

Article

An Active Multi-Object Ultrafast Tracking System with CNN-Based Hybrid Object Detection

Qing Li , Shaopeng Hu , Kohei Shimasaki  and Idaku Ishii * 

Smart Robotics Laboratory, Graduate School of Advanced Science and Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima 739-8527, Hiroshima, Japan

* Correspondence: iishii@robotics.hiroshima-u.ac.jp; Tel.: +81-82-424-7692; Fax: +81-82-422-7158

Abstract: This study proposes a visual tracking system that can detect and track multiple fast-moving appearance-varying targets simultaneously with 500 fps image processing. The system comprises a high-speed camera and a pan-tilt galvanometer system, which can rapidly generate large-scale high-definition images of the wide monitored area. We developed a CNN-based hybrid tracking algorithm that can robustly track multiple high-speed moving objects simultaneously. Experimental results demonstrate that our system can track up to three moving objects with velocities lower than 30 m per second simultaneously within an 8-m range. The effectiveness of our system was demonstrated through several experiments conducted on simultaneous zoom shooting of multiple moving objects (persons and bottles) in a natural outdoor scene. Moreover, our system demonstrates high robustness to target loss and crossing situations.

Keywords: high-speed vision; convolutional neural network (CNN); template matching (TM); multi-object tracking



Citation: Li, Q.; Hu, S.; Shimasaki, K.; Ishii, I. An Active Multi-Object Ultrafast Tracking System with CNN-Based Hybrid Object Detection. *Sensors* **2023**, *23*, 4150. <https://doi.org/10.3390/s23084150>

Academic Editors: Krzysztof Bernacki and Mateja Novak

Received: 10 March 2023
Revised: 9 April 2023
Accepted: 11 April 2023
Published: 21 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Multi-target tracking and high-definition image acquisition are important issues in the field of computer vision [1]. High-definition images of many different targets can provide more details, which is helpful for object recognition and improves the accuracy of image analysis. It has been widely used in traffic management. [2], security monitoring [3], intelligent transportation systems [4], robot navigation [5], auto pilot [6], and video surveillance [7].

However, there is a contradiction between wide field of view and high-definition resolution, as shown in Figure 1. The discovery and tracking of multiple targets depends on a wide field of view. While a panoramic camera with a short focal length can provide a wide field of view, the definition of the image is low. A telephoto camera is the exact opposite of a panoramic camera. Using a panoramic camera with a larger resolution is a feasible solution; however, it requires greater expenditure and larger camera size [8]. With the rapid development of deep learning in the image field, the super-resolution reconstruction method based on autoencoding has become the mainstream, and its reconstruction accuracy is significantly better than that of traditional methods [9]. However, due to the huge network model and large amount of model training required in the super-resolution method based on deep learning, there are defects in the reconstruction speed and the flexibility of the model [10].

Therefore, researchers have tried to use telephoto cameras to obtain a larger field of view and track multiple targets. A feasible solution is to stitch the images obtained from a telephoto camera array together into high-resolution images and track multiple targets [11]. Again, this results in greater expenditure and an increase in device size. Another research method is to make the telephoto camera an active system by mounting it on a gimbal. Through the horizontal and vertical movement of the gimbal, the field of

view of a pan-tilt-zoom (PTZ) camera can be changed to obtain a wide field of view [12]. However, the original design of such a gimbal camera is not intended for multi-target tracking. Due to the limited movement speed of the gimbal and the size of the telephoto lens, it is difficult for gimbal-based PTZ cameras to move at high speeds and observe multiple objects [13]. Compared to traditional camera systems operating at 30 or 60 fps, high-speed vision systems can work at 1000 fps or more [14]. The high-speed vision system acquires and processes image information with extremely low latency and interacts with the environment through visual feedback control [15]. In recent years, a galvanometer-based reflective camera system has been developed that can switch the perspective of a telephoto camera at hundreds of frames per second [16]. This reflective PTZ camera system is able to virtualize multiple virtual cameras in a time-division multiplexing manner in order to observe multiple objects [17]. Compared with traditional gimbal-based and panoramic cameras, galvanometer-based reflective PTZ cameras have the advantages of low cost, high speed, and high stability [18], and are suitable for multi-target tracking and high-definition capture.

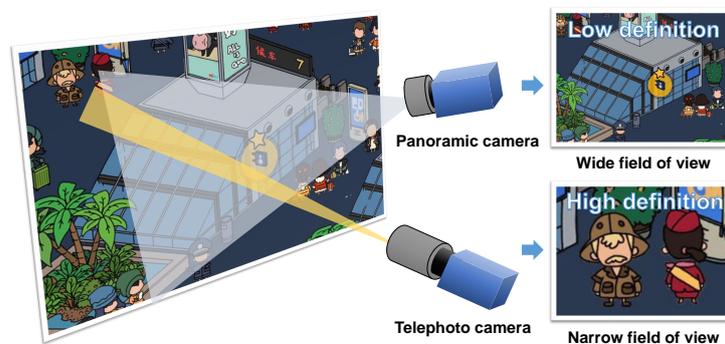


Figure 1. Contradiction between wide field of view and high-definition images.

However, the current galvanometer-based PTZ cameras rarely perform active visual control in the process of capturing multiple targets. Instead, they mainly rely on panoramic cameras, laser radars, and photoelectric sensors to obtain the positions of multiple targets, and finally use reflective PTZ cameras for multi-angle capture [19]. Due to the impact of detection delay and accuracy, it is difficult for multiple objects to be tracked smoothly. With the victory of AlexNet in the visual competition, CNN-based detectors continue to develop, and can now detect various objects in an image at a dozens of frames per second [20]. For high-speed vision at a speed of hundreds of frames per second, however, it is difficult to achieve real-time detection with deep learning.

This study aims to utilize a reflective PTZ camera system to track multiple objects and to capture high definition images with low latency. A reflective PTZ camera system switches perspectives to track multiple objects at 500 fps per second by implementing 2-ms-latency visual feedback control. The high-speed vision feedback control relies on CNN-based hybrid detection methods [21]. Compared with the previous system, this work achieves the following: (1) the acquisition of images with large field of view and high resolution, (2) simultaneous observation of up to 20 objects at a speed of 25 fps; and (3) active tracking of multiple fast-moving objects with no-latency detection. The subsequent parts of this study are organized as follows: related works and research are presented in Section 2; a detailed algorithmic analysis and concept illustration are presented in Section 3; Section 4 presents a full description of the experimental test platform, followed by a discussion of the test; finally, our conclusions are presented in Section 5.

2. Related Works

This study aims to track multiple targets by switching the perspectives of PTZ cameras simultaneously. This is closely related to the research on object detection, multiple object

tacking, and high-speed vision. In the next two sections, we provide a brief review of related works.

2.1. Object Detection

Object detection is a computer vision task that involves detecting instances of semantic objects of a certain class (such as a person, bicycle, or car) in digital images and videos [22]. The earliest research in the field of object detection can be traced back to the Eigenface method for face detection proposed by researchers at MIT [23]. Over the past few decades, object detection has received great attention and achieved significant progress. Object detection algorithms are roughly divided into two stages, namely, traditional object detection algorithms and the object detection algorithms based on deep learning [24].

Due to the limitations of computing speed, traditional object detection algorithms focus mainly on pixel information in images. Traditional object detection algorithms can be divided into two categories, sliding window-based methods [25] and region proposal-based methods [26]. The sliding window-based approach achieves object detection by sliding windows of different sizes over an image and classifying the contents within the different windows. Many such feature extraction methods have been proposed, such as histogram of oriented gradients (HOG), local binary patterns (LBP), and Haar-based classifiers. These features, in conjunction with traditional machine learning methods such as SVM and BOOST, have been widely applied in pedestrian detection [27] and face recognition [28]. Region proposal-based methods achieve object detection by proposing regions in which objects may exist in generated images, then classifying and regressing these regions [29].

Traditional algorithms have been proven effective; however, with continuous improvements in computing power and dataset availability, object detection technologies based on deep learning have gradually replaced the traditional manual feature extraction methods and become the main research direction. Thanks to continuous development, convolutional neural network (CNN)-based object detection methods have evolved into a series of high-performance structural models such as AlexNet [30], VGG [31], GoogLeNet [32], ResNet [33], ResNeXt [34], CSPNet [35], and EfficientNet [36]. These network models have been widely employed as backbone architectures in various CNN-based object detectors. According to the differences in the detection process, object detection algorithms based on deep learning can be divided into two research directions, One-Stage and Two-Stage [37]. Two-stage object detection algorithms transform the detection problem into a classification problem for generated local region images based on region proposals. Such algorithms generate region proposals in the first stage, then classify and regress the content in the region of interest in the second stage. There are many efficient object detection algorithms that use a two-stage detection process, such as R-CNN [38], SPP-Net [39], Fast R-CNN [40], Faster R-CNN [41], FPN [42], R-FCN [43], and DetectoRS [44]. R-CNN was the earliest method to apply deep learning technology to object detection, reaching an MAP of 58.5% on the VOC2007 data. Subsequently, SPP-Net, Fast R-CNN, and Faster R-CNN sped up the running speed of the algorithm while maintaining the detection accuracy. One-stage object detection algorithms, on the other hand, are based on regression, which converts the object detection task into a regression problem for the entire image [45]. Among the one-stage object detection algorithms, the most famous are single shot multibox detector (SSD) [46], YOLO [47], RetinaNet [48], CenterNet [49], and Transformer [50]-based detectors [51]. YOLO was the earliest one-stage target detection algorithm applied to actual scenes, obtaining stable and high-speed detection results [52]. The YOLO algorithm divides the input image into $S \times S$ grids, predicts B bounding boxes for each grid, and then predicts the objects in each grid separately. The result of each prediction includes the location, size, confidence of the bounding box, and the probability that the object in the bounding box belongs to each category. This method of dividing the grid avoids a large number of repeated calculations, helping the YOLO algorithm to achieve a faster detection speed. In follow-up studies, algorithms such as YOLOv2 [53], YOLOv3 [54], YOLOv4 [55], YOLOv5 [56], and

YOLOv6 [57] have been proposed. Owing to its high stability and detection speed, in this study we use yolov4 as the AI detector.

2.2. Multiple Object Tracking (Mot)

Multiple object tracking detects and tracks multiple targets in videos, such as pedestrians, vehicles, and animals. It is an important research direction in the field of computer vision, and has been widely applied in intelligent surveillance and behavior recognition [58]. Early classical tracking methods, such as Meanshift [59], particle filtering [60], KCF [61], and MOSSE [62], mainly focused on single-object tracking. With the rapid development of CNNs, detection-based multi-object tracking methods have quickly become the mainstream research direction.

Currently, there are three popular research directions in multi-object tracking: detection-based MOT, detection and tracking-based joint MOT, and attention-based MOT. In detection-based MOT algorithms, object detection is performed on each frame to obtain image patches of all detected objects. A similarity matrix is then constructed based on the IoU and appearance between all objects across adjacent frames, and the best matching result is obtained using a Hungarian or greedy algorithm; representative algorithms include SORT [63] and DeepSORT [64]. In detection and tracking-based joint MOT algorithms, detection and tracking are integrated into a single process. Based on CNN detection, multiple targets are fed into the feature extraction network to extract features and directly output the tracking results for the current frame. Representative algorithms include JDE [65], MOTDT [66], Tracktor++ [67], and FFT [68]. The attention mechanism-based MOT is inspired by the powerful processing ability of the Transformer model in natural language processing. Representative algorithms include TransTrack [69] and TrackFormer [70]. TransTrack takes the feature map of the current frame as the key and the target feature query from the previous frame and a set of learned target feature queries from the current frame as the input query of the whole network.

2.3. High-Speed Vision

High-speed vision is a computer vision technology that aims to realize real-time image recognition and analysis through the use of fast and efficient image processing algorithms at high frame rates of 1000 fps or more. It is an important direction in the field of computer vision and is used in a variety of applications, such as intelligent transportation [71], security monitoring [72], and industrial automation [73].

High-speed vision has two properties: (1) the image displacement from frame to frame is small, and (2) the time interval between frames is extremely short. In order to realize vision-based high-speed feedback control, it is necessary to process massive images in a short time. Unfortunately, current image processing algorithms, such as noise reduction, tracking, and recognition, are all based on traditional image data involving dozens of frames per second. An important idea in high-speed image processing is that it reduces the amount of work required for small-scale shifts between high-speed frames. Field programmable gate arrays (FPGAs) and graphics processing units (GPU), which support massively parallel operations, are ideal for processing two-dimensional data such as images. In [74], the authors presented a high-speed vision platform called H^3 vision. This platform employs dedicated FPGA hardware to implement image processing algorithms and enables simultaneous processing of a 256×256 pixel image at 10,000 fps. The hardware implementation of image processing algorithms on an FPGA board provides high performance and low latency, making it suitable for real-time applications that require high-speed image processing. In [75], a super-high-speed vision platform (HSVP) was introduced that was capable of processing 12-bit 1024×1024 grayscale images at a speed of 12,500 frames per second using an FPGA platform. While the fast computing speed of FPGA is ideal for high-speed image processing, its programming complexity and limited memory capacity can pose significant challenges. Compared with FPGA, GPU platforms

can realize high-frame-rate image processing with lower programming difficulty. A GPU-based real-time full-pixel optical flow analysis method was proposed in [76].

In addition to high-speed image processing, high-speed vision feedback control is very important. Gimbal-based camera systems are specifically designed for image streams with dozens of frames per second. Due to the limited size and movement speed of the camera, it is often difficult to track objects at high speeds while simultaneously observing multiple objects. Recently, a high-speed galvanometer-based reflective PTZ camera system was proposed in [77]. The PTZ camera system can acquire images from multiple angles in an extremely short time and virtualize multiple cameras from a large number of acquired image streams. In [78], a novel dual-camera system was proposed which is capable of simultaneously capturing zoomed-in images using an ultrafast pan-tilt camera and wide-view images using a separate camera. The proposed system provides a unique capability for capturing both wide-field and detailed views of a scene simultaneously. To enable the tracking of specific objects in complex backgrounds, a hybrid tracking method that combines convolutional neural networks (CNN) and traditional object tracking techniques was proposed in [21]. This hybrid method achieves high-speed tracking performance of up to 500 fps and has shown promising results in various applications, such as robotics and surveillance.

3. Active Multi-Object Ultrafast Tracking with CNN-Based Hybrid Object Detection

We propose the concept of wide field of view registration and high-speed multi-object active tracking by virtual cameras using a galvanometer-based reflective PTZ camera, as shown in Figure 2. This high-speed reflective PTZ camera can change the view thousands of times per second to scan the monitoring area. Objects detected during scanning are registered as tracking targets, then the reflective PTZ camera system switches perspectives between different objects at an ultrafast speed for tracking. By classifying and combining frames of different views, multiple virtual cameras with a frame rate of hundreds of frames can be formed. As mentioned in Section 1, current CNN-based object detectors often have dozens of milliseconds of latency from input frames to output results. Compared to ultra-high-speed galvano-mirror control, detection latencies can negatively impact vision-based feedback control, causing skipped frames without object detection. To address this issue, we propose a framework called HFR multiple target tracking, which combines high-speed TM-based trackers with CNN-based object detectors to achieve low-latency visual feedback at hundreds of Hz. This framework enables the tracking of multiple fast-moving objects, and can be used in real-time applications such as robotics, surveillance, and autonomous navigation.

As shown in Figure 3, the whole multi-object tracking process mainly includes two processes, namely, the new object registration process and the multi-object tracking process. The object registration process first scans the surveillance area at an ultrafast speed and stitches together a high-resolution panoramic image. Subsequently, the CNN-based detector detects the image frame-by-frame, looking for objects of interest to complete the registration. After object registration process, the multi-target tracking process changes the field of view to observe different objects at an extremely fast speed. Meanwhile, we use a CNN-based hybrid detection method in each virtual camera for low-latency visual feedback control. If the object is lost, the object registration process is restarted to register new objects.

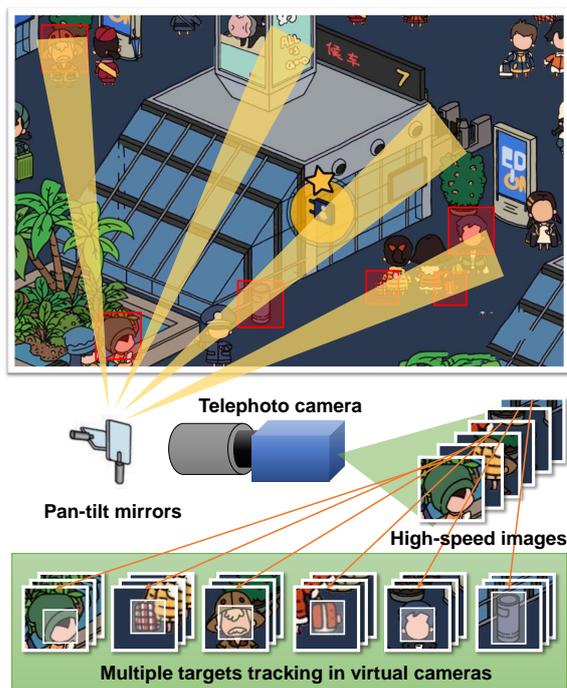


Figure 2. Wide field of view registration and multi-object tracking by virtual cameras using a galvanometer-based reflective PTZ camera.

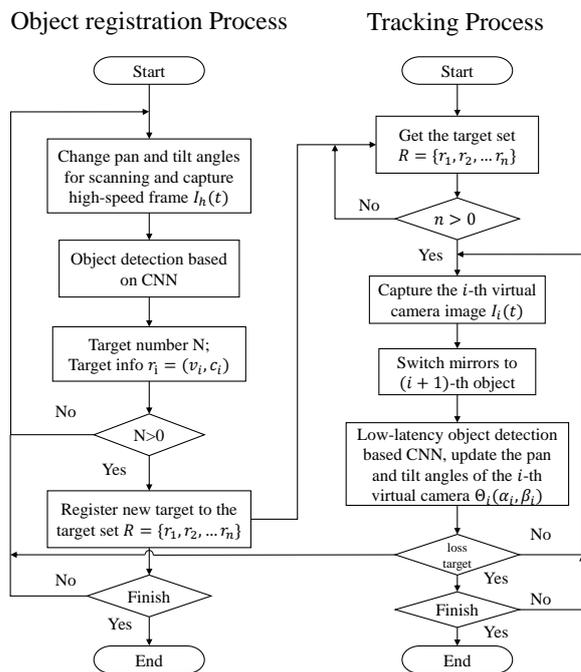


Figure 3. Flowchart of object registration process and multi-object tracking process.

3.1. New Object Registration Process

The high-speed reflective PTZ camera system captures frames of different angles through ultra-fast rotating two-axis galvano-mirrors. During the scanning process of the monitoring area, all captured high-speed frames, denoted as $I_h(t)$, and control angles, denoted as $v(t) = \{u_{pan}(t), u_{tilt}(t)\}$, are stored in the frame set F and angle set V . Meanwhile,

the CNN-based detector performs object detection on the frame set F during the scanning process. The detected objects in an input frame $I_h(t)$ at time t are expressed as follows:

$$D(I_h(t)) = \{d^1(t), d^2(t), \dots, d^j(t), \dots, d^J(t)\}, \quad (1)$$

where D denotes an operator of the CNN-based object detection. For the j -th detected object ($j = 1, \dots, J$), each detection result $d^j(t)$ is composed of the following parameters:

$$d^j(t) = \{o_x^j(t), o_y^j(t), w^j(t), h^j(t), p^j(t), c^j(t)\}, \quad (2)$$

where $o^j(t) = (o_x^j(t), o_y^j(t), w^j(t), h^j(t))$ denotes the bounding box of j -th detection object. In addition, $p^j(t)$ and $c^j(t)$ denote its detection confidence and object class, respectively. In this article, the detection algorithm used in AI detection is YOLOv4, which is currently a very mature detection algorithm with low latency and stable detection time. If an object is detected, we obtain the control angle v_i of the i -th object as follows:

$$v_i = \{u_{pan}^i(t) + \varepsilon_{pan}d_{pan}, u_{tilt}^i(t) + \varepsilon_{tilt}d_{tilt}\}, \quad (3)$$

where $u_{pan}^i(t)$ and $u_{tilt}^i(t)$ denote the control angle of the pan and tilt when the frame is captured at time t , ε_{pan} and ε_{tilt} denote the pixel deviation between the center of the object and the frame center, and d_{pan} and d_{tilt} denote the gain between the pixel and the angle. Then, the detected object info $r_i = \{v_i, c_i\}$ is registered to the target set $R = \{r_1, r_2, \dots, r_n\}$ for high-speed multi-object tracking, while c_i denotes the label of the i -th object.

3.2. Multi-Target Tracking Process

The multi-object tracking process is initiated when the target set size R is greater than 0. In the image acquisition process of the galvanometer-based reflective PTZ camera (as depicted in Figure 4a, the galvano-mirror movement and camera exposure represent the two primary stages. To cope with high-speed image streams of hundreds of frames per second, we parallelize the image acquisition and processing process and divide the stream into multiple virtual cameras.

YOLOv4 can process only about 30 frames per second, and struggles to keep up with the real-time processing demands of the hundreds of frames per second in each virtual camera. To overcome this challenge, we developed a hybrid algorithm that combines template matching and CNN-based object detection. Specifically, the object template image obtained from the CNN detector is matched with the current image through template matching to update the object position in each virtual camera. Our hybrid algorithm comprises two modes, as illustrated in Figure 4b: (1) playback mode, which performs real-time playback tracking in all intermediate images from the detected frame to the current frame once CNN obtains the object position, and (2) frame-by-frame forward tracking, which matches the template obtained from CNN frame-by-frame with the new input image.

To reduce the processing delay caused by the CNN, we activate the instant playback tracking mode to address the deviation caused by the difference between a newly detected fast-moving object's position and its position in the current frame. After updating the template of the TM-based tracker with the detected object area, the TM-based tracker estimates the target position in the current image and the new object position in all frames from the detected input frame to the current frame. The estimated target area of the current frame during playback tracking is used to determine the template of the TM-based tracker and the initial position of frame-by-frame tracking.

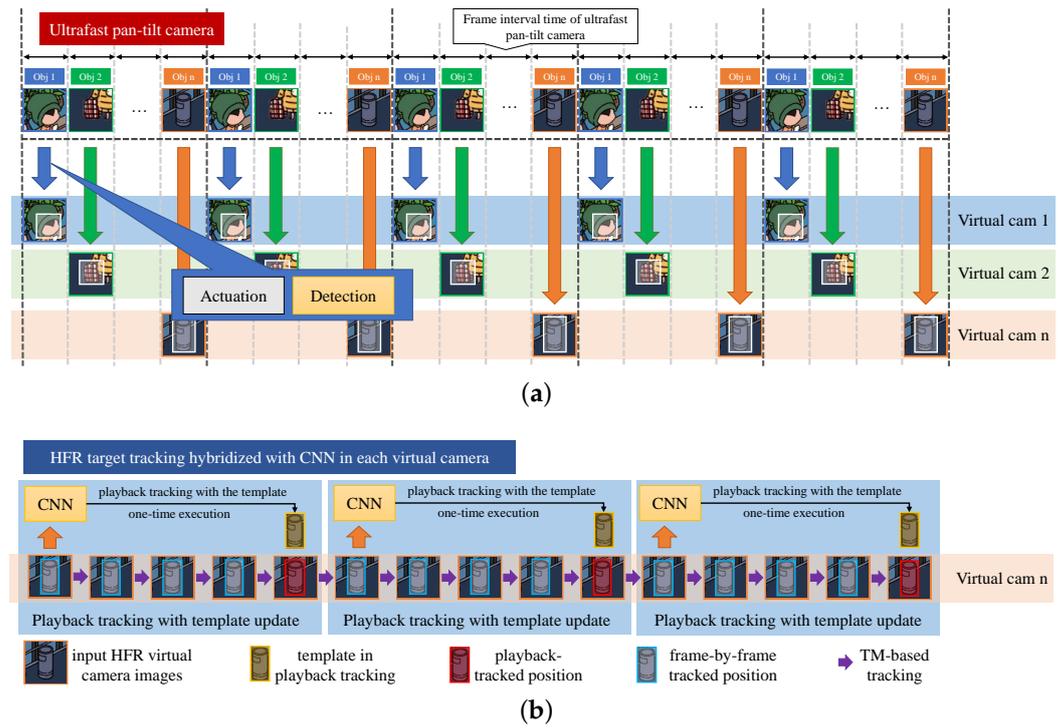


Figure 4. Time-division threaded gaze control process for multiple target tracking based on HFR object detection hybridized with CNN: (a) time-division threaded gaze control process for simultaneous multi-object observation and (b) HFR target tracking hybridized with CNN in each virtual camera.

Below, we describe the algorithm used in the hybridized object-tracking approach. We denote the time intervals of the input HFR images and CNN-based object detection as τ_h and τ_d , respectively, with τ_d being much larger than τ_h and equal to $m\tau_h$.

(1) Updating object template using CNN

Due to detection latency, CNN detectors skip frames and continuously detect images from the n th virtual camera. The objects detected in an input image $I_n(t_d)$ from the n th virtual camera at time t_d can be described as follows:

$$D(I_n(t_d)) = \{d_n^1(t_d), d_n^2(t_d), \dots, d_n^J(t_d)\}. \quad (4)$$

The definition of the detection results $d_n^j(t_d)$ is the same as that in Equation (2). Results that differ from the target labels c_n tracked by the n th virtual camera are initially eliminated. To update the template I_n of the n th virtual camera for the S detection results for which the class is the same as c_n , we use the following method:

$$T_n = \begin{cases} \operatorname{argmax}_{T_s} \operatorname{NCC}(T_s, T'_n) (S > 0) \\ T_s \\ \text{don't update}, (S = 0), \end{cases} \quad (5)$$

$$\operatorname{NCC}(T_s, T'_n) = \frac{\operatorname{Cov}(T_s, T'_n)}{\sqrt{\operatorname{Var}(T_s) \operatorname{Var}(T'_n)}}. \quad (6)$$

Here, T_n and T'_n are the current and last template of the n th virtual camera, respectively. The NCC [79] (Normalized Cross Correlation) algorithm is used to measure the similarity of templates.

(2) TM-Based HFR Tracking

(a) Frame-by-frame forward tracking: when there is no template update the position $p(t_n)$ of the tracked target in the image is obtained directly through the SDS (standard deviation of squares) equation, as shown below:

$$p(t_n) = p'(t_n) + \arg \min_{|x| \leq Ran, |y| \leq Ran} E(x, y), \quad (7)$$

$$E(x, y) = \frac{\sum_{x', y'} (T_n(x', y') - I_n(x'_n + x + x', y'_n + y + y'))^2}{\sqrt{\sum_{x', y'} T_n(x', y')^2 \cdot \sum_{x', y'} I_n(x'_n + x + x', y'_n + y + y')^2}}. \quad (8)$$

Here, $p'(t_n)$ is the position of the object in the last image. Due to the advantage of high-speed vision, objects move more slowly between high-speed image sequences and there is less displacement between frames. Accordingly, we can set the search range Ran for object detection in the image to a few pixels. Here, (x'_n, y'_n) represents the top-left coordinate of the target region in the image from the previous time step in the n th virtual camera, while I_n represents the new image from the n th virtual camera. Because the algorithm is applied to a real-time high-speed system, we prioritize speed over accuracy and robustness.

(b) Playback tracking during template updating: there is a fatal problem with frame-by-frame template matching, which is that the appearance of a moving object often changes. As time progresses, the tracking becomes unstable. Therefore, when updating the template it is necessary to perform a playback operation. The playback operation refers to the process of performing a sub-forward TM through all image sequences from the time $t'_n = t_n - t_d$ at which the previous input image was passed to the CNN until the current time t_n .

$$p(t'_n + (k + 1)\tau_n) = p(t'_n + k\tau_n) + \arg \min_{|x| \leq Ran, |y| \leq Ran} E(x, y), (0 \leq k\tau_n \leq t_d). \quad (9)$$

Here, τ_n represents the time interval between frames for the n th virtual camera. As shown in Figure 4b, we perform a replay operation every time we update the template to avoid the problem of changes in the object's appearance. To reduce latency in CNN-based object detection at t_d intervals, playback tracking functions are utilized as delay compensators, while frame-by-frame forward tracking functions serve as frame interpolators, converting target positions from t_d intervals to τ_n intervals that match the HFR input images.

4. Experiments

4.1. System Configuration

To enable tracking of multiple fast-moving targets distributed across a wide area, we developed a high-speed pan-tilt camera system that utilizes an ultrafast galvanometer mirror. The system is capable of tracking multiple moving objects simultaneously with a frame rate of 500 fps. The system includes a high-speed CMOS camera head from Image Source, Bremen, Germany (DFK37BUX287), a two-axis pan-tilt Galvano-mirror from Cambridge Technology, Kansas City, MO, USA (6210H), and a control computer with an Intel i9-9900K processor (3.6 GHz), 64-GB DDR4 RAM, and Windows 10 Home (64-bit). Control signals are sent to the Galvano-mirror via a D/A board (PEX-340416) from Interface Corporation, Hiroshima, Japan.

In this paragraph, we describe the technical specifications of the galvanometer-based reflective PTZ camera system. The camera head has a 50 mm telephoto lens and a color CMOS sensor measuring 720×540 pixels. The sensor has a size of 4.96×3.72 mm and a pixel size of 6.9×6.9 μm . It can capture 8-bit RGB 720×540 images at 539 fps and transfer them to a PC via USB 3.1. The galvanometer mirror provides two degrees of freedom gaze control, with a range of -20 to 20 degrees for pan and -10 to 10 degrees for tilt. The mirror can be controlled within 2 ms in the ten-degree range. The the overview and geometry of the galvanometer-based reflective PTZ camera system are presented in Figure 5. Real-time

control signals from the computer to the galvano-mirror via the D/A board enable the system to zoom in on and track multiple objects.

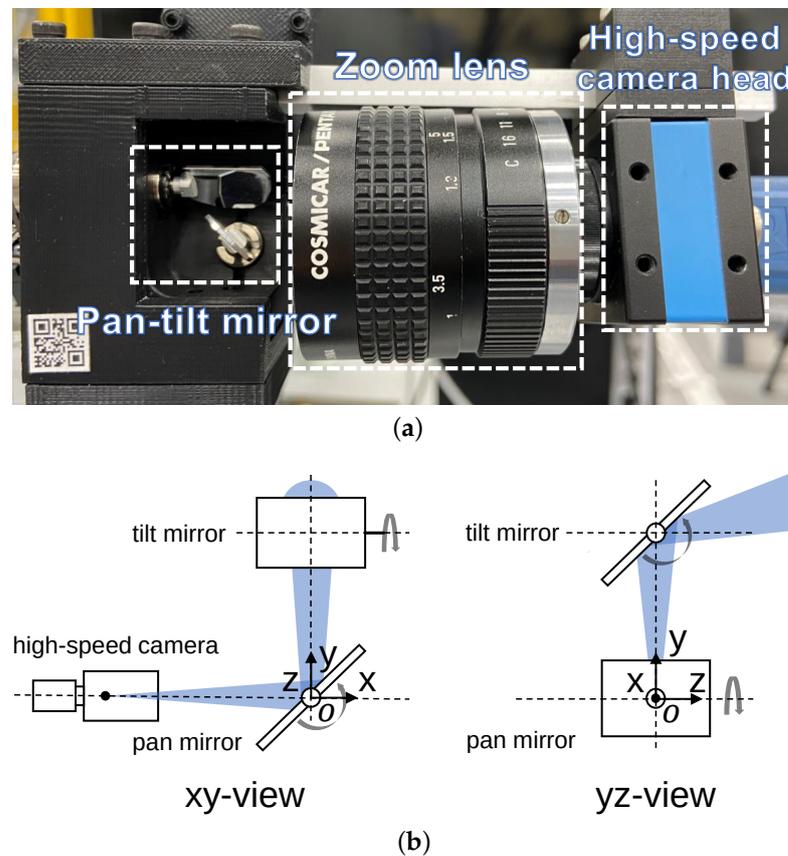


Figure 5. (a) Overview and (b) geometry of the galvanometer-based reflective PTZ camera.

4.2. Execution Times of Visual Tracking Algorithm

The system captures 640×480 images at a rate of 500 fps ($\tau_h = 2$ ms), with the initial search areas determined by the position of the moving object by the new object registration process.

Using YOLOv4 for CNN detection, we can execute object detection with a delay of about $\tau_d = 33$ ($m = 16$) ms using multiple high-speed PTZ virtual cameras. In the implemented YOLOv4, eighty object categories (car, bicycle, sports ball, apple, mouse, etc.) were pre-trained using the COCO dataset, with color images resized to 416×416 for the purpose of estimating object regions and labels. We fine-tuned the YOLOv4 pre-trained model to incorporate facial detection, allowing the network to detect human faces. The average latency of the processing pipeline for object detection in our system is $\tau_l = 30$ ms ($L = 15$). Considering that the frame rate of a high-speed virtual PTZ camera increases with the number of objects to be tracked, the frame rate drops by almost 100 frames. Thus, the displacement of the object between adjacent frames in a virtual PTZ camera is slightly larger than the continuous image stream of 500 fps. The search range in TM-based tracking is set to the 5×5 neighborhood ($R = 4$) in both instant playback tracking and forward tracking modes. The sub-images obtained from the template image are adaptively down-sampled according to image size before completing the template matching, thereby speeding up the playback tracking process and keeping the time within 2 ms. Our algorithm enables pan-tilt tracking with 500-fps visual feedback control to track multiple moving targets at the image center $(c_x, c_y) = (320, 240)$. We evaluated the execution times of our algorithm with template sizes of 64×64 , 128×128 , 256×256 , and 512×512 pixels and compared the results with those of the following single-object tracking algorithms prepared as tracking APIs in OpenCV 4.5: MIL, BOOSTING, Median Flow, TLD, KCF, GOTURN

pre-trained on the ALOV300++ dataset, and MOSSE. Table 1 summarizes the execution times for implementation on 640×480 input images using the same PC used for our proposed system. For our algorithm with $R = 4$ and $L = 6$, we show the execution time for (i) playback tracking with template updating, (ii) frame-by-frame forward tracking, and (iii) YOLOv4; YOLOv4 is executed in parallel with the template matching track, and the largest processing delay occurs in playback tracking. Thus, it is necessary to increase the robustness of object tracking under large processing delays. Our algorithm can deliver target positions in real-time, achieving a speed of hundreds of fps or higher through parallel execution with YOLOv4. Compared with other single-object tracking approaches based on either online adaptive templates or pre-trained deep neural networks, our algorithm offers an advantage in terms of processing speed.

Table 1. Execution times of tracking algorithms.

Tracker	Size			
	64×64	128×128	256×256	512×512
BOOSTING	17.73	54.85	29.85	8.11
KCF	5.39	5.05	20.6	90.22
MOSSE	0.26	1.12	1.55	17.26
MIL	79.63	76.67	72.68	67.71
TLD	30.15	22.14	26.45	26.99
MEDIANFLOW	2.12	2.21	2.10	2.23
GOTURN	23.22	24.54	24.74	29.60
ours(playback)	0.27	0.41	0.77	1.82
(forward)	0.021	0.056	0.222	0.62
(YOLOv4)			33	

(unit: ms).

4.3. Simultaneous Tracking of Twenty Different Objects

We first tested the proposed multi-object tracking system based on reflective mirrors for tracking a large number of targets. Figure 6a shows the overview of the experimental scene. We placed twenty different types of targets, such as cars, bicycles, clocks, and sports balls, on the wall and whiteboard approximately 6 m away from the multi-object tracking system. Among these, the whiteboard was movable and the objects on the whiteboard were able to move along with the motion of the whiteboard. During the object registration process, the galvanomirror-based reflective camera system scanned the monitoring area at a speed of 500 fps. Figure 6b shows the panoramic image stitched together from the reflective camera system following completion of the object registration process. The resulting twenty detected objects were mapped onto the panoramic image in the form of overlays, and their positions in the panoramic image were updated in real-time. The virtual camera labels are shown in “v-cam:n”, where different virtual cameras are assigned different numbers. The virtual camera responsible for scanning is displayed as a green bounding box in the top right corner of the image. If object tracking failed, the virtual camera responsible for scanning was reactivated to search for new objects. Figure 7 presents high-definition images of all tracked objects continuously updated within the image frames at 30 fps. Figure 8 shows the pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking twenty different targets. The other objects on the wall remained stationary while the objects (the bicycle and car) attached to the whiteboard (virtual cameras 8, 13, 18) moved left and right along with the whiteboard. Because high-speed resources are divided equally, the frame rate of each target is low and certain fast-moving objects cannot be tracked accurately due to their high motion speed.

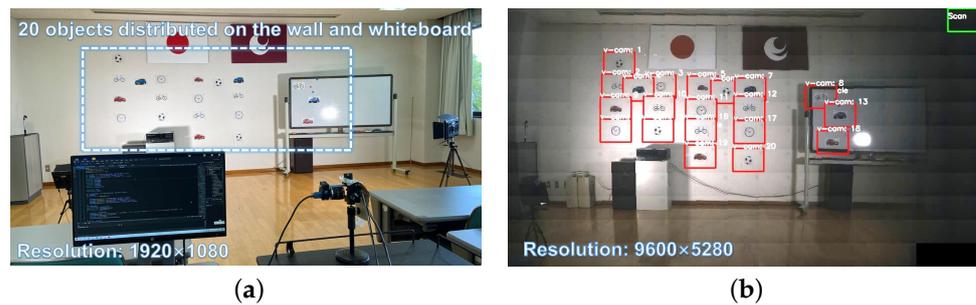


Figure 6. The 1920×1080 input images from the digital camera and the panoramic stitched 9600×5280 images from the PTZ camera: (a) overview of the experimental scene and (b) panoramic stitched image from the PTZ camera (targets are pasted on the panoramic image in the form of a red frame texture).



Figure 7. HD images of twenty objects tracked simultaneously.

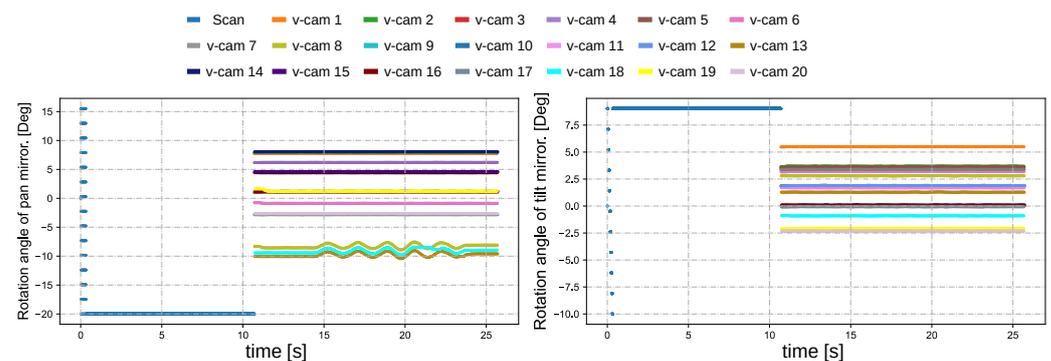


Figure 8. Pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking twenty different targets.

4.4. Low-Latency Pan-Tilt Tracking of Multiple Moving Bottles

Next, we verified the multi-object visual tracking performance of our proposed system at 500 fps. In this experiment, we employed a visual search to automatically detect bottles that were distributed in the surveillance area. By changing the viewpoint, we were able to track multiple bottles at the image centre of a 640×480 input image. The experimental environment is depicted in Figure 9. Three bottles were strategically positioned at a distance of about 8 m from the PTZ camera system. Subsequently, two of the bottles were released from a height of approximately 1.7 m while the camera system finished searching and needed to track multiple bottles. A digital camera (model DSC-RX10M3, focal length

30 mm) was positioned adjacent to the camera system to capture 1920×1080 images of the surveillance area at 60 frames per second.

The initial stage of the experiment involved a zigzag scan of the monitoring area using the PTZ camera system. Specifically, the pan mirror was adjusted by 2.54 degrees and the tilt mirror by 1.9 degrees for each scan. The final observation range of the pan mirror was set between -17.78 degrees and 17.78 degrees, while the tilt mirror was set to observe within a range of -9.5 degrees to 9.5 degrees. Figure 10 shows the 1920×1080 input images from digital camera and panoramic stitched 9600×5280 images from the PTZ camera. Our PTZ camera system can obtain $24\times$ higher-definition images of the surveillance area at 3 fps compared with digital cameras. Subsequently, we utilized YOLOv4 to analyze and detect objects in the 165 images captured during the zigzag scan process. Following the completion of object registration, we started a low-latency tracking process to track the detected bottles in real time. Figure 11 shows the 145×108 ROI images around the targets from the digital wide-view camera and 640×480 input images from the virtual PTZ cameras at $t = 19.1$ s. In the image obtained from PTZ camera, the characters on the bottles can be clearly read while being robustly tracked. In contrast, only the approximate outline and color of the bottles can be seen in the digital camera image. During tracking, we first picked up two bottles from the table, then released the bottles into a free fall at $t = 19.1$ s.

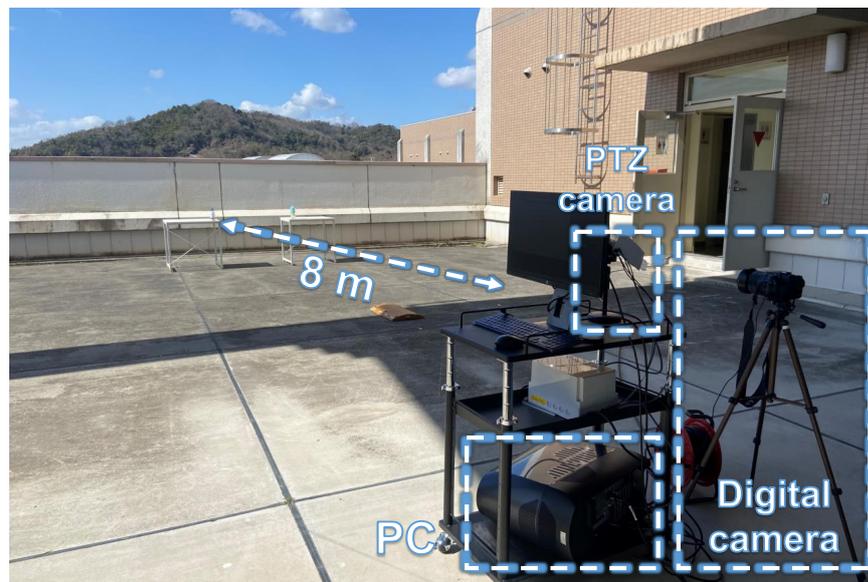


Figure 9. Experimental environment used for tracking multiple moving objects in an outdoor scene.



Figure 10. The 1920×1080 input images from the digital camera and panoramic stitched 9600×5280 images from the PTZ camera: (a) input image of digital camera and (b) panoramic stitched image from the PTZ camera.

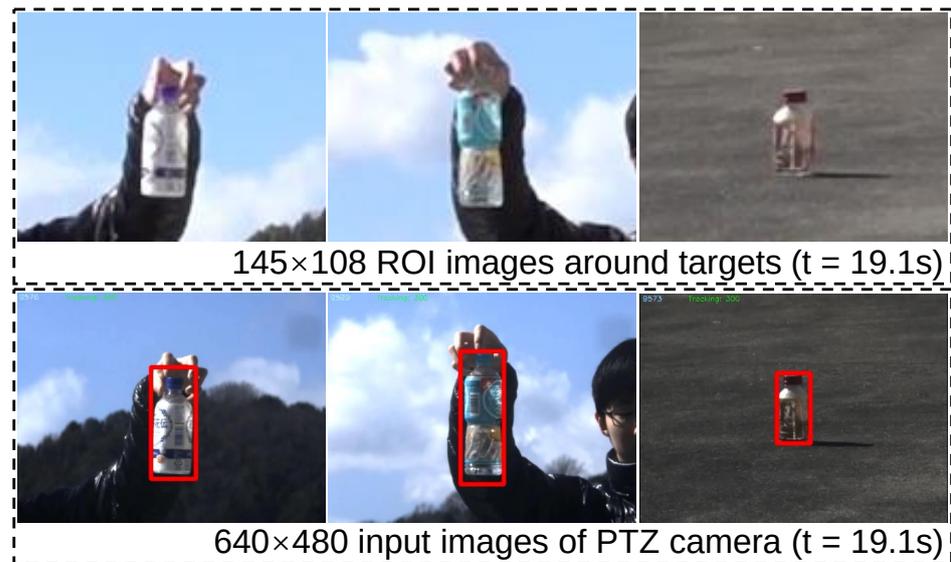


Figure 11. The 145×108 ROI images around targets from the digital wide-view camera and 640×480 input images from the virtual PTZ cameras (red boxes are the test results).

Figure 12 shows the pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking multiple bottles. The system undergoes object registration from 0 to 9 s, after which it transitions to the multi-target tracking process. We simultaneously released two water bottles from a height of about 1.7 m at 19.1 s. The two bottles experienced about 0.6 s of freefall. Figure 13 illustrates the x and y coordinates of the centroids for the regions of interest (ROIs) that were tracked in the input images during the period $t = 9$ –23 s. Except in the process of falling, the deviation from the center of the image gradually increases, and is otherwise very close to the center of the image (320×240).

Figure 14 depicts the tracking status during free fall of bottle 1 when tracking three bottles simultaneously. Figure 14a,b shows the tracking situations based on the CNN hybrid algorithm and YOLOv4, respectively. When using only YOLO tracking, objects leave the field of view quickly at $t = 0.3$ s as their speed increases. Nevertheless, the CNN-based hybrid tracking algorithm exhibits superior performance in tracking the falling bottle. The velocity of the free-falling bottle is directly proportional to the time it takes to fall. Figure 15 shows the relationship between the velocity and distance from the detection ROI to the image center during free-fall bottle tracking. YOLOv4 tracking is limited in its ability to track objects with a speed greater than 3 m per second. Using our CNN-based hybrid algorithm for tracking, a moving object with a speed of 5.5 m/s is located approximately 45 pixels away from the image center. In theory, it is possible to track three objects moving at a speed of 30 m/s and maintaining a distance of 8 m simultaneously. Finally, we conducted experiments to track three free-falling bottles simultaneously using different tracking methods. Figure 16 shows pixel deviation values between the object position calculated by different algorithms and the object's actual position during the falling process. The actual position of the object was obtained from offline videos recorded at 30 fps during online system operation. The deviation values shown in the figure represent the specific error at each time step. After being dropped from a height of approximately 1.7 m, the water bottle impacted the ground after approximately 0.6 s. The Boosting, TLD, KCF, and Medianflow tracking methods lost the target within 0.2 s, 0.3 s, 0.4 s, and 0.5 s, respectively. The GOTURN and MOSSE tracking methods, along with our proposed method, were able to maintain tracking until the end of the tracking task. Among these methods, the deviation of the tracked object's real position is smaller when using our proposed hybrid tracking method based on CNN.

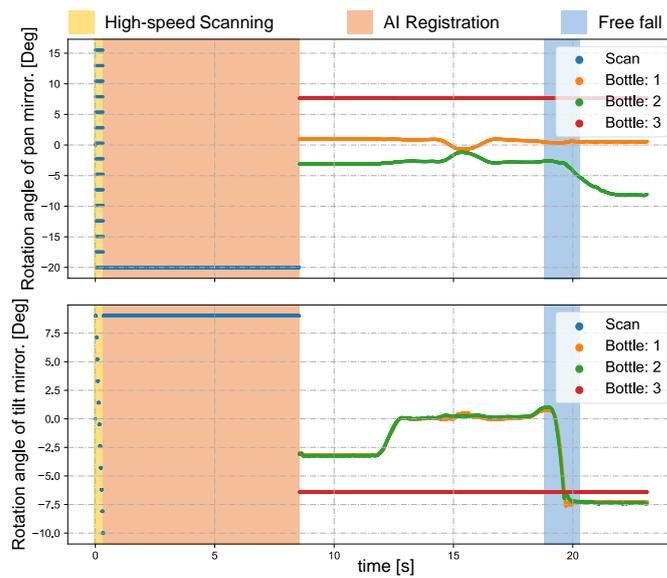


Figure 12. Pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking multiple bottles.

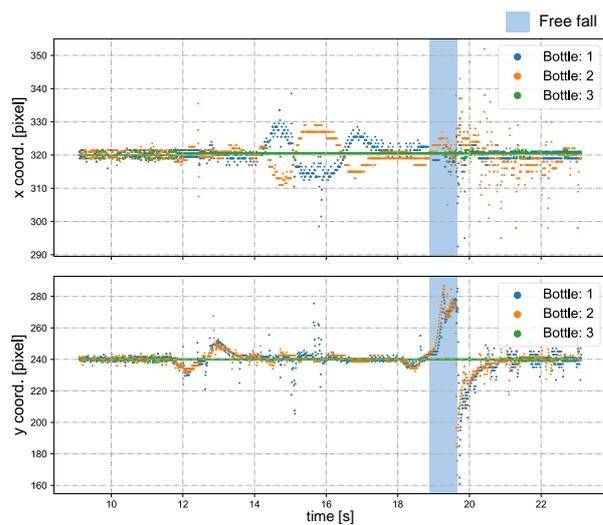


Figure 13. The x and y centroids of tracked bottle regions.

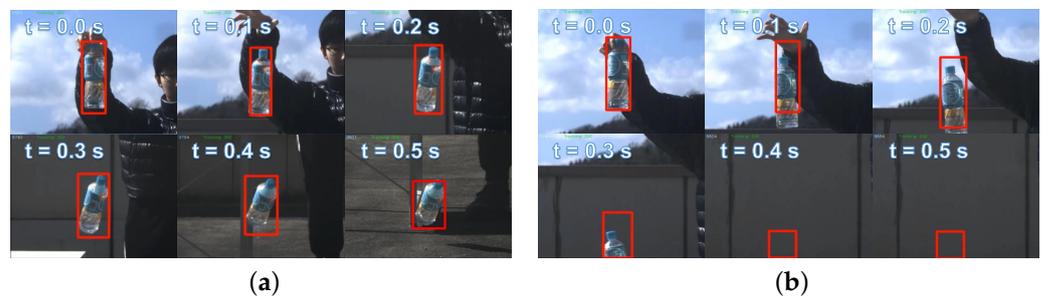


Figure 14. Tracking status of the free-fall of bottle 1 when tracking three bottles simultaneously: (a) free-fall of bottle 1 based on CNN hybrid tracking and (b) free-fall of bottle 1 based on YOLOv4 tracking.

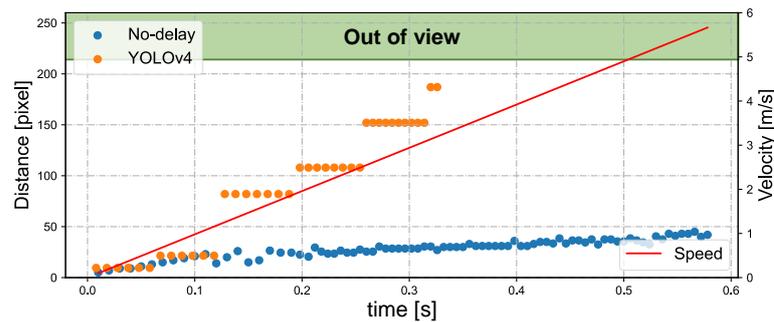


Figure 15. Relationship between velocity and distance from the detection ROI to the image center during free-fall bottle tracking.

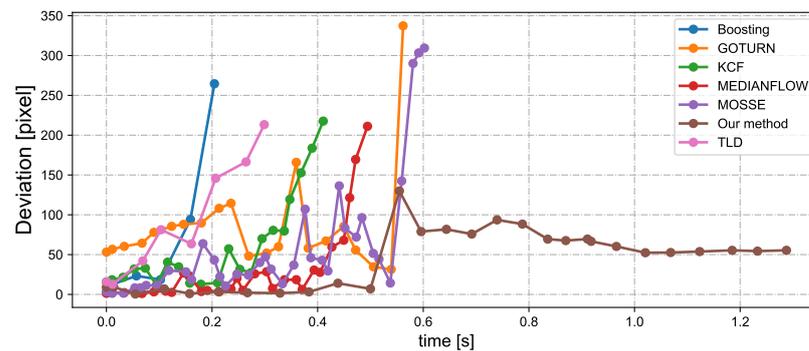


Figure 16. Pixel deviation value between the object position calculated by different algorithms and the object's real position during free-falling.

4.5. Multi-Person Pan-Tilt Tracking in Wide-Area Surveillance

Subsequently, we designed a scenario more commonly encountered in reality involving multiple individuals in motion. Figure 17a shows the experimental environment captured by the digital camera (focal length = 40 mm). Five individuals were positioned approximately 8 m away from the PTZ camera system. Prior to the completion of the scans and detection, all participants stood still. Figure 17b depicts the scanning and detection outcome at the start of the experiment. Figure 18 depicts the pan and tilt angles of the galvanometer during the experiment as it scanned and tracked different individuals. The upper right corner shows a thread for detecting lost targets and initiating rediscovery. The tracking of multiple individuals commenced at 8.5 s. Between 10 and 13 s, all individuals performed vertical jumps, while from 13 to 15 s, they swayed their bodies horizontally. A loss waiting time of 300 frames was established for each individual. After a waiting time of 300 frames, person 1 was lost at 16.5 s and the scanning thread was restarted; person 1 was eventually rediscovered at 21 s. Two crossings occurred between person 2 and person 3 during 23 to 27 s and 30.5 to 34 s. Sface [80] was employed to extract facial features and conduct similarity matching. In this study, we only utilized facial features for facial discrimination, and did not assign individual IDs for recognition of individuals. During the cross-tracking process, a virtual camera always followed a single person. As shown in Figure 19, person 3 was briefly occluded by person 2 during the initial cross-tracking process and was subsequently re-identified and tracked. In the second crossing process, person 3 was not identified during the short occlusion; thus, the scanning thread had to be restarted, leading to the rediscovery of person 3 at 33 s, as shown in Figure 20. Due to lighting conditions, person 5 was not detected at 26 s, prompting the scanning thread to remain active from that point forward in an attempt to locate person 5. The x and y coordinate values of the image centroids of the tracked ROIs are depicted in Figure 21. These results demonstrate that our PTZ camera system can track multiple fast-moving objects simultaneously at a high speed. Moreover, the system exhibits high robustness to object loss and occlusion.

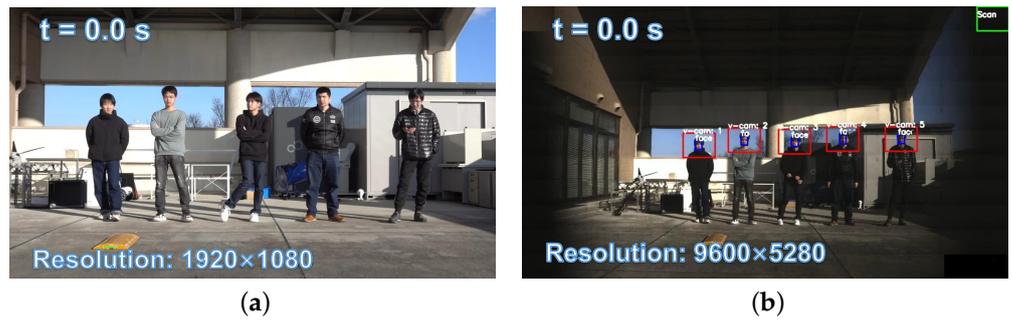


Figure 17. The 1920×1080 input images from the digital camera and panoramic stitched 9600×5280 images from the PTZ camera: (a) input image of the digital camera and (b) panoramic stitched image from the PTZ camera.

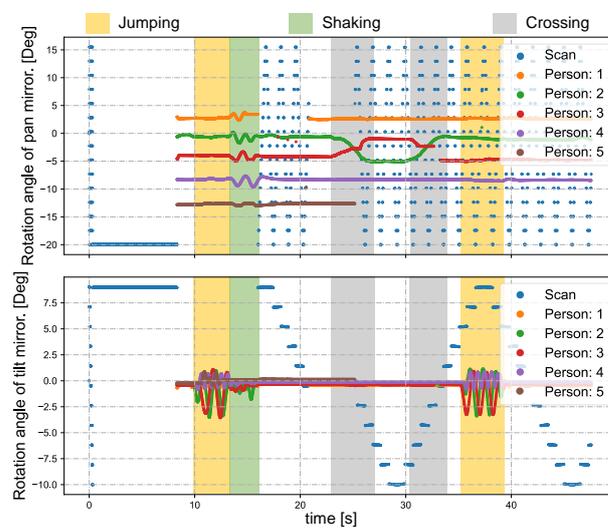


Figure 18. Pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking multiple persons.

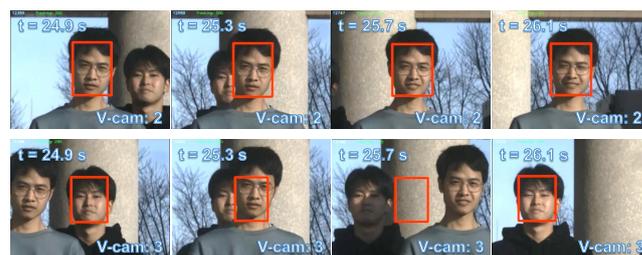


Figure 19. Cross-tracking of person 2 and person 3 between 24.9 and 26.1 s.

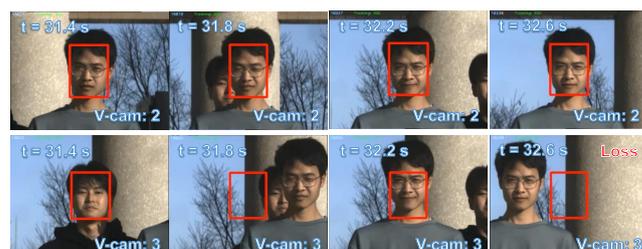


Figure 20. Cross-tracking of person 2 and person 3 between 31.4 and 32.6 s.

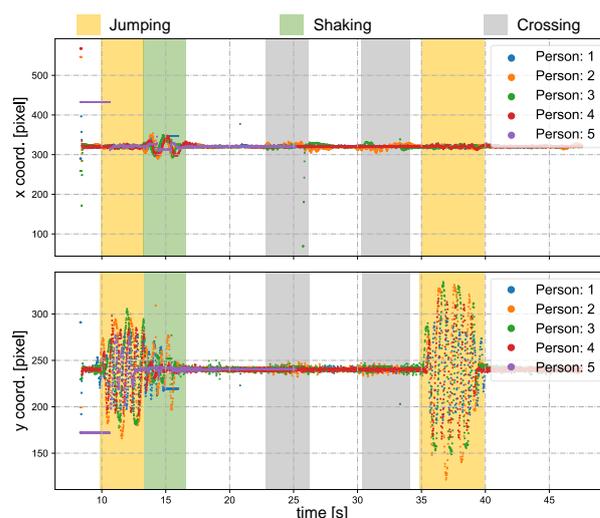


Figure 21. The x and y centroids of the regions with tracked people.

5. Conclusions

In this study, we developed a multi-object visual surveillance system with 500-fps image processing capabilities able to robustly track multiple objects within a wide area. The effectiveness of our system was demonstrated through two experiments: (1) tracking of multiple free-falling water bottles, and (2) tracking of multiple freely moving individuals. Our system offers three key advantages: (1) rapid generation of high-definition wide-view images; (2) the ability to track up to twenty low-speed moving targets at a maximum rate of 25 fps; and (3) simultaneous tracking of multiple high-speed moving targets with high robustness against object occlusion and loss.

However, there are several limitations faced by the current system; for example, the object registration process in the early stage requires several seconds, during which time the object may continue to move, resulting in target loss during the tracking process. In future work, we plan to incorporate a panoramic camera for pre-detection of interesting targets within the monitoring area.

Author Contributions: All authors contributed to designing this study and writing the manuscript. I.I. contributed to the concept of a reflective active visual tracking system with multiple targets. S.H. and K.S. designed the ultrafast mirror-drive active vision system that can switch hundreds of different views in a second. Q.L. developed a multi-target high-speed tracking system based on a galvanometer camera, and conducted multi-target tracking experiments in complex scenes. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [China Scholarship Council] grant number [202006540007].

Institutional Review Board Statement: I think this study did not require ethical approval.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study. And I think this study did not involve humans.

Data Availability Statement: Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Conflicts of Interest: No potential conflict of interest was reported by the authors.

References

1. Luo, W.; Xing, J.; Milan, A.; Zhang, X.; Liu, W.; Kim, T.K. Multiple object tracking: A literature review. *Artif. Intell.* **2021**, *293*, 103448. [\[CrossRef\]](#)
2. Mwamba, I.C.; Morshedi, M.; Padhye, S.; Davatgari, A.; Yoon, S.; Labi, S.; Hastak, M. *Synthesis Study of Best Practices for Mapping and Coordinating Detours for Maintenance of Traffic (MOT) and Risk Assessment for Duration of Traffic Control Activities*; Technical Report; Joint Transportation Research Program; Purdue University: West Lafayette, IN, USA, 2021.

3. Ghafir, I.; Prenosil, V.; Svoboda, J.; Hammoudeh, M. A survey on network security monitoring systems. In Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, Austria, 22–24 August 2016; pp. 77–82.
4. Qureshi, K.N.; Abdullah, A.H. A survey on intelligent transportation systems. *Middle-East J. Sci. Res.* **2013**, *15*, 629–642.
5. Weng, X.; Wang, J.; Held, D.; Kitani, K. 3d multi-object tracking: A baseline and new evaluation metrics. In Proceedings of the 2020 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, 24–30 October 2020; pp. 10359–10366.
6. Rządca, K.; Findeisen, P.; Swiderski, J.; Zych, P.; Broniek, P.; Kusmierk, J.; Nowak, P.; Strack, B.; Witusowski, P.; Hand, S.; et al. Autopilot: Workload autoscaling at Google. In Proceedings of the Fifteenth European Conference on Computer Systems, New York, NY, USA, 8–12 May 2023; pp. 1–16.
7. Bohush, R.; Zakharava, I. Robust person tracking algorithm based on convolutional neural network for indoor video surveillance systems. In Proceedings of the Pattern Recognition and Information Processing: 14th International Conference, PRIP 2019, Minsk, Belarus, 21–23 May 2019; Revised Selected Papers 14, pp. 289–300.
8. Kaiser, R.; Thaler, M.; Kriechbaum, A.; Fassold, H.; Bailer, W.; Rosner, J. Real-time Person Tracking in High-resolution Panoramic Video for Automated Broadcast Production. In Proceedings of the 2011 Conference for Visual Media Production, Washington, DC, USA, 16–17 November 2011; pp. 21–29. [[CrossRef](#)]
9. Fukami, K.; Fukagata, K.; Taira, K. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* **2019**, *870*, 106–120. [[CrossRef](#)]
10. Anwar, S.; Khan, S.; Barnes, N. A deep journey into super-resolution: A survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [[CrossRef](#)]
11. Carles, G.; Downing, J.; Harvey, A.R. Super-resolution imaging using a camera array. *Opt. Lett.* **2014**, *39*, 1889–1892. [[CrossRef](#)] [[PubMed](#)]
12. Kang, S.; Paik, J.K.; Koschan, A.; Abidi, B.R.; Abidi, M.A. Real-time video tracking using PTZ cameras. In Proceedings of the Sixth International Conference on Quality Control by Artificial Vision, Gatlinburg, TN, USA, 19–22 May 2003; Volume 5132, pp. 103–111.
13. Ding, C.; Song, B.; Morye, A.; Farrell, J.A.; Roy-Chowdhury, A.K. Collaborative sensing in a distributed PTZ camera network. *IEEE Trans. Image Process.* **2012**, *21*, 3282–3295. [[CrossRef](#)] [[PubMed](#)]
14. Li, Q.; Chen, M.; Gu, Q.; Ishii, I. A Flexible Calibration Algorithm for High-speed Bionic Vision System based on Galvanometer. In Proceedings of the 2022 IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 4222–4227. [[CrossRef](#)]
15. Gu, Q.Y.; Ishii, I. Review of some advances and applications in real-time high-speed vision: Our views and experiences. *Int. J. Autom. Comput.* **2016**, *13*, 305–318. [[CrossRef](#)]
16. Okumura, K.; Oku, H.; Ishikawa, M. High-speed gaze controller for millisecond-order pan/tilt camera. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 6186–6191.
17. Aoyama, T.; Li, L.; Jiang, M.; Inoue, K.; Takaki, T.; Ishii, I.; Yang, H.; Umemoto, C.; Matsuda, H.; Chikaraishi, M.; et al. Vibration sensing of a bridge model using a multithread active vision system. *IEEE/ASME Trans. Mechatron.* **2017**, *23*, 179–189. [[CrossRef](#)]
18. Hu, S.; Shimasaki, K.; Jiang, M.; Takaki, T.; Ishii, I. A dual-camera-based ultrafast tracking system for simultaneous multi-target zooming. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 521–526.
19. Jiang, M.; Sogabe, R.; Shimasaki, K.; Hu, S.; Senoo, T.; Ishii, I. 500-fps omnidirectional visual tracking using three-axis active vision system. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [[CrossRef](#)]
20. Chua, L.O.; Roska, T. The CNN paradigm. *IEEE Trans. Circuits Syst. Fundam. Theory Appl.* **1993**, *40*, 147–156. [[CrossRef](#)]
21. Jiang, M.; Shimasaki, K.; Hu, S.; Senoo, T.; Ishii, I. A 500-fps pan-tilt tracking system with deep-learning-based object detection. *IEEE Robot. Autom. Lett.* **2021**, *6*, 691–698. [[CrossRef](#)]
22. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [[CrossRef](#)]
23. Turk, M.; Pentland, A. Eigenfaces for Recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86. [[CrossRef](#)] [[PubMed](#)]
24. Amit, Y.; Felzenszwalb, P.; Girshick, R. Object detection. In *Computer Vision: A Reference Guide*; Springer International Publishing: Cham, Switzerland, 2020; pp. 1–9.
25. Li, G.; Xu, D.; Cheng, X.; Si, L.; Zheng, C. Simvit: Exploring a simple vision transformer with sliding windows. In Proceedings of the 2022 IEEE International Conference on Multimedia and Expo (ICME), Taipei, Taiwan, 18–22 July 2022; pp. 1–6.
26. Ghasemi, Y.; Jeong, H.; Choi, S.H.; Park, K.B.; Lee, J.Y. Deep learning-based object detection in augmented reality: A systematic review. *Comput. Ind.* **2022**, *139*, 103661. [[CrossRef](#)]
27. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893. [[CrossRef](#)]
28. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. 1. [[CrossRef](#)]

29. Nguyen, V.D.; Tran, D.T.; Byun, J.Y.; Jeon, J.W. Real-time vehicle detection using an effective region proposal-based depth and 3-channel pattern. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 3634–3646. [[CrossRef](#)]
30. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
32. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
33. Targ, S.; Almeida, D.; Lyman, K. Resnet in resnet: Generalizing residual architectures. *arXiv* **2016**, arXiv:1603.08029.
34. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [[CrossRef](#)]
35. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
36. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
37. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [[CrossRef](#)]
38. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
39. Purkait, P.; Zhao, C.; Zach, C. SPP-Net: Deep absolute pose regression with synthetic views. *arXiv* **2017**, arXiv:1712.03452.
40. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
41. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)]
42. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
43. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*.
44. Qiao, S.; Chen, L.C.; Yuille, A. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10213–10224.
45. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
46. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14, pp. 21–37.
47. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
48. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [[CrossRef](#)]
49. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6568–6577. [[CrossRef](#)]
50. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787. [[CrossRef](#)]
51. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
52. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [[CrossRef](#)]
53. Sang, J.; Wu, Z.; Guo, P.; Hu, H.; Xiang, H.; Zhang, Q.; Cai, B. An improved YOLOv2 for vehicle detection. *Sensors* **2018**, *18*, 4272. [[CrossRef](#)] [[PubMed](#)]
54. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
55. Yu, J.; Zhang, W. Face mask wearing detection algorithm based on improved YOLO-v4. *Sensors* **2021**, *21*, 3263. [[CrossRef](#)]

56. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788. [[CrossRef](#)]
57. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
58. Soleimanitaleb, Z.; Keyvanrad, M.A.; Jafari, A. Object Tracking Methods: A Review. In Proceedings of the 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Khorasan Razavi, Iran, 24–25 October 2019; pp. 282–288. [[CrossRef](#)]
59. Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 790–799. [[CrossRef](#)]
60. Carpenter, J.; Clifford, P.; Fearnhead, P. Improved particle filter for nonlinear problems. *IEEE Proc.-Radar Sonar Navig.* **1999**, *146*, 2–7. [[CrossRef](#)]
61. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the ECCV Workshops (2), Zurich, Switzerland, 6–7 September 2014; Volume 8926, pp. 254–265.
62. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
63. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
64. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.
65. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards real-time multi-object tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XI 16, pp. 107–122.
66. Chen, L.; Ai, H.; Zhuang, Z.; Shang, C. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
67. Bergmann, P.; Meinhardt, T.; Leal-Taixe, L. Tracking without bells and whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 941–951.
68. Zhang, J.; Zhou, S.; Chang, X.; Wan, F.; Wang, J.; Wu, Y.; Huang, D. Multiple object tracking by flowing and fusing. *arXiv* **2020**, arXiv:2001.11180.
69. Sun, P.; Cao, J.; Jiang, Y.; Zhang, R.; Xie, E.; Yuan, Z.; Wang, C.; Luo, P. Transtrack: Multiple object tracking with transformer. *arXiv* **2020**, arXiv:2012.15460.
70. Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; Feichtenhofer, C. Trackformer: Multi-object tracking with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 8844–8854.
71. Dimitrakopoulos, G.; Demestichas, P. Intelligent transportation systems. *IEEE Veh. Technol. Mag.* **2010**, *5*, 77–84. [[CrossRef](#)]
72. Loianno, G.; Scaramuzza, D.; Kumar, V. Special issue on high-speed vision-based autonomous navigation of uavs. *J. Field Robot.* **2018**, *1*, 1–3. [[CrossRef](#)]
73. Huang, S.; Bergström, N.; Yamakawa, Y.; Senoo, T.; Ishikawa, M. Applying high-speed vision sensing to an industrial robot for high-performance position regulation under uncertainties. *Sensors* **2016**, *16*, 1195. [[CrossRef](#)] [[PubMed](#)]
74. Ishii, I.; Taniguchi, T.; Sukenobe, R.; Yamamoto, K. Development of high-speed and real-time vision platform, H3 vision. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, New Orleans, LA, USA, 18–24 June 2009; pp. 3671–3678. [[CrossRef](#)]
75. Sharma, A.; Shimasaki, K.; Gu, Q.; Chen, J.; Aoyama, T.; Takaki, T.; Ishii, I.; Tamura, K.; Tajima, K. Super high-speed vision platform for processing 1024 × 1024 images in real time at 12500 fps. In Proceedings of the 2016 IEEE/SICE International Symposium on System Integration (SII), Shiroishi, Japan, 13–15 December 2016; pp. 544–549. [[CrossRef](#)]
76. Gu, Q.; Nakamura, N.; Aoyama, T.; Takaki, T.; Ishii, I. A Full-Pixel Optical Flow System Using a GPU-Based High-Frame-Rate Vision. In Proceedings of the 2015 Conference on Advances In Robotics, Goa, India, 2–4 July 2015; Association for Computing Machinery: New York, NY, USA, 2015. [[CrossRef](#)]
77. Kobayashi-Kirschvink, K.; Oku, H. Design principles of a high-speed omni-scannable gaze controller. *IEEE Robot. Autom. Lett.* **2016**, *1*, 836–843. [[CrossRef](#)]
78. Hu, S.; Shimasaki, K.; Jiang, M.; Senoo, T.; Ishii, I. A simultaneous multi-object zooming system using an ultrafast pan-tilt camera. *IEEE Sens. J.* **2021**, *21*, 9436–9448. [[CrossRef](#)]
79. Yoo, J.C.; Han, T.H. Fast normalized cross-correlation. *Circuits Syst. Signal Process.* **2009**, *28*, 819–843. [[CrossRef](#)]
80. Zhong, Y.; Deng, W.; Hu, J.; Zhao, D.; Li, X.; Wen, D. SFace: Sigmoid-Constrained Hypersphere Loss for Robust Face Recognition. *IEEE Trans. Image Process.* **2021**, *30*, 2587–2598. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.