 17, using the CycleGAN model, the difference of scale and position between the target and ground truth images is significantly affected on the detected contour. Therefore, depending on the application field, it may be effective to match the shapes of the target and ground truth image by preprocessing.

Evaluations of the effectiveness of the proposed method in environments with these various shapes and backgrounds are feature challenges.

7 CONCLUSIONS

To recognize one object in objects arranged in high density, contour detection is often required to preprocess. In this field, various methods have been proposed utilizing GAN to perform pixel-by-pixel detection. However, with these methods, it is necessary to prepare ground truth images that have a one-to-one correspondence with the target images. Thus, there is often an obstacle to their application. Meanwhile, the CycleGAN model can translate images from two different domains to each other without preparing such one-to-one correspondent ground truth images.

Therefore, this study proposed a method to use the ground truth images automatically generated by CG tool utilizing the advantages of CycleGAN. Furthermore, the comparative evaluations between this method and the approach of using the ground truth images that had a one-to-one correspondence with the target images were conducted. The accuracy of image recognition using template matching based on detected contours was compared. Consequently, the proposed metho's accuracy could be close to the case of using the conventional one-to-one correspondence ground truth images, even when the ground truth CG images were used.

Future challenges include the improvement of contour detection accuracy using the proposed method and the evaluation of the accuracy in the case of applying this method to other application fields.

ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 19K11985.

REFERENCES

[1] X. Chu, A. Zheng, X. Zhang, and J. Sun, "Detection in crowded scenes: One proposal, multiple predictions," Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition, pp. 12214–12223 (2020).

[2] D. Foster, "Generative deep learning: Teaching machines to paint, write, compose, and play," O'Reilly Media (2019), https://github.com/davidADSP/GDL_code (referred May 24, 2021).

[3] P. Gao, T. Tian, L. Li, J. Ma, and J. Tian, "DE-CycleGAN: An object enhancement network for weak vehicle detection in satellite images," IEEE J. Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 14, pp. 3403–3414 (2021).

[4] X. Y. Gong, H. Su, D. Xu, Z. Zhang, F. Shen, and H. Yang ,"An overview of contour detection approaches," Int. J. Automation and Computing, Vol. 15, No. 6, pp. 656–672 (2018).

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial networks," Communications of the ACM, Vo. 63, No. 11, pp.139–144 (2014).

[6] K. He, X, Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Conf. Computer Vision and Pattern Pecognition, pp. 770–778 (2016).

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," Proc. IEEE Int. Conf. Computer Vision, pp. 2961–2969 (2017).

[8] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 1125–1134 (2017).

[9] T. Kudo, and R. Takimoto, "CG utilization for creation of regression model training data in deep learning." Procedia Computer Science, Vol. 159, pp. 832–841 (2019).

[10] T. Kudo, "A proposal for article management method using wearable camera," Procedia Computer Science Vol. 178, pp. 1338–1347 (2020).

[11] T. Kudo, "Moving object detection method for moving cameras using frames subtraction corrected by optical flow," Int. J. Informatics Society, Vol. 13, No. 2, pp. 79–91 (2021).

[12] M. Li, Z. Lin, R. Mech, E. Yumer, and D. Ramanan, "Photo-sketching: Inferring contour drawings from images," 2019 IEEE Winter Conf. Applications of Computer Vision (WACV), pp. 1403–1412 (2019).

[13] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2117–2125 (2017).

[14] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection." Proc. IEEE Int. Conf. Computer Vision, pp. 2980–2988 (2017).

[15] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," Int. J. Computer Vision, Vol. 128, No. 2, pp. 261–318 (2020).

[16] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," IEEE Communications Surveys and Tutorials, Vol. 20, No. 4, pp. 2923-2960 (2018).
[17] M. Rajiput, "YOLOv5 is here! elephant detector training using custom dataset & YOLOV5," https://towardsdatascience.com/yolo-v5-is-here-b668ce2a4908 (referred May 24, 2021).
[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 779–788 (2016).
[19] D. Rukhovich, K. Sofiiuk, D. Galeev,O. Barinova, and A. Konushin, "IterDet: Iterative scheme for object detection in crowded environments," arXiv preprint arXiv:2005.05708 (2020).
[20] K. Saleh, A. Abobakr, M. Attia, J. Iskander, D. Nahavandi, M. Hossny, and S. Nahvandi, "Domain adaptation for vehicle detection from bird's eye view LiDAR point cloud data," Proc. IEEE/CVF Int. Conf. Computer Vision Workshops, pp. 3235–3242 (2019).
[21] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks," Scientific Reports, Vol. 9. No. 1, pp. 1–9 (2019).
[22] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 3982–3991 (2015).
[23] A. Sindel, A. Maier, and V. Christlein, "Art2Contour: Salient contour detection in artworks using generative adversarial networks," 2020 IEEE Int. Conf. Image Processing (ICIP), pp. 788–792 (2020).
[24] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Detection of apple lesions in orchards based on deep learning methods of cyclegan and YOLOv3-dense," J. Sensors, Vol. 2019, Article 7630926 (2019).
[25] M. Verhelst, and B. Moons, "Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to IoT and edge devices," IEEE Solid-State Circuits Magazine, Vol. 9, No. 4, pp. 55–65 (2017).
[26] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. 2001 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, Vol. 1, pp.511–518 (2001).
[27] H. Yang, Y. Li, X. Yan, and F. Cao, "ContourGAN: Image contour detection with generative adversarial network, Knowledge-Based Systems, Vol. 164, pp. 21–28 (2019).
[28] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," ACM Computing Surveys (CSUR), Vol. 38, No. 4, Article 13 (2006).
[29] Z. Zeng, Y. K. Yu, and K. H. Wong, "Adversarial network for edge detection," Int. Conf. Informatics, Electronics & Vision (ICIEV), pp. 19-23 (2018).
[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 2881–2890 (2017).
[31] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," Proc. IEEE Int. Conf. Computer Vision, pp. 2223–2232 (2017).

(Received: October 31 , 2021)
(Accepted: January 11, 2022)

Tsukasa Kudo received the BSc and ME from Hokkaido University in 1978 and 1980, and the Dr. Eng. from Shizuoka University in 2008. In 1980, he joined Mitsubishi Electric Corp. He was a researcher of parallel computer architecture and engineer of business information systems. Since 2010, he is a professor of Shizuoka Institute of Science and Technology. Now, his research interests include deep learning and database application. He is a member of IEIEC and IPSJ.

Regular Paper

A Scalable Video-on-Demand System on Edge Computing Environments

Satoru Matsumoto* and Tomoki Yoshihisa*

*Cybermedia Center, Osaka University, Japan
{smatsumoto, yoshihisa}@cmc.osaka-u.ac.jp

Abstract -Due to the recent increase in the users of video-on-demand (VoD) services, many clients such as smart phones or laptop computers request video data to a video distribution server. Such large-scale VoD systems utilize the edge computing technology to distribute the communication load and the processing load on the video distribution server. In most of the VoD systems utilizing edge computing, the edge servers receive a part of the video data from the video distribution server and caches them for other transmissions. However, the edge servers can receive them before receiving the requests from the clients (pre-cache). Moreover, the edge servers can transmit pre-cached video pieces to other edge servers. Here, the research challenges are: which pieces the edge servers should pre-cache and how to redistribute the pieces among the edge servers to effectively distribute the loads. In this paper, we propose a scalable VoD system on edge computing environments. Our evaluation results revealed that our proposed system reduces the probability that the transmissions overlap with other transmissions and increases the shortest arrival interval of the clients that the system can work by 26% compared with the conventional system in the simulated situation.

Keywords: Streaming media distribution, interruption time, webcasting, Internet broadcasting, cloud computing

1 INTRODUCTION

Recently, video-on-demand (VoD) services such as YouTube or Netflix are widely used. In most VoD services, the clients request the video data to the video distribution server. The video distribution server sends the video data pieces sequentially to the clients so that they can play the video while receiving the pieces. When the clients cannot finish receiving each piece before starting playing it, the video playback is interrupted. A longer interruption time more annoys the viewers. Here, the interruption time means the total time that the playback is interrupted while a client is playing the video. Therefore, various interruption time reduction schemes for VoD services have been proposed.

In large-scale VoD systems, many clients request the video data and thus the transmissions of data pieces frequently overlap with other transmissions. Here, the transmission means a series of distributing the pieces to a client. A transmission starts when a client requests playing a video to the video distribution server and finishes when the client finishes receiving all the pieces of the video data. If the times required for transmissions increase and the transmissions continue to overlap with others, the interruption times also continue to lengthen. Therefore, existing schemes for interruption time reduction aim to reduce the transmission time to avoid the overlapping of transmissions. Major techniques for this are pre-caching ([1]-[4]), redistributions of data pieces ([5]-[7]), etc. and are adopted in various CDN (Contents Delivery Networks) for scalable VoD systems ([8]). Unfortunately, these traditional approaches cause the communication and processing loads on the clients. Such extra loads decline the users' operability of the clients and further consume the batteries if they are mobile devices.

Edge computing is one of the approaches that relief the computational loads on the clients since the edge servers, i.e., the servers on the edge of the network and geometrically close to the clients, are often managed by CDN companies such as Akamai or Cloudflare. In most of the VoD systems utilizing edge computing, the edge servers receive some pieces from the video distribution server and caches them for other transmissions. However, the number of the pieces that the edge servers need to send decreases by adopting the above both techniques (pre-caching and redistributions) to the edge servers. Here, the research challenges are: which pieces the edge servers should pre-cache and how to redistribute the pieces among the edge servers to effectively reduce the communication load and the processing load.

In this paper, we propose a scalable VoD system on edge computing environments. Our proposed system adopts a distributed edge caching scheme. In our proposed system, the edge servers store some pieces before starting the VoD service. When a new client requests a video data, the video distribution server selects an edge server for sending the pieces of the requested video data to the client. The edge server sends its stored pieces to the client. After finishing sending the stored pieces, the video distribution server sends the subsequent pieces to the client. Thus, our proposed system reduces the probability that the transmissions overlap with other transmissions and increases the maximum number of the clients of that interruption time converges. The novelty of the proposed system is the adopting both the pre-caching technique and the redistribution technique to the VoD systems utilizing edge computing. The contributions of the paper are (1) the increase of the number of the clients that the system can accommodate, (2) the proposition of a scalable video-on-demand system, (3) the confirmation of the effectiveness of the proposed system.

The paper is organized as follows. Some related work are introduced in Section 2. The proposed scheme is explained in Section 3, analyzed in Section 4, and evaluated in Section 5. Finally, we conclude the paper in Section 6.

2 RELATED WORK

Many studies focus on fast data reception for VoD services.

ISSN1883-4566 © 2022 - Informatics Society and the authors. All rights reserved.

2.1 Pre-caching Techniques for VoD Services

Abuhadra et al. proposed a proactive caching technique for mobile devices [1]. The probability that the clients encounter interruptions decreases by sending more video data to the mobile devices while their network connections are available because they sometimes disconnect from the network. The proposed technique reduces in-network transmission delays by caching the video data. Feng et al. found the optimal cache placement for the system with wireless multicasting [2]. My research group proposed a broadcasting method for pre-caching video data pieces predicting the video data that the client will play [3]. Coutinho et al. proposed a proactive caching technique for DASH video streaming [4]. In the proposed technique, the clients select a proxy caching server based on the network conditions. However, these pre-caching techniques for VoD services require the clients' storage capacity.

2.2 Redistribution (Peer-to-Peer Sharing) Techniques for Video-on-Demand Services

Sheshjavani et al. proposed a peer-to-peer data sharing mechanism for VoD services [5]. In the proposed mechanism, the clients manage their buffer map. The buffer maps inscribe the received video data pieces and non-received pieces of each client. In the mechanism, the clients exchange the pieces based on the buffer map to receive the pieces that each client does not have. In the method proposed by Zhang et al., the clients send the pieces considering the bandwidth consumption [6]. Fratini et al. analyzed the efficiency of using replicated video servers [7].

However, similar to the pre-caching techniques, these redistribution techniques cause communication and processing loads on the clients. Such extra loads decline the users' operability of the clients and consume the batteries if they are mobile devices.

2.3 Video-on-Demand Services on Edge Computing

Due to the recent prevalence of edge computing, some researchers focus on edge computing for VoD services. Mehrabi et al. proposed an edge computing assisted adaptive mobile video streaming [9]. The mechanism determines the video resolution and the video data rate based on the network conditions between the clients and the edge servers. The performance of an edge computing enhanced video streaming is investigated by Yang et al. [10].

3 PROPOSED SYSTEM

In this section, we explain our proposed system.

3.1 Proposed System Architecture

Figure 1 shows the assumed system. The system consists of three layers, the top layer, the edge layer, and the client layer. One video distribution server is in the top layer. CDN (Contents Delivery Network) companies or VOD service companies provide the machines in the edge layer. The edge layer includes some edge servers. The clients are in the client layer. Similar to other researches for the VoD systems utilizing edge computing, the networks for these three layers are application layer networks and we assume that the influences of the underlaying session/transport layers are sufficiently small.

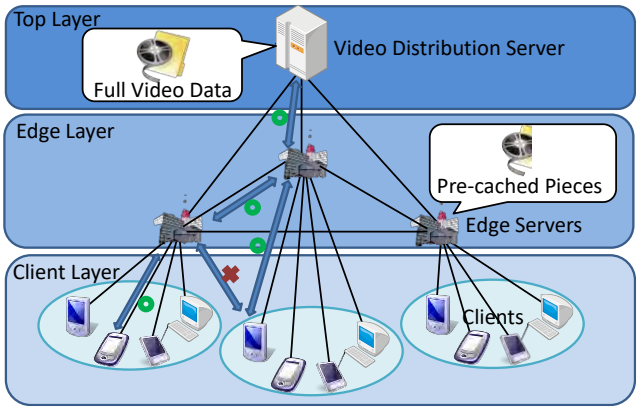


Figure 1: Assumed Environment

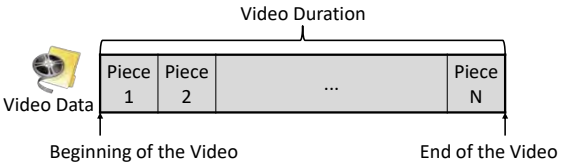


Figure 2: Video Data Division

The single video distribution server has full video data for all the videos and connects to the Internet. The edge servers also connect to the Internet and can communicate with the video distribution server. They can store a part of some video data (pieces). The clients connect to the geometrically closest edge server and can communicate with the edge server. The clients request the video data to the video distribution server. Each video data are divided into some pieces, as shown in Fig. 2. The clients receive the requested video data pieces from the video distribution server and the edge servers.

The assumed system model is general and practical. One of the applications is that the video distribution server provides the videos to the clients in a prefecture, and each edge server serves for each region in the prefecture. For example, there are eight local regions in Osaka, Japan. In this case, the number of the edge servers is eight, and the edge servers provides 100 videos.

3.2 Target Issue

We explain our target issue in this subsection.

3.2.1 Issues in Conventional Systems

In the VoD systems, the video playback is interrupted when the clients cannot finish receiving each piece before starting playing it. A longer interruption time more annoys the viewers. Here, the interruption time means the total time that the playback is interrupted while a client is playing the video. In

the VoD system in that a video distribution server distributes the video data to the clients, the communication load and the processing load for the distribution concentrate on the server. Therefore, in the cases that such VoD system hold many clients (scalable) and the clients frequently requests to play the videos to the video distribution server, the server is easy to overload. The overloading results in long video data transmission times and causes long interruption times. In most of the VoD systems utilizing edge computing, the edge servers receive pieces from the video distribution server and caches them for other transmissions.

3.2.2 Pre-Caching Pieces

The edge servers can receive pieces from the video distribution server before the requests for them come from the clients. That is, the edge servers can pre-cache the pieces. The pre-caching can reduce the communication load and the related processing load on the video distribution server and the edge servers. This is because they receive the pieces without the requests for the pieces and can receive them before starting the VoD service. Pre-caching more pieces reduce more loads, but require more storage capacity on the edge servers.

3.2.3 Redistributing Pieces

Moreover, the edge servers can redistribute pieces to other edge servers. The redistribution can distribute the communication load and the related processing load on the video distribution server to the edge servers because the edge servers send the pieces instead of the server. However, if an edge server frequently redistributes the pieces, the loads concentrate on the edge server, results in long interruption time. Therefore, the edge servers need to redistribute the pieces without causing the overloads on it.

3.2.4 Our Objective

Based on the discussion in this subsection, we aim to reduce the interruption time by adopting pre-caching and redistributing pieces on edge computing environments. For this, we propose a scheme which determines which pieces the edge servers should pre-cache and how to redistribute the pieces among the edge servers.

3.3 Proposed Scheme

Our proposed scheme runs on our proposed system architecture explained in Subsection 3.1. In the scheme, the edge servers pre-cache several preceding pieces of popular videos. When a client requests playing a video to the connected edge server, the edge server checks whether it has pre-cashed the requested video already. If the piece that the client is going to receive is pre-cached, the client receives the piece from the connected edge server. If the piece is pre-cached by another edge server, the connected edge server receives it from the edge server and after that sends it to the client. If the pieces is not pre-cached by any edge server, the connected edge server receives it from the video distribution server and after that sends it to the client.

3.3.1 How to Pre-Cache Pieces

To solve the first issue, the edge servers pre-cache preceding several pieces of popular videos, i.e., the pieces that are close to the beginning of the videos. This is because the time to start playing the preceding pieces is early, and the possibility that the clients encounter interruptions is high. The preceding pieces of the videos that are frequently requested by the connected clients are pre-cached. To avoid redundant pre-caching, the edge servers do not pre-cache the pieces that are pre-cached by other edge servers. The edge servers do not cache the pieces that are transmitted to the clients except for the pre-cached pieces to reduce the required storage capacity for caching. The number of the pieces to be pre-cached is a parameter for the scheme.

For example, imagine the case that the VoD system provides 100 videos and owns eight edge servers. When the number of the pieces to be pre-cached is set by 10 per each video, each edge server pre-caches the preceding 10 pieces of the 100/8=25 videos.

3.3.2 How to Redistribute Pieces

To solve the second issue, in the cases that the edge server does not pre-cache the requested video, it requests the redistribution of the preceding pieces of the video to the edge server that pre-caches the requested video. This is because the edge layer and the client layer are separated in our proposed system, and thus, the clients cannot communicate with the edge servers that they do not belong to. Therefore, the edge servers redistribute the pieces to other edge servers, not directly to the clients.

For example, imagine the case that the client 1 directly connects to the edge server 1 and requests the video 2. The preceding pieces of video 2 is pre-cached by the edge server 2. In this case, the edge server 2 redistribute the pre-cashed pieces of the video 2 to the edge server 1 and the edge server 1 sends the received pieces to the client 1.

3.4 Flow of Procedures

In this subsection, we explain the flow of the procedures for the video distribution server, the edge servers, and the clients.

3.4.1 Video Distribution Server

The video distribution server has all the pieces of all the video data. When it receives a request for a piece, it sends the requested piece to the requesting edge server.

Moreover, the video distribution server manages the statistics of the VoD system and measure the popularity of the video data to determine which video data each edge server should pre-cache. This can be performed by calculating popularity of the video data. The video distribution server can calculate this by getting the information about the video requests from all the edge servers.

3.4.2 Edge Servers

Figure 3 shows the procedure flow of the edge servers. In the figure, $p_{j,i}$ indicates the ith piece of the video j. When an

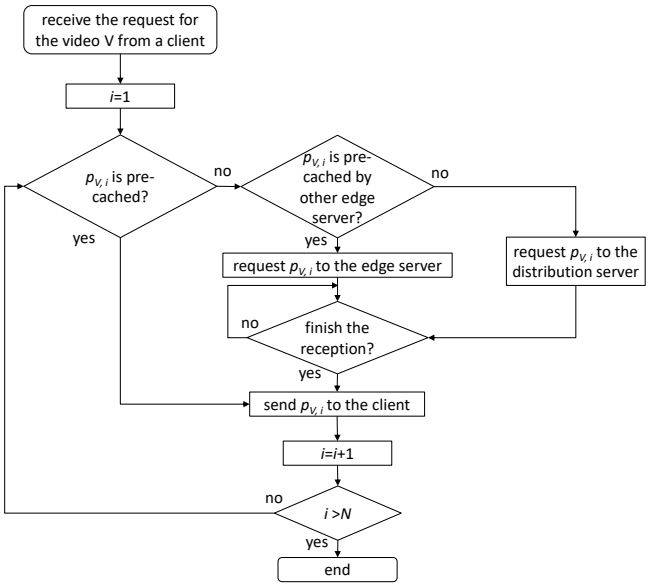


Figure 3: The procedure flow of the edge servers

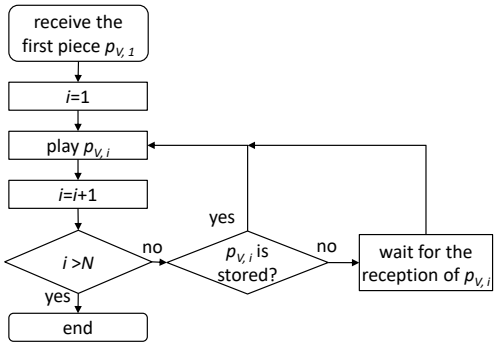


Figure 4: The procedure flow of the clients

edge server receives the request for the video from a client, it checks whether it pre-caches the first piece of the video or not. If the edge server pre-caches the piece, it sends the piece to the client. Otherwise, it finds the edge server that pre-caches the piece. If there is the edge server that pre-caches the piece, it requests the piece reception to the edge server and waits for the reception. When the reception completes, it sends the received piece to the client. If no edge servers pre-cache the piece, it request the piece reception to the video distribution server and sends it to the client when the piece reception completes. The edge server that receives the request of the video continue to this procedure until it sends the last piece. When it completes sending all the pieces of the video to the client, the flow for the request finishes.

3.4.3 Clients

Figure 4 shows the procedure flow of the clients. Similar to Fig. 3, $p_{j,i}$ indicates the ith piece of the video j. When a client receives the first piece, it starts playing the piece. After playing the piece, the client try to continuously play the next piece and checks whether it has stored the next piece or not. If the next piece has already stored in its storage, it starts playing the next piece. Otherwise, it waits for the reception of the next piece. In this case, interruption occurs. The client continue to this procedure until finishing playing all the pieces.

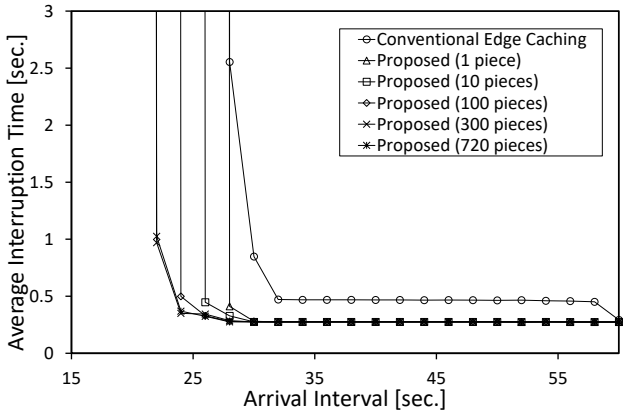


Figure 5: Average arrival interval and the interruption time

4 MATHEMATICAL ANALYSIS

In this section, to find the necessary bandwidth among the video distribution server, the edge servers, and the clients, we analyze the communication situation under the proposed scheme.

In our proposed scheme, each video data is divided into some pieces. The data amount for each piece is the same and is denoted by P. For example, if the number of the pieces of the video i is NP_i, the data size of the video is $P \times NP_i$.

4.1 Necessary Bandwidth Between Clients and Edge Servers

Suppose the case when an edge server receives the new video request before finishing sending all the pieces to the current client. In this case, the interruption time diverges since the piece transmissions overlap. The average arrival interval of the clients for an edge server is given by λE. λ is the average global arrival interval of the request for the videos from the clients, and E is the number of the edge servers. Therefore,

$$\frac{PM}{B_{EC}} < \lambda E . \quad (1)$$

Here, M is the maximum number of the pieces of all the videos. B_{EC} is the bandwidth between the edge server and the clients. For the analysis, I assume that the bandwidth is the same for all the edge servers. From (1), the following inequality should be satisfied to converge the interruption time.

$$B_{EC} > \frac{PM}{\lambda E} \quad (2)$$

4.2 Necessary Bandwidth Among Edge Servers

We assume that the video data are requested with a same probability. Each edge server has the same number of the videos. Let V denote the number of the videos that the system provides. Then, the probability that a video is requested when a client requests playing a video is $1/V$. Therefore, the probability that the video data that an edge server has are requested is $(V / E) / V = 1/E$. Contrary, the probability that the directly connected edge server does not have the requested video data is $(E - 1)/E$. Therefore, a redistribution occurs with the probability $(E - 1)/E$. The time that the edge server can consume for

the redistribution is λE if the edge server does not have the next requested video. The probability that the edge server does not have the next requested video is (E - 1)/E. If the edge server has the next requested video (the probability is 1/E), the time that the edge server can consume for the redistribution is $2\lambda E$. Thus, the average time that the edge server can consume for the redistributions is given by

$$\sum_{i=1}^{\infty} i\lambda E \left(\frac{E-1}{E}\right)^2 \left(\frac{1}{E}\right)^{i-1}. \quad (3)$$

Let J the number of the pieces allocated to each edge server. The time needed to send J pieces among the edge servers should be shorter than this average arrival interval to avoid transmission overlapping. Therefore,

$$\frac{PJ}{B_{EE}} < \lambda(E-1)^2 \sum_{i=1}^{\infty} i \left(\frac{1}{E}\right)^i. \quad (4)$$

Hence, the bandwidth among the edge servers B_{EE} should satisfy the following inequality.

$$B_{EE} > \frac{PJ}{\lambda(E-1)^2 \sum_{i=1}^{\infty} i \left(\frac{1}{E}\right)^i} \quad (5)$$

4.3 Necessary Bandwidth for Video Distribution Server

The edge servers receive the pieces that are not allocated to any other edge servers from the video distribution server. The time needed to send $M - J$ pieces from the video distribution server to the edge server should be shorter than the average global arrival interval. Therefore,

$$\frac{P(M-J)}{B_{DE}} < \lambda. \quad (6)$$

B_{DE} is the bandwidth between the video distribution server and the edge servers. Hence, the following inequality should be satisfied to converge the interruption time.

$$B_{DE} > \frac{P(M-J)}{\lambda} \quad (7)$$

5 EXPERIMENTAL EVALUATION

To check the performances of the proposed scheme, we measured the interruption time using our developed simulator.

5.1 Evaluation Setting

Based on the application example in Subsection 3.1, the number of the edge servers is eight, and the edge servers provide 100 videos. The bandwidth between each edge server and the clients is 100 [Mbps] considering a realistic situation. The bandwidth between the video distribution server and the edge servers is 600 [Mbps], and that among the edge servers is also 600 [Mbps], considering that these are in the backbone network. I set the same bandwidth to all the edge servers to make the experiments precisely understandable. The video distribution server can communicate with each edge server directly, and the edge servers can communicate with each other. The clients connect to the closest edge server.

The video duration is 60 [min.], and the bitrate is 5 [Mbps] based on the videos provided by practical services. The data amount of a piece is the same as the video data for 5 [sec.] based on HLS (HTTP Live Streaming [15]) and is 3125 [Kbytes]. The number of the pieces in each video data is 720.

The users request playing one of the videos according to a fixed arrival interval to make the results be easily understandable. In the case that the requests non-uniformly arrive, the average interruption time increases because the maximum value increases. The popularity of the video data is given fairly.

We compare the proposed scheme with a conventional edge caching scheme, an often used caching technique for CDN. In the scheme, the pieces are cached at the edge servers, but not redistributed among them. The edge servers receive the pieces that need to be sent to the clients and are not cached from the video distribution server.

5.2 Influence of Arrival Interval

More frequent arrivals of the requests for playing the video data cause more transmission overlaps, and thus, the interruption time can continue to increase with a higher probability. Therefore, we investigate the influence of the arrival intervals of the clients' requests.

5.2.1 Interruption Time

Figure 5 shows the average interruption time under different arrival intervals. The horizontal axis is the arrival interval and the vertical axis is the average interruption time. In the legend, `Conventional Edge Caching' indicates the average interruption time under the conventional edge cashing scheme explained in the previous subsection. `Proposed (J pieces)' indicated the average interruption time under our proposed scheme. In the scheme, each edge server pre-caches J pieces.

We can see that the average interruption times under each scheme are almost the same when the arrival interval is longer than a certain value. This is because the transmissions does not overlap with others and the interruption time does not continue to increase. Under the conventional scheme, the average interruption time when the arrival interval is longer than 30 [s] is a little bit longer than that under our proposed scheme. This is because the bandwidth between the edge servers and the clients are not sufficient and the transmissions sometimes overlap with others. On the other hand, we can see that the average interruption times under all schemes suddenly increases when the arrival interval is shorter than a certain value. This is because the transmissions always overlap with the next transmission and the interruption time continue to lengthen. In such cases, the VoD system abandons and cannot provide their services.

For example, when the arrival interval is 25 [s], the conventional scheme cannot provide the service, but our proposed scheme can provide it and the average interruption time is approximately 325 [ms]. At the shortest, our proposed system

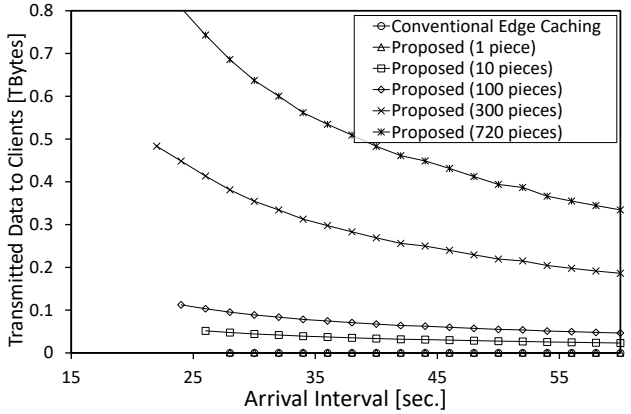


Figure 6: Average arrival interval and the transmitted data to the clients

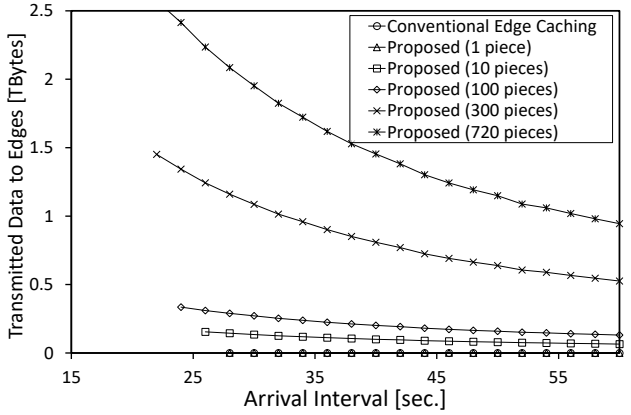


Figure 7: Average arrival interval and the transmitted data to the edge servers

can work even when the arrival interval is 22 [s]. The conventional method can provide service only when the arrival interval is longer than 30 [s] in this situation. Therefore, our proposed system can improve the shortest arrival interval under that the system can work 26% compared with the conventional system.

5.2.2 Edge Servers' Data Transmission

One of the indexes for the communication load and the processing load on the edge servers is the data amount transmitted to others. Therefore, we investigate the amount changing the arrival interval.

Figure 6 shows the total data amount transmitted to the clients by the edge servers. Since the average interruption time diverges when the arrival interval is excessively short, the lines stop at the shortest arrival interval under that the interruption time converges. The data amount decreases as the arrival interval increases because the number of the clients that receive data per time decreases. The data amount increases as the edge servers pre-cache more data because they need to transmit them instead of the video distribution server.

Figure 7 shows the total data amount transmitted to other edge servers. The result is similar to Fig. 6, the total data amount transmitted to the clients. But, the amount differs. The data amount transmitted to other edge servers is generally larger than that to the clients because the edge servers request the data transmissions to other edge servers in proportional to

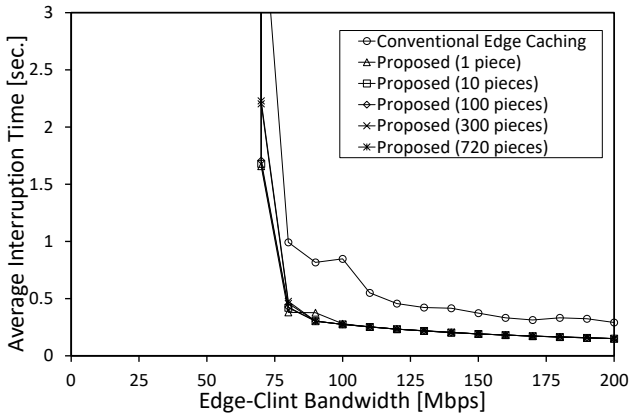


Figure 8: Edge-client bandwidth and the interruption time

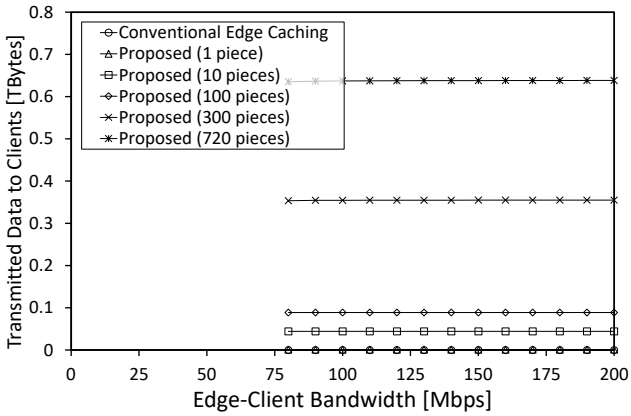


Figure 9: Edge-client bandwidth and the transmitted data to the clients

the number of the clients and each edge server respond to the requests from all the other edge servers.

We only show the total data amount transmitted to the clients because we confirmed that the total data amount transmitted to the edge servers is similar to this.

5.3 Influence of Edge-Client Bandwidth

A less communication bandwidth between the edge servers and the clients (edge-client bandwidth) cause more transmission overlaps, and thus, the interruption time can continue to increase with a higher probability. Therefore, we investigate the influence of the bandwidth between the edge servers and the clients.

Figure 8 shows the average interruption time under different client-edge bandwidth. The arrival interval is 30 [s]. The horizontal axis is the client-edge bandwidth, i.e., the bandwidth between each edge server and the clients. The vertical axis is the average interruption time. We can see that the average interruption times under our proposed scheme are almost the same when the edge-client bandwidth is larger than a certain value. Similar to the discussion for the previous subsection, this is because the transmissions does not overlap with others when the edge-client bandwidth is large. For the same reason as the previous subsection, the average interruption time under the conventional scheme is a little bit longer than that under our proposed scheme. Since the behavior between the edge servers and the clients are the same between

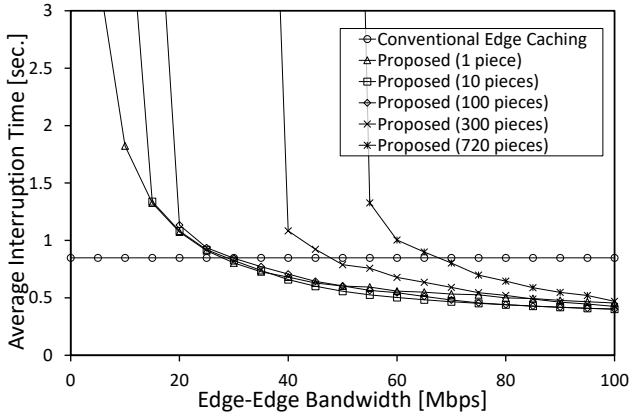


Figure 10: Edge-edge bandwidth and the interruption time

Table 1: Transmitted data to the clients under Figure 10.

Pre-cached pieces	0	1	10	100	300	720
Transmitted Data Amount	0	888 [Mbytes]	44.4 [GBytes]	88.8 [GBytes]	354 [GBytes]	636 [GBytes]

our proposed scheme and the conventional scheme, the edge-client bandwidth under that the interruption time diverges is the same, approximately 75 [Mbps].

Figure 9 shows the total data amount transmitted to the clients. Since the average interruption time diverges when the edge-client bandwidth excessively small, the lines stop at the smallest bandwidth under that the interruption time converges. The total data amount does not change even when the edge-client bandwidth changes because the data amount only depends on the number of the clients and the number of the pre-cached pieces.

5.4 Influence of Edge-Edge Bandwidth

A less communication bandwidth among the edge servers (edge-edge bandwidth) cause more transmission overlaps. Thus, the interruption time can continue to increase with a higher probability.

Figure 10 shows the average interruption time. The arrival interval is 30 [s]. The horizontal axis is the edge-edge bandwidth. The average interruption time under the conventional scheme is constant even when the edge-edge bandwidth changes because the edge servers do not redistribute the pieces in the scheme. The average interruption time under our proposed scheme decreases as the edge-edge bandwidth increases because the edge servers can faster redistribute the pieces to another edge server. Moreover, the average interruption time continue to lengthen when the edge-edge bandwidth is smaller than a certain value because the transmissions overlap.

The total data amount transmitted to the clients and the edge servers do not depend on the edge-edge bandwidth and these have a similar tendency as shown in Figs. 6 and 7. Therefore, we show the total data amount transmitted to the clients for the situation of Fig. 10 in Table 1. As shown in the table, the data amount increases as the number of the pre-cached pieces increases because the edge servers own more data. At the maximum case, i.e., the edge servers pre-cache all the 720 pieces, the data amount is 636 [GBytes] for 100 video data.

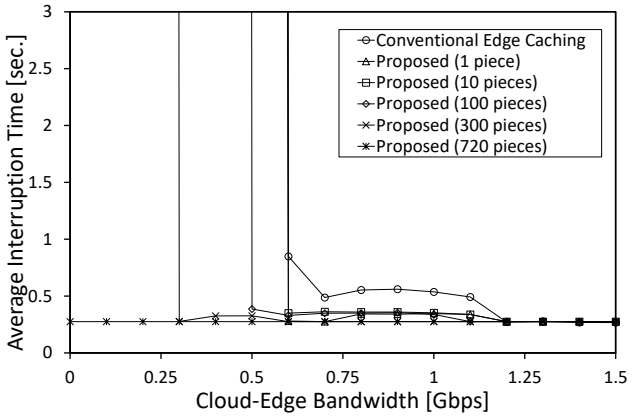


Figure 11: Cloud-edge bandwidth and the interruption time

5.5 Influence of Cloud-Edge Bandwidth

A less communication bandwidth between the video distribution server and the edge servers (cloud-edge bandwidth) cause more transmission overlaps. Therefore, we investigate the interruption time changing the cloud-edge bandwidth.

Figure 11 shows the average interruption time. The arrival interval is 30 [s]. The horizontal axis is the cloud-edge bandwidth. The average interruption time under the conventional scheme is longer than that under our proposed scheme because the bandwidth between the edge servers and the clients are not sufficient and the transmissions sometimes overlap with others as shown in Fig. 5. Similar to previous results, the average interruption time suddenly increases when the cloud-edge bandwidth is smaller than a certain value because the transmissions continue to overlap. The cloud-edge bandwidth under that the average interruption time converges is larger as the number of the pre-cached pieces increases because the edge servers receive less pieces from the video distribution server as the pre-cached pieces increase. Since the edge servers pre-cache all the pieces when they pre-cache 720 pieces, the average interruption time does not change even when the cloud-edge bandwidth changes in this case.

5.6 Influence of Edge Servers

The load on the edge servers can distribute as the system equips with a larger number of the edge servers. However, it takes much monetary cost as the system equips with a larger number of the edge servers. Therefore, we investigate the interruption time changing the number of the edge servers.

Figure 12 shows the result. The arrival interval is 30 [s] and the horizontal axis is the number of the edge servers. We can see that the average interruption time is almost constant under each scheme when the number of the edge servers is large because the transmissions do not overlap if the system equips with a sufficient number of the edge servers. Otherwise, the transmissions overlap and the interruption time continue to lengthen. For example, in the evaluated situation, the interruption time diverges when the number of the edge servers is less than 6. Therefore, it is better for the system to find the appropriate number of edge servers under that the interruption time converges.

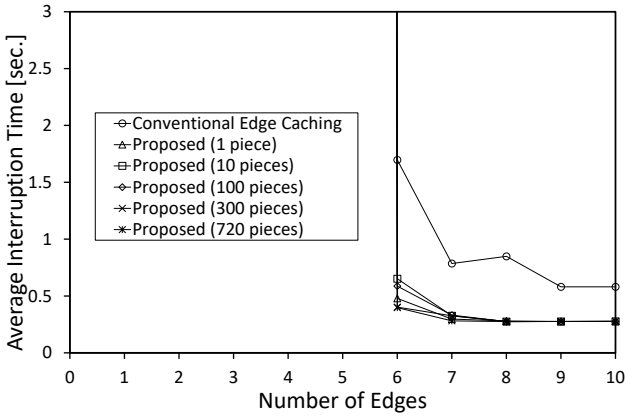


Figure 12: Number of the edges and the interruption time

5.7 Summary of Evaluation Results

We confirmed that the average interruption time did not largely depend on the number of the pre-cached pieces when the communication network was not congested, i.e., a longer arrival interval, a larger bandwidth (edge-client/edge-edge/cloud-edge). Meanwhile, the arrival interval or the bandwidths under that the system could work, i.e., the average interruption time converges, could be improved by pre-caching more pieces on the edge servers. This was because the transmissions did not overlap with others when the communication network was not congested. On the other hand, a larger number of pre-cached pieces causes more computational loads on the edge servers.

6 CONCLUSION

In this paper, we proposed a scalable VoD system on edge computing environments. Our proposed system adopted the pre-caching and the redistribution techniques. Our proposed system reduces the probability that the transmissions overlap with other transmissions and increases the maximum number of the clients of that interruption time converges. Our simulation evaluation revealed that our proposed system can transmit the video data more than the conventional scheme. Our proposed system can improve the shortest arrival interval under that the system can work by 26% compared with the conventional system in the simulated situation.

In the future, we will consider the popularity of the videos and adopt the dynamic caching technique to the edge servers. Moreover, we will consider the communication loads on the edge servers and the distribution server.

ACKNOWLEDGMENTS

This work was partially supported by JSPS KAKENHI Grant Numbers JP21H03429, JP18K11316, and by G-7 Scholarship Foundation.

REFERENCES

[1] R. Abuhadra and B. Hamdaoui, “Proactive In-Network Caching for Mobile On-Demand Video Streaming,” in Porc. IEEE International Conference on Communications (ICC), pp. 1-6 (2018).

[2] H. Feng, Z. Chen, H. Liu, and D. Wang, “Optimal Cache Placement for VoD Services with Wireless Multicast and Cooperative Caching,” in Proc. IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6 (2018).

[3] Y. Gotoh, T. Yoshihisa, and M. Kanazawa, Y. Takahashi, “A Broadcasting Protocol for Selective Contents Considering Available Bandwidth,” IEEE Transactions on Broadcasting, Vol. 55, No. 2, pp. 460–467 (2009).

[4] R. Coutinho, F. Chiariotti, D. Zucchetto, and A. Zanella, “Just-in-time proactive caching for DASH video streaming,” in Proc. Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), pp. 1-6 (2018).

[5] A.G. Sheshjavani, B. Akbari, and H.R. Ghaeini, “An Adaptive Buffer-Map Exchange Mechanism for Pull-based Peer-to-Peer Video-on-Demand Streaming Systems,” Springer International Journal of Multimedia Tools and Applications, Vol. 76, No. 5, pp. 7535–7561 (2016).

[6] Y. Zhang, C. Gao, Y. Guo, K. Bian, X. Jin, Z. Yang, L. Song, J. Cheng, H. Tuo, and X.M. Li, “Proactive Video Push for Optimizing Bandwidth Consumption in Hybrid CDN-P2P VoD Systems,” in Proc. IEEE International Conference on Computer Communications (INFOCOM), p. 2555-2563 (2018).

[7] R. Fratini, M. Savi, G. Verticale, and M. Tornatore, “Using Replicated Video Servers for VoD Traffic Offloading in Integrated Metro/Access Network,” in Proc. IEEE International Conference on Communications (ICC), pp. 3438-3443 (2014).

[8] E. Ghabashneh, S. Rao, “Exploring the Interplay between CDN Caching and Video Streaming Performance,” in Proc. IEEE Int’l Conference on Computer Communications (INFOCOM), p. 2555-2563 (2018).

[9] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, “Edge Computing Assisted Adaptive Mobile Video Streaming,” in IEEE Transactions on Mobile Computing, Vol. 18, No. 4, pp. 787-800 (2019).

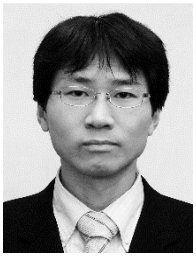
[10] S. Yang, Y. Tseng, C. Huang, and W. Lin, “Multi-Access Edge Computing Enhanced Video Streaming: Proof-of-Concept Implementation and Prediction/QoE Models,” in IEEE Transactions on Vehicular Technology, Vol. 68, No. 2, pp. 1888-1902 (2019).

[11] J.-P. Hong, W. Choi, “User Prefix Caching for Average Playback Delay Reduction in Wireless Video Streaming,” IEEE Transactions on Wireless Communications, Vol. 15, Issue 1, pp. 377-388, 2015.

[12] RFC 8216, HTTP Live Streaming, https://tools.ietf.org/html/rfc8216 (2017).

(Received: November 8, 2021)
(Accepted: December 24, 2021)

Satoru Matsumoto received his Diploma's degrees from Kyoto School of Computer Science, Japan, in 1990. He received his Master's degree from Shinshu University, Japan, in 2004. From 1990 to 2004, he was a teacher in Kyoto School of Computer Science. From 2004 to 2007, he was Assistant Professor of The Kyoto College of Graduate Studies for informatics. From 2007 to 2010, he was Assistant Professor of Office of Society Academia Collabo-ration, Kyoto University. From 2010 to 2013, he was Assistant Professor of Research Institute for Economics & Business Administration, Kobe University. From 2015 to 2016, he was a specially appointed assistant professor of Cybermedia Center, Osaka University. From April 2016 to September 2016, he became a specially appointed researcher. Since November 2016, he became an assistant professor. His research interests include distributed processing systems, rule-based systems, and stream data processing. He is a member of IPSJ, IEICE, and IEEE



Tomoki Yoshihisa received the Bachelor's, Master's, and Doctor's degrees from Osaka University, Osaka, Japan, in 2002, 2003, 2005, respectively. Since 2005 to 2007, he was a research associate at Kyoto University. In January 2008, he joined the Cybermedia Center, Osaka University as an assistant professor and in March 2009, he became an associate professor. From April 2008 to August 2008, he was a visiting researcher at University of California, Irvine. His research interests include video-on-demand, broadcasting systems, and webcasts. He is a member of the IPSJ, IEICE, and IEEE.

Regular Paper

Continual Lengthening of Titles: Implications for Deep Learning Named Entity Recognition

Yukihisa Yonemochi[†] and Michiko Oba[‡]

[†]Graduate School of Future University Hakodate, Japan
[‡]Future University Hakodate, Japan
{g3119008, michiko}@fun.ac.jp

Abstract - The objective of this paper is to highlight a problem of deep learning (DL) named entity recognition (NER) for titles of various works. Extracting information from a text input is an important task for interactive text interfaces such as chatbots and voice interfaces. In the field of natural language processing, NER is known as a type of information retrieval. Most of the latest NER methods utilize deep learning with the highest accuracy. As the standard, in most cases, only word sequences have been used as input features. However, these methods have a problem in recognizing unknown longer compound words. Longer titles can be found in titles of works such as novels, manga, animation, and movies. We verified this phenomenon using the following three aspects: First, we verified how standard DL NER has a problem with longer titles. Second, assuming that the addition of lexical features improves the performance, we verified its effectiveness. Third, we verified that such longer titles are distributed within actual real-world titles. Herein, we report the results of the verification and suggest the necessity of considering countermeasures.

Keywords: Named entity recognition, deep learning, feature selection

1 INTRODUCTION

The extraction of proper nouns or unique names such as movie titles from input text is an important part of natural language processing (NLP) when developing interactive systems such as chatbots or voice interfaces. Recognizing the named entity of an artifact, as defined through Message Understanding Conference (MUC) [1] and Information Retrieval and Extraction Exercise (IREX)[2], is the standard method for extracting proper nouns. This is known as the NER in the NLP area. NER has two tasks: tagging and disambiguation. Tagging is the process of generating tag data for the start and end positions of a fragment in a text. Disambiguation involves choosing the correct data from several candidate types.

Table 1 shows an example text that has the fragment "The Bridge on the River Kwai," which can be the title of either a novel or a movie.

B-NOVEL and B-MOVIE indicate the beginning of the title of a novel and a movie, respectively. Likewise, I-NOVEL and I-MOVIE show a range of titles. In addition, O refers to OTHER. This is called the BIO-style label, which is typically used for NER tasks. The NER process must first obtain the correct range, and in this example, the range is 2-7 along with the index number. The NER is then disambiguated, choosing between novels and movies. In this example, B-MOVIE and I-MOVIE are the correct choices.

Although the text initially seems to be saying, "I saw the bridge," it actually states, "I saw the movie." Knowledge of the title can help us reconcile this meaning. Next, we need to guess whether it is a NOVEL or MOVIE. Humans can intuitively recognize that the title can be a movie utilizing the verb "saw."

In NER systems, statistical methods are applied to recognize the range and choose the correct type. In recent years, some DL methods have obtained high NER scores. NLP-progress [3] reported accuacy of more than 90% for several datasets.

We encountered a problem in which unknown, longer unique names often cause errors in the recognition process. In this context, "unknown" indicates a unique name that was not included in the set of training data, and "longer" means a unique name that has several more words of different types than the known name. Such unknown, longer names are commonly seen in novel, manga, cartoon, or movie titles. We refer to such names as "UnknownLonger" throughout this paper. We can assume that some UnknownLonger names will not be correctly extracted in the actual system under the following situation. DL is trained using texts including a list of existing titles. It can extract the existing titles from an input text with high accuracy. However, if a new longer title is announced, it is not correctly extracted.

A typical method for evaluating the accuracy randomly separates a dataset into subsets of 70% for training and 30% for testing. It can hide the problem from the evaluation score because most titles are included in the training data.

The objective of this study is to determine whether it is necessary to add lexical information to a feature when using DL NER in an interactive interface. In this paper, we verify the effectiveness of adding a lexical feature to DL NER for extracting UnknownLonger names from texts. The effectiveness of the technique was measured through experiments conducted using our original dataset with manga, novel, cartoon, and movie titles. For comparison purpose, Japanese and English titles are used because longer titles occur more frequently in Japanese. The dataset was generated using several spoken text patterns with real titles. The text pattern was manually created, and title names were collected from Wikidata

ISSN1883-4566 © 2022 - Informatics Society and the authors. All rights reserved.

Table 1: Input and Label of NER

index	0	1	2	3	4	5	6	7	8	9
input	**I**	**saw**	**The**	**Bridge**	**on**	**the**	**River**	**Kwai**	**yesterday**	**.**
Label as Movie	O	O	B-MOVIE	I-MOVIE	I-MOVIE	I-MOVIE	I-MOVIE	I-MOVIE	O	O

[4].

In this paper, we verify the problem for the following three aspects.

Verification 1: Existence of the problem
The DL model is prepared, and it is confirmed whether it works well. (Test-1)
Problems are intentionally created by changing the data selection. (Test-2)

Verification 2: Effectiveness of adding lexical features
The DL job is modified to add lexical features and confirm whether the accuarcy improves. (Test-3)

Verification 3: Investigation into the distribution
The distribution of the title lengths is visualized based on the time and type. The difference in the length of the work titles are compared based on the nationality and type.

Detailes of the verification procedure are described in the following sections.

2 DEEP LEARNING NAMED ENTITY RECOGNITION

DL methods have recently been utilized for the NER. NLP-progress reports have shown that the top NER rankings for the CoNLL [5] dataset are CNN [6], RNN [7]+CRF, LSTM [8], Bi-LSTM [9], BERT [10].

The language model GPT-3 [21] by OpenAI [20] or GPT-J [24] by EleutherAI [23] are getting as much attention as BERT. They have numerous more parameteres than BERT and have reported a better performance. They can also be applied for NER too [22][25].

These DL methods are suitable for time-series data. The same set of features are used for the NER, regardless of the method applied. Simultaneously, label data indicating the tag are prepared, which are referred to as "tagged," "annotated," or "labeled." The sequence of tokens is the explanatory variable and the sequence of labels is the objective variable.

Figure 1 shows an example of how DL NER processes input text and labels. DL NER is processed through following steps.

Tokenization
First, the input text needs to be tokenized as a sequence of words or morphemes. Latin-derived languages can be tokenized using space characters, and the Japanese language can be tokenized using a Japanese tokenizer [11].

Encoding
Once the text has been tokenized, the sequence of tokens is translated into vector data, which is called a tensor. The method used to generate a distributed representation has a strong influence on the NER results. The distributed representation of tokens is used as the input feature for DL tools to improve the accuracy. The labels are also translated into tensors, but in simple through a one-hot vector.

Training
The deep learning model is trained to take a word tensor as the input and output a label tensor.

Most studies have used large tagged corpora to create a trained model. However, in [12] and [13], the authors proposed utilizing Wikidata to create large training datasets. In so doing, they support a wide range of vocabulary but still use the same features and labels.

3 DEFINING THE PROBLEM

As mentioned in section 2, standard NER methods use only the surfaces of the word and word sequences as the input feature. This can cause tagging errors in the UnknownLonger titles.

Consider the classification model for the NER, which is trained using training data. The underlined part is the title and must be extracted as an argument for certain applications.

- I want to watch **Star Wars** next week.
- When will **Harry Potter** be released?

The number of words in the titles are both 2, and all of the words are nouns. Using this example, unknown titles such as "Star Trek" can be extracted as a title even if it was not included in the training data. This is called unknown word extraction.

- I want to watch **Star Trek** next week.

However, the trained model failed to extract the title when it was not in the training data and was longer. See the following example:

- I saw **The Bridge on the River Kwai** yesterday.

input	feature tensor						label	label tensor			
I	0	1	0	1	0	0	O	1	0	0	0
saw	0	0	1	1	1	0	O	1	0	0	0
the	1	0	1	0	1	0	B_MOVIE	0	1	0	0
bridge	0	1	1	1	0	0	I_MOVIE	0	0	1	0
on	1	1	1	0	1	0	I_MOVIE	0	0	1	0
the	0	1	0	1	1	1	I_MOVIE	0	0	1	0
river	1	1	1	1	0	1	I_MOVIE	0	0	1	0
kwai	0	1	0	1	1	0	I_MOVIE	0	0	1	0
yesterday	1	0	1	1	1	0	O	1	0	0	0
.	0	0	0	0	1	1	O	1	0	0	0

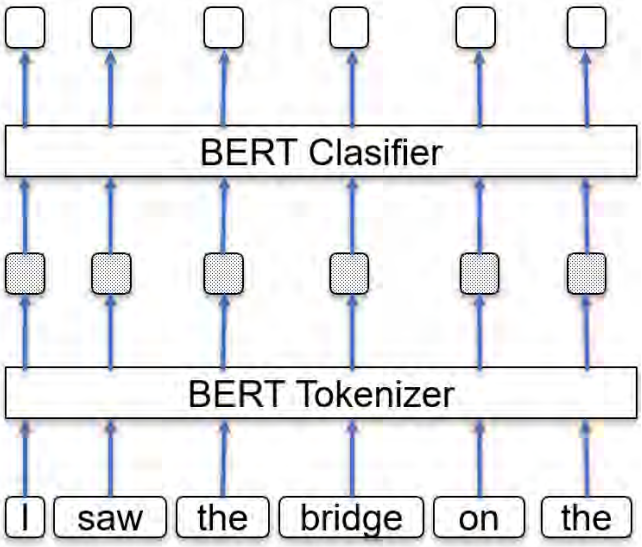


Figure 1: DL NER of a title

This title has six words, i.e., article-noun-preposition-article-noun-noun. This is more complex and longer than that of the examples mentioned above, which can confuse the classification module. This type of situation poses a problem. Artifacts such as manga (comic books), animation (cartoons), novels, and movies are being added daily. However, we cannot train classification models daily, because the computing cost is high.

We conducted a verification using datasets that were arbitrarily divided by the number of words in the target title. Shorter titles were used for the training model, and the remaining longer titles were used for testing.

The existence of this problem in NER tasks was verified through the experimental results described in Section 6.

4 ADDING A BOOLEAN FEATURE

To improve the accuracy of the DL NER of UnknownLonger titles, we propose injecting vocabulary information into the feature. This method, which have previously proposed for improving the NER for gazetteer [14] adds a feature, which is simple a Boolean flag, to the input feature. A flag indicates whether a series of words can be found in the database. In the training data, flags are generated from the labels. In the test data, flags can also be generated from the labels. In the production input data, the flags can be added by searching the word sequences in the database. **Figure 2** shows an example of the addition of flags to a feature. Column "b" indicates the binary feature flag adding vocabulary information to the input.

In this example, "The Bridge on the River Kwai" is the title of a movie. Flags can be generated from the B-MOVIE and I-MOVIE labels during training and testing. During the production time, the flag must be added by searching the name from a database. More precisely, the flags are added to the tensor matrix after the words are translated into a distributed expression.

5 EXPERIMENT

For the first and second verifications, we conducted an experiment using our newly prepared test data.

The purpose of this experiment is to verify whether a typical NER method has a problem with UnkownLonger. Adding lexical features is an effective way to improve the problem.

In desiging the experiment, we need to make sure that the typical NER model does indeed have the UnkonwLonger problem. For this purpose, we prepare a typical NER model that works well on the prepared dataset and we intentionally create the UnknownLonger problem. To intentionally recreate a problems, we will focus on how to split the test data to measure the performance. In measuring the performance of a typical NER model, the test data are divided randomly. In this case, the UnknownLonger problem is hidden. In our experiment, we recreate the UnknownLonger problem by splitting the test data only for sentences that contain long entity names as the test data. This must recreate the problem, which we will observe. Next, we will confirm the improvement by adding lexical features described in Chapter 4.

Through a series of experiments, we aimed to confirm that an NER model with high accuracy becomes less accurate on UnknownLonger, and that the lexical features improve the accuracy significantly. The steps of the experiment are as follows:

(Test-1) A typical valid deep-learning model is prepared.
(Test-2) It is confirmed whether the problem actually occurs.
(Test-3) Whether the problem can be improved by adding a lexical feature is then confirmed.

As the first verification, the result of (Test-1) becomes (Test-2) owing to the influence of UnkownLonger, and as the sec-

input	feature tensor						b	label	label tensor			
I	0	1	0	1	0	0	0	O	1	0	0	0
saw	0	0	1	1	1	0	0	O	1	0	0	0
the	1	0	1	0	1	0	1	B_MOVIE	0	1	0	0
bridge	0	1	1	1	0	0	1	I_MOVIE	0	0	1	0
on	1	1	1	0	1	0	1	I_MOVIE	0	0	1	0
the	0	1	0	1	1	1	1	I_MOVIE	0	0	1	0
river	1	1	1	1	0	1	1	I_MOVIE	0	0	1	0
kwai	0	1	0	1	1	0	1	I_MOVIE	0	0	1	0
yesterday	1	0	1	1	1	0	0	O	1	0	0	0
.	0	0	0	0	1	1	0	O	1	0	0	0

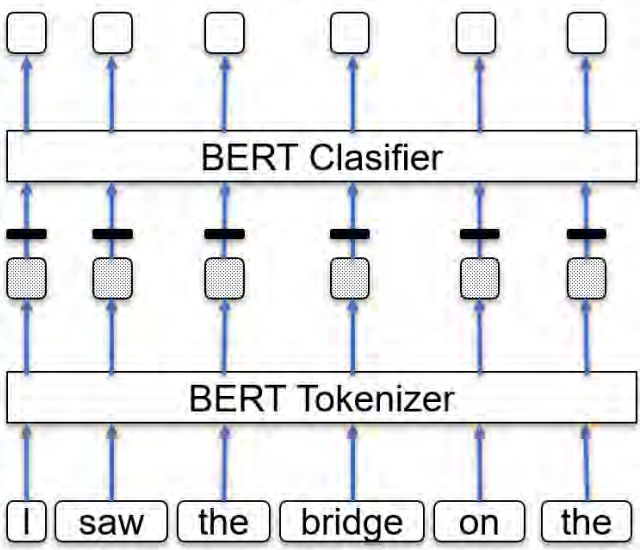


Figure 2: Adding lexical feature for long title

ond verification, (Test-2) extends to (Test-3) owing to the effect of the lexical feature. A more detailed description of the experiment environment is provided in this section.

5.1 Test Dataset

Because we have a problem with long titles, the dataset was specially designed. The dataset was generated by combining the following two types of text:

- spoken statement patterns
- titles

We chose the titles of mangas, novels, cartoons, and movies as the target areas. The spoken statements were manually created with 20 statements for each area. List 1 shows examples of statements in the movie context.

List 1. Statement patterns

I can't wait until x is released
Will you go see x next week?
I need to buy a ticket for x
The movie x will be coming out in theaters
x world premiere

Titles were collected from each area of the Wikidata. The work titles on Wikidata cover a wide range of genres, from the very old to the very new. All the necessary information for aggregation is available, such as language, year of release, and nationality. It is updated on a daily basis, making it extremely worthwhile to connect to the actual interactive interface. Of course, if there is a highly comprehensive database of titles of wrok, we can use it for our experiments. However, Wikidata is free, easy to connect, and convenient for scientific experiments. For these reasons, we used Wikidata to collect data for our experiments.

When collecting titles from Wikidata, the instance_of(P31) property is used. For example, when we collect movie titles, the movie item on Wikipedia is wd:Q11424. Wikidata can be searched using a SPARQL[15] query. Code 1 shows the SPARQL query used to collect movie items for US movie titles, with ID and label names.

Code 1. SPARQL query for US movie titles

```
SELECT ?item ?itemLabel
WHERE {
  ?item wdt:P31 wd:Q11424;
  wdt:P495 wd:Q30.
  SERVICE wikibase:label
   {bd:serviceParam wikibase:language "en". }
}
LIMIT 500
```

In the query, P495 denotes the "country of origin," Q30 represents the "USA." The SERVICE argument limits the items to those that have a label for locale "en." Therefore, the query collects English labels of instances of movies with the property, i.e., country of origin, being the USA. For the Japanese work labels from Japan, Q17 is used instead of Q30.

The query times out if a LIMIT is not included. The query limits the number of lines to 500. List 2 shows examples of English movie titles from the United States.

List 2. Examples of movie titles

The Brain That Wouldn't Die
Puppet Master: The Legacy

Puppet Master 4
The Lusty Men
Curse of the Puppet Master

Table 2 shows the list of items used in the SPARQL query to collect titles from Wikidata.

Table 2: List of id and target labels of Wikidata

Type	instance_of(P31)	Japanese	USA
Manga	wd:Q21198342	MANGA	
TV Animation	wd:Q63952888	ANIME	
Written work	wd:Q47461344		NOVEL
Animated series	wd:Q581714		ANIME
Movie	wd:Q111424	MOVIE	MOVIE

Different sets of items were used in both English and Japanese. Movies, animations, and manga are in Japanese. Because mangas (comic books) were originally created in Japan, it is obvious that TV animation shows and movies will be created from them. However, we tried to collect an instance_of "comic book series" for English, and only 172 items were collected, which is an insuffcient number. Therefore, we chose "written work" (Q47461344) and more than 6,000 instance items were collected. This indicates that they were classified as novels.

By using patterns and titles as examples, and combining the first pattern and first title, the first line of the dataset is as follows:

"I can't wait until *The Brain That Wouldn't Die is released.*"

We used 500 titles and 20 statements for each of the three areas, with a total of 30,000 lines of statements for English and Japanese works.

5.2 Tools

The experiment environment used the existing components. We chose the BERT Tokenizer which is a state-of-the-art technology for NER. **Table 3** lists the environments and tools used during the experiment.

We utilized Python3.0 and TensorFlow [16] 2.0 on Google Colaboratory [17]. The input texts are tokenized using space characters for the English titles, and using Japanese Tokenizer Janome for Japanese titles. The distributed representation is applied as the tensor input feature, and the BertTokenizer with a pre-trained model bert-base-uncased is used for the English titles. In addition, BertJapaneseTokenizer with a pretrained model bert-base-japanese-whole-word-masking[19], which is published by Tohoku University, is used for Japanese titles. TFBertForTokenClassification was used for classification.

5.3 Experimental Steps

The experiments were conducted through the following steps:

1. Dataset preparation
   - Prepare statement patterns
   - Collect titles of written work (USA), including manga (Japanese), animation, and movie from Wikidata
   - Merge, combine patterns and titles, and generate statements and labels
2. Prepare standard NER job
   - TensorFlow
   - BERT Tokenizer (English / Japanese), BERT Classification
3. Test the standard method (Test-1)
   - Randomly divide the dataset into a 7:3 ratio
   - Confirm whether the NER model works well (high score)
4. Verify UnknownLonger problem (Test-2)
   - Divide dataset into shorter and longer titles
   - Verify that the problem exists (lower score)
   - This result is the baseline
5. Verify the effectiveness of the lexical feature (Test-3)
   - Add lexical Boolean feature to the input
   - Compare with the baseline (improve the score)

Table 3: Environment and tools applied

Computing environment	Google Colaboratory
Platform	Python3.0
Tokenized by	space character (for English) Janome (for Japanese)
DL Tool	TensorFlow 2.0
Distributed Representation	BertTokenizer BertJapaneseTokenizer
Classifier	TFBertForTokenClassification
Pretrained model	bert-base-uncased(for English) cl-tohoku/bert-base-japanese-whole-word-masking(for Japanese)
batch size	32

5.4 Dividing database based on the number of words

Figure 3 shows how to intentionally create a problem.

In Test-1, similar to the typical method of testing the accuracy of deep learning models, the dataset is randomly devided into a ratio of 70% to 30%. In Test-2, the set of titles is divided into training and testing data using the number of words in each title. Designating the number of words in the title by N, the following were used as the training data for the experiments.

- For English text: $N < 3$.
- For Japanese text: $N < 4$.

Table 4: Experiment results(test-1,2,3)

language	division	count of data training	testing	lexical feature	Epoc	F1 score NOVEL	ANIME	MOVIE	weighted avg.
English	random 7:3	21264	9114		20	0.997	1.000	1.000	0.999
	train w/N <3	21198	9180		15	0.7987	0.7593	0.8111	0.7889
	train w/N <3	21198	9180	added	24	**1.0000**	**0.9998**	**0.9959**	**0.9988**
language	division	training	testing	feature	Epoc	MANGA	ANIME	MOVIE	weighted avg.
Japanese	random 7:3	21264	9114		20	0.997	0.998	0.986	0.994
	train w/N <4	21771	8607		6	0.5837	0.6212	0.3045	0.4605
	train w/N <4	21771	8607	added	47	**0.9932**	**0.9820**	**0.9734**	**0.9803**

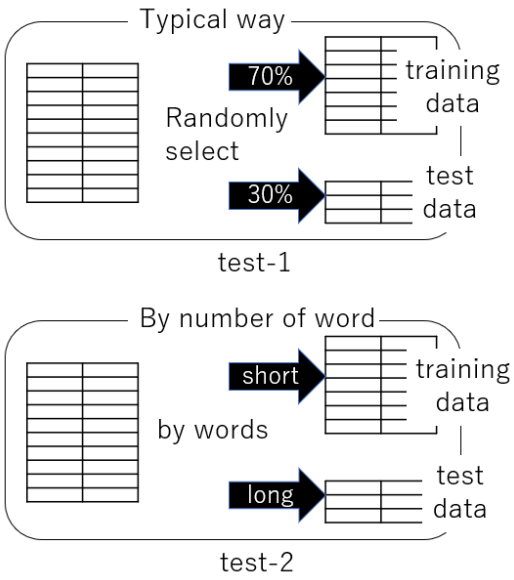


Figure 3: Intentionally Re-create the Problem

We chose these numbers to split the dataset into a ratio of approximately 7:3. This indicates that English titles are shorter than Japanese titles.

6 RESULTS AND DISCUSSION

Table 4 presents the experimental results. The experiment results, listed in **Table 4**, are described in this section. The lines for each language in the results of Tests 1, 2, and 3 are discussed in Section 5.3.

6.1 Preparation of Classification Job

The first lines for each language in **Table 4** show the results of Test-1, where the training job using the standard method randomly divided the dataset into 70% for training and 30% for testing. The weighted average F1 score was 99.9% for English and 99.4% for Japanese. These results are excellent, and we thus assume that the dataset was prepared arbitrarily and is too easy for the classification task. However, it shows that the prepared job works for this dataset.

6.2 Verifying the Existence of the Problem

The second line for each language shows the results of Test-2, the training job using the traditional method of dividing the dataset using the number of words in the title as the criterion. The test data lines with a title having less than three words for English and four words for Japanese were used as the training data. The remaining data were used as the test data. The weighted average F1 score was 78.9% for English and 46.1% for Japanese. From these results, we can verify that the accuracy of the classification decreases when testing only UnkownLonger titles. This difference indicates that Japanese texts are more strongly affected by UnknownLonger titles than English texts, which was considered the baseline.

6.3 Verifying the Effectiveness of the Lexical Features

The third line for each language in **Table 4** shows the results of Test-3, i.e., the training job with an added lexical feature that divides the dataset according to the criteria of the number of words in the title. The data division criteria were the same as those for Test-2. The weighted average F1 score was 99.9% for English and 98.0% for Japanese. This indicates that adding lexical features is effective for this job and the dataset.

6.4 Discussion and the Mechanism

Figure 4 shows a visualization of the results of Tests-1,2, and 3 for Verifications 1 and 2.

When testing UnknownLonger, we can observe a clear decrease in accuracy (Verification1,) and we can observe that it improves by adding lexical features (Verification2.)

From the experiment, we confirmed that the problem exists and we can improve it by adding a lexical feature. Now discuss the mechanism. A typical NER has only a sequence of words as input. The names labeled during training are stored in the trained model as a lexicon, but only peripheral words are used for an unknown word estimation. UnknownLonger names often appear like excerpts from sentences, and they may include words that are included in the periphery. In such cases, the boundary between the name and the peripheral words is extremely vague. The lexical features as a Boolean vector that strongly suggest the possibility of a boundary between names and peripheral words, and the discriminator re-

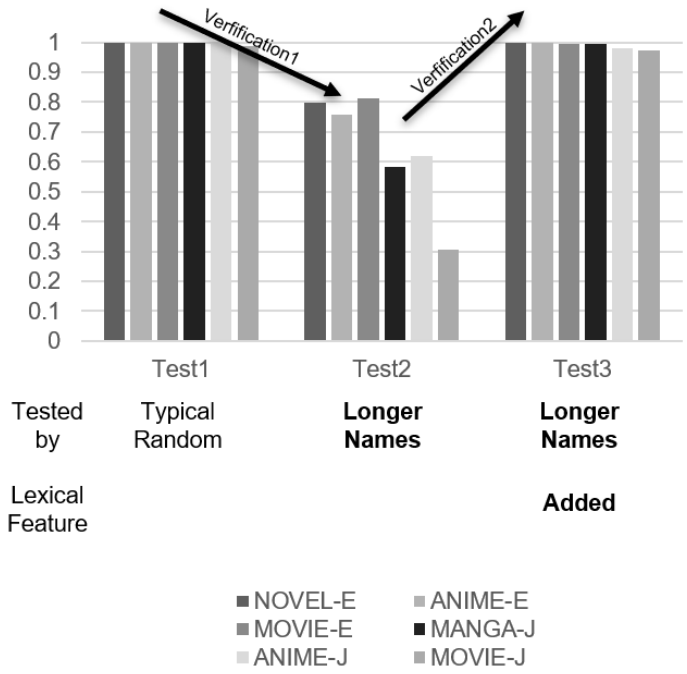


Figure 4: Verification 1 and 2

sponds strongly to them, which is considered to be a significant contribution to solving the problem.

7 VERIFYING THE DISTRIBUTION OF LONGER TITLES

As a third verification, we investigated the distribution of the length of the titles. We assume that not only the number of words, but also the types of words that make up the name of an entity, affect the recognition. The types of words are known as parts of speech. Parts of speech include nouns, verbs, adjectives, adverbs, conjunctions, and articles. Words that fall into different parts of speech, such as nouns, verbs, adjectives, and adverbs, should be placed far apart in a distributed representation space by the BERT Tokenezer. UnknownLonger names are often composed of many parts of speech, and the type of part of speech is a matter of interest.

A POS analysis was studied earlier. In the experiments conducted in this study, we use OpenNLP for English names and a Japanese morphological analyzer for Japanese names. The words and POS were counted for each title. In English titles, words are separated by space characters, and the POS is tagged using Apache OpenNLP [18]. In Japanese titles, words are separated using the Japanese Tokenizer Janome, and the POS is tagged at the same time.

Table 5 shows an example of the POS of the title of the work, as analyzed by Apache OpenNLP.

Table 5: Example POS

The	Bridge	on	The	River	Kwai
DT	NNP	IN	DT	NNP	NNP

In this example, DT is a determiner, NNP is a singlar proper noun, and IN is a preposition or subordinating conjunction.

There are six words. In addition, there are three types of POS, i.e., articles, nouns, and prepositions.

The number of words and the POS are summarized by year to confirm the trend of these lengths. The number is also summarized by these work types to confirm which type has a long name.

7.1 Increasing length

Figure 5 shows a summary of the published year. From the visualization, the trend in the length of the work titles increases yearly. The trend is the same for the main title and the subtitles in both Japan and the US. Thus, we verified that the length of work titles increased annually. This means that UnknownLonger titles may appear after the recognition model is trained.

7.2 Longer Titles for Japanese Manga and Movie

Figure 6 shows the distribution of the number of words and POS based on the type of work. **Table 6** shows the independent t-test results for each work type. The t-test calculations confirmed that Japanese manga and movie titles are significantly longer than English titles. This confirms that the accuracy of the results for the Japanese manga and movie titles in **Table 4** was strongly affected.

8 CONCLUSION AND FUTURE WORK

We verified that the accuracy of the BERT classifier is affected by the length of unknown words and can be recovered by adding a lexcal feature. As indicated by the existing titles of various works, Japanese manga and movie titles are longer than English titles. The lengths of such titles are increasing every year. It was therefore suggested that measures such as adding lexical features are needed to improve the accuracy of identifying UnknownLonger titles. Particular attention should be paid to the Japanese titles of manga and movies.

We assume that GPT-3 and GPT-J have the same problems and rather exacerbate the problem in terms of training costs. We will also need to make sure that GPT-3 and GPT-J have the same problem and that the proposed method is effective.

As an assumption, adding lexical features has an unintended effect on short names. If the title of the database is only a single common word, it may be labeled a title whenever the word is used in the input text. We need to study how to work around this problem to utilize lexical features.

In a text interface, people do not accurately or perfectly input long names. Abbreviations or shortened names are used. In the future, we will study the tendency of people to abbreviate or shorten long titles, and devise a method for recognizing shortened names from input text using lexical information.

9 ACKNOWLEDGEMENT

We thank Google LLC for providing the Google Colaboratory used in our experiment. We would like to thank Editage

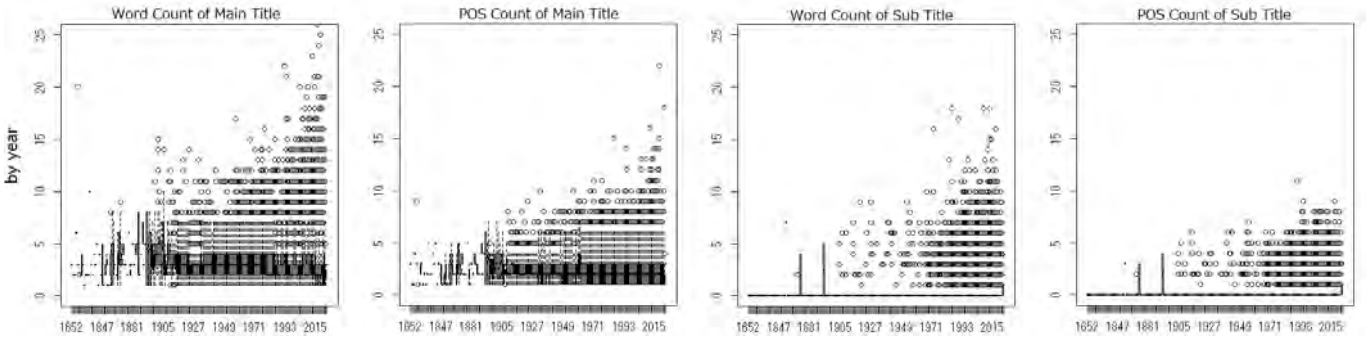


Figure 5: Count of Words and POS of titles by year

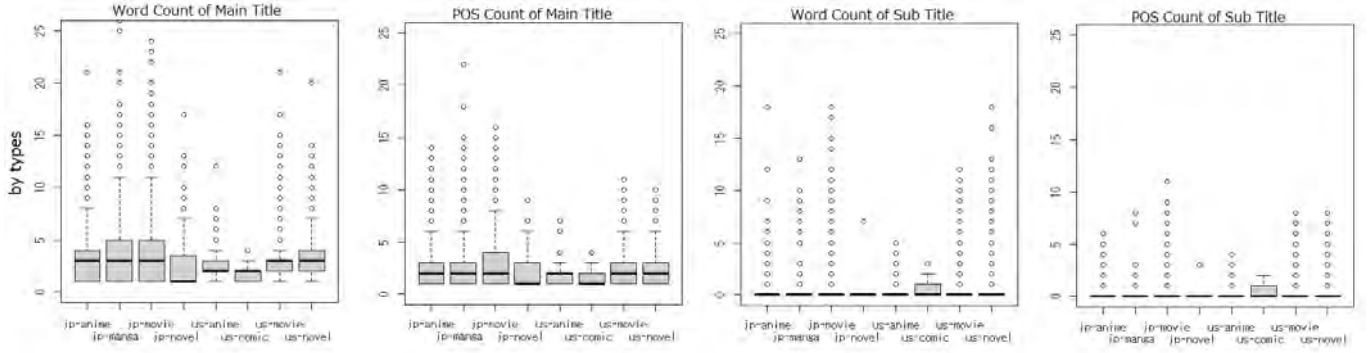


Figure 6: Count of Words and POS of titles by types

(www.editage.com) for English language editing.

REFERENCES

[1] N. Chinchor and E. Marsh. MUC-7 information extraction task definition, In Proceeding of the Seventh Message Understanding Conference (MUC-7), Appendices, pp. 359-367 (1998).

[2] S. Sekine and H. Isahara. Irex: Irie evaluation project in Japanese. In Proceedings of the 13th International Conference on Language Resources and Evaluation (LREC), pp. 1977–1980 (2000).

[3] S. Ruder. Nlp-progress. London (UK): Sebastian Ruder (accessed 2020-01-18). https://nlpprogress.com (2020).

[4] F. Erxleben, M. Günther, Markus Krötzsch, Julian Mendez, and Denny Vrande ˇci ´c. Introducing wikidata to the linked data web. In Proceedings of International Semantic Web Conference (ISWC), pp. 50–65. (2014).

[5] ACL SIGNLL. The signll conference on computational natural language learning. https://conll.org/ (2020).

[6] Y. LeCun et al. LeNet-5, Convolutional Neural Networks. URL: http://yann.lecun.com/exdb/lenet, Vol. 20, No. 5, p. 14 (2015).

[7] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, Vol. 79, No. 8, pp. 2554–2558 (1982).

[8] S. Hochreiter and J̈ Schmidhuber. Long shortterm memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780 (1997).

[9] M. Schuster and K. K Paliwal. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, Vol. 45, No. 11, pp. 2673–2681 (1997).

[10] J. Devlin, M.-W. Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[11] Janome. https://mocobeta.github.io/janome/ (2020).

[12] A. L. F. Shanaz and R. G. Ragel. Named entity extraction of wikidata items. In 2019 14th Conference on Industrial and Information Systems (ICIIS), pp. 40–45. IEEE, (2019).

[13] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 708–716, (2007).

[14] S. Magnolini, V. Piccioni, V. Balaraman, M. Guerini, and B. Magnini. How to use gazetteers for entity recognition with neural models. In Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5), pp. 40–49 (2019).

[15] w3c. Sparql query language for rdf. (2008).

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learn-

Table 6: t-test Movie Titles between Japan and US(P=0.05)

	ANIME		MANGA		MOVIE		NOVEL	
	ja-JP	en-US	ja-JP	en-US	ja-JP	en-US	ja-JP	en-US
Mean	3.05	2.74	**3.49**	2.04	**4.00**	2.78	2.83	3.17
Variance	5.89	2.57	10.45	1.08	10.76	1.82	7.59	2.71
Count	1225	139	1295	26	6978	65371	192	7153
t	2.00		6.56		31.03		-1.72	
P($T \leqq t$)	0.023		6.24E-08		7E-199		0.043	

ing. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283 (2016).

[17] E. Bisong. Google colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64, (2019).

[18] Apache, https://opennlp.apache.org/ (2021).

[19] Tohoku University, cl-tohoku models, https://huggingface.co/cl-tohoku (2021)

[20] OpenAI, https://openai.com/ (2021)

[21] Floridi, L., Chiriatti, M., GPT-3: Its Nature, Scope, Limits, and Consequences., Minds & Machines 30, 681–694 (2020), https://doi.org/10.1007/s11023-020-09548-1(2021)

[22] R. Ma, Using GPT-3 for Named Entity Recognition, https://ricky-ma.medium.com/using-gpt-3-for-named-entity-recognition-83a95389408e (2021)

[23] EleutherAI, https://www.eleuther.ai/ (2021)

[24] EleutherAI, Mesh Transformer JAX, https://github.com/kingoflolz/mesh-transformer-jax/ (2021)

[25] A. Diaz, NLP – What happens in Entity Extraction and its value, https://www.marscrowd.com/blog/text/nlp-entity-extraction-and-its-value/ (2021)

(Received: November 3, 2021)
(Accepted: February 10, 2022)

Yukihisa Yonemochi Graduated from Gunma National College of Technology, the Department of Mechanical Engineering in 1987. In the same year, he joined IBM in the area of software product support, selling, marketing, and research. From 2012, he was a lecturer at Future University Hakodate. In 2015, he joined Honda Research Institute Japan as a manager. In 2020, he founded Pandrbox. He is a member of the Information Processing Association of Japan (IPSJ), Japanese Society for Artificial Intelligence (JSAI), and The Association for Natural Language Processing (ANLP).

Michiko Oba Michiko Oba received a PhD. in engineering from Osaka University, Japan, in 2001. She worked in the Systems Development Laboratory and the Software Division of Hitachi Ltd. from 1982. She is currently a professor in the Department of Media Architecture, Future University Hakodate, Japan. Prof. Oba is a member of IEEE Computer Society, the Information Processing Society of Japan (IPSJ), and the Institute of Electrical Engineers of Japan (IEEJ). She became a Council Member of Science Council of Japan in 2020. She became a IPSJ fellow in 2020.

Regular Paper

Assistant Devices for Presentation of Distinctive Viewers' POV in 360-degree Internet Live Broadcasting

Masaya Takada* and Yoshia Saito*

*Graduate School of Software and Information Science, Iwate Prefectural University, Japan
m-osawada@iwate-jh.ed.jp, y-saito@iwate-pu.ac.jp

Abstract –360-degree Internet live broadcasting has been widespread in Internet live broadcasting services such as YouTube. However, there is an issue in the 360-degree Internet live broadcasting. The broadcaster cannot be aware of viewers' point of view (POV) because the 360-degree Internet live broadcasting uses an omnidirectional camera and the camera lens do not show the viewers' POV unlike conventional web cameras. The role of gaze information in remote communication is very important, as it shows the focus of the conversation and the object of interest. If the broadcaster cannot be aware of the viewers' POV, it is difficult to perform smooth communication between the broadcaster and the viewers. To solve this issue, we have studied an analysis algorithm of viewers' POV in 360-degree Internet live broadcasting. The algorithm used characteristics about the viewers' viewing behavior and could detect distinctive POV which represented viewers' interests. In the previous research, the distinctive viewers' POV were simply presented as red circles on the equirectangular video using a laptop PC for the broadcaster. The presentation method was not easy to comprehend and the use of a laptop PC would increase the risk of accidents. In this paper, we propose three assistant devices for presentation of the distinctive viewers' POV in 360-degree Internet live broadcasting. The first one is a belt-type device which presents directions of the POV using vibration motors. The second one is a LED-type device which presents directions of the POV using LED light sources cylindrically. The third one is a robot-type device which presents directions of the POV using movement of the robot's head. We developed the three assistant devices and conducted an evaluation experiment. From the experiment, we found the broadcasters did not prefer assistant devices in visual form and they had a positive impression of the robot-type device.

Keywords: 360-degree Internet live broadcasting, Viewers' POV, Assistant devices

1 INTRODUCTION

In recent years, many people use Internet live broadcasting services. In the Internet live broadcasting services, the viewers can enjoy real-time communication with the broadcaster. Besides, YouTube started a 360-degree Internet live broadcasting service which supports omnidirectional cameras from 2016. It enables anyone to easily use the 360-degree Internet live broadcasting service now.

In the 360-degree Internet live broadcasting, a broadcaster takes a 360-degree video using an omnidirectional camera and distributes it to viewers in real-time via the Internet. The broadcaster does not need to care about the view angle of the camera. The viewers can change their point of view (POV) while watching the 360-degree live video and they can watch the video from POV which they are interested in.

The 360-degree Internet live broadcasting, however, has an issue that the broadcaster cannot check the viewers' POV. In the conventional Internet live broadcasting, it uses a web camera which has a single lens and the single lens definitely shows the rectangular photographing range. The broadcaster can be aware of the viewers' viewing range and what they are watching by direction of the lens. On the other hand, in the 360-degree Internet live broadcasting, it uses an omnidirectional camera which has a wide-angle lens or multiple lens and the broadcaster cannot know what the viewers are watching by direction of the lens.

There are many studies about the role of gaze information in the remote communication [1][2]. In the studies, it is turned out that the communicatee's gaze information indicates the target of interest or center of the topic. The gaze information in the remote communication is similar to the viewers' POV in the 360-degree Internet live broadcasting. Therefore, the viewers' POV are not only information which indicates where the viewers are watching but also information which indicates what the viewers are interested in. Because of that, the broadcaster sometimes cannot understand the context of the viewers' comments and it can be a factor which causes communication errors between the broadcaster and the viewers.

To solve this issue, we have studied about an algorithm which detects distinctive viewers' POV to grasp viewers' interests [3]. In this research, the algorithm could detect useful viewers' POV for the broadcaster. We have also studied the effect of presentation of the distinctive viewers' POV to the broadcaster [4]. In this research, we found the presentation of the distinctive viewers' POV could be effective for the broadcaster and it gave positive effects to the communication between the broadcaster and the viewers. It enabled the broadcaster to know what the viewers were interested in. The broadcaster also had a chance to communicate with passive viewers who sent few comments to the broadcaster by the presentation of the distinctive viewers' POV. Even if the distinctive viewers' POV which were not useful to know the viewers' interests were displayed, it did not have a significant negative impact on communication and broadcasting.

ISSN1883-4566 © 2022 - Informatics Society and the authors. All rights reserved.

The use case which we envisioned for the previous researches were that a single broadcaster delivered the situation of walking through a tourist spot. The broadcaster would visit a tourist spot and report about the spot to the viewers. The equipment used for the broadcasting were a laptop computer and an omnidirectional camera. The broadcaster carried a backpack with a camera mounter to fix the omnidirectional camera and handed the laptop PC. The distinctive viewers' POV were simply presented as red circles on the equirectangular video using a laptop PC for the broadcaster. The presentation method was not easy to comprehend and the use of a laptop PC would increase the risk of accidents.

To solve these issues, we study several assistant devices to present the distinctive viewers' POV to the broadcaster in an effective manner. In this research, we propose three assistant devices for presentation of the distinctive viewers' POV in 360-degree Internet live broadcasting. The first one is a belt-type device which presents directions of the POV using vibration motors. The second one is a LED-type cylindrical device which presents directions of the POV using LED light sources. The third one is a robot-type device which presents directions of the POV using movement of the robot's head.

The contributions of this paper are summarized as follows:

- We developed three assistant devices for presentation of distinctive viewers' POV.
- We clarified the effects of each assistant device in experiments and which device the broadcaster preferred.

The rest of this paper is organized as follows. Section 2 describes our previous work about viewers' POV introducing related work. Section 3 describes three assistant devices for presentation of the distinctive viewers' POV in 360-degree Internet live broadcasting. Section 4 describes implementation of the assistant devices. Section 5 describes evaluation experiments to clarify the effects of each assistant device and discussion about the experimental results. Section 6 summarizes this study.

2 PREVIOUS WORK

2.1 Importance of Gaze Information and POV Analysis

There have been many studies on the role of non-verbal information in communication. In particular, gaze information has been shown to play an important role in communicating mutual intentions. The GAZE Groupware is a study of gaze information in communication [1]. In this study, the non-verbal information of the remote communication in a teleconference system is analyzed. He verified whether natural communication can be performed by conducting a meeting with nonverbal information in a virtual conference room. In addition, he discovered a problem that it is difficult to present gaze information because the space in which the conference participants reside is different in the remote meeting systems. He concludes that it is possible to analyze who talks about what by talking about the gaze directions of the communicatees.

Another study on mutual gaze in remote communication using videoconferencing systems [2] has revealed some interesting findings. The authors argue that the eye contact information of the communicatee is an important factor in the outcome of collaborative work with remote communication. Furthermore, the study also examined the method of presenting gaze information and concluded that the presentation of images including the eyes of the communicatee requires a certain size of images. In 360-degree Internet live broadcasting, the POV is the information that indicates the viewing direction and viewing range of the viewers, and it plays the same role as the gaze in remote communication.

On the analysis of viewers' POV in 360-degree video, a study of Yen-Chen Lin et al. examined on the correction of viewing direction in 360-degree video [5]. In this study, they examined a method of correcting the viewer's direction to the direction of the main story of a 360-degree video. They have implemented and evaluated two patterns of corrections: an automatic correction function and a correction with annotations. The results showed that there were multiple purposes and patterns in the viewer's viewing behavior and emphasize the need to analyze the viewer's viewing direction to provide a higher quality viewing experience.

YouTube provides a heat map analysis function for posted 360-degree videos, and the results of the analysis of the entire 360-degree videos are also available [6]. An analysis of the viewer's POV during viewing revealed the characteristics of watching a 360-degree video. The viewer's POV was directed most toward the 90-degree horizontal range centered on the front of the video, where 75% of the playback time was spent. It was also shown that only 20% of the users watched the full 360-degree range, even for the most popular videos.

2.2 Proposed Algorithm

We have built the following three hypotheses concerning the characteristics about the viewers' POV in a 360-degree Internet live broadcasting [3].

(1) The viewers' POV is concentrated on the direction of the broadcaster's way in mobile environment.
(2) If the viewers' POV directs at other direction except the direction of the broadcaster's way, the viewing behaviors have meanings and there are some interesting objects for the viewers in the direction.
(3) The viewers' POV returns to the direction of the broadcaster's way after the viewers' interests are satisfied

Summarizing the above hypotheses, in a 360-degree Internet live broadcasting, the viewer's POV is directed in

the direction where the broadcaster is going. It changes from the frontal direction to the other direction when a target of interest is found. Thereafter, when the interest is satisfied or the target is no longer visible, the viewer's POV is expected to return to the direction where the broadcaster is going.

By conducting an experiment about these three hypotheses, we found they were true. The viewers changed their POV according to their own interests when the POV was directed to the other range in the experiment. Therefore, we developed an algorithm to detect POV viewing within other ranges as the distinctive viewers' POV. We determined the classification of the viewer's state. The viewer's state was classified into the following four categories. The state in which the viewer was viewing the front range was called the "normal viewing". The state in which the POV changed from the front range to other range was called "start of other range viewing". The state in which the viewer was continuously viewing the other range is called "other range viewing". The state that returns to the front range was "end of other range viewing". The "normal viewing" was the state which the viewer's POV changed only within the front range, and we were expected to remain in this state for the longest period of time during the broadcast. The algorithm detected the POV in the state of "other range viewing" by checking the start of other range viewing and end of other range viewing.

Figure 1 shows a flowchart of the algorithm which we created and Fig. 2 shows the variables and conditional expressions used in the flowchart. In the algorithm, the front range is defined as horizontal polar coordinates of broadcaster's way $\varphi_f \pm 60$. It monitors horizontal polar coordinates of viewers' POV every 100 milliseconds. φ_{i_n} gives the horizontal polar coordinates of i-th viewer's POV in n-th monitoring by the algorithm. If one of the viewers' POV goes beyond the front range by satisfying the conditional expression 1 and 4, the algorithm sets the state to "start of other range viewing" and considers it is the distinctive viewers' POV until the state goes to "end of other range viewing" by satisfying the conditional expression 2 and 3. From an experiment, we found the algorithm could detect the distinctive viewers' POV which could be utilized for broadcasting with 89.76% accuracy.

We conducted an experiment for evaluation of presentation of distinctive viewers' POV [4]. As shown in Fig. 3, The distinctive viewers' POV were simply presented as red circles on the equirectangular video using a laptop PC for the broadcaster in the previous work. As a result of the experiment, we found that the presentation of the distinctive viewers' POV made it easy for the broadcaster to understand interests of the viewers. On the other hands, the presentation method was not easy to comprehend and the use of a laptop PC would increase the risk of accidents. We have to study some methods to present the distinctive viewers' POV to the broadcaster in an effective manner.

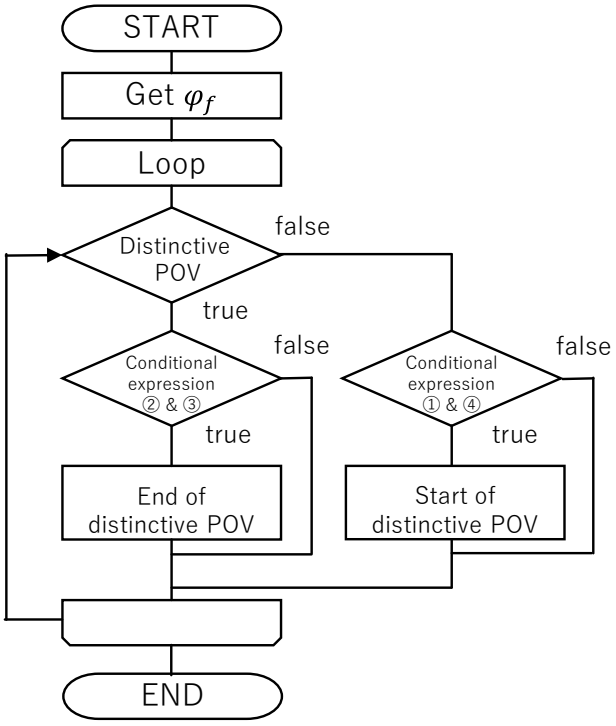


Figure 1: Flowchart of the algorithm

Variables:
Horizontal polar coordinates of viewers' POV
$$\varphi_i = \{\varphi_{i_1}, \varphi_{i_2}, \ldots, \varphi_{i_n}\}$$
Horizontal polar coordinates of broadcaster's way
$$\varphi_f$$

Conditional expression ① :
$$\varphi_f - 60 < \varphi_{i_{n-9}} < \varphi_f + 60$$

Conditional expression ② :
$$\varphi_{i_{n-9}} < \varphi_f - 60 \;||\; \varphi_f + 60 < \varphi_{i_{n-9}}$$

Conditional expression ③ :
$$\varphi_f - 60 < \varphi_{i_n} < \varphi_f + 60$$

Conditional expression ④ :
$$\varphi_{i_n} < \varphi_f - 60 \;||\; \varphi_f + 60 < \varphi_{i_n}$$

Figure 2: The variables and conditional of flowchart

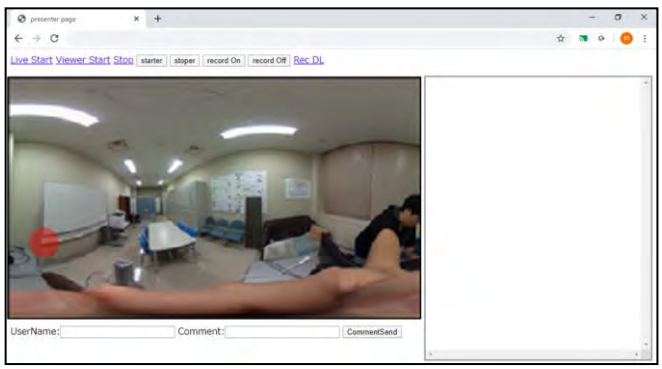


Figure 3: Presentation of distinctive viewers' POV in the previous work

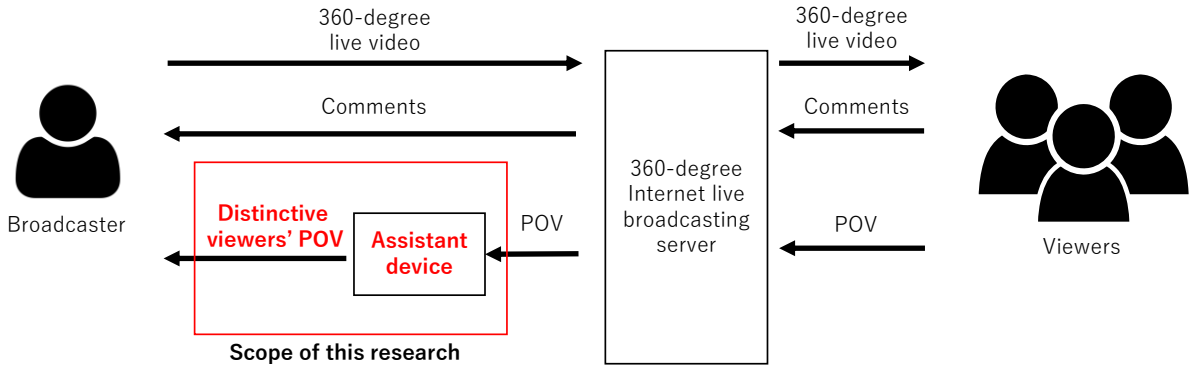


Figure 4: A model of the assistant devices in the 360-degree Internet live broadcasting

3 ASSISTANT DEVICES

3.1 A Model of the Assistant Devices

We propose assistant devices for presentation of distinctive viewers' POV in 360-degree Internet live broadcasting. Figure 4 shows a model of the assistant devices. In the model, a broadcaster sends 360-degree live video to the broadcasting server and viewers can watch the broadcasting and send comments to the broadcaster accessing the broadcasting server as same as typical 360-degree Internet live broadcasting services. The viewers can watch the video changing their POV. The coordinates of the POV are also sent to the broadcasting server in real time as same as our previous work. In this research, the broadcasting server sends the POV to an assistant device. The assistant device uses the algorithm of the previous work to detect distinctive viewers' POV. The assistant device presents the distinctive viewers' POV to the broadcaster by using a presentation means. The broadcaster can communicate with the viewers smoothly estimating their interests from the presentation of the distinctive viewers' POV.

3.2 Requirements of the Assistant Devices

The broadcaster's motivation to use the assistant devices is to make the walk in the tourist spot better. The idea is that the broadcaster can gain a companion from the viewers, even if the broadcaster is traveling alone. In this environment, there are the following three requirements.

1. It should be short time to check the assistant device.
2. It should not restrain the broadcaster's physical activity.
3. It should improve the broadcaster's experience.

The first and second requirements are needed not to increase the risk of accidents, for example, in case of using a laptop PC. The third requirement means the assistant device has great potential to become a companion for the broadcaster in order to improve his/her experience.

3.3 Ideas of the Assistant Devices

Based on the requirements, we present three ideas of the assistant devices. The first one is a belt-type device which presents directions of the POV using vibration motors. The second one is a LED-type cylindrical device which presents directions of the POV using LED light sources. The third one is a robot-type device which presents directions of the POV using movement of the robot's head.

The belt-type device has an advantage that the broadcaster does not need to look the assistant device and can pay attention with his/her surroundings. It would highly satisfy requirement 1 and 2. We use a single line belt and the belt is bound around broadcaster's waist so that it can keep fashionability. Although the belt-type device can present the distinctive viewers' POV in the horizontal direction, it is difficult to present the distinctive viewers' POV in the vertical direction.

The LED-type cylindrical device has an advantage that it can present the distinctive POV both in the horizontal and vertical direction. However, the degree of achievement of the requirement 1 and 2 would be lower than the belt-type device because the LED-type device makes the broadcaster look the assistant device in order to check the distinctive viewers' POV.

The robot-type device specializes in requirement 3. It has an advantage that the robot can be a companion as if the broadcaster walked with a friend together and it would improve the broadcaster's experience. On the other hand, it has a disadvantage in terms of requirement 1 and 2 because it makes the broadcaster look and carry the device.

4 IMPLEMENTATION

We implemented prototypes of the assistant devices evaluate their effectiveness in an experiment. We used a 360-degree Internet live broadcasting system which was developed in our previous work [4] and added some functions to the broadcasting system so that it can handle the assistant devices. In this section, we describe the system architecture and prototypes of the assistant devices.

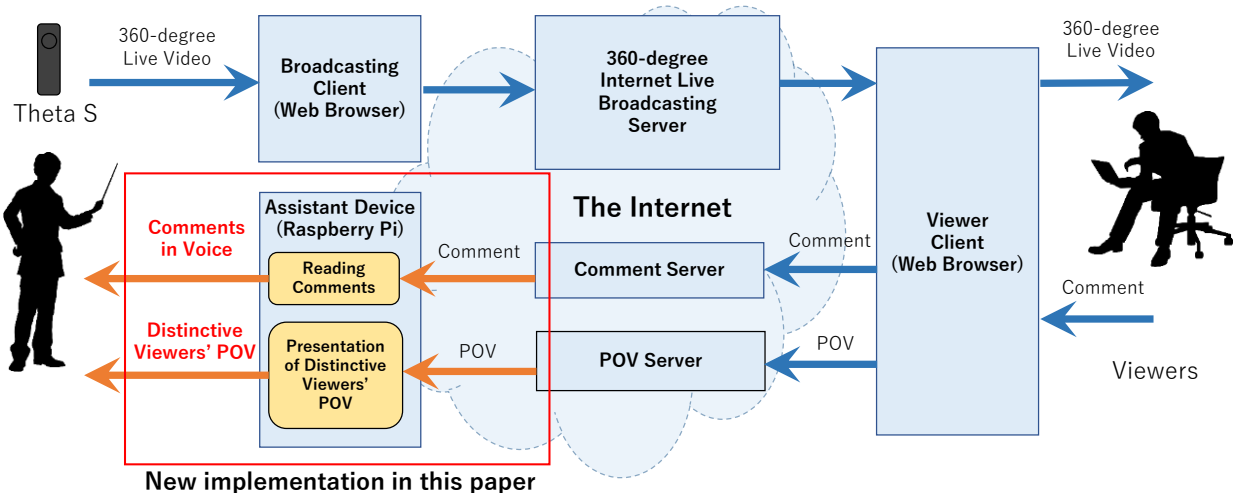


Figure 5: System architecture

4.1 System Architecture

The basic features of the 360-degree Internet live broadcasting system was similar to that in our previous work. In the broadcasting system, it consists of two components which are a 360-degree Internet live broadcast system and a POV server which collects the viewers' POV to deliver it to an assistant device of the broadcaster. Figure 5 shows the architecture of the system. We use the Ricoh Theta S as an omnidirectional camera for the system.

The broadcaster accesses a broadcasting system and starts broadcasting by a web browser. Viewers access the broadcasting system and watch the live broadcasting. The viewers' POV information is sent to the POV server periodically.

We implement the 360-degree Internet live broadcast system by adopting WebRTC that realizes the video streaming on HTML5. We use the Kurento Media Server [7] for the video streaming server. The WebRTC is an API to provide real-time communication functions such as voice communication and video distribution without requiring any plug-in and installation of special software. We use a javascript library called Three.js [8] for processing images. The image acquired from the omnidirectional camera is mapped to a spherical object by using the library. The viewers' POV are managed by the angular coordinates of the two axes which are acquired every 100 ms and the data is sent to the POV server. In this system, the POV is represented by the polar coordinate (ϕ , θ , r) of the spherical surface. The POV sever receives the POV from the viewers and saves the collected data in the database. The Web server and the POV server are implemented using node.js [9]. The algorithm which detects the viewers' distinctive POV is implemented in the POV server.

4.2 Prototypes of the Assistant Devices

Described below are the new functions in this paper. The POV server analyzes the collected POV with the algorithm and send the distinctive viewers' POV to the assistant device.

Figure 6: Prototypes of assistant devices

The assistant device has functions of presentation of the distinctive viewers' POV and reading comments in voice. The prototypes of the assistant devices are controlled by software on Raspberry Pi.

Figure 6 shows the prototypes of the assistant devices. The belt-type device has several vibration motors inside the belt at regular intervals. The broadcaster binds a belt around his/her waist and grasp the distinctive viewers' POV in a horizontal direction by the vibration. The LED-type device

has several LED tapes which can light individually. The LED tapes are wound around a cylindrical can. The bottom of the can has a clip to fix it on something such as a laptop PC. The broadcaster can grasp the distinctive viewers' POV both in horizontal and vertical direction by the light. The robot-type device has a robot head and it can be turned up, down, right or left using two servo motors. The bottom of the robot head also has a clip to fix it on something. The broadcaster can grasp the distinctive viewers' POV both in horizontal and vertical horizontal direction by the direction of the robot head.

The reading comments in voice is a function which enables the broadcaster to confirm the viewers' comments in voice without looking a display device such as a smartphone and a laptop PC. In the previous work, the viewers' comments were displayed on a laptop PC. Since the prototypes of the assistant devices do not need the laptop PC, the broadcaster does not need to look the laptop PC. This function makes the broadcaster free from looking the laptop PC. In the outdoor environment, the broadcaster must to be able to hear the surrounding sound for safety. We use a bone conduction headphone for hear the viewers' comments.

5 EVALUATION

5.1 Experimental Procedure

We conducted two sets of experiments to evaluate the prototypes of the assistant devices. In each set of the experiments, there were one broadcaster and five viewers and the total experimental participants were 12 people. The location of the broadcasting was Takamatsu Pond in Morioka City, Iwate Prefecture, which was famous as a place where swans flied. Each broadcaster performed broadcasting for approximately 15 minutes and it was performed three times changing the assistant devices. The broadcaster walked around the pond and communicated with the viewers talking about the situation of the circumference. The viewers sent comments about the situation of the circumference of the broadcaster. We asked the broadcaster to stop at a safe place when checking the assistant device for his/her safety and to reply the comments from the viewers as many times as possible.

After three times of the experiments changing the assistant devices, we interviewed to the broadcaster about the presentation of the distinctive viewers' POV. In the interview, we asked about "Awareness of the presentation of the distinctive viewers' POV", "Awareness of the POV direction" and "Usefulness of the POV". Moreover, we asked the broadcaster to evaluate about "understandability of the POV direction", "Easiness of the POV check", "Utilization degree of the POV" and "Do you want to use it again?" on a scale of one to five. After that, we asked the broadcaster to give his/her impression about the assistant device through the whole experiment.

Table 1: Result of the interview after the experiment

Presentation Method	Experiment	Presentation Times	Awareness of the POV	Awareness of the POV direction	Usefulness of the POV
Belt-type	1st	24	24 (100%)	19 (79.2%)	14 (58.3%)
	2nd	23	23 (100%)	20 (87.0%)	11 (47.8%)
LED-type	1st	18	17 (94.4%)	**8 (44.4%)**	9 (50%)
	2nd	21	19 (90.5%)	**12 (57.1%)**	10 (47.6%)
Robot-type	1st	21	21 (100%)	18 (85.7%)	16 (76.2%)
	2nd	28	28 (100%)	24 (85.7%)	18 (64.3%)

Table 2: Result of the comparison

Presentation Method	Understandability of the POV direction	Easiness of the POV check	Utilization degree of the POV	Do you want to use it again?
Belt-type	3.5	5	3.5	4
LED-type	2	2.5	3	2
Robot-type	4	4	3	4

Table 3: Comments about the assistant devices

Presentation Method	Experiment	Comments from the Broadcaster
Belt-type	1st	The vibration **reduced labor and time for checking the POV**. Since the space between the vibration motors was long, it was **difficult to understand what the viewers were looking at**. I could **use the belt-type device without worrying about what surrounding people thought of my equipment** comparing with the other devices.
	2nd	I was **easy to understand** the vibration. **The direction of the POV was a little vague**.
LED-type	1st	It was **difficult to check which LED were glowing** because of the sunlight reflected by the pond. The vertical direction of the POV was vague and it was not useful in this experiment.
	2nd	It was **difficult to understand which LED were glowing** because the time of light was short. It **took time and effort to check the glowing LED** because I needed to make a shadow portion for the device.
Robot-type	1st	Although I was **worried about what surrounding people thought of my equipment** comparing the other devices, it was **easy to understand the direction of the POV**. I **felt friendship for the robot** because the voice of reading comments sounded to me as if the robot had spoken.
	2nd	The head movement was **cute**. It **took time to wait** until the head movement was completed. **The direction of the POV was best understandable**.
Impression about all methods	1st	Since the device read the comments, it was easy to perform the broadcasting. **The assistant devices did not impose a burden to me**.
	2nd	**I could not check the assistant device on several occasions** because it was slippery.

5.2 Experimental Results

Table 1 shows the experimental result of the interview after the experiment. The presentation of the distinctive viewers' POV by the belt-type device was 24 times in 1st experiment and 23 times in 2nd experiment. The

presentation by the LED-type device was 18 times in 1st experiment and 21 times in 2nd experiment. The presentation by the robot-type device was 21 times in 1st experiment and 28 times in 2nd experiment. There was no significant difference among the assistant devices and it could be fair comparison. In terms of the awareness of the POV, they were aware of almost all of the POV presentations. In terms of the awareness of the POV direction, the POV presentations by the belt-type and robot-type devices could be understandable about 80%. However, that of the LED-type device was remarkably low, that is, around 50%.

Table 2 shows the experimental result of the comparison of the assistant devices. The belt-type and robot-type devices achieved scores above average in all comparison items. This result showed these devices were useful to the broadcaster. However, the LED-type device was low score in most comparison items without "utilization degree of the POV". The LED-type device especially would have problems of the usability.

Table 3 shows the comments from the broadcaster written by a free format about the assistant devices. In terms of positive comments about the belt-type device, "it reduced labor and time for checking the POV" and "it can be equipped inconspicuously". In terms of negative comments about the belt-type device, "it was difficult to understand what the viewers were looking at" and "the direction of the POV was a little vague". These results showed the belt-type device was usable but there was an issue with understandability of the POV direction. The LED-type device got only negative comments. They were "difficult to check which LED were glowing", "it was difficult to understand which LED were growing" and "it took time and effort to check the glowing LED". These results showed the LED-type device was not suitable outdoors in the daytime. In terms of positive comments about the robot-type device, "it was easy to understand the direction of the POV", "I felt friendship for the robot", "it was cute" and "The direction of the POV was best understandable". In terms of negative comments about the robot-type device, "I was worried about what surrounding people thought of my equipment" and "it took time to wait for the robot movement" These results showed the robot-type device gave friendship feeling to the broadcaster and it had an advantage in understandability of the POV direction though it stood out outside and the delay time was existed to present the POV.

5.3 Discussion

In the experiment, the belt-type device was highly evaluated by the broadcaster because the device could be used without attention to the device disturbing the broadcasting. The robot-device was also highly evaluated by the broadcaster because it gave friendship feeling to the broadcaster in addition to good understandability of the presentation of the distinctive viewers' POV. We found a new possibility that it became a companion for the broadcaster in order to improve his/her experience. On the other hand, the LED-device got significant low evaluation because it disturbed the broadcasting to check the device and was difficult to see the light outside. To summarize the experimental results, we found the broadcasters did not prefer assistant devices in visual form and they had a positive impression of the robot-type device.

The prototypes of the assistant device in this paper can present one presentation of the distinctive viewers' POV at a time. However, in some cases, there are several distinctive viewers' POV at the same time. In such cases, the several distinctive viewers' POV should be weighted by using the importance because multiple presentations of the distinctive viewers' POV may confuse the broadcaster. The weighting function should be studied and implemented into the assistant devices.

The other point to be discussed is whether a voice assistant is effective for the presentation of the distinctive viewers' POV. In the experiment, the voice function was only utilized for reading out the comments from the viewers. It could be utilized for the recognition of the distinctive viewers' POV by reading out the position by voice such as "the distinctive viewers' POV is right-upper direction". It is necessary to evaluate the voice assistant comparing these assistant devices. We consider there is a possibility of combination of the robot-type device with the voice assistant in order to improve its understandability and friendship feeling.

6 CONCLUSION

In this paper, we proposed assistant devices for presentation of the distinctive viewers' POV in 360-degree Internet live broadcasting. we study and developed three prototypes of the assistant devices; a belt-type device which presented directions of the POV using vibration motors, a LED-type device which presented directions of the POV using LED light sources cylindrically and a robot-type device which presented directions of the POV using movement of the robot's head. We conducted an evaluation experiment. From the experiment, we found the broadcasters did not prefer assistant devices in visual form and they had a positive impression of the robot-type device.

For the future work, we will improve the robot-type device to make it more understandable and friendlier. We also study other methods to present the distinctive viewers' POV to the broadcaster using mixed reality (MR) technologies without disturbing the broadcasting.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number JP20K11794.

REFERENCES

[1] R. Vertegaal, "The GAZE groupware system: mediating joint attention in multiparty communication and collaboration", CHI '99 Proceedings of the

SIGCHI conference on Human Factors in Computing Systems, pp.294-301 (1999).
[2] D. M. Grayson, A. F. Monk, "Are you looking at me? Eye contact and desktop video conferencing", ACM Transactions on Computer-Human Interaction (TOCHI) Volume 10 Issue 3, September 2003, pp.221-243 (2003).
[3] M. Takada, D. Nishioka, and Y. Saito, "A Detection Method of Viewers' Interests Based on POV for 360-Degree Internet Live Broadcasting in Mobile Environment", IEEE GCCE OS-VDP: 2D/3D Video Data Distribution and Processing (2019).
[4] M. Takada and Y. Saito, "A Study on Presentation of Viewers' Interests based on POV Analysis in Mobile 360-degree Internet Live Broadcasting", International Workshop on Informatics (IWIN2020), pp.3-8 (2020).
[5] Y.C. Lin, Y. J. Chang, H. N. Hu, H. T. Cheng, C. W. Huang, M. Sun, "Tell Me Where to Look: Investigating Ways for Assisting Focus in 360° Video", Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 2535-2545 (2017).
[6] YouTube Creator Blog, "Hot and Cold: Heatmaps in VR", available from <https://youtube-creators.googleblog.com/2017/06/hot-and-cold-heatmaps-in-vr.html> (accessed 2022-03-01).
[7] Kurento, available from <http://www.kurento.org/> (accessed 2022-03-01).
[8] three.js - Javascript 3D library, available from <https://threejs.org/> (accessed 2022-03-01).
[9] Node.js, available from <https://nodejs.org/ja/> (accessed 2022-03-01).

(Received: October 30, 2021)
(Accepted: March 31, 2022)

Masaya Takada. received his Ph.D. degree from Iwate Prefectural University, Japan, in 2021. His research interests include 360-degree Internet live broadcasting. He is a member of IPSJ and IEEE.

Yoshia Saito. received his Ph.D. degree from Shizuoka University, Japan, in 2006. He had been an expert researcher of National Institute of Information and Communications Technology (NICT) from 2004 to 2007, Yokosuka, Japan. He was a lecturer from 2007 to 2011 at Iwate Prefectural University and he is currently an associate professor at the University. His research interests include computer networks and Internet broadcasting.

Regular Paper

Towards Resistance to Memory Inspection Attacks on Plausibly Deniable Distributed File Systems

Ryouga Shibazaki†, Hiroshi Inamura‡, and Yoshitaka Nakamura*

†Graduate School of Systems Information Science, Future University Hakodate, Japan
‡School of Systems Information Science, Future University Hakodate, Japan
* Faculty of Engineering, Kyoto Tachibana University, Japan
{g2120017, inamura}@fun.ac.jp, nakamura-yos@tachibana-u.ac.jp

Abstract - Data protection has become an important issue in Internet services. In storage systems, conventional methods such as full disk encryption are generally used, but this alone cannot protect against forced attacks of key disclosure. PDE (Plausibly Deniable Encryption), which enables the denial of the existence of confidential information, has been proposed, and by disclosing the decoy key, it has become possible to protect the user from the force to disclosure the key. It is an issue to be considered that the main memory is attacked at runtime due to the use in the cloud and the spread of virtualization technology. Therefore, we are proposing PTEE FS that realizes an encrypted file system using the concept of PDE in a trusted execution environment (TEE). To provide the resistance to exploiting the knowledge from the use of disclosed the decoy key, we introduce FID unification mechanisms. Regarding the performance of PTEE FS, we will evaluate the estimated performance given by the overhead of using TEE by using a model that imitates actual use on the cloud and using file synchronization between the server and client as the actual use model on the cloud.

Keywords: Plausibly Deniable Encryption, OS Security, Trusted Execution Envirionment.

1 BACKGROUND

Leakage of confidential data related to privacy endangers the privacy of data owners and leads to the loss of social credibility of the leaked organization, so protection of such data has become an important issue. Traditional methods such as full disk encryption are commonly used in storage systems, but these methods make it difficult to maintain confidentiality when access to computer hardware or administrator privileges is stolen by an attacker. On the other hand, Plausibly Deniable Encryption (PDE), which is a new concept of encryption, has been proposed [1]. PDE protects confidential information sufficiently by allowing the existence of information to be denied. By disclosing the decoy key, PDE protects against the extortion of the decryption key by an attacker. While admitting that the encrypted file system exists in the system, the attacker is given the decoy key to access the decoy area, but the existence of the hidden area and its contents are kept secret. From the perspective of storage system configuration, PDE's existing research primarily protects sensitive information in persistent storage, and it is assumed that the main memory, which controls the existence of confidential information at runtime, will not be attacked. As an attack on the main memory, a memory inspection attack is assumed in this paper. This is an attack that illegally takes a snapshot of the main memory and obtains confidential information.

According to the white paper of the Japanese Ministry of Internal Affairs and Communications [2], it can be seen from Fig. 1 that data storage and backup are performed using the cloud. In this way, cloud services have become widespread due to the spread of virtualization technology in recent years, and attacks on main memory have become an issue to be considered. There is a risk that someone who understands the system configuration attempts to attack the main memory with a vulnerability on the premise of using technology that controls hardware privileges such as virtualization technology and hypervisor used in cloud services. An example of incidents involving suspected administrative compromise is the illegal access to data for about 100 million people stored on Amazon Web Services at Capital One, an American bank, in 2019 [3].

With the development of virtualization technology, hardware support functions are being incorporated into CPUs to be able to perform processing that guarantees data confidentiality even when such privileged users and terminal administrators are not credible [4].

So far, the purpose of this study is to construct an encrypted file system that is resistant to attacks not only on the persistent storage device but also on the main memory and that can deny the existence using the concept of PDE. We proposed a system using Intel SGX as a hardware-protected execution an environment in the realization of an encrypted file system [5].

In this paper, we examine countermeasures against attacks that are established on the premise that the attacker knows the existence of the decoy key and decoy data for PTEE FS (PDE with Trusted Execution Environment File System). In Chapter 4, we describe the problems of an attack using knowledge from the use of disclosed decoy key. First, introduce a PDE system and threat analysis as assumptions for our previous proposal. Then, we explain an attack using knowledge from the use of disclosed decoy key, which is a threat to be solved in this paper. Chapter 5 we propose FID (file ID) unification, which is a method for solving attacks using knowledge from the use of disclosed decoy key explained in Chapter 4. The basic concept, procedures, and integrated structure of the hidden and

ISSN1883-4566 © 2022 - Informatics Society and the authors. All rights reserved.

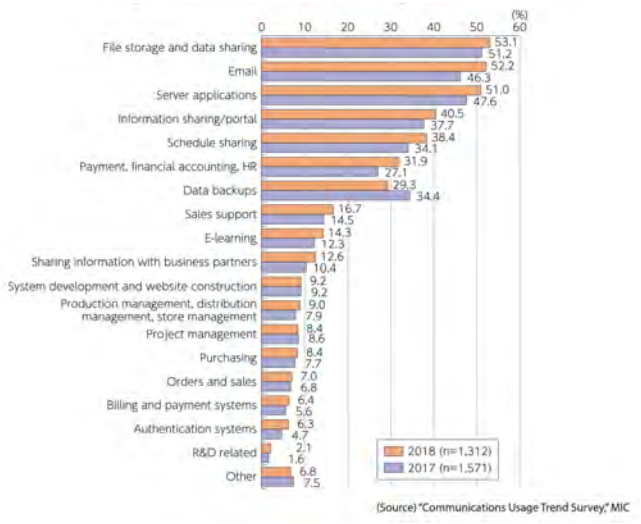


Figure 1: Breakdown of cloud service usage in enterprises

decoy file will be presented. In Chapter 6 and 7, as an evaluation of the processing time in normal access of PTEE FS, a model that imitates the actual use on the cloud is used, and the performance is evaluated in consideration of the overhead due to the use of TEE. In addition, as the processing time of the program started on-demand, the performance is evaluated in consideration of the additional latency due to calling the FID merge processing described in Chapter 5. In the evaluation of processing time in normal access, file synchronization between server and client is used as an actual usage model on the cloud. In the evaluation of the processing time of the program started on-demand, the local file created by referring to the existing research [6] by Leung et al. is used.

2 RELATED RESEARCH AND RELATED TECHNOLOGY

This section describes the concept of Plausibly Deniable Encryption, its application to file systems, and Intel SGX, which is being examined for application to the realization of attack resistance to main memory.

2.1 Plausibly Deniable Encryption

Plausibly Deniable Encryption (PDE) was proposed by Canetti et al. [1] as one of the encryption methods. Traditional disk encryption methods including full disk encryption have the problem that they cannot be protected if the owner is forced to disclose the decryption key by an attacker. Therefore, PDE, which was proposed as one of the methods to protect the owner from the key disclosure extortion attack, enables the protection attack by using the decoy key. PDE is a characteristic of using a decoy key, which enables protection from key disclosure extortion attacks. As shown in Fig. 2, PDE applies special encryption to confidential information that can be decrypted with both a decoy key and a private key, unlike conventional encryption. Decryption with the decoy key gives the decoy plaintext, and decryption with the private key gives the original plaintext. When the legitimate user is attacked by

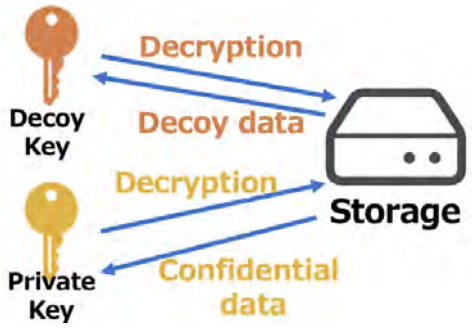


Figure 2: Overview of PDE

an attacker forcing key disclosure, the user can give the decoy key to the attacker. Since the attacker thinks that the decoy key is the original private key, it allows the original confidential information unnoticed and be kept secret.

On the other hand, the disadvantage is that the size of the ciphertext becomes extremely large, which may make the attacker suspicious of applying a special cipher. Furthermore, traces of confidential information may be obtained from the file system and the physical storage medium layer, etc., and considering these, it cannot be said to be a practical method. However, the idea of PDE is that the decoy key gives decoy information and the private key gives the confidential information that can be used.

2.2 Applications of the PDE Concept

Using the idea of PDE, a method was proposed to bring confidentiality using two types of techniques, steganography and hidden volume, instead of using simple encryption. First, a PDE method using the concept of steganography was proposed by Anderson et al. [7] and Chang et al. [8]. The basic idea is to hide confidential information in ordinary information. For example, confidential information is embedded and saved in a part of a large file such as an image file. In steganography, there is a risk that the confidential information will be overwritten when the file in which such confidential information is embedded is changed. To avoid overwriting confidential information, the risk is alleviated by copying and saving multiple confidential information, but it has the disadvantage that the usage efficiency of the storage device deteriorates and a large amount of confidential information cannot be retained. PDE using hidden volume technology has been proposed by Jia et al. [9] and Zuck et al. [10]. File system using hidden volume technology creates a decoy volume on a storage device with a decoy key and a hidden volume with a private key. The decoy volume is placed throughout the storage device, and the hidden volume is usually placed from the hidden offset, which is the initial position of the hidden volume on the storage device, toward the end of the storage device. When using the PDE file system using hidden volume technology, the user logs in public mode or PDE mode and uses the file system. In public mode, the user only operates decoy volumes and in PDE mode, the user can operate hidden volumes. When forced to disclose the key, the owner discloses the login password of the public volume and the decoy key, to protect the hidden volume and

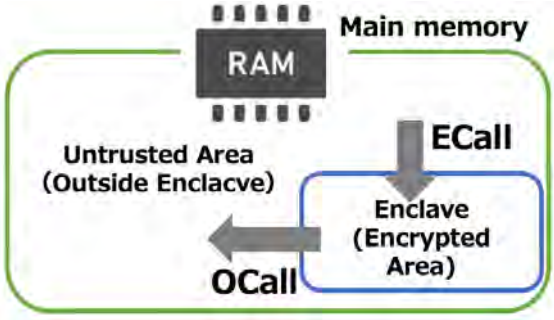


Figure 3: Overview of Intel SGX

the confidential data from the attacker. In the hidden volume technology, the existence of the hidden volume and the hidden offset are unknown in the system that operates the decoy volume, so the data stored in the decoy volume may overwrite the hidden volume.

2.3 Intel Software Guard Extensions

Intel Software Guard Extensions (Intel SGX) [11] is a CPU extension architecture provided by Intel Corporation. Intel SGX can perform processing that guarantees the confidentiality of data even if the privileged user or terminal administrator is not credible. As shown in Fig. 3, Intel SGX creates an encrypted area called Enclave on memory. Enclave provides a trusted execution environment (TEE) to enable program execution while maintaining data confidentiality provided at the hardware level. Intel SGX can protect the programs and data in the enclave from memory inspection attacks. Enclave is called using ECall from untrusted areas. Then, the result processed in the enclave is passed to the untrusted area using OCall. Enclave is executed by the CPU in a special mode which deny cannot be inspected and tampered with by a program outside Enclave. ECall and OCall can achieve confidentiality by denying access from cached addresses to Enclave's private memory by a program outside Enclave. Intel Corporation provides the Intel Software Guard Extensions SDK [12] as an environment for using Intel SGX technologies.

However, Enclave has a limited size that included both program and data, the size is about 100MB. Therefore, the content to be processed by the enclave must be minimized. For example, In the existing research [13] using Intel SGX by Ahemed et al., The policy is to keep only the private key and perform only the related processing in the enclave. A study measuring the performance of Intel SGX by Gjerdrum et al. [14] has shown that the overhead increases when the size of the buffer sent to the enclave exceed 64 kB.

2.4 Measured Traffic for File Server on the Cloud

In the evaluation, we need to assume a usage model of file sharing on the cloud. Leung et al. [6] measured traffic for two file-sharing servers used in NetApp data centers for three months. One of the servers was used by the marketing, sales, and finance departments, and the other was used by the engineering department. This time, we referred to the statistical data of the servers used in each department of marketing, sales, and finance. This server received 364.3GB Read and 177.7GB Write access in 3 months. The ratio of Read, Write, and Delete requests was 540: 170: 1 in this order. The request size when accessing the file was about 70 % for less than 1kB, about 10 % for 1kB or more and less than 100kB, and about 20 % for those over 100kB.

2.5 TCG Storage Security Subsystem Class: Opal

MBR Shadow, one of the functions of "TCG Storage Security Subsystem Class: Opal" [15], allows users to create and switch between the two areas such as decoy and hidden, realized by PDE in the boot process. The area division and access control functions provided by MBR Shadow will solve the issues in overwriting confidential data by decoy data, discussed in Jia et al. [9] and Zuck et al. [10]. However, MBR Shadow configures LBA (Logical Block Addressing) mapping at the time of authentication in the boot process. Therefore, if we configure a storage system with this device to the proposed system, the following usage scenarios required for this system cannot be realized. When a legitimate user and an attacker use the system at the same time, it is required that simultaneous and parallel access needs to be allowed to each of the hidden and decoy areas.

3 PLAUSIBLY DENIABLE DISTRIBUTED FILE SYSTEMS

The purpose of this research is to realize a plausibly deniable distributed file system that is resistant to key disclosure attacks and also resists memory inspection attacks in virtual environments.

3.1 Base Design

In our research so far [5], we have designed a prototype of a distributed file system for key disclosure attacks as follows.

The basic idea of PDE is that using a decoy key or passphrase will give you information that is allowed to be disclosed and using the original private key or passphrase will give you highly confidential information. To realize the basic idea of PDE, the proposed system provides a mechanism to switch the contents of the file handled based on the key and passphrase used for logging in to the file system.

PTEE FS server operates only the encrypted file and does not operate the plaintext file, but the PTEE FS client encrypts and decrypts the data and operates plaintext files. The server manages the decoy space and the hidden space. In the hidden area, highly confidential data such as access keys and passphrases for other systems that should not be leaked are stored. The decoy area does not include the data to be saved in the hidden area, and the data with low risk even if leakage occurs to the outside is saved. PTEE FS sever has the authorization control unit that determines whether the key sent from the client is a decoy or authentic and switches the operation to protect it using TEE (Trusted Execution Environment) and

performs processing. We use the NFS (Network File System) protocol with necessary modifications.

In the proposed configuration, it is necessary to switch the access destination into the decoy area and the hidden area by the key presented by the client and switch the structure of the file system. Code of the structure operation executes in TEE to prevent leakage and inspection by a snapshot of the main memory.

Since Intel SGX is used as the TEE, the confidentiality of the code for these structural operations can be maintained even when the attacker is a privileged user or terminal administrator. Therefore, this configuration can be resistant to infringement from snapshots of the main memory when accessing the file system. However, with the TEE built using Intel SGX, there is a limit to the size of the enclave that can be used, and there are some that cannot be used for kernel functions such as standard input/output in the enclave. In this research, we consider the security of the parts that are not protected by TEE and propose the system configuration that protects them.

It is possible to obtain resistance to infringement from snapshots of persistent storage devices by performing processing such as filling empty areas on the file system with random bits as by Jia et al. [9]. This is because NAND flash devices such as SSD have the characteristic that the entire block is filled with " 1" bit when the block is deleted. Therefore, it is possible to judge whether the area is free or used from the snapshot of persistent storage. If the block is not filled with " 1" bit even though it is a free area on the file system that handles the decoy area, the attacker can suspects the existence of confidential data. Therefore, processing such as filling the free area on the file system with random bits is performed.

3.2 Key Authorization

In this configuration, decryption is performed by the client, so the key or password between server and client send for access authentication and authorization at the time of mounting. A key used for encryption/decryption, a key, and a passphrase used for authorization with the PTEE FS server is not the same. The key for encryption/decryption is handled only on the client terminal. The authorization control unit determines whether the key sent from the client is a decoy or authentic and switches the operation to protect it using TEE and performs processing. The system configuration at this time is as shown in Fig. 4. Figure 4 shows the data flow when the client access the persistent storage device mounted by the Read call. The system that decrypts and encrypts at the client terminal is called PTEE FS Client, and the system that determines the key and switches the operation at the server is called PTEE FS Sever. The user who uses the client terminal receives the encrypted data as shown in Fig. 4 at the client terminal, decrypts it with the PTEE FS Client, and uses it. A legitimate user who uses the proposed system usually uses the private key, and when the attacker forces the decryption key to be disclosed, the decoy key is disclosed to protect the confidential data.

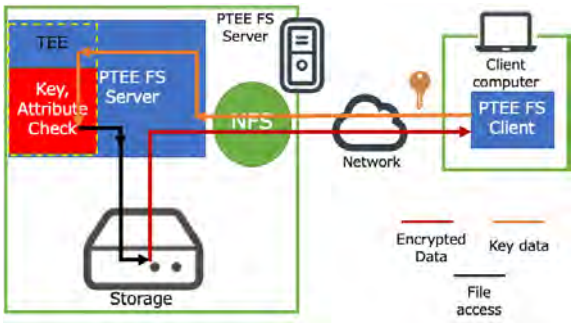


Figure 4: Data flow using TEE in the proposed system

4 PROBLEM

We give a design to resist attacks using knowledge from the disclosure of the decoy key and obtain a practical prospect from the performance estimation when applied to cloud services. In addition to the attack methods we have examined so far, we describe attacks that use knowledge from the disclosure of decoy keys that have not been examined so far.

We introduce the PDE system and threat analysis as assumptions for our previous proposal in Section 4.1. We have provided improvements to these threats in our previous proposals. Then, we explain an attack using knowledge from disclosed decoy key in Section 4.2. We present the design of this system and estimate the performance given by TEE when operating with the access pattern of the file synchronization service that is often seen in cloud storage services. In addition, we evaluate the performance of the system proposed in Chapter 5 when it is used in a typical workload when using cloud storage based on the existing research by Leung et al. [6].

4.1 Threat Analysis

Make some assumptions for threat analysis. First, a legitimate user who accesses the server on which the proposed system is running always mounts the server and handles the data. Next, the attacker has the same access rights to the cloud server as the cloud provider, and the attacker can take snapshots of the main memory and persistent storage at any time. Finally, the attacker does not have access to the legitimate user's client terminal. Based on these assumptions, the vulnerable points of the system configuration will be described. Figure 5 shows the server's basic configuration of the file system which is the conventional method using the PDE concept and the places where the denial of the existence of the hidden area may be lost. In Fig. 5, it is assumed that the persistent storage device is encrypted by the hidden volume and the main memory does not use a protection mechanism such as TEE. In Fig. 5, the location is indicated by a red circle and labeled with the number written on the speech balloon. In the following, in Fig. 5, the part with label 1 is referred to as "vulnerable point 1", and the part with labels 2 and 3 is referred to as "vulnerable point 2" and "vulnerable point 3". The attack model for each vulnerable point is as follows.

vulnerable point 1： Infringement from a running applica-

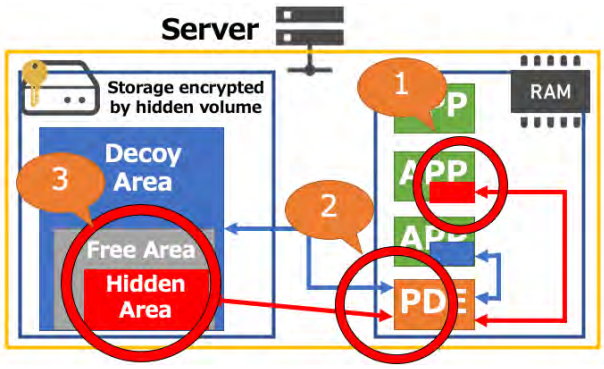


Figure 5: Vulnerable points in the basic configuration of a hidden volume file system

tion for user use

vulnerable point 2： Infringement when accessing the file system

vulnerable point 3： Infringement from persistent storage snapshots

These attacks were given resistance by the base design described in Section 3.1. For infringement from a running application for user use, the proposed system has a server/client configuration., and resistance is given by using the data only on the client terminal that cannot be observed by the attacker. For infringement when accessing the file system, resistance to attacks is given by executing a program related to hidden areas such as code that handles the file structure, on the TEE of the server. For Infringement from persistent storage snapshots, resistance is given by filling the free space with random bits as shown in Section 3.1. However, in the prototype design, there is no discussion about attacks using knowledge from the use of disclosed the decoy key.

4.2 Exploiting Knowledge from the Use of Disclosed Decoy Key

We explain an attack that uses knowledge from the disclosure of the decoy key. When an attacker whose decoy key is disclosed can acquire the time series of attacker's access information to the decoy area by network traffic or a memory inspection attack on the server, the time series of access information to the hidden area by the private key by the legitimate user can be obtained, and the existence of the hidden area is revealed by comparing and collating these.

Regarding attacks using the knowledge of decoy key disclosure in PTEE FS, we will consider how the attacks are possible by monitoring the data exchange at the interface of TEE, and how to protect them. Figure 6 shows the data flow in the TEE interface. There are two interfaces, one between the network and TEE and the other between the persistent storage device and TEE. The information that can be observed in each interface is defined as follows.

TS1: (TimeSeries1) In the operation time series between the network and TEE, the exchange of the modified NFS protocol is observed.

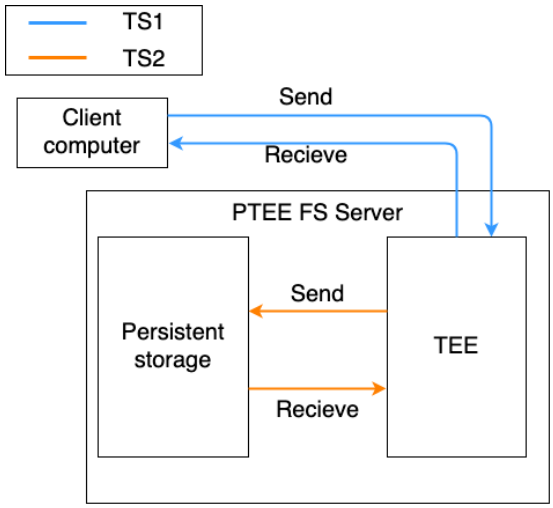


Figure 6: Data flow in TEE interface

Table 1: Example of TS1 and TS2

TS1	Send	Recieve
	getattr File	(OK, Error) Result
	getattr Dir	(OK, Error) Result
	readdirplus Dir	(OK, Error) Result
	write File data	(OK, Error) Result
	read File	(OK, Error) Result data
TS2	Send	Recieve
	fetch ObjectID data	Result ObjectID (OK, Error)
	store ObjectID data	Result ObjectID (OK, Error)

TS2: (TimeSeries2) In the operation time series between the persistent storage device and TEE, operation sequences such as fetch and store to the persistent storage device are observed.

Table 1 below shows examples of the contents observed by TS1 and TS2.

TS1 is represented by the blue line in Fig. 6, and TS2 is represented by the orange line. TS1 and TS2 are arbitrarily generated by an attacker as TS1_m and TS2_m (m: malicious), and those generated by legitimate user operations are TS1_l and TS2_l (l:legitimate). At this time, the following two attacks can be considered from the information observable on the TEE interface.

Attack Possibility 1: Because of the attacker observing the difference between TS1_m and TS1_l, the existence of the hidden area is revealed

Attack Possibility 2: When TS2_m is externally observable as an operation result of TS1_m, it is possible to judge the match between TS1s from the unification of the pair of TS2_m and TS2_l, and the hidden area Existence is exposed

We place two assumptions are made as conditions for establishing "attack possibility 2".

Attacker Assumption 1: Correspondence between TS1 and TS2
$TS2 = TEE\_exposed\_func(TS1)$
can be estimated. This means that it is possible to associate the operation series from the NFS RPC time series with the operations for the persistent storage device.

Attacker Assumption 2: It is possible to judge the match between the elements of TS2. In other words, it means that the operations on the persistent storage device can be identified and the unification can be observed.

Therefore, the following two are required to protect confidential information from attackers using the proposed method.

1. "Attack Possibility 1" is not established
2. Defend "Attack potential 2" by disabling either "Attacker Assumption 1" or "Attacker Assumption 2".

4.2.1 Eliminating Attack Possibility 1

By encrypting the payload part of the RPC of the packet, which is a component of TS1, the difference other than the data size becomes unobservable, and the occurrence of "Attacker Possibility 1" can be prevented.

4.2.2 Eliminating Attack Possibility 2

For "Attack Possibility 2", the following system configuration is adopted to prevent the "Attacker Assumption 1" from being established. As with the countermeasure for "Attack Possibility 1", the part related to RPC of the packet is encrypted. Regarding TS2, the data itself stored in the persistent storage device will be encrypted. In this configuration, the persistent storage device side assumes a general disk or a normal file system, so the object ID used when specifying the target in the persistent storage device is not protected from memory inspection attacks. The information obtained by the attacker at this time is the operation and object ID, the input/output timing to TEE, and the size of the encrypted part. The appearance pattern of the object ID in the IO traffic at the TEE mustn't provide any clue for the attacker tracks hidden volume using TS2.

5 DESIGN OF PTEE FS

FID is the object ID used by the attacker to specify in the persistent storage device. To solve the problem of attacks using the knowledge from the use of disclosed decoy key, the file ID (FID) observed by the attacker in TS2 should be the same in the decoy and hidden area. We propose FID unification to achieve and maintain this. This way, even when accessing sensitive data, only FID known to the attacker in the decoy area is observed. Specifically, the file block specified by one FID has a structure that holds decoy data and confidential data inside. Figure 7 shows the structure of a file block with the same file ID. Place sensitive metadata immediately after decoy metadata, then write decoy data, then sensitive data. Confidential metadata and confidential data are encrypted with the private key, and for users who only know the decoy key,

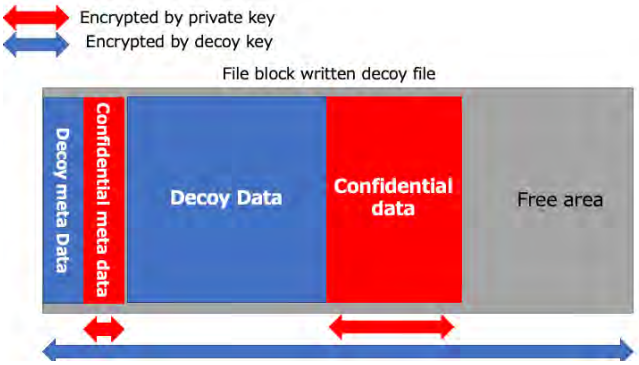


Figure 7: About File block that decoy and hidden FID become same

these data look like a random bit. The state where the FID is the same in the decoy and the hidden area may be destroyed by the file operation by the user. For example, when a decoy file is deleted, a file block that holds only confidential data occurs. After that, by accessing that confidential data, the deleted FID is observed. Section 5.2 defines the procedure for FID unification to maintain the FID unification during operation and to perform FID unification for the entire file system by initialization and so on.

5.1 Discussion about FID Unification

When considering an attack using knowledge from the use of disclosed decoy key, the worst case is when the attacker obtains all the information of the files existing in the decoy area, and the amount of knowledge about the storage area is maximized. This situation is realized when the decoy key is disclosed to only one attacker, there is no access by a legitimate user, and there is no use other than this attacker who changes the decoy area. At this time, the attacker can count and examine all the file IDs existing in the decoy area. After that, the access of the legitimate user is monitored and the file ID that appears based on this knowledge is inspected, and if the attacker observes a file ID that does not exist in the list of counted IDs, the attacker can suspect the existence of a hidden file. Here, consider another strategy, for example, a mechanism in which a common area accessible from a legitimate user and an attacker is created and a decoy file ID and a common file ID appear randomly. Assuming the worst-case above, where there is no legitimate user access and only this attacker to change the decoy area. The attacker can count and examine all the file IDs that exist in the common area and the decoy area in this case. Therefore, the common area is equivalent to the expanded decoy area and is not an effective measure. However, for example, assuming the existence of parallel access accompanied by an update to a file by a legitimate user when observed by an attacker, it seems possible to create a certain confidential margin.

To solve the problems of these methods, propose a method for a file in any hidden area to be embedded internally in one of the files in the decoy area. Here, the hidden file is recognized as a free area by the system that handles only the decoy area. Similarly, one directory in a hidden area should be

embedded inside one of the directories in the decoy area. As a premise, the data placed on the decoy side or hidden side in the persistent storage object read is recorded in the encrypted area of the persistent storage object. Which one should be accessed by the system can be safely confirmed and operated within TEE.

5.2 FID Unification Procedure

The operation of embedding a hidden file inside a file in a decoy area in an appropriate directory structure is called FID unification processing. In the FID unification process, the same FID can be used by embedding the contents of the hidden area file in the file located in the appropriate directory structure of the decoy area. Embedding this file is called a merge operation.

The ideal design of the FID unification process is executed on the server-side, monitoring the existence of non-identical FIDs for each file operation and merging files as necessary. However, if the merge process requires an amount of calculation in the design, the client may be blocked for a long time. To evaluate the amount of the processing, we implemented the FID unification process on the client-side to simplify the design enough for performance estimations. Figure 8 shows a simple client-side flow of the FID identification process. It is assumed that the FID unification process will be called after some extent of updates are performed on the client-side and when the client's PC idle state continues.

The FID unification process and merge operation are shown below. The FID unification process is used for initialization immediately after the proposed system is applied and for the reunification of unmerged files caused by file changes during operation. A simplified flow of the processing in the FID unification process is shown in Fig. 9. To merge the files in the hidden area into the files in the appropriate decoy area, the combination is searched to identify the appropriate location of the directory structure in the decoy area by the method shown in Algorithm1.

Algorithm1 operates as follows. First, get the pathname list of all directories in the decoy area and the hidden area, and pass them to Function Search as an argument. In Function Search, the directory position of the decoy area, which is the starting point of the FID unification process, is determined from the combination of all the directories of the decoy area and the hidden area. The determination method is as follows. From the directory position of the decoy area that is the starting point, each directory of the decoy area and the hidden area has a one-to-one correspondence, and the following unification suitability evaluation is calculated by the operation shown in Alogorithm2. The calculation method of the unification conformity assessment in Algorithm2 is explained in Section 5.2.1. The unification relevance evaluation consists of a mergeable flag and a conformance score. The mergeable flag is expressed by a boolean value indicating whether the directory combination can be merged, and when true, it indicates that the merge condition is satisfied. The calculation of the unification suitability evaluation is made into a memo, and when it is necessary to calculate the score of the same combination, it is called from the memo to shorten the

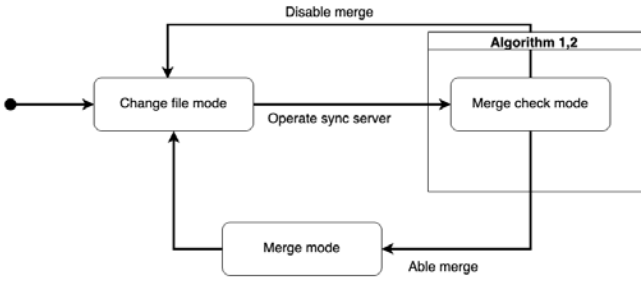


Figure 8: FID unification and operation mode on client PC

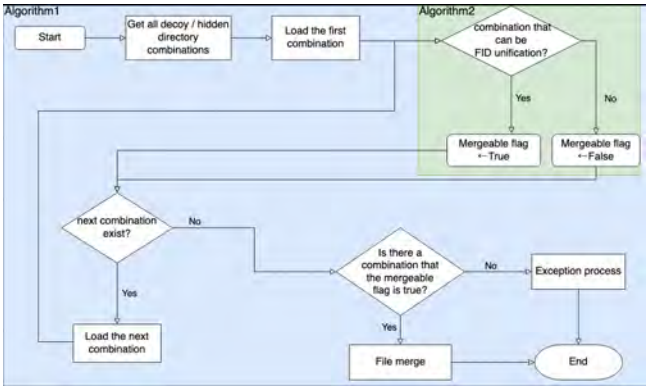


Figure 9: Simplified flow in FID unification

calculation.

Among all combinations, the one with the maximum optimal score is selected from the ones for which the mergeable flag is true, and the directory position of the decoy area that is the starting point of the FID unification process is determined. If none of all combinations have the mergeable flag set to true, the one with the highest matching score is taken out and judged to be at risk based on that combination. Algorithm1 performs the processing up to this point and returns the directory location of the decoy area that is the starting point of the FID unification process, or risk. The FID unification process recursively merges files or notifies the user of the risk based on the result received from Algorithm1. There is a risk, that is, the mergeable flag obtained by Algorithm2 is not true because there are not enough decoy files in the decoy directory to be merged. Therefore, it calculates how many files should be added to which directory in the decoy area, and also notifies the user.

5.2.1 Conformance Score

The conformance score integrates the conformance values for a specific file to be merged, and the larger the conformance score, the better the combination of the corresponding directories. A high match score means that the percentage of files in the decoy area where hidden area files are not embedded is high. In other words, if the conformance score is high, even if a new file on the hidden side is added or a file on the decoy side is deleted, there is a high possibility that the FID unification process can be performed only within the combination of the corresponding directories. It is used as a conformance

score of the FID unification process. The average size of the files in the directory is the size of the decoy area as pSize, and the size of the hidden area is as sSize. The number of files in the directory is the number of decoy areas as pNum, and the number of hidden areas as sNum. The mergeable flag is boolean valule of the following predicate equation.

$$(pSize/sSize + pNum/sNum)/2 >= 2$$

The conformance score is given by;

$$pNum/sNum$$

5.3 Invocation of FID Unification Process

We have two cases in the FID unification process invoked. First, FID unification processing is performed in the initialization of this system. Start by migrating the server to the same mode for FID unification processing. Perform the above FID unification, notify the user of the result if necessary, and then switch to normal mode. Next, the processing after the changes are made to the file will be explained. Think about the directory containing the file that is unmerged in the hidden area due to the modified file. Target hidden directory has already been merged into the decoy area directory. When re-merging a file that has been changed and is out of the merged state, first check whether it can be resolved in the corresponding directory. Algorithm1 is operated by passing only the corresponding directory as an argument. The relevance evaluation of the directories of the corresponding decoy area and the hidden area is calculated, and if the mergeable flag is true, remerge is performed in the corresponding decoy area directory and the hidden area directory. If the mergeable flag is false, the mode shifts to the unification mode in the same way as the initialization, and the FID unification process is started for all directories.

6 EXPERIMENT

In this section, to consider the validity of the design of PTEE FS, the evaluation is performed using the verification case from the following two points.

1. Processing time in normal access :
For the performance when applied to the cloud service of Section 6.1, first, we get the trace data of the file system acquired under assuming a realistic file group workload. The processing time is estimated applied our performance model [5] to the trace data got.

2. Processing time of FID unification invoked on demand
: Regarding the FID unification processing shown in the proposed method, the processing time when applied to the local file system is measured by the evaluation program. The evaluation program is implemented in Algorithm1and Algorithm2, algorithms are implemented in python. Estimate from actual measurement and extrapolation of processing time using the evaluation program.

6.1 Processing Time for Normal Access

For the implementation model of the proposed method, we estimate the performance when operating with the access pattern of the file synchronization service that is often seen in cloud storage services. We obtained Equation (1) as a model for calculating the overhead on the processing time per call to NFS RPC that needs protection [5].

OH (ms) represents the overhead of additional response time, and i (MB) represents the size of the transfer buffer during the enclave call. o (MB) represents the size of the transfer buffer when escaping the enclave.

Since it is known that the overhead increases when the transfer buffer at the time of enclave call exceeds 64 kB, the equation is divided into 2 by $i = 64kB$ when calculating the overhead. When the proposed system is applied to a cloud service, it can be seen that the effect on the performance of the file system can be evaluated from the difference in response time depending on whether TEE is used or not.

File synchronization is a practical example of cloud applications. Therefore, we will evaluate TEE performance again and use RSYNC [16] as a concrete file synchronization system. Trace the RSYNC execution traffic, give the size to be sent/received as an argument of Equation (1), calculate the estimation overhead, and evaluate its effectiveness.

$$OH = \begin{cases} 0.0097 + 0.0354o + 0.0115 & i < 0.064 \\ 0.9560i + 0.0354o - 0.0002 & i >= 0.064 \end{cases} \quad (1)$$

6.2 Processing Time of FID Unification Process

Regarding the FID unification process, we evaluate the effect of the smallest process among the FID unification processes that occur when used in a typical workload. When the same usage as the file system using the existing PDE concept is used, it is the most called process in the proposed method, and the impact on the user is significant. So, we evaluate the effect of the additional latency to gave by the FID unification process.

Create an environment that assumes the use case described in Section 7 of the proposed system, operate the FID unification process under that environment, and perform an evaluation experiment.

The performance is estimated and evaluated from the response time by the Python-based code for the effect of the FID unification process on the performance in a steady state. In this experiment, we evaluate a part of the performance of FID unification processing. In the experiment, we perform the process of passing only one directory to each argument of Alogrithm1, which occurs most in the FID unification process.

The FID unification process may be performed on all directories existing in the volume, but this is not the target of this experiment. As a PTEE FS system, it is conceivable that decryption and encryption will be performed before and after the FID unification process, but this is not the case in this experiment. The performance of the implementation in Python may be lower or almost unchanged than that implemented in

Algorithm 1 Algorithm to search for the best directory combination

```
function Serach(secretDirs, publicDirs)
    if secretDirs.length > publicDirs.length then ▷ If the hidden area has more directories than decoy area, no search is performed because there is no matching pattern.
        result ⇐ noMatch
        return result
    allMatch ⇐ allPermutaitionPatern(publicDirs) ▷ Calculate and substitute permutation patterns for directories in all decoy areas
    for i = 1, ··· , allMatch.length do ▷ Repeat the process for the number of allMatch
        for j = 1, secretFileNum do ▷ Repeat the process for the number of file in hidden area
            if resultMemo[j][allMatch[i][j]] = null then
                score ⇐ CheckMatchDir(secretDirs[j], publicDirs[allMatch[i][j]]) ▷ Get the mergeable flag and optimal value for a combination of a directory in a decoy area and a directory in a hidden area
                resultMemo[j][allMatch[i][j]] ⇐ score ▷ Save the score you have done once in a memo
            else
                score ⇐ resultMemo[j][allMatch[i][j]] ▷ When the same combination appears, call it from the memo
            throghScore[i].optimal ⇐ score.optimal ▷ Accumulate scores in the current permutation pattern
            if score.conform = false then
                throghScore[i].conform ⇐ false
                throghScore[i].optimal ⇐ −1
    if max(throughScore[i].optimal)! = 1 then ▷ Check if there is a mergeable combination
        result ⇐ argmax(throughScore.optimal) ▷ Get the permutation pattern with the highest conformance score
    else
        result ⇐ noMatch
    return result
```

C ++. Therefore, if a sufficient value is obtained in the evaluation performance based on Python, it is expected that the evaluation performance will be sufficient in C ++. The proposed system will be implemented in C ++, but this time we will evaluate it using the one implemented in Python.

To understand the additional latency provided by the FID unification process, compare the processing time of the FID unification process in one directory in the local directory to which PTEE FS has already been applied and the processing time of file synchronization with PTEE FS. The following use cases are assumed when applying the FID unification process.

It is assumed that the client uses this system when saving highly confidential data such as password lists and keys in the storage on the cloud for backup purposes. For this reason, the data saved in the hidden area is sufficiently smaller than the decoy area, and even if the FID unification process occurs due to file changes/deletion, it is assumed that most of the data can be resolved only in the relevant directory.

The assumed decoy use case workload is as follows. The ratio of requests for create: read: write is set to about 1: 40.5: 12.75, referring to the previous research by Leung et al. [6]. The size ratio of bytes flowing on the traffic by read: write is 2.1: 1. 70% of the file size to be handled is a file less than 1KB, 10% is 1 to 100KB, and the remaining 20% is 100KB to 1MB. The hidden area use case assumes pgp key management, and the key used is 2048 bits of RSA encryption key. The processing time of the FID unification process is evaluated as the additional latency that gives the proposed system as it is, and the effect of the additional latency on the file system is estimated and evaluated concerning an example of file sharing.

6.3 Experimental Method

6.3.1 Experiment of Processing Time with Normal Access

We prepared a server/client capable of RSYNC communication and synchronized the files. The server configuration of RSYNC is mainly done by directly using the file server in the company or school, but it is assumed that the changes are synchronized on the client terminal at home for remote work. Assuming that the size of the data loaded for uploading and downloading the modified file is the same, here we trace the traffic between the uploading client and server. Figure 10 shows the configuration for data acquisition. The server was mounted on the proxy server, rsync was performed from the client to the proxy server, and NFS traffic between the proxy server and server was observed. From the trace data, the RPC READ and WIRTE call get the series of buffer sizes to be passed.

The series of buffer sizes was used as the buffer size for calling and escaping the enclave in Equation (1), the execution time was measured, and the evaluation was performed as the overhead of Intel SGX when using this system.

The response time of the entire process was measured 10 times for file synchronization of the same size, and the average process time was used. The files shown in Table 2 were used as the files that are changed synchronously by RSYNC. The average size of the text file was 11263 bytes, and the average size of the binary file was 216269 bytes. Four of the files have a file size that exceeds 64 kB, which increases the TEE response time to input/output these files.

Algorithm 2 Algorithm for calculating the unification aptitude score

```
1: function CheckMatchDir(secretDir, publicDir)
2:     publicFiles ⇐ getAllFiles(publicDir)          ▷ Get the file entry for the target decoy area directory
3:     secretFiles ⇐ getAllFiles(secretDir)          ▷ Get the file entry for the target hidden area directory
4:     publicFileSizeMean ⇐ publicFiles.sumSize/publicFiles.fileNum   ▷ Get the average file size of the decoy area directory
5:     secretFileSizeMean ⇐ secretFiles.sumSize/secretFiles.fileNum   ▷ Get the average file size of the hidden area directory
6:     if publicFileSizeMean/secretFileSizeMean > 1 then   ▷ Check if the average file size meets the conditions
7:         sizeScore ⇐ true
8:     else
9:         sizeScore ⇐ false
10:    if publicFiles.fileNum > secretFiles.fileNum then   ▷ Check if number of files meets the conditions
11:        fileNumScore ⇐ true
12:    else
13:        fileNumScore ⇐ false
14:    if sizeScore&fileNumScore then
15:        conform ⇐ true
16:    else
17:        conform ⇐ false
18:    if conform then   ▷ Check if both the average file size and number of files meets the conditions
19:        optimal ⇐ publicFiles.fileNum/secretFiles.fileNum   ▷ If the mergeable flag is true, the optimum value is calculated.
20:    else
21:        optimal ⇐ 0
22:    return (conform, optimal)   ▷ Returns mergeable flag, conformance score
```

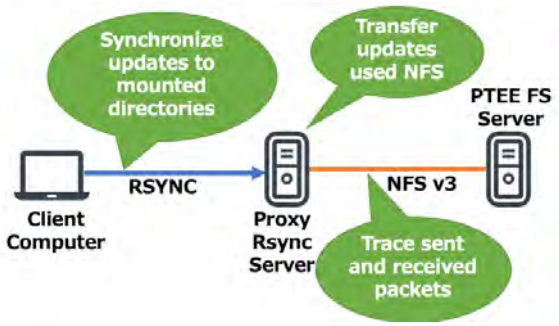


Figure 10: Server configuration in the experiment

Table 2: File size and type used in the experiment

File size (byte)	File type	File size (byte)	File type
55358	Text	1444438	Binary
22113	Text	223065	Binary
11602	Text	97252	Binary
11101	Text	96516	Binary
4284	Text	35184	Binary
2651	Text	33033	Binary
2607	Text	10244	Binary
2317	Text	6527	Binary
567	Text	165	Binary
28	Text		

6.3.2 Experiment of Processing Time of FID Unification Processing

We prepared a decoy area directory and a hidden area directory according to the workload of the use case and performed the FID unification processing. For the decoy area directory, referring to the existing research [6] by Leung et al., We prepared 70% for files with file sizes from 1 byte to 1 kB, 10% for files with a file size of 1 kB to 100 kB, and 20% for files with a file size of 100 kB or more. We prepared three types of files, 30 and 50, contained in one decoy directory. For the hidden directory, referring to the key management of pgp, it was decided that the public key and private key pair of public key authentication, which is asymmetric authentication, is assigned to each directory. Two types of files, 10 and 20, are prepared in one hidden directory. If the number of files is 10, there are 5 public/private key pairs, and if the number of files is 20, there are 10 public/private key pairs. In the actual experiment, assuming that the user uses so that the number of files on the hidden area side is sufficiently small in the PDE file system. We experiment with 2 pairs of decoy area directories and hidden area directories. One pair is that the number of files in the decoy area directory is 30 and the number of hidden area directories is 10. The other is that number of files in the decoy area directory was 50 and the number of hidden area directories was 20. We execute the FID unification processing in these 2 pairs and the execution time was measured.

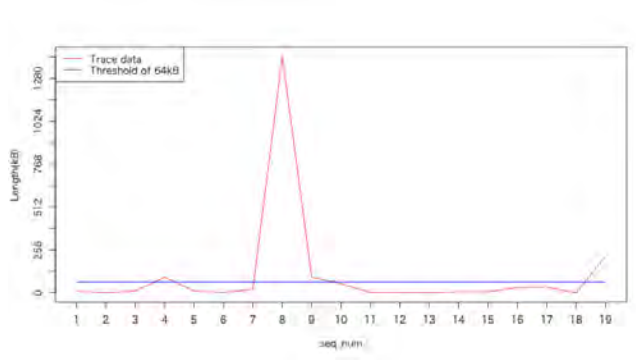


Figure 11: Transfer data size and part of the NFS Call series observed during rsync

6.4 Experiment Environment

A computer with RSYNC [16] and NFS Version 3 [17] installed was used as the server and client for the experiment. Wire Shark [18] was used to trace the traffic. The network bandwidth in the experimental environment was 6.90 MB/s.

7 EVALUATION

7.1 Experiment of Processing Time with Normal Access

The average response time of the entire process on the client/server in the experimental environment was 18336.2 ms, and the standard deviation was 2373. Figure 11 shows a series of transfer data sizes that appear in the NFS traffic trace when RSYNC is executed. [5]. Substituting this transfer data size sequence into Equation (1) for calculating the increase in response time, which is the overhead (OH) per NFS call, gives 1.9 ms from the total sequence. Therefore, the estimated processing time when using the proposed system is 18338.1 ms. Therefore, the ratio of overhead to the processing time is 0.010%, and it is considered that the overhead required when using TEE is acceptable.

7.2 Experiment of Processing Time of FID Unification Processing

The average time required for FID unification is 133.8 ms with 30 files in the decoy area and 10 files in the hidden area, and 134.6 ms with 50 files in the decoy area and 20 files in the hidden area.

8 CONCLUSION

We improved our design of Plausibly Deniable Distributed File Systems to obtain resistance to key disclosure attacks. Two experiments were conducted and evaluated in terms of performance to validate the design. In the experiments and evaluations, we discussed the processing time for normal access in use cases applied to cloud services. In the file synchronization use case using rsync, the increased ratio in response time by the use of TEE is estimated with a measured figure. The result is 0.010%. Increase for the whole operation, which is considered to be acceptable overhead by TEE. To provide the resistance to exploiting the knowledge from the use of disclosed the decoy key, we added new functionalities of the FID unification as a countermeasure to memory inspection attacks. The processing time of the FID unification process invoked on-demand was tested and evaluated using a program implemented in python.

The processing time of the FID unification process is 1133.8 ms in an environment with 30 files in the decoy area and 10 files in the hidden area, and 134.6 ms with 50 files in the decoy area and 20 files in the hidden area. Therefore, the additional latency due to the FID unification process may be tolerated. However, in this evaluation, the cost of encryption processing is not added to the processing time. Examination of a performance model that includes these is future work.

REFERENCES

[1] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. “Deniable Encryption”. In B. S. Kaliski, editor, Advances in Cryptology — CRYPTO ’97, Lecture Notes in Computer Science, pp. 90–104. Springer Berlin Heidelberg, (1997).

[2] Ministry of Internal Affairs and Communications. “Ministry of Internal Affairs and Communications | 2019 WHITE PAPER Information and Communications in Japan | ICT Data on the ICT Field”. https://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2019/chapter-3.pdf#page=1. (accessed 2019-10-22).

[3] G. Kumparak. “Capital One hacked, over 100 million customers affected”. https://social.techcrunch.com/2019/07/29/capital-one-hacked-over-100-million-customers-affected/. (accessed 2021-06-11).

[4] H. Ryo, T. Sasaki, Y. Morita, K. Miyoshi, and T. Kobayashi. “A Study on Access Control for Devices with Trusted Execution Environments”. Proceedings of Computer Security Symposium 2017, Vol. 2017, No. 2, (2017/10/16).

[5] R. Shibazaki, H. Inamura, and Y. Nakamura. “Design of Encrypted File System Using the Concept of PDE”. Proceedings of the 82th National Convention of IPSJ, Vol. 82, No. 1, pp. 103–104, (2020).

[6] A. W. Leung, S. Pasupathy, G. Goodson, and E. L. Miller. “Measurement and Analysis of Large-Scale Network File System Workloads”. In 2008 USENIX Annual Technical Conference (USENIX ATC 08), (2008/6).

[7] R. Anderson, R. Needham, and A. Shamir. “The Steganographic File System”. In Information Hiding, pp. 73–82. Springer, Berlin, Heidelberg, (1998/4/14).

[8] B. Chang, F. Zhang, B. Chen, Y. Li, W. Zhu, Y. Tian, Z. Wang, and A. Ching. “MobiCeal: Towards Secure and Practical Plausibly Deniable Encryption on Mobile Devices”. In 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 454–465, (June 2018).

[9] S. Jia, L. Xia, B. Chen, and P. Liu. “DEFTL: Implementing Plausibly Deniable Encryption in Flash Translation Layer”. In Proceedings of the 2017 ACM SIGSAC

Conference on Computer and Communications Security, CCS '17, pp. 2217–2229, New York, NY, USA, (2017). ACM.

[10] A. Zuck, U. Shriki, D. E. Porter, and D. Tsafrir. "Preserving Hidden Data with an Ever-Changing Disk". In Proceedings of the 16th Workshop on Hot Topics in Operating Systems, HotOS '17, pp. 50–55, New York, NY, USA, (2017). ACM.

[11] "Intel® Software Guard Extensions (Intel® SGX)". https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html. (accessed 2021-06-11).

[12] Intel® Software Developer Zone. "SDK — Intel® Software Guard Extensions". https://software.intel.com/en-us/sgx/sdk. (accessed 2019-10-26).

[13] R. Ahmed, Z. Zaheer, R. Li, and R. Ricci. "Harpocrates: Giving Out Your Secrets and Keeping Them Too". In 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp. 103–114, Seattle, WA, USA, (10/2018). IEEE.

[14] A. Gjerdrum, R. Pettersen, H. D. Johansen, and D. Johansen. "Performance of Trusted Computing in Cloud Infrastructures with Intel SGX:". In Proceedings of the 7th International Conference on Cloud Computing and Services Science, pp. 696–703, Porto, Portugal, (2017/4). SCITEPRESS - Science and Technology Publications.

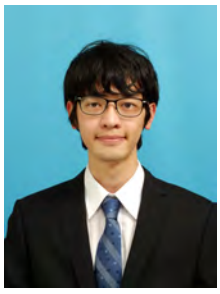
[15] Trusted Computing Group. "TCG Storage Security Subsystem Class: Opal", (2022/1/24). (accessed 2022-02-16).

[16] rsync. "rsync". https://rsync.samba.org/. (accessed 2020-05-08).

[17] B. Callaghan, B. Pawlowski, and P. Staubach. "NFS Version 3 Protocol Specification". https://www.ietf.org/rfc/rfc1813.txt, (1995 June). (accessed 2019-12-24).

[18] WIRESHARK. "Wireshark". https://www.wireshark.org. (accessed 2020-02-13).

(Received: October 26, 2021)
(Accepted: April 26, 2022)



Ryouga Shibazaki received B.E. and M.S. degree in from Future University Hakodate in 2020 and 2022. His research interests include cloud file system and OS security. He is currently an engineer in MEITEC CORPORATION.

Hiroshi Inamura He is a professor of School of Systems Information Science, Future University Hakodate, since 2016. His current research interests include mobile computing, system software for smart devices, IoT network and their security. He was an executive research engineer in NTT docomo, Inc. He received B.E., M.E. and D.E. degree in Keio University, Japan. He is a member of IPSJ, IEICE, ACM and IEEE.

Yoshitaka Nakamura received B.E., M.S., and Ph.D. degrees from Osaka University in 2002, 2004 and 2007, respectively. He is currently an associate professor at the Faculty of Engineering , Kyoto Tachibana University. His research interest includes information security and ubiquitous computing. He is a member of IEEE, IEICE, and IPSJ.

Submission Guidance

About IJIS

International Journal of Informatics Society (ISSN 1883-4566) is published in one volume of three issues a year. One should be a member of Informatics Society for the submission of the article at least. A submission article is reviewed at least two reviewer. The online version of the journal is available at the following site: http://www.infsoc.org.

Aims and Scope of Informatics Society

The evolution of informatics heralds a new information society. It provides more convenience to our life. Informatics and technologies have been integrated by various fields. For example, mathematics, linguistics, logics, engineering, and new fields will join it. Especially, we are continuing to maintain an awareness of informatics and communication convergence. Informatics Society is the organization that tries to develop informatics and technologies with this convergence. International Journal of Informatics Society (IJIS) is the journal of Informatics Society.

Areas of interest include, but are not limited to:

- Internet of Things (IoT)
- Smart Cities, Communities, and Spaces
- Big Data, Artificial Intelligence, and Data Science
- Network Systems and Protocols
- Computer Supported Cooperative Work and Groupware
- Security and Privacy in Information Systems
- Intelligent Transportation System
- Distributed Computing
- Multi-media communication
- Information systems
- Mobile computing
- Ubiquitous computing

Instruction to Authors

For detailed instructions please refer to the Authors Corner on our Web site, http://www.infsoc.org/.

Submission of manuscripts: There is no limitation of page count as full papers, each of which will be subject to a full review process. An electronic, PDF-based submission of papers is mandatory. Download and use the LaTeX2e or Microsoft Word sample IJIS formats.

http://www.infsoc.org/IJIS-Format.pdf

LaTeX2e

LaTeX2e files (ZIP) http://www.infsoc.org/template_IJIS.zip

Microsoft Word™

Sample document http://www.infsoc.org/sample_IJIS.doc

Please send the PDF file of your paper to secretariat@infsoc.org with the following information:

Title, Author: Name (Affiliation), Name (Affiliation), Corresponding Author. Address, Tel, Fax, E-mail:

Copyright

For all copying, reprint, or republication permission, write to: Copyrights and Permissions Department, Informatics Society, secretariat@infsoc.org.

Publisher

Address: Informatics Laboratory, 3-41 Tsujimachi, Kitaku, Nagoya 462-0032, Japan

E-mail: secretariat@infsoc.org

CONTENTS