Regular Paper

Moving Object Detection Method for Moving Cameras Using Frames Subtraction Corrected by Optical Flow

Tsukasa Kudo†

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

Abstract - With the progress of the Internet of Things (IoT), numerous videos have been recorded by various mobile cameras, such as in-vehicle and wearable cameras. Therefore, it has become necessary to organize target information automatically using image recognition. This often requires the extraction of the target area from the video frames by object detection as preprocessing. However, it often becomes difficult to detect a moving object efficiently in a video recorded by a moving camera depending on the video's environment, which can include complex backgrounds. In this study, I propose a method for extracting the target area by creating a subtraction of target images between adjacent frames. In this method, the background-position is aligned based on the displacement vector in the optical flow and then subtracted. Moreover, when the target areas are consecutively extracted from the video, I show its accuracy can be improved by comparing the extracted area size with its moving average.

Keywords: optical flow, moving camera, wearable camera, video, object detection, frame subtraction, IoT

1 INTRODUCTION

With the progress of the Internet of Things (IoT), various sensors are connected to the network, and a large amount of data is collected and analyzed. For example, regarding videos, a large number of cameras have been deployed and used for various purposes, such as monitoring traffic conditions and the insides of buildings. This has led to a rapid increase in the number of videos that need to be processed. Therefore, it has become necessary to extract target information automatically from such a vast amount of video data.

On the other hand, deep learning's effectiveness in image recognition has been demonstrated [4], [9], [23] and applied to various fields, such as handwritten character recognition and face recognition [5], [1]. Furthermore, it has been used for image recognition from a large number of videos, such as automatic target or object recognition and abnormality detection [16].

In this study, I have been attempting to automate inventory management in factories. Factory workers are equipped with wearable cameras and discriminate their current location and target objects using deep learning from images extracted from videos. Since this is for indoor use, backgrounds do not move but are instead diverse, including various kinds of walls and equipment. As a result, for the location, I showed that the training data could be collected efficiently, and the discrimination accuracy could be improved by continuous discrimination against the same target while comparing the results [11]. For the objects, they were held in hand and recorded by the cameras. However, when the objects were small, I found that the accuracy deteriorated due to the background's influence, especially in the case of a complex background.

The image is often preprocessed for deep learning to suppress the influence of the background, wherein a relatively small area including the target (hereinafter, target area) is extracted from the image. For example, in face recognition, the face area in an image is first extracted using Haar-like features then the face is recognized within this area [24]. Furthermore, various studies have been conducted on object detection in images, including video frames. However, I could not find an efficient method for the detection of a moving object from frames recorded by a moving camera, as mentioned above.

For this problem, I propose a method for extracting the target from backgrounds utilizing the optical flow in this study. The two frames are superimposed, and their background positions are matched based on the optical flow, then the subtraction between the frames is generated. As a result, the target area is extracted in this subtraction because the target's motion is different from the background. Along with this, I propose a method to identify the frame of the video, in which the target is observed, by utilizing the difference between the adjacent frames. I also show that the extraction accuracy of the target area can be improved by comparing its individual size with the moving average.

The remainder of this paper is organized as follows. Section 2 shows the motivation of this study and related works, and Sec. 3 proposes a target area extraction method based on the optical flow. Section 4 shows the implementation of this method in the experimental system, and Sec. 5 shows the experimental results. Section 7 concludes this paper.

2 MIOTIVATION AND RELATED WORKS

2.1 Motivation of This Study

I have been working on improving inventory management in a machinery factory, where various parts are stored in bulk containers. Since these inventory quantities cannot be counted visually from outside the container, their stock-taking is a heavy workload for the workers. For this problem, I showed the inventory quantity can be estimated with practical accuracy from the image of the bulk container by applying deep learning [10]. However, the next challenge was to find an

efficient way to collect these images, since a factory usually houses more than one thousand bulk containers.

Consequently, I focused on the fact that inventory changes when workers replenish or ship the parts. I conceived to estimate stocks using images extracted from videos recorded by cameras that the workers wore. Firstly, it was necessary to detect a worker's approach to a bulk container and the handling of machine parts. I used images extracted automatically from videos to detect the former with a certain accuracy by using the deep learning model trained to distinguish the target room's entrance and equipment [11].

I noticed for the latter that the worker needed to hold the parts in his hand for the inventory work. In other words, as shown in Fig. 1 (a), if the object held in hand can be recognized, the target parts can be distinguished automatically. So, I collected various such images to train the model and evaluate the accuracy of distinguishing the target. As a result, I found that the accuracy deteriorated for small targets because of the background's influence.

Concretely, I evaluated the discrimination accuracy of three groups of objects (large, small, and thin) using a multi-class classification model of deep learning. Firstly, the photos taken in the same room were divided into training and test data. Then, the model was trained with the training data and evaluated with the test data. Its accuracy was 84.8%. Next, the model was evaluated with photos taken in another room. As a result, the accuracy was 100% for the large objects, which occupy more than half of each photo, but it deteriorated to 72.5% as a whole.

Preprocessing is usually performed for such problems to extract the target area from the image, and then the target is recognized by using this area image. Various methods using optical flow have been proposed to detect a moving object in a video image recorded by a moving camera [26]. The optical flow shows the displacement vector between a pixel in one frame and another frame's corresponding one. This applies to all frame pixels in the dense optical flow [7].

First, I performed an experiment in which the target was moved in front of the wearable camera. The target was then extracted based on the difference in the optical flow between the background and the target. As a result, the target could be extracted with high accuracy when the background was flat. However, I found a problem in the case shown in Fig. 1 (a), where the background was complex, and the target was flat. As shown in Fig. 1 (b), the target area was split, and it was difficult to extract it completely.

The process of this extraction is as follows. Figure 1 (c) shows the optical flow's displacement direction where brightness increases counterclockwise from zero (black) pointing to the right. Figure 1 (d) shows the normalized displacement distance, in which the higher the brightness, the larger the displacement distance. As shown in (c) and (d), the brightness distribution was not separated between the background and target. Furthermore, the target's flat part was not distinguished as the moving part, because both of its displacement direction and distance were almost zero (black).

Figure 1 (e) is a binarization of the brightness in (c), created by making 20% of the luminance range in the target area

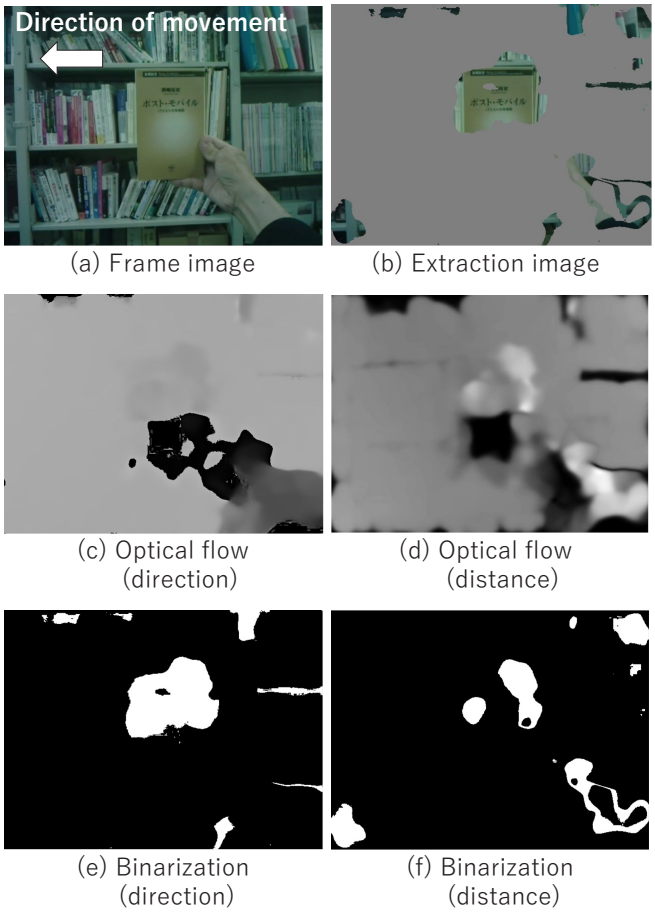

Figure 1: Problem of target extraction using optical flow

white. The central white area corresponds to the top part of the book in (a). Similarly, in Fig. (f), the white areas at the center and bottom right correspond to the book's top part and the hand in (a). These white areas were the target area estimated from the optical flow, and (b) was created by superimposing the original image on the combination of the white areas in (e) and (f).

This study's motivation is to develop an efficient method that can accurately extract the target area, including the moving target shot by a moving camera, even in the case of complex backgrounds. This study also targets stationary backgrounds and rigid objects that can be held and moved by the hand, as shown in Fig. 1 (a). Regarding the efficiency requirements mentioned above, the target areas must be consecutively extracted from the video, although there are some intervals.

2.2 Related Works

Currently, various kinds of mobile cameras have become ubiquitous, such as in-vehicle cameras and mobile phone cameras, in addition to the wearable cameras in this study. Therefore, many studies have been conducted to detect and track a moving object from videos recorded by mobile cameras.

The following three methods are widely used to extract a target from the frames of video: the background subtraction method, the frame subtraction method, and utilization of op-

tical flow. In the background subtraction method, the target is extracted by the difference between the background image and the image in which the target appears in front of the background. In the frame subtraction method, the target is extracted using the subtraction between each frame. However, in both cases, it is assumed that the background image is fixed, and it is difficult to apply directly to a moving camera [15].

On the other hand, various studies using optical flow have been conducted for free-moving cameras [26]. The most direct ways use the displacement difference in the optical flow between the background and the moving target [17], [25]. However, as shown in Fig. 1, this creates a problem in the case of a complex background.

Some methods have been proposed for this problem. For example, combining different methods, performing analysis over many frames, and utilizing deep learning [2], [8], [15]. However, there were issues in terms of efficiency, such as processing complexity and model training. Moreover, methods to estimate the camera motion by utilizing optical flow have been proposed [6], [21]. However, their aim was motion recognition. Furthermore, some methods utilize the optical flow to reconstruct the background, and the target is detected by the background subtraction [22], [27]. However, these methods target seamless backgrounds or pan-tilt-zoom cameras. Therefore, it is challenging to apply these methods to the wearable camera shooting the complex background shown in Fig. 1.

Regarding object detection and classification of detected objects, studies utilizing deep learning have been progressing rapidly. Faster R-CNN performed both of them in a lump by collective end-to-end training of both models [20], and YOLO executed them with a single neural network to improve efficiency [19]. Concerning different scale objects, SSD made it possible to process them collectively [14], and RetinaNet improved efficiency by introducing the Feature Pyramid Network (FPN) and improving the error function [12], [13]. Then, M2Det has further improved accuracy and efficiency by introducing the new FPN and error function [28]. However, since these methods target each image, they are not suitable for object detection targeted by this study, which detects the objects moving in front of the background using multiple frames of a video. Furthermore, it is necessary to prepare a large amount of training data and train the model to apply these deep learning methods to the individual target shown in Fig. 1.

In summary, an efficient method has not yet been proposed to extract an area that includes a moving target in front of a complex background from a video recorded by a moving camera.

3 PROPOSED METHOD

This study aims to extract a target area from frames of a video shot by a wearable camera, as shown in Sec. 2.1, by utilizing the optical flow. First, it is necessary to extract suitable frames to calculate the optical flow between them, namely, frame pairs without considerable blur and observing the same target. The proposed method determines those frame pairs based on the difference between adjacent frames. Figure 2

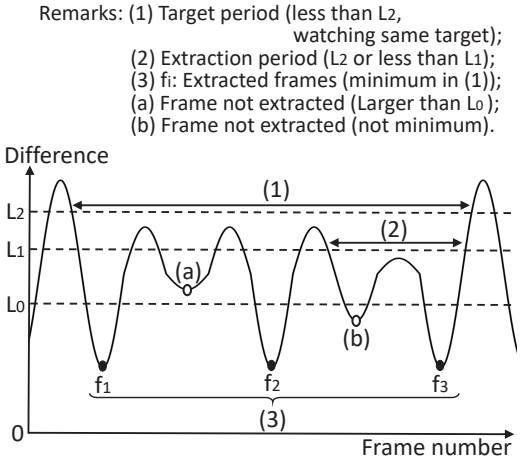

Figure 2: Transition of difference between adjacent video frames

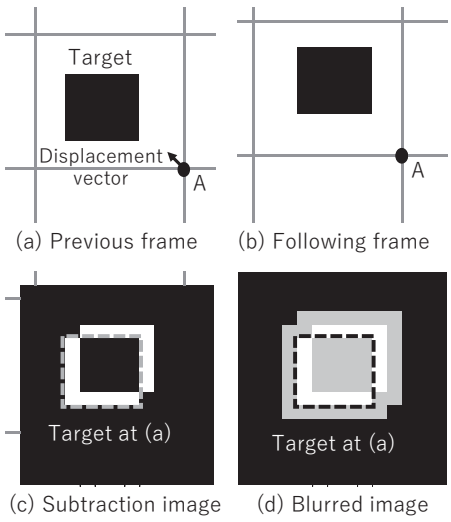

Figure 3: Target extraction method using optical flow and frame subtraction

shows an example of the transition of the difference between adjacent frames.

Figure 2 (1) shows a target period surrounded by large difference points exceeding the threshold L_2. It is supposed the same target is continuously observed in this period. Incidentally, the viewpoint is supposed to have moved from one object to another at both ends. Only the frame with the least difference is then extracted for each extraction period shown in Fig. 2 (2) to obtain slightly different images for the same target. In the case of Fig. 2, they are f_1, f_2, and f_3. Here, the differences in the frames of (2) are L_1 or less, and the difference in each frame of (3) is L_0 or less to extract an image with little blur. The values of these thresholds are $L_0 < L_1 < L_2$. The frames (a) and (b) in Fig. 2 are not extracted because the difference of (a) is greater than L_0, and (b) is not minimum in (2).

Figure 3 shows the target area extraction process in the proposed method. (a) shows the video's previous frame f_1, and (b) shows the following frame f_2. Here, the black rectangle is the target. As shown in Sec. 2, the background is assumed

to be stationary, so the difference in the background between (a) and (b) is only the parallel translation. Therefore, when the displacement vector of the optical flow between (a) and (b) is obtained for one point A in the background, (b) can be superimposed to (a) so that their background positions match. As a result, their subtraction in the background becomes zero (black), as shown in (c).

At this time, if the target was moved, there would be a gap in the target area between these two frames. This gap becomes the difference from the background, as shown in the white area in Fig. 3 (c). Note that there is no difference in the central area of the gap where the target overlaps in both frames, so it becomes black. Therefore, the target at (a), illustrated by the gray-dashed rectangle in (c), is included in this gap. Here, a part of the gap is outside this rectangle. However, since this study aims to narrow down the area where the target exists, this is acceptable.

As shown in the gray area in Fig. 3 (d), the image is blurred to enlarge the target area and connect the two white areas in (c); the entire interior of the area is also targeted. Further, some areas are set as no difference (black) to exclude the different areas between frames (a) and (b), which is caused by the displacement of (b). In the case of (c), they are at the left end and the top end (white). Finally, the target area can be extracted, which includes the target at (a) indicated by the black dashed line in (d) by extracting these white and gray areas as a continuous area.

4 IMPLEMENTATION

The functions described in Sec. 3 were implemented on a PC running Windows 10 to evaluate the proposed method. The programming language used was Python Ver.3.6; opencv-python Ver.4.1.0.25 was used for image processing. Below, the implementation of the proposed method is shown, along with two improvements.

4.1 Implementation of Proposed Method

First, the transition of the difference between the adjacent frames shown in Fig. 2 is calculated to extract target frames from a video. The frame image is converted to grayscale, and a histogram of the number of pixels with brightness j is created. This number is expressed by n_{ij} using the frame number i and brightness j. Then, as shown in Eq. (1), the absolute difference of D_i between previous and the following frames is calculated by weighting with luminance j and dividing by the number of pixels (N). Here, the division is to prevent fluctuations on D_i due to the number of pixels.

$$D_i = \sum_{j=0}^{255} |n_{i+1j} * j - n_{ij} * j|/N \tag{1}$$

D_i corresponds to "Difference" in Fig. 2. It is determined that the same target is observed while D_i is less than the threshold L_2. The frame with the smallest D_i is selected for each period where D_i is equal to or less than the threshold L_1. When it is equal to or less than the threshold L_0, this frame is extracted such as f_1 in Fig. 2.

The optical flow was calculated using the calcOpticalFlowFarneback method of opencv-python [18]. This is an implementation of the polynomial expansion algorithm [3]. In this method, I set the parameter as follows: the polynomial area was 5, the polygon width was 0.5, the window size was 60, the pyramid size was 0.5, its level was 3, and Gaussian kernel was used for prior blurring. Figure 4 shows an example of intermediate processing results of the proposed method for the same frame image shown in Fig. 1 (a). (1) and (2) show the displacement direction and distance of the optical flow. They are similar to Fig. 2 (b) and (c). Rectangles with a 10% margin from the edges are added for the following explanation. The background displacement is calculated using the median luminance of outside each rectangle. Note, the influence of the hand in the bottom right can be excluded by using the median.

If the target's displacement is too small, a clear subtraction image is not created, as shown in Fig. 3 (c). When the maximum displacement distance of the optical flow is less than the threshold, the subsequent frame is rejected, and the following one is tested. The threshold was set to 12 pixels based on the experimental results described later in Sec. 4.2. If the displacement was sufficient, the subsequent frame shifted based on the optical flow's background displacement, and its subtraction image with the previous frame is created. The distribution of luminance is then expanded by histogram equalization, as shown in Fig. 4 (3). If there is a difference between both frame regions due to the subsequent frame's shift, this part is filled with black, as shown at the left and bottom edges of (3).

The image is blurred by the median filter to combine image fragments as continuous regions, such as the book title, as shown in Fig. 3 (4). The kernel size of this median filter was set to 15×15 pixels. However, the corners of the targe are rounded by applying the median filter, as shown in (4). The measure for this is mentioned later in (7) convex hull and Sec. 4.3. (5) is a binarization image of (4) in which the area of the brightness above the threshold was extracted as white parts. This threshold was 159, which was the median value 127 plus the error 32. As a result, the part corresponding to the target was extracted as a continuous area, as shown in (5). Then, the white area was eroded and dilated to exclude noise and separate unnecessarily combined regions outside the target, as shown in (6). They were each performed five times with kernel size set to 3×3 pixels.

As shown in Fig. 3 (7), the target area is created by the following processing from (6). First, the largest continuous region of white parts is selected as the area, including the target. In the case of (6), the central region is applicable. Second, the entire part surrounded by the convex hull contour was extracted. Third, to recover the corners, which are rounded by the median filter shown in (4), the area is dilated. This was performed five times with the same kernel size as (6). Finally, the target region is extracted by superimposing the original image on (7), as shown in Fig. 3 (8). The part other than the target area is painted gray.

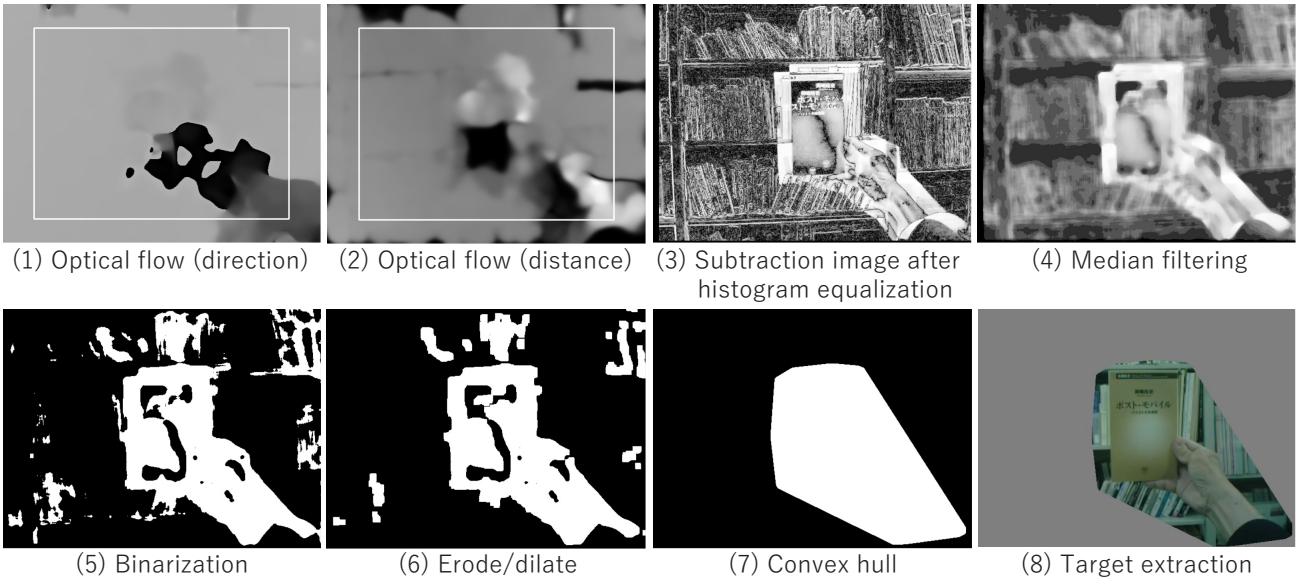

Figure 4: Intermediate processing result images in the proposed method

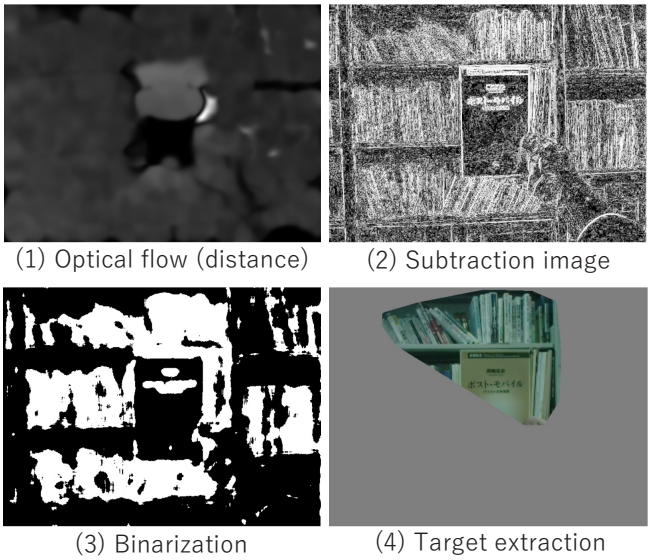

Figure 5: Analysis of partial missing of target area

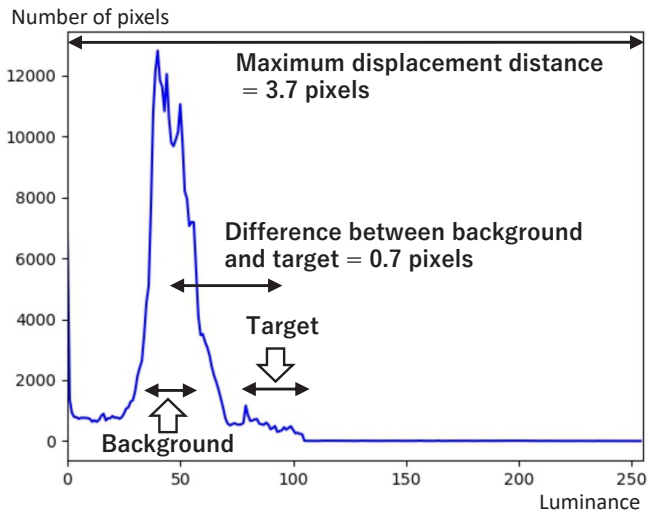

Figure 6: Histogram of distance in optical flow

4.2 Determination of Displacement Distance Threshold

As mentioned in Sec. 4.1, when the displacement of the target is too small, a clear subtraction image is not created, as shown in Fig. 3 (c). Figure 5 shows an example of such a case, and (1) shows the displacement distance of the optical flow. Figure 6 shows its histogram of luminance, and the maximum displacement distance is 3.7 pixels, which corresponds to a luminance of 255. The part with the maximum number of pixels corresponds to the background; the right part corresponds to the target. The difference in displacement distance between them is 0.7 pixels, that is, less than one pixel.

In this case, the luminance of the target subtraction image was small, and it was nearly the same as that of the background, as shown in Fig. 5 (2). Therefore, the background difference became too large in (3) binarization. As a result, the whole target area was not extracted, as shown in (4).

To find the displacement distance's threshold, I created simple images with a white background and a rectangle whose horizontal position changed stepwise. Then, the correlation between the displacement distances of the rectangle and the clarity of the subtraction image was evaluated. Besides, this image was created using the same procedure as in Fig. 5 (2). As a result, the difference between the target and background clearly appeared when the maximum displacement distance was 12 pixels or more. Based on this result, the threshold was set to 12 pixels in this implementation, as mentioned in Sec. 4.1

4.3 Restoration of Target's Corners

Figure 7 shows the cause and influence of the rounding of the target's corners by the median filter's blur shown in Fig.

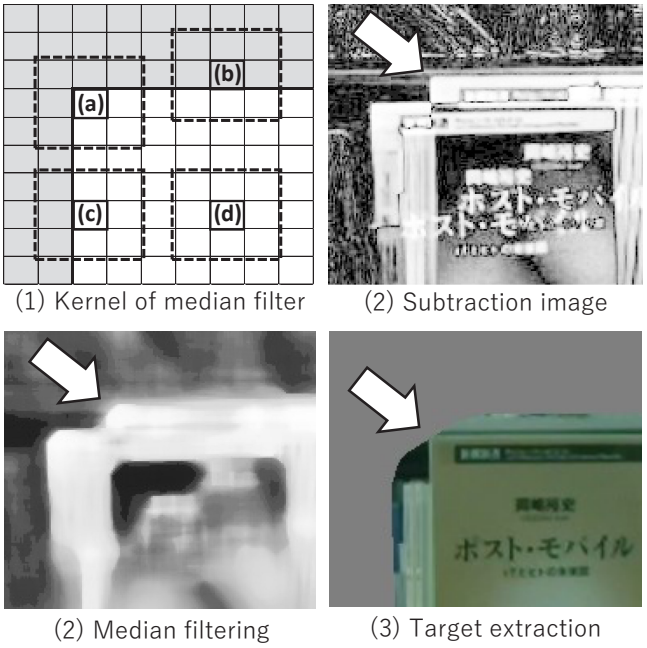

Figure 7: Rounded corners due to median filter

4 (4). Figure 7 (1) shows the target's boundary in the image. Each square corresponds to a pixel, and the white and gray areas correspond to the white and black areas of the image in Fig. 4 (3), respectively. Each broken line shows the median filter's kernel for the points (a) to (d), respectively, and the values of these points after blur are the median values of the broken line region. The value of the point (d), whose kernel is in the white area, and the boundary points (b) and (c) do not change even after blur. However, the kernel of the corner point (a) consists of five black pixels and four white pixels, so it becomes black by the median filter, and the corner is rounded.

Figure 7 (2) and (3) show the images before and after blur respectively, and the corner in (3) was rounded. As a result, in the final target extraction image, the corner was missing as shown in (3).

In this implementation, dilation of the convex hull shown in Fig. 4 (7) was performed eight times with a kernel of 3×3 pixels to prevent this missing of corners. This number corresponded to half of the median filter kernel size, which was 15×15 pixels. Note that dilation was also performed at the stage of Fig. 4 (6). However, if additional dilation was performed here, binding with fragments in the outside target area might occur. This process was performed in Fig. 4 (7).

5 EXPERIMENTS AND EVALUATIONS

The effectiveness of the proposed method was evaluated by performing the following two experiments. The first one aimed to evaluate the effectiveness in plural environments, and the extractions of the target area were performed for the combinations of three targets and four backgrounds. The second experiment aimed to evaluate the achievement of this study's purpose. In the environment shown in Fig. 1, automatic target extraction was performed continuously for a certain period of time.

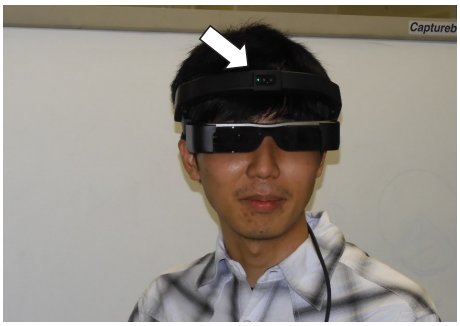

Figure 8: Wearable camera used in experiment

Table 1: Target object in experiment

No.	Type	Used target
B1	Clear contour	Book 1
B2	Clear figure	Book 2
B3	Flat	Book 3

A wearable camera recorded videos in a laboratory, for which headset EPSON MOVERIO Pro BT-2000B shown in Fig. 8 was used. It secured a video camera to the forehead, as shown by the arrow in Fig. 8. The video was displayed on the see-through glasses. It was set up with a frame size of 640×480 dpi at 30 frames per second. Images were extracted from videos shot by this camera using the experimental system mentioned in Sec. 4.

5.1 Evaluation in Plural Environments

For the experiment, I used three types of targets, namely books shown in Table 1, and four types of backgrounds shown in Table 2. The effectiveness of the proposed method was evaluated by combining these targets and backgrounds. As shown in Fig. 9, books consisted of the following: B1 had a clear outline in the bottom half, B2 had a clear form and an unclear outline, and B3 had a relatively flat image. Similarly, backgrounds consisted of the following: W1 was a flat wall; W2 was a relatively simple wall with equipment placed in front of it; W3 was a background with a clear boundary by the monitor; W4 was a complex background of the bookshelf. Note that the case of B3 and W4 shown in Fig. 9 is the one shown in Fig 1 (d).

First, a video of each target was shot with the background changing. The transition of difference between the adjacent frames was calculated using Eq. (1). Figure 10 shows the case

Table 2: Background in experiment

No.	Type	Used background
W1	Flat	Wall without equipment
W2	Sparse	Wall with equipment
W3	Bordered	Wall with large monitor
W4	Complex	Book shelf

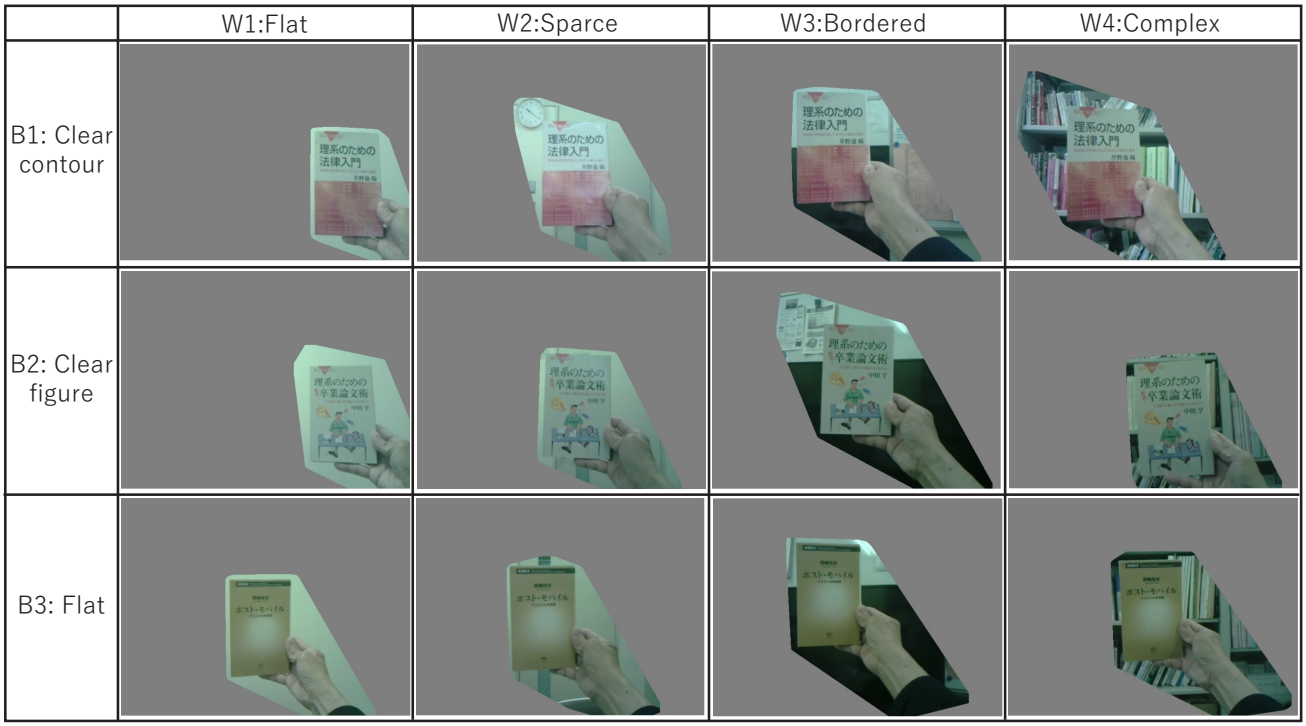

Figure 11: Experimental result in each combination between target object and background

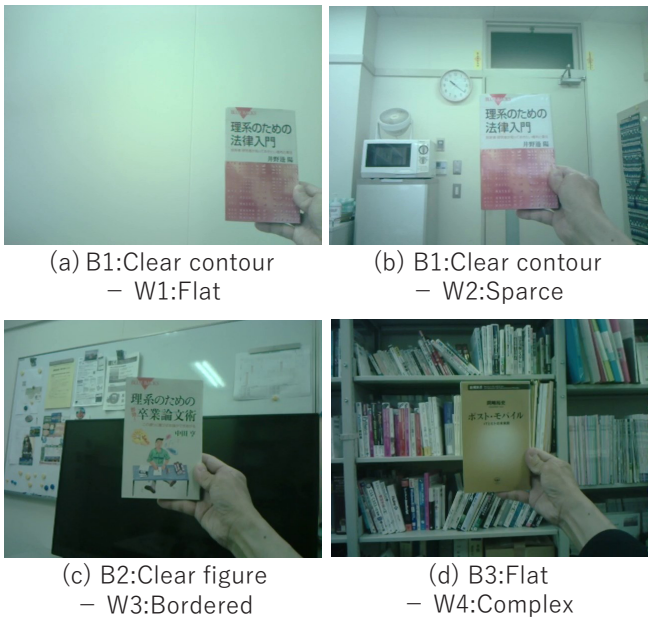

Figure 9: Combination examples of targets and backgrounds

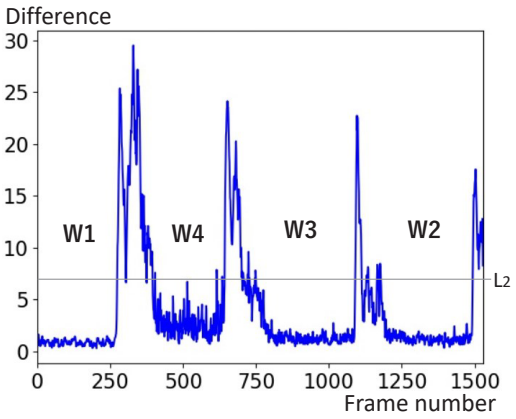

Figure 10: Transition of difference between adjacent frames as for target B1

of book B1. The relatively flat periods in Fig. 10 correspond to when the book was moved in front of each background, that is, the background was the same. The relatively large fluctuation corresponds to the camera's movement from one background to another. The movement could be detected based on the magnitude of the fluctuation, exceeding the threshold L_2. W1 to W4 in Fig. 10 correspond to each background, respectively.

I extracted a frame pair from these videos for each combination with a target displacement distance of 12 pixels or more based on the results described in Sec. 4.2. The target area was then extracted from each frame pair using the experimental system described in Sec. 4.1. Figure 11 shows the results where each row corresponds to the target, and each column corresponds to the background. The target area was extracted in every combination.

However, in combinations B1-W2, B1-W4, and B2-W3, areas other than the target were included. Furthermore, in the case of B2-W1, in which the background was flat, the top left part outside the target area was also extracted, although it was a relatively narrow range. Therefore, I investigated the intermediate results of the extraction process.

Regarding the former, Fig. 12 shows the case of B1-W4, where (1) shows the result of blur by the median filter, and

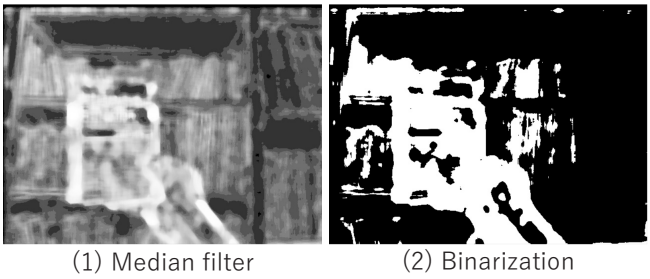

Figure 12: Intermediate processing result images in B1-W4

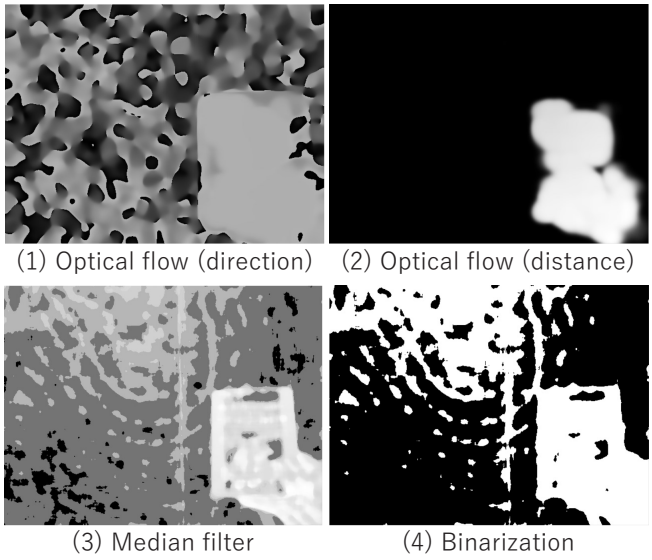

Figure 13: Intermediate processing result images in B2-W1

(2) shows the result of binarization. These correspond to (4) and (5) of Fig. 4, respectively. As shown in (1) of Fig. 12, the background subtraction was relatively bright on the left side and dark on the right. In other words, the distances of the displacement of the background were different depending on their position. Therefore, the upper left part of the background, where the distance was relatively large, was also extracted in binarization and connected to the target area, as shown in (2). As a result, this part was also extracted as the target area.

Regarding the latter, images of displacement direction and distance of optical flow are shown in Fig. 13 (1) and (2). They correspond to Fig. 4 (1) and (2), respectively, though the rectangles are not drawn. Additionally, Fig. 13 (3) and (4) is the same as Fig. 12 (1) and (2). This case used background B1, namely, the flat wall, so the background's displacement distance due to optical flow was calculated as zero, as shown in (2). However, there was a background difference between frames, as shown in (3). As a result, similar to the former case, the upper left part of the background was extracted in binarization and connected to the target. In other words, the displacement of the background could not be detected by the optical flow in the case of a flat wall.

However, the luminance of the background subtraction was smaller than that of the target, as shown in Fig. 13 (3). Therefore, it is expected that the background's influence can be suppressed by increasing the threshold of the binarization. This is evaluated in the next section.

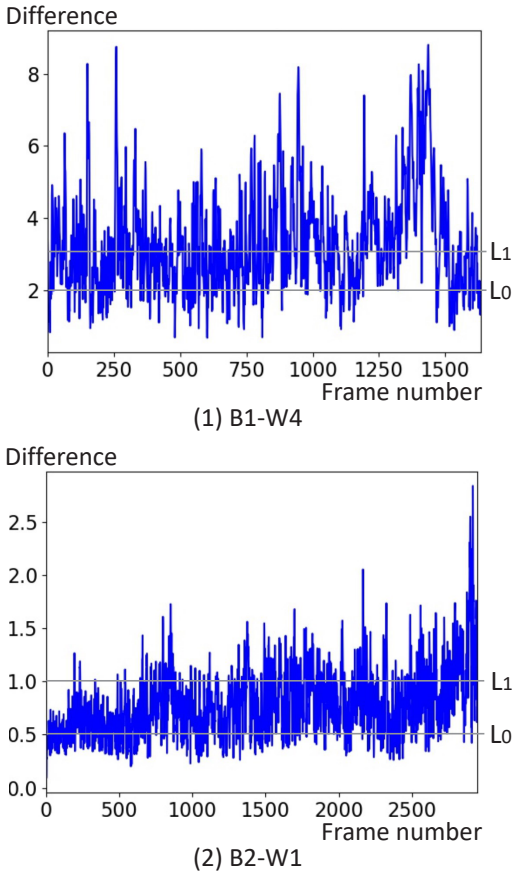

Figure 14: Transition of difference between adjacent frames of target period

5.2 Evaluations of Successive Extraction from Video

An experiment was performed to automatically extract the target area from the video and evaluate the accuracy for combination cases B1-W4 and B2-W1 shown in Fig. 11. A relatively wide area was extracted in these cases, as mentioned in Sec. 5.1.

First, the transition of the difference shown in Fig. 2. was graphed for each case. As shown in Fig. 14, the magnitude of the difference depends on the background. The magnitude was relatively larger in B1-W4, which had a complex background. Therefore, the threshold L_1 and maximum value L_0 were set to 3.0 and 2.0 in (1) B1-W4 and 1.0 and 0.5 in (2) B2-W1, respectively. As a result, 72 frames were extracted from about 1,630 frames of the video in (1), and 69 frames from about 2,950 frames in (2).

The target areas were then extracted using the adjacent pairs of extracted frames. Here, only the pairs with a maximum displacement distance of 12 pixels or more were used, similar to Sec. 5.1. Accordingly, each previous frame's subsequent frames were sequentially tested, and the first frame with 12 pixels or more pixels was selected as its pair. Two thresholds for the binarization of 159 and 191 were used in (2), which

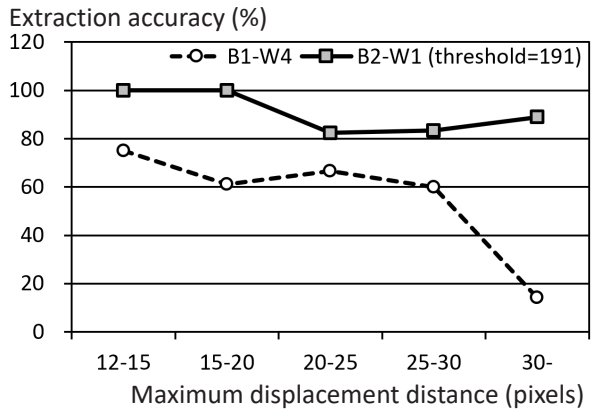

Figure 15: Correlation between accuracy and maximum displacement distance

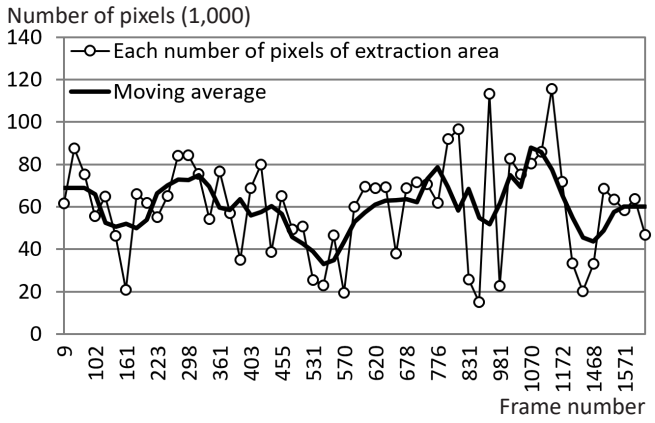

Figure 16: Transition of each number of pixels of extraction area and its moving average (window size=5)

had a flat background, as mentioned in Sec. 5.1. Whereas, only one of 159 was used in (1), similar to Sec. 5.1. Here, the error 32 is doubled for 191, namely, $191 = 127 + 32 \times 2$. The number and accuracy of target area extraction were 57 and 59.6% in B1-W4, and 59 and 88.1% in B2-W1 with the threshold 191; 59 and 100.0% in B2-W1 with the threshold 159.

The correlation between the extraction accuracy and displacement distance was evaluated to clarify the appropriate displacement distance range. Figure 15 shows that for B1-W4 with complex backgrounds, the accuracies were 60% or more in the cases with 30 pixels or less. However, the accuracies deteriorated to about 14% in cases with more than 30 pixels. On the other hand, in the case of B2-W1 with flat backgrounds and threshold 191, no significant deterioration was observed within the experimental range. In addition, the case of B2-W1 with threshold 159 was omitted, because all the accuracies were 100%.

In this method, since the target area is extracted from the video consecutively, it can be assumed that the target areas have almost the same size in the nearby frames. In other words, when the target area is missing or too large, it is expected that there is a certain gap in the ratio between each target area's number and their moving average (hereinafter,

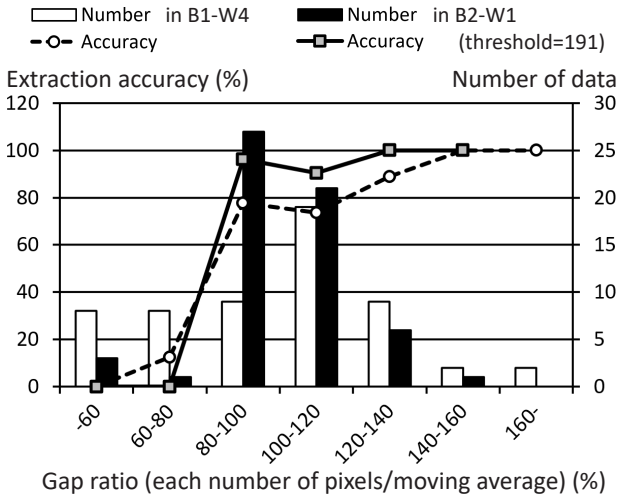

Figure 17: Correlation between gap ratio and extraction accuracy

gap ratio). Figure 16 shows the transition of each number of pixels of the extraction area in B1-W4 and its moving average with a window size of 5. The former fluctuates considerably, while the latter fluctuates gently. Regarding fluctuation factors of the moving average, there was the distance fluctuation between the camera and target, and the size fluctuation of the hand area, depending on the position of the target.

The correlation between the gap ratio and extraction accuracy was then evaluated. Figure 17 shows the results. The horizontal axis is the gap ratio, and when it is 100%, the size of the extraction area is equal to the moving average; The left vertical axis shows the extraction accuracy indicated by the line graph, and the right vertical axis shows the number of data indicated by the bar graph. B2-W1 is the case with a threshold of 191. The case of 159 is omitted because the accuracy was always 100%. As shown in Fig. 17, the extraction accuracies deteriorated significantly when the gap ratio was less than 80% in both cases. In the case of B1-W4, eight data with a maximum displacement distance of 30 pixels or more, shown in Fig. 15, were included; only one was not included.

Figure 18 shows examples of the extracted target areas in B1-W4. Both (a) and (b) show the cases where the gap ratios were around 1.0. While (a) is extracted without any parts missing, a part is missing in (b). Although (a) and (b) are close in the gap ratio, there is a large difference in the extracted area. This is because large or small extracted areas of nearby frames affected the moving average. Furthermore, (c) shows the case where the gap ratio is large, and a wide area was extracted. Conversely, (d) shows the case of a small gap ratio where a part is missing.

In B2-W1, there was a difference in the extraction accuracy, depending on the threshold. Therefore, the correlation between the size of the extraction area and the threshold was also evaluated. Figure 19 shows the result in a scatter plot. The vertical axis shows the number of pixels in the extraction area with threshold 159. The horizontal axis shows the case with threshold 191. Here, the data is excluded where a part was missing in the case of threshold 191. The diagonal line corresponds to the case where the number of pixels is equal in

Figure 18: Examples of extraction results in B1-W4

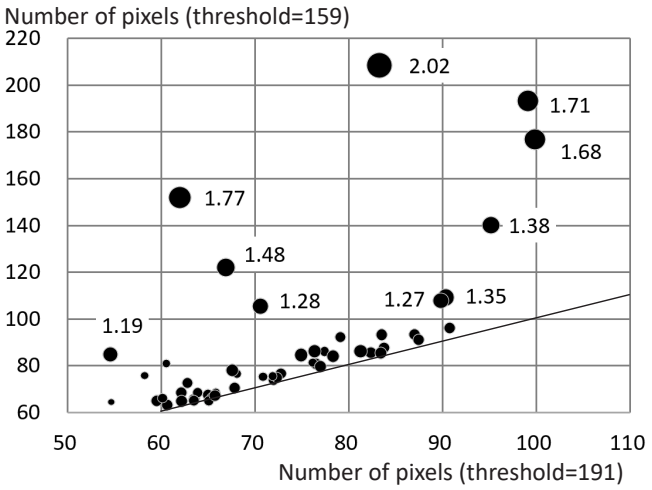

Figure 19: Correlation of the number of pixels of the extraction area between threshold 159 and 191 in B2-W1

both. The extraction area size of threshold 159 was equal to or more than that of threshold 191 in all cases. Furthermore, point sizes in the graph show the gap ratio of each extraction area of the threshold 159. The gap ratio increased when the deviation from the diagonal line increased. In other words, the size was very large compared to that of threshold 191.

Figure 20 shows an example of the extracted target area in B2-W1. The top row shows the case of threshold 191, and the bottom row shows that of threshold 159. The left column shows the case where extraction was accurate without missing parts or too-wide, and the two ((a), (c)) are the results for the same frame. As shown in (a) and (c), the smaller the threshold was, the larger the target area was. Also, similar to the complex background in B1-W4, missing occurred when the gap ratio was too small, as shown in (b), and a too-wide area was extracted when the gap ratio was too large, as shown in (d).

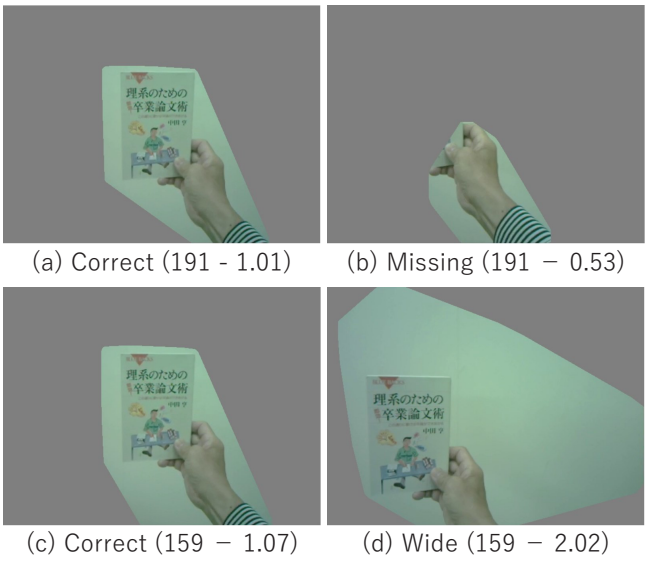

Figure 20: Examples of extraction results in B2-W1

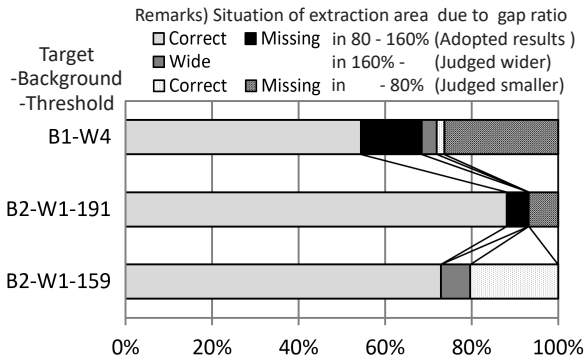

Figure 21: Percentage of extraction area situation due to gap ratio

The above results show that the extraction accuracy can be improved by adopting a gap ratio between 80% and 160%. Figure 21 shows the following percentages of accuracy: accurate and missing extractions when the gap ratio was between 80% and 160%; too-wide extraction defined the case where the ratio exceeded 160%; accurate and missing extractions when the ratio was less than 80%. Here, accurate extraction indicates the case without missing or too-wide extraction. In B1-W4 and B2-W1 with threshold 191, the missing ratio increased when the gap ratio was less than 80%. On the other hand, in B2-W1 with threshold 159, there were only the accurate extractions, even when the gap ratio was less than 80%. However, there were too-wide extractions.

Figure 22 shows a decrease in the error ratio when the adopted data were narrowed down based on the gap ratio, namely, between 80% and 160%. The error was defined as missing or too-wide extraction. In B1-W4 and B2-W1 with threshold 191, both error rates improved to less than half. In B2-W1 with threshold 159, the error rate also decreased to 0% by removing too-wide extraction. However, the data removal rate became about 27% because normal data were also

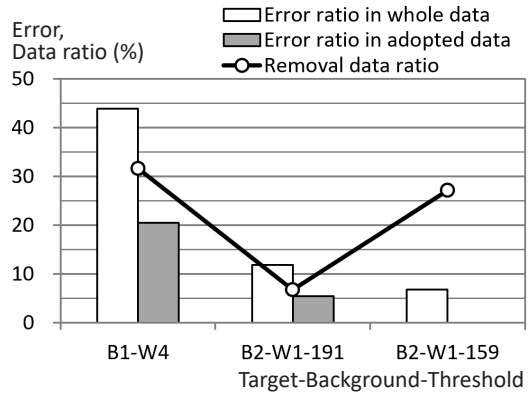

Figure 22: Improvement of error rate by narrowing data based on gap ratio

removed.

5.3 Evaluation Results

I proposed and evaluated a method for extracting a target area from a video shot by a wearable camera. The evaluation results are summarized below.

First, it was shown that the target area could be extracted with various backgrounds and targets, as shown in Fig. 11. In Fig. 11, an area surrounding the target was also extracted. As long as it is a narrow area, this is acceptable since this method aims to extract the area that includes the target from an image as a preprocessing of image recognition using deep learning. Besides, when the binarization's threshold increased, the ratio of missing parts from the extracted target area also increased, as shown in Fig. 21. When the threshold decreased, the relatively wide area was extracted.

Second, the target frames could be extracted from the video automatically and consecutively using the difference between adjacent frames. In this method, only the frames with small differences in which the object was observed were extracted. In other words, they had the maximum value (L_0) or less, as shown in Fig. 2. Therefore, a frame was not extracted when the difference between adjacent frames was greatly changed, such as during movement, as shown in Fig. 10. On the other hand, as shown in Fig. 14, the magnitude of the fluctuation of the difference between adjacent frames depended on the background. Therefore, the threshold (L_1) and maximum value (L_0) had to be set for each background.

Third, frames that might be missing or too-wide could be removed by comparing the size of the extracted target areas with their moving averages. Consequently, the target area could be extracted with an error rate of about 20% for a complex background, and about 5% or less for a flat background, as shown in Fig. 22. On the other hand, about 30% of the frames were removed in some cases of this improvement. However, this method's advantage is that a large number of frames could be easily collected since a video camera continuously recorded the target. For example, in B1-W4, 31 accurate target areas were automatically extracted from 1,630 frames, namely, about 54 seconds of video. In other words, one correct extraction image could be acquired every 1.7 seconds. Therefore, it is acceptable to reject a certain percentage of data in this method.

6 DISCUSSION

Firstly, I mention below that the comparison between the related works shown in Sec. 2.2 and the proposed method. Whereas the background subtraction and frame subtraction method targeted stationary cameras, the proposed method was able to extract the target even with the wearable camera, that is, the moving camera, shown in Fig. 8. Similarly, with the conventional method using optical flow, extracting the target was difficult in a complicated background as shown in Fig. 1. But, with the proposed method, it was possible as shown in Fig. 11. Therefore, as methods utilizing a few video frames of moving camera, I consider that the moving object detection accuracy could be improved by the proposed method.

In addition, comparing with the method combining different methods or analyzing many frames, the proposed method uses only two adjacent frames. So, for example, it is possible to repeatedly extract the target from the continuously shot video and improve the accuracy by comparing the moving average as shown in Fig. 16. Similarly, whereas applying deep learning to a specific environment requires model training, the proposed method can be applied without such preparation. From the above, the proposed method is considered a more efficient method than the related works' methods.

Secondly, some parameters had to be adjusted according to the background. The thresholds L_0 and L_1 to extract frames from the video had to be changed depending on the complexity of the background, as shown in Fig. 14. Since the frame with fluctuation through L_0 and L_1 is extracted, if the setting for the flat background in Fig. 14 (2) is applied to the complicated background in Fig. 14 (1), the number of extracted frames decreases. Its reverse is also the same. Therefore, in this experiment, I created a tool to monitor the transition of the difference shown in Fig. 14 and set the threshold manually.

Similarly, as mentioned in Sec. 5.2, as for the threshold for the binarization shown in Figs. 12 and 13, higher accuracy was obtained by setting a small threshold of 159 than a large one of 191 for the flat background. It is expected that the above parameters can be set automatically by analyzing the transition of the difference in Fig. 14. This is a future challenge.

7 CONCLUSIONS

We can efficiently collect data for training and discrimination for deep learning using mobile camera videos such as wearable cameras. For a small target, it is necessary to extract a relatively small area that includes the target from the video frame, since the background's influence decreases the discrimination accuracy. However, it was often difficult to extract a moving target area efficiently from a video shot by a moving camera in a complex environment.

Hence, I proposed a method for extracting the target area by creating a subtraction of target images between adjacent frames. In this method, the backgrounds are aligned based

on the displacement vector in the optical flow and superimposed. Moreover, through the experiment, I showed that a certain accuracy could be achieved even in complex backgrounds. Furthermore, taking advantage of consecutively extracting the target area from the video, I also showed the accuracy could be improved by comparing the extracted area size with its moving average.

Future studies will focus on applying this method to image recognition using deep learning and verifying its effectiveness.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] F. Chollet, "Deep learning with python," Manning Publications Co. (2017).

[2] A. Elqursh, and A. Elgammal, "Online moving camera background subtraction," European Conference on Computer Vision, pp. 228–241 (2012).

[3] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," Scandinavian conference on Image analysis, Springer, pp. 363–370 (2003).

[4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, "Deep learning," MIT press (2016).

[5] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, and T. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," Proc. IEEE Int. Conf. on computer vision workshops, pp. 142–150 (2015).

[6] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2555–2562 (2013).

[7] A. Kaehler, and G. Bradski, "Learning OpenCV 3: computer vision in C++ with the OpenCV library," O'Reilly Media, Inc. (2016).

[8] A. I. Károly, R. N. Elek, T. Haidegger, K. Széll, and P. Galambos, "Optical flow-based segmentation of moving objects for mobile robot navigation using pre-trained deep learning models," 2019 IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 3080–3086 (2019).

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, pp. 1097–1105 (2012).

[10] 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).

[11] T. Kudo, "A proposal for article management method using wearable camera," Procedia Computer Science, Vol. 176, pp. 1338–1347 (2020).

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

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

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector,".European Conf. on Computer Vision, pp. 21–37, Springer. (2016).

[15] K. Makino, et al., "Moving-object detection method for moving cameras by merging background subtraction and optical flow methods," 2017 IEEE Global Conf. on Signal and Information Processing. pp. 383-387 (2017).

[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 & Tutorials, Vol. 20, No. 4, pp. 2923–2960 (2018).

[17] M. Narayana, A. Hanson, and E. Learned-Miller, "Coherent motion segmentation in moving camera videos using optical flow orientations," Proc. IEEE Int. Conf. on Computer Vision, pp. 1577–1584 (2013).

[18] OpenCV team, "OpenCV modules," https://docs.opencv.org/4.3.0/index.html (referred May 25, 2020).

[19] J. Redmon, S.Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 779–788.(2016).

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems, pp. 91–99 (2015).

[21] S. Singh, C. Arora, and C. V. Jawahar, "Trajectory aligned features for first person action recognition," Pattern Recognition, Vol. 62, 45-55 (2017).

[22] M. Unger, M. Asbach, and P. Hosten, "Enhanced background subtraction using global motion compensation and mosaicking," 2008 15th IEEE Int. Conf. on Image Processing, pp. 2708–2711 (2008).

[23] M. Verhelst, and B. Moons, B., "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).

[24] P. Viola, and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. 2001 IEEE computer society conference on computer vision and pattern recognition, Vol. 1, pp.511-518 (2001).

[25] H. Wang, H, and C. Schmid, "Action recognition with improved trajectories," Proc. IEEE Int. Conf. on Computer Vision, pp. 3551–3558 (2013).

[26] M. Yazdi, and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," Computer Science Review, Vol. 28, pp. 157–177 (2018).

[27] W. Zhang, X. Sun, and Q. Yu, "Moving object detection under a moving camera via background orientation reconstruction," Sensors, Vol. 20, Issue 11, 3103 (2020).

[28] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object de-

tector based on multi-level feature pyramid network," Proc. AAAI Conf. on Artificial Intelligence, Vol. 33, pp. 9259–9266.(2019).

(Received October 16, 2020)
(Accepted May 11, 2021)

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.