In multi-object tracking of identical objects, it is difficult to return to tracking after occlusions occur due to three-dimensional intersections between objects because the objects cannot be distinguished by their appearances. In this paper, we propose a method for multi-object tracking of identical objects using multiple high-speed vision systems. By using high-speed vision, we take advantage of the fact that tracking information, such as the position of each object in each camera and the presence or absence of occlusion, can be obtained with high time density. Furthermore, we perform multi-object tracking of identical objects by efficiently performing occlusion handling using geometric constraints satisfied by multiple high-speed vision systems; these can be used by appropriately positioning them with respect to the moving region of the object. Through experiments using table-tennis balls as identical objects, this study shows that stable multi-object tracking can be performed in real time, even when frequent occlusions occur.

Keywords: identical objects tracking, multiple high-speed vision, occlusion handling

1. Introduction

Motion analysis is an elemental technology in various fields, including robotics. Object tracking is particularly important for measuring the position of an object at each instant and capturing its motion as its trajectory. In computer vision, object tracking using video images (or a sequence of images) obtained from an image sensor as the input has been actively studied. Because object tracking using video images does not require a sensor to be attached to the object, it is used in various applications. Examples include human-machine interaction [1], surveillance to prevent accidents and crimes [2], motion tracking in sports [3], and motion capture to obtain 3D posture information [4].

However, occlusion is a challenging issue in object tracking using video images. Occlusion is a situation in which an object in 3D space is mapped onto a 2D image space but is obscured by an object in front of it, making it invisible in the image space. This occlusion can occur in various situations; however, it is especially frequent in situations where multiple objects are tracked (multi-object tracking) because different objects come across each other. In this case, the tracked object is lost due to occlusion, and tracking the object that occluded the target object may lead to incorrect measurements.

In particular, in the case of identical objects that cannot be distinguished from each other by appearance or in the case of privacy-conscious people who do not retain object-specific information, it is difficult to correctly supplement the situation before and after occlusion with video input at a typical video rate. Furthermore, it is difficult to recover the tracking target because it is not possible to use the information on the target object’s appearance. Therefore, it is necessary to develop a technology that enables continuous tracking of each object in an environment where multiple identical objects occlude each other.

Occlusion handling has become a major issue in recent object-tracking research. The MOT [5], a common object-tracking benchmark, is often used in these studies. However, because MOT is composed of video images at approximately 30 frames per second (fps) and is based on a single viewpoint, many of the studies implicitly assume a hardware environment. From the viewpoint of system configuration, it is desirable to optimize the system for object tracking using video images based on both the hardware design of the types of sensors used and the configuration and the algorithm design for extracting target location information from the information obtained with the hardware configuration. In other words, because the information that can be used in the algorithm in the later stage depends on the information obtained from the hardware, it is desirable to construct the algorithm after selecting the appropriate hardware configuration according to the target and situation to be analyzed while satisfying various conditions.

In this study, we constructed an occlusion-handling...
system for multi-object tracking of identical objects by focusing on high space / time denseness by preparing high-frame-rate video images from multiple viewpoints. By using high-frame-rate video images, it is possible to assume that the changes in the object between frames are small. Thus, the need to estimate the changes in appearance between frames is eliminated, and the subsequent image processing, such as the correspondence of the object between frames, is simplified.

Furthermore, we show that real-time occlusion handling is possible through simple processing that takes advantage of the camera geometry by using multi-view video images based on appropriately located cameras. As an example of occlusion handling of identical objects, we performed multi-object tracking of table-tennis balls moving randomly in space and evaluated the robustness and real-time performance of the process of recovery from occlusion.

2. Related Work

2.1. Tracking Identical Objects

When tracking identical objects, it is difficult to distinguish objects in close proximity, and there is a risk of incorrectly recognizing the IDs assigned to the objects (ID switch). To address this problem, Dicle et al. [6] proposed a method for associating tracklets of identical objects while reducing the number of ID switches by associating several tracklets (i.e., associating target objects between several consecutive frames) already obtained in video images based on their motion information. However, this method is applied during post-processing and is not suitable for real-time execution.

Wang et al. proposed an approach for the multi-object tracking of vehicles with similar appearances that utilizes the smoothness of the vehicle’s movement trajectory instead of appearance information [7]. However, this approach is not effective when the preconditions are not satisfied because of collisions between objects or when complete occlusions occur, which is the target of this study.

2.2. Multi-Camera Tracking and Occlusion Handling

Tracking using multi-view video images is called multi-camera tracking. In multi-camera tracking, it is common to add markers to objects to facilitate the detection and identification of each tracked object. For example, OptiTrack [a] can detect and identify each object by adding infrared retroreflective markers to the object and performing three-dimensional measurements; however, it is challenging to operate in situations in which adding markers is difficult.

Therefore, multi-camera tracking methods that do not use markers have been studied. Black et al. used two-view motion images and applied a Kalman filter to predict and correct the position of an object at each viewpoint detected by background subtraction on a two-dimensional plane and in a three-dimensional space, respectively. They successfully improved the accuracy of tracking and handled occlusion [8]. Wu et al. extended the greedy method to a larger number of objects and to making correspondences from each viewpoint and proposed an efficient method to solve the problem in 3D space [9]. Hofmann et al. proposed a more optimal association of objects by assuming a tracking-by-detection situation in which observations were made at each viewpoint, treating them as nodes of a graph, and performing MAP estimation by considering 3D constraints [10]. In these studies, occlusion handling was performed by post-processing, and none was suitable for identifying identical objects.

We also introduce a study that considers multiple cameras as a sensor network and aims to extract more information than a simple collection of single-sensor information. Liu et al. proposed a method for associating viewpoints using appearance and motion information, without using the relative positional relationship between viewpoints [11]. Previtali et al. realized not only more accurate tracking by correcting observations with a particle filter at each viewpoint and then observing again with the particle filter at multiple viewpoints but also real-time (video rate) processing [12]. In the present study, we achieved a processing speed of more than 1,000 fps by tracking with high-speed vision, which is a simpler method than those in previous studies.

2.3. Tracking with High-Speed Vision

Fundamental technologies and applications of high-speed vision have been studied in various fields, including robot vision [13]. In high-speed vision tracking, the self-windowing method, which limits the region of interest based on the upper bound of the object’s velocity, is widely used. Recently, an extension of the self-windowing method for tracking identical objects was proposed. Sueishi et al. [14] added the restriction of ellipses to the area extracted by self-windowing for small fish (cyprinodont) tracking and showed that when occlusion occurs between individuals, the effect of the occlusion can be reduced by the posture information parameterized by the ellipses. Jiang et al. showed that information obtained through parallel execution of object detection processing at a normal frame rate and high-speed tracking using high-speed vision complemented each other and enabled tracking recovery, even in the case of occlusion [15]. However, real-time multi-object tracking in a situation where many identical objects exist and occlusion occurs, which is the target of this study, has not yet been achieved.

3. Occlusion Handling of Identical Objects Using Multi-View High-Speed Vision

Figure 1 shows the overall pipeline of the proposed method. The method proposed in this study is divided into two major parts. The first part is a high-speed tracking method that updates the target position for the cur-
rent frame based on the target position in the previous frame and simultaneously determines whether occlusion occurred. The second is occlusion handling, which corrects the target positions that were incorrectly updated during occlusion based on the updated target positions and the results of the occlusion judgments.

3.1. Camera Arrangement Robust Against Occlusion

We considered appropriate camera layouts to realize a simple and effective occlusion handling. In recent years, studies [16] have used genetic algorithms to optimize the camera layout in the measurement space, primarily for motion capture in virtual reality. In this study, for simplicity, we referred to the method proposed by Olague et al. [17] and determined the camera layout according to the following criteria: any area in which the object to be tracked fits into the field of view of at least three cameras, in which the angles between the lines of sight of the camera pairs are sufficiently large, and the epipolar lines of other viewpoints intersect in the field of view at a sufficiently large angle of intersection.

By arranging the camera pairs so that their line-of-sight directions have sufficient angles, the presence of objects that are simultaneously invisible to all cameras owing to occlusion can be avoided, and the state of occlusion described in Section 3.3 can be achieved. In addition, making the intersection angle of the epipolar lines sufficiently large eliminates ambiguity in the correction process based on the epipolar geometry, as described later. In the following sections, we discuss in detail high-speed tracking and occlusion handling when the camera layout satisfies these conditions.

3.2. High-Speed Tracking Under Occlusion

In the high-speed tracking, we used two types of tracking algorithms that take advantage of the high frame rate of high-speed vision. In addition, because the update of the target position during tracking was performed in parallel for each viewpoint, information from multiple viewpoints was not used. Before providing the details of the tracking algorithms, we explain their characteristics using high-speed vision under occlusion.

When using a high-speed vision camera, the frame interval is so short that the position and orientation of the object hardly change during the frame interval. Fig. 2 shows the differences in multi-object tracking at low and high frame rates. At a low frame rate, the position, posture, and appearance of the object may change significantly during a relatively long frame interval. In such a situation, the objects are associated by estimating their movement between frames or by identifying them using appearance information. However, when the movement is difficult to estimate or when the objects are similar in appearance, these solutions may not be possible and the target may be lost. It is also possible that the ID that distinguishes the target switches, resulting in incorrect tracking.

However, when a high frame rate is used, the movement between frames is limited; thus, in many cases, the possible objects in object association are limited to those that are closest to each other in position on the images in the previous and next frames. Therefore, a target-associating operation can be achieved by searching only the nearest neighbors. This has the advantage that the targets can be easily identified, except in the case of occlusion, and determining the presence or absence of occlusion caused by the targets is also easier.

Furthermore, in high-speed tracking, multi-object tracking can be decomposed into multiple instances of local single-object tracking, which can be accelerated by simplifying the process. In addition, when occlusion occurs between target objects in tracking-by-detection, which performs multi-object tracking by associating the same object among the detected objects, the number of detected objects in the frame is reduced by the number of occlusions; however, it is not known at the time of detection which target objects are occluded. By contrast, in multiple instances of local single-object tracking, occlusion can be determined by the position of each updated object, and it is possible to know which object has been
Therefore, the tracking algorithm proposed in this study utilizes high-speed and local single-object tracking. In this study, we used tracking based on moment-based center-of-gravity calculation and tracking based on template matching. The tracking algorithm is iterated over the number of viewpoints multiplied by the number of objects; however, these processes can be sufficiently accelerated by parallel processing.

### 3.2.1. High-Speed Tracking Using Moment-Based Center-of-Gravity Calculation

**Figure 3** shows the flow of high-speed tracking using moment-based center-of-gravity calculation. This method quickly updates the position of the target by calculating its center of gravity using the moment calculation of the binarized image. The input frames received from the cameras are set to $I(i, j)$ and binarized with a threshold value $t$. The resulting binarized images are denoted as $B(i, j)$ and are expressed by Eq. (1).

$$B(i, j) = \begin{cases} 0 & \text{if } I(i, j) < t \\ 1 & \text{otherwise} \end{cases} \ldots \ldots \ldots (1)$$

Let $c_{t-1}$ be the position of the object in the previous frame. Crop the binarized image $B(i, j)$ to a square region with a size of $2r$, centered at $c_{t-1}$. Let the region of interest $b(i, j)$ be the cropped area. The moment $m_{pq}$ in this binarized region of interest $b(i, j)$ is defined as

$$m_{pq} = \sum_j \sum_l I^p j^q b(i, j) \ldots \ldots \ldots \ldots \ldots \ldots \ldots (2)$$

and its center of gravity $C_t$ is calculated as $C_t = (m_{10}/m_{00}, m_{01}/m_{00})$. The target position in the previous frame is updated by the difference between the calculated $C_t$ and the center of the region of interest.

In this method, the region of interest $r$ is adaptively changed with respect to the size of the target using the area $m_{00}$ of the target area used in the moment calculation. The formula for this change is shown in Eq. (3) using the three hyperparameters $a$, $b$, and $c$.

$$r = \max\{a \sqrt{m_{00}} + b, c\} \ldots \ldots \ldots \ldots \ldots \ldots \ldots (3)$$

However, $a$ and $b$ are parameters that determine the ratio and offset of the area of the target region. Because a large region of interest increases the possibility that other objects may be included and makes the moment calculation unstable, $a$ and $b$ are set appropriately using a linear function with respect to the size of the target to prevent excessive expansion of the region of interest. Moreover, to prevent the region of interest from becoming too small and to secure a minimum region of interest, a lower limit $c$ of the region is set. By keeping the region of interest as small as possible with the parameter set based on the above assumptions, the space for other objects to enter the region of interest can be reduced. Therefore, tracking based on moments with high-speed vision is more robust than tracking with a normal frame rate when other objects are located near the target.

In addition, when the center of gravity calculated by moment does not change depending on the posture of the object, as in the case of a sphere, this method can capture almost the same 3D point as the object position from multiple viewpoints, which is a feature that enables accurate recovery from occlusion using 3D information in the occlusion-handling stage.

### 3.2.2. High-Speed Tracking Using Template Matching

**Figure 4** shows the process flow of the tracking method using template matching. In template matching, the area around the target position in the previous frame is used as the template, and the corresponding point in the current frame that best matches the template is used as the updated position of the target. After updating the position...
of the target, a new template is set in each frame, with that position as the center. Although template matching is generally computationally expensive because of pixel-to-pixel operations, we assumed that the movement of the target is small between frames of a high-speed vision camera and limited the region of interest to the center position of the previous frame and its surroundings to accelerate the process. In this study, the normalized correlation is used as an index for template matching, and the point with the highest correlation is considered to be the updated position of the target.

Although the method is slightly more computationally expensive than the previous method (Section 3.2.1), it has the advantage of being robust even when other objects are very close.

3.3. Occlusion Handling

In the above high-speed tracking methods, an occlusion judgment is made at the end. In the occlusion judgment, the occluding and occluded targets are identified. The Euclidean distance on the image coordinates of the viewpoint is calculated for all combinations of all object positions obtained for each viewpoint, and it is determined whether the distance exceeds a certain threshold value. In this study, the threshold was set as approximately the radius of the observed object. The result of the occlusion judgment is maintained for each object as a container representing the occluded objects. In this study, this container is referred to as the occlusion list.

Figure 5 shows the application flow. If there is no occluded target, the container for that target is assumed to be empty; if there is an occluded target, the ID of the occluding counterpart is stored in the container. By checking this container, it can be determined whether an occlusion occurred and which target was involved in the occlusion. The container is initialized as empty at the beginning of each frame.

When occlusion occurs between targets from each viewpoint while each tracker tracking targets, multiple trackers often track the front target after the occlusion. We correct this false tracking using information from other viewpoints whose relative camera positions are known. In this study, we considered the following cases:

1. Two or more viewpoints can track targets without occlusion.
2. No occlusion occurs from one viewpoint. Occlusion occurs from the other viewpoints where one of the trackers still tracks.

We then propose occlusion handling for each of the above cases. To facilitate an explanation, we use three viewpoints in this section.

3.3.1. Situation Where No Occlusion Occurs from Multiple Viewpoints

Figure 6 shows a situation in which no occlusion is found from two or more viewpoints. In this case, the trackers to be corrected are A and B in view #1. By using the target position and relative camera position from the two viewpoints where no occlusion occurred, two epipolar lines can be drawn from the viewpoint of occlusion (Fig. 7). The cameras were positioned such that these epipolar lines intersected in the image and had sufficiently large crossing angles. The intersection of these two epipolar lines can be used to recover the positions of occluded objects. If occlusion does not occur for a tracked object from two or more viewpoints, the trackers are corrected by preferentially performing the restoration process described above.

3.3.2. Situation Where No Occlusion Occurs from One Viewpoint and One of the Trackers Continues Tracking

By contrast, Fig. 8 shows a situation where every target is tracked by one tracker, while no occlusion occurs from
one viewpoint. In this case, the trackers to be corrected are A and B, from views #2 and #3, respectively. With information from only one viewpoint where no occlusion has occurred, even if we use the epipolar constraint, we only know that the targets must be somewhere in a straight line on the image plane from the other viewpoints, and we cannot determine the target positions. However, if each target is tracked by one tracker from other viewpoints, recovery can be achieved.

It is highly likely that one of the trackers that overlaps during occlusion is tracking the correct target position. The overlapping tracker tracking the correct target position is determined by the epipolar constraint of the target position from the viewpoint of which the trackers can be trusted (view #1 in this case).

The following explanation is based on Target A from view #2 in Fig. 9. First, by checking the occlusion list of Target A of view #2, we can confirm that the tracker of Target A overlaps with that of Target B. Next, when the epipolar lines of Targets A and B involved in the occlusion of reliable view #1 are drawn in view #2, the tracker that tracks the correct target is on the epipolar line. The overlapping trackers track the target that corresponds to the epipolar line running over them. In some cases, the epipolar line could be significantly close to the trackers, and the targets could not be identified. However, the camera positions were arranged to prevent such a case.

If the overlapping tracker that tracks the correct target location is identified, the target position can be corrected using the intersection of the epipolar lines, as in the case in which no occlusion occurs from two or more viewpoints. In other words, the position of Target A from view #3 is corrected as the intersection of the epipolar line in view #3, where occlusion has occurred, and the epipolar line of Target A drawn from view #1, where no occlusion has occurred, using the position of Target A, which was newly found to be reliable in view #2, where occlusion has occurred.

The recovery process using the above procedure requires tracking with at least one more camera than the number of occlusions for one object. Conversely, an object can be tracked up to one fewer occlusion than the number of cameras used.

4. Evaluation

4.1. Configuration of Experimental Apparatus

As shown in Fig. 10, we used three cameras equipped with a high-speed image sensor (resolution: 320 × 320 pixels, frame rate: 1,000 fps) from Ximea, a wide-angle lens (focal length: 1.28 mm, F-value: 1.8) from Nittoh Kogaku, and a ring-shaped LED light. The three cameras were arranged to satisfy the conditions described in Section 3.1. The internal and external parameters of the
cameras were obtained using prior calibration.

The computer used in the experiments had an Ubuntu 18.04 LTS OS with an Intel® Xeon® CPU E5-2637 v3 (16 cores) and 64 GB of memory. The implementation was performed in C++; OpenCV 3.4.3, an open-source library, was used for various image processing, and OpenMP 4.5, an open-source library, was used for parallel processing.

To quantitatively evaluate the tracking performance, we used three units of OptiTrack PrimeX22 (resolution: 2048 × 1088 pixels, frame rate: 200 fps), which is capable of 3D measurement by adding markers to the tracking target, to acquire reference values. As shown in Fig. 10, OptiTrack captures the targets from the bottom angle, which is the most difficult angle for occlusions to occur in this experimental setup. Such an arrangement is possible only when the motion model of the targets is known in advance and not necessarily for more general motion of the targets. OptiTrack tracked the bottom of the table-tennis ball because the marker detected by OptiTrack was placed there. However, the high-speed multi-camera system tracked the center of the table-tennis ball, which resulted in a vertical offset of the radius of the table-tennis ball.

In this experiment, the threshold of binarization in the moment-based center-of-gravity calculation was set to \( t = 50 \), and the hyperparameters for setting the region of interest were set to \( a = 0.6, b = 1.6, \) and \( c = 3.0 \). The size of the template in template matching was set to 8, and the size of the region of interest was set to 10.

4.2. Evaluation Procedure

In this experiment, we employed table-tennis balls as an example of objects that are difficult to distinguish based on appearance information (Fig. 11). To generate occlusions, we randomly moved 10 table-tennis balls as objects to be tracked. Each table-tennis ball was given an initial vibration with random timing and an oscillation angle of approximately 20°, and then was allowed to vibrate freely. As shown in Fig. 10, retroreflective markers were attached to the table-tennis balls and used as references of 3D positions for 3D measurements using OptiTrack. We also measured the processing time of the proposed method when the number of table-tennis balls to be tracked changed, and the real-time performance of the method was evaluated. The following four options were applied for each of the two tracking methods proposed in this study; thus, eight combinations were compared.

1. With 1,000 fps input (no occlusion handling).
2. With 1,000 fps input + occlusion-handling Rule 1.
3. With 1,000 fps input + occlusion-handling Rules 1 and 2.
4. With 30 fps input + occlusion-handling Rules 1 and 2.

Here, we define occlusion-handling Rule 1 as the recovery process for the case where no occlusion occurs from two or more viewpoints. Rule 2 is defined as the recovery process for the case where no occlusion occurs from one viewpoint, and every target is tracked by one tracker.

Options 1, 2, and 3 aim to determine whether the tracking performance is improved by the occlusion-handling rule, and Options 3 and 4 aim to determine how much the tracking method using the assumption for the high frame rate contributes to the tracking accuracy. However, when changing the input frame rate, hyperparameters, such as the size of the region of interest, were set to appropriate values for each case. The initial position of each target in the proposed method was manually set.

From the previously acquired videos, we chose and used 5-s video clips that had no frame loss and were successful in the measurement with OptiTrack. The target table-tennis balls were suspended using transparent wires and manually randomly vibrated to artificially generate occlusions. The average pixel error in the image coordinates between the OptiTrack method and the proposed method for each frame and target was used as the index for quantitative evaluation.

4.3. Results

4.3.1. Comparison with Reference Values

Figure 12 shows the average pixel error between the proposed method and OptiTrack for each frame and object. The horizontal axis shows Options 1–4 from left to right, and for each option, both proposed tracking methods were used. The vertical axis represents the average error per pixel, which is shown in a box plot.

The method with the smallest error was the moment-based tracking method with Options 2 and 3, and the error was approximately 4.5 pixels. The radius of the object in the images is approximately 4 pixels, which indicates that the proposed method can track the object accurately, considering the offset of the radius between OptiTrack and the proposed method, as mentioned in Section 4.1. The smallest error for the method using template matching is

![Fig. 11. Identical objects to be tracked (table-tennis ball).](image-url)
approximately 5.0 pixels with Options 2 and 3. Comparing the error variances of the two methods, it was found that the method using template matching tends to have larger errors. The reasons for this are discussed below.

Next, we compare the results for Options 1, 2, and 3 to check the effects of the occlusion-handling rules. The comparison shows that the error variance of tracking with occlusion-handling rules is smaller than that without the rules. This confirms that the proposed occlusion-handling rules can appropriately correct the trackers.

Finally, we compared the results for Options 3 and 4 to check the effect of changing the frame rate used for the input. In Option 4, where the input frame rate is reduced to 30 fps, both the average and variance of the errors are larger when moments are used for tracking. However, there was no statistically significant difference between Options 3 and 4 when template matching was used. 

**4.3.2. Tracking Trajectory of Each Target from Each Viewpoint**

**Figure 14** shows the trajectory of the target from the viewpoint with no occlusion. Each trajectory is colored for the measurement by OptiTrack (reference), the measurement by the moment-based tracking method, and the measurement by the template-matching tracking method. The X-coordinate values are indicated by solid lines, and the Y-coordinate values are indicated by dashed lines. Option 1 was applied to each proposed method. The Y-coordinates show the measurement offset between OptiTrack and the proposed methods. Except for this offset, the measurement with each proposed method is consistent with that with OptiTrack, confirming that single-object tracking works properly with the proposed method. 

**Figure 15** shows the object trajectories from the viewpoint with occlusion. Each trajectory is colored according stably in the case of Option 3 with 1,000 fps input. However, in the case of Option 4 with 30 fps input, the targets did not fit into the region of interest, and the template of the targets was gradually replaced by the background, resulting in a gradual shift of the trajectory. This indicates that the same tracking algorithm can track the object appropriately if a high-frame-rate input is used because of the assumptions made about the changes between frames.
to Options 1, 2, and 3 applied to the template-matching tracking method, and according to the method with OptiTrack. It can be seen that the trajectory with Option 3 reproduces the OptiTrack trajectory, whereas the trajectory with Option 1, which does not apply the occlusion-handling rule, tracks the wrong target and deviates significantly from the OptiTrack trajectory. In addition, Option 2 fails to track the target at around 0.8 s, and the tracker fails to recover. In this case, there is only one viewpoint that does not have occlusion, and it is considered that the tracker could not be restored by applying Rule 1 alone. The trajectory indicates that each of the occlusion-handling rules contributes to the tracker’s recovery.

We also examined the tracking of the two proposed tracking methods and discussed the difference in the magnitude of the error. Fig. 16 shows the tracking of table-tennis balls using the moment method. It can be seen that the region of interest is adaptively adjusted according to the size of the object, and that the original object is accurately tracked before and after occlusion. The bounding box of the target was also accurately centered on the target, indicating that the table-tennis ball could be tracked with high accuracy. Furthermore, although occlusions occur at multiple locations in the image, each occlusion is processed independently; thus, the tracking of the target is successfully performed continuously.

Figure 17 shows the tracking of the table-tennis balls using template matching. It can be seen that, similar to the moment-based method, this method can correctly track the original object, even when occlusion occurs. Unlike with the moment-based center-of-gravity calculation, the object is not captured at the center of the region of interest, although the object itself can be tracked. This is because once an off-center template is created in template matching, it is difficult to return the target to the center, and the error from the true value increases. Therefore, although it is considered effective for tracking more general shapes, it will be necessary to combine it with the center-returning process in the future.

4.3.3. Processing Speed

We measured the processing speed of the tracking when the number of targets changed. We used table-tennis balls as the objects to be tracked and prepared two videos for each of the three patterns with 5, 10, and 15 objects. The average speed of the 10 processing runs was taken as the processing speed of each pattern. The method used in this experiment is based on the two proposed tracking methods with Option 3 (using occlusion-handling Rules 1 and 2 together).

The results are shown in Fig. 18. In this figure, the vertical axis represents the processing speed in fps, and the horizontal axis represents the number of targets. When the number of targets was 15, each method achieved approximately 10,000 fps. This was measured with OpenMP parallel processing; however, even with-
out parallelization, a processing speed of 3,000 fps was achieved. In the tracking proposed in this study, the most computationally time-consuming multi-object tracking is decomposed into multiple instances of single-object tracking, and then low-load processing of occlusion handling is used, resulting in high scalability.

If the targets are moving quickly, it is necessary to expand the region of interest to continue tracking. Because the time required for the occlusion-handling process is very small, assuming that the amount of computation required for tracking at each time is proportional to the square of one side of the region of interest, tracking at 1,000 fps is possible even if the region of interest is expanded approximately three times when the number of objects is 15. However, because the possibility of mistakenly capturing other objects increases with the expansion of the region of interest, it is desirable to use a high-speed camera as much as possible to minimize the amount of movement between frames and to narrow the region of interest.

4.4. Discussion

However, there are issues that remain to be solved to handle more occlusions than those handled in this study. When the duration of occlusion is long or when the tracking is unstable owing to the influence of the background or surrounding light source environment, the stability of tracking may be improved by combining the prediction with a filter. Furthermore, in the case of complex occlusions involving more than two objects, which may occur when tracking a larger number of objects, it is necessary to deal with the cases that were excluded in this study, such as a case in which occlusion occurs from all view-

Fig. 17. Tracking of identical objects using template matching.

Fig. 18. Comparison of computation speeds with different numbers of tracking targets.

Hirano, M., Iwakuma, K., and Yamakawa, Y.

Journal of Robotics and Mechatronics Vol.34 No.5, 2022
points or a case in which there is an object that is not tracked by any tracker and there is one viewpoint from which no occlusion has occurred. Because the proposed method has high scalability to maintain a high processing speed for an increase in the number of viewpoints, increasing the number of viewpoints with no occlusion by increasing the number of camera viewpoints is considered to be effective; however, it is necessary to consider how to handle cases in which there is an object that is completely occluded from any viewpoint.

In addition, it is necessary to consider the case of processing failure owing to interruptions by the OS. If tracking continues even after a drop frame occurs, it is possible to detect the drop frame based on the distance traveled between frames and perform interpolation, as appropriate. If tracking fails, re-detection and reidentification must be performed separately. However, they may not be effective for objects with the same appearance, which is an issue to be addressed in the future.

5. Conclusions

In this study, we developed a multi-object tracking system using multiple high-speed vision systems in a situation where multiple identical objects, which are difficult to identify by their appearance, are occluded by each other. The tracking system is based on the fact that the tracking information (position and occlusion) for each object can be efficiently obtained using high-speed vision. The geometric constraints satisfied by multiple high-speed vision systems appropriately positioned for the motion area of the object were used for efficient occlusion handling based on the results of the tracking performed by each vision system. In this study, we used table-tennis balls as identical objects and showed that stable multi-object tracking was possible even when occlusion occurred. We also achieved a processing speed of more than 1,000 fps and showed that real-time tracking processing at 1,000 fps, the input frame rate of high-speed vision, was possible.

However, as mentioned in Section 4.4, it is necessary to deal with more severe occlusion situations in the future. In addition, although we used relatively simple shapes as tracking objects in this study, it is necessary to make the system robust to objects that undergo more flexible external changes, such as tracking players in sports. For this purpose, it is necessary to combine a tracking method with a processing speed that matches the input frame rate and is more optimized for the tracking target.

References:


Supporting Online Materials:


Journal of Robotics and Mechatronics Vol.34 No.5, 2022
Hirano, M., Iwakuma, K., and Yamakawa, Y.

<table>
<thead>
<tr>
<th>Name: Masahiro Hirano</th>
<th>Name: Yuji Yamakawa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affiliation: Assistant Professor, Institute of Industrial Science, The University of Tokyo</td>
<td>Affiliation: Associate Professor, Interfaculty Initiative in Information Studies, The University of Tokyo</td>
</tr>
<tr>
<td>Address: 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan</td>
<td>Address: 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan</td>
</tr>
<tr>
<td>Brief Biographical History: 2018- Project Assistant Professor, The University of Tokyo 2020- Assistant Professor, The University of Tokyo</td>
<td>Brief Biographical History: 2011- Project Assistant Professor, The University of Tokyo 2014- Assistant Professor, The University of Tokyo 2017- Lecturer, The University of Tokyo 2020- Associate Professor, The University of Tokyo</td>
</tr>
<tr>
<td>Membership in Academic Societies: ● Institute of Electrical and Electronics Engineers (IEEE) ● Information Processing Society of Japan (IPSJ) ● The Society of Instrument and Control Engineers (SICE)</td>
<td>Membership in Academic Societies: ● The Japan Society of Mechanical Engineers (JSME) ● The Robotics Society of Japan (RSJ) ● The Society of Instrument and Control Engineers (SICE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name: Keigo Iwakuma</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Affiliation: Assistant Research Engineer, GA technologies</td>
<td></td>
</tr>
<tr>
<td>Address: 3-2-1 Roppongi, Minato-ku, Tokyo 106-6290, Japan</td>
<td></td>
</tr>
<tr>
<td>Brief Biographical History: 2019 Received Bachelor degree (Mechanical Engineering) from The University of Tokyo 2021 Received Master degree (Mechanical Engineering) from The University of Tokyo 2021- GA technologies</td>
<td></td>
</tr>
<tr>
<td>Membership in Academic Societies: ● The Japan Society of Mechanical Engineers (JSME)</td>
<td></td>
</tr>
</tbody>
</table>

1084 Journal of Robotics and Mechatronics Vol.34 No.5, 2022