A study on high-speed stereo multi-object tracking using ultra-fast active vision

（超高速アクティブビジョンを用いた
高速ステレオ複数物体追跡の研究）

by

Qing Li
李　慶

Graduate School of Engineering
Hiroshima University
September, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Stereo multi-object tracking is a technique for simultaneously tracking and localizing multiple objects in stereo vision (i.e., stereo images acquired from multiple cameras or sensors). It combines the concepts of multi-target tracking and spatial positioning, and aims to achieve accurate tracking and position estimation of targets in a stereoscopic environment. By fusing geometric and semantic information in stereo images, stereo multi-object tracking can provide more accurate and robust object tracking and position estimation results. At present, stereo multi-target tracking has been widely used in the fields of robot navigation [1], automatic driving [2], path planning [3], behavior analysis [4] and real-time monitoring [5]. It improves perception and understanding, providing critical support for realizing intelligent, autonomous systems.

Stereo multi-target tracking is mainly divided into two steps, multi-target tracking within a single camera and stereo correspondence across cameras. Multi-object tracking is a key task in computer vision and robotics, which aims to continuously track the position, shape and motion state of a specific object from video, image or sensor data. Then, use the cross-camera data association algorithm to correspond to multiple targets tracked in different cameras, and perform spatial positioning and pose estimation among multiple targets according to the set relationship between cameras or sensors.

Multiple object tracking is a challenging problem in the field of computer vision. Currently, it faces numerous challenges such as occlusion, object loss, complex backgrounds, real-time tracking, and obtaining high-definition images of the tracked objects. Among these challenges, tracking over a large area with high definition contributes to continuous observation and analysis, making it an urgent problem to be solved. There has been extensive research dedicated to this area. Using higher-resolution cameras is a simple and feasible solution. However, high-definition camera equipment typically implies higher cost investment. Another feasible solution is to use a dual-camera system composed of a wide-angle camera and a long-focus PTZ camera. Traditional PTZ cameras mostly use a pan-tilt platform, which is difficult to drive large-sized long-focus cameras to switch between multiple viewpoints. Therefore, they are commonly used for single-object tracking and high-definition photography.

Recently, an ultra-fast gaze control system using high-speed mirrors has been developed, which can switch between hundreds of viewpoints to observe multiple objects within one second. In our laboratory, a dual-camera system based on a wide-angle camera and an ultra-fast mirror camera has been developed, enabling high-definition photography of multiple objects at a frame rate of dozens per second. However, in the current work, due to the lack of real-time detection and visual feedback control in the ultra-fast mirror camera system, the tracking may fail when objects move at a relatively fast speed. Therefore, in the process of multi-object tracking based on an ultra-fast mirror camera, real-time visual feedback control is an urgent problem that needs to be addressed.

Secondly, in multi-camera stereo correspondence, how to match the same object across different cameras is also a challenging problem. Matching based on object appearance information is a common approach. Traditional appearance-based matching methods primarily rely on manually extracted features, such as color histograms, scale-invariant feature transform (SIFT), and oriented FAST and rotated BRIEF (ORB), which have been proven effective. With the development of convolutional neural networks, appearance

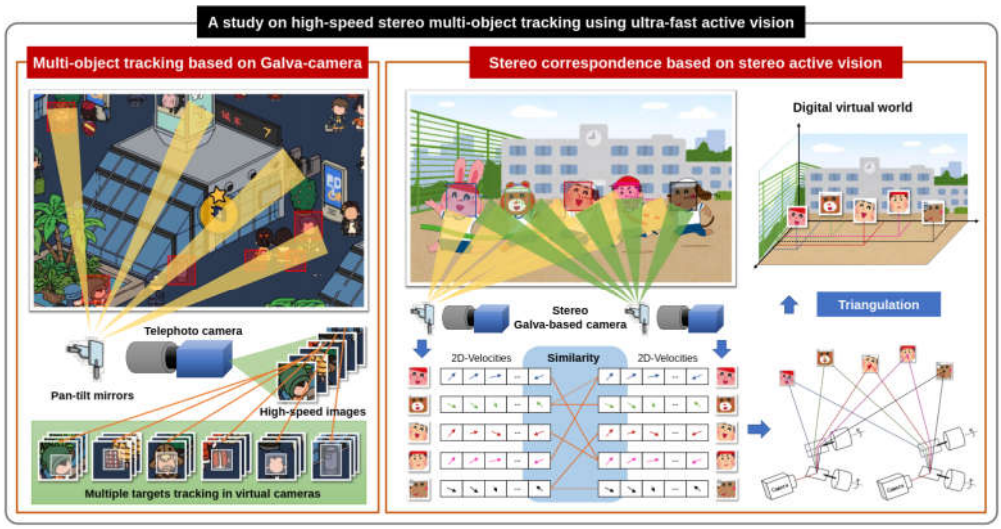


Figure 1.1: Concept overview of this study.

matching methods based on deep learning have demonstrated higher performance. However, appearance-based matching methods are vulnerable to changes in viewing angle, lighting, and pose. This problem becomes even more complicated especially for multiple objects with similar appearance, such as human faces, farm animals, or workers in the same clothing. At the same time, the movement of objects is unique and difficult to completely replicate, which can be used for multi-target matching across cameras. Sensors based on high-speed vision systems have unparalleled advantages, and we can obtain highly synchronized motion information of multiple moving targets in a stereo camera system in a short time.

1.2 Purpose of the research

As mentioned in the previous section, stereo multi-target tracking based on ultra-fast active vision still faces two problems. Existing methods lack low-latency and robust tracking algorithms in complex scenes or fast-moving targets. At the same time, there is still a gap in the cross-camera multi-target stereo correspondence for ultra-high-speed active vision.

As shown in Figure 1.1, our research goal is to achieve high-definition tracking and depth information acquisition of multiple targets in a wide range. Therefore, we decompose the whole process into two goals, 1) develop a fast and accurate multi-object tracking algorithm based on ultra-fast active vision, which can handle challenges such as object occlusion and scale changes in complex scenes. 2) develop a fast stereo correspondence algorithm based on ultra-high-speed active stereo vision, which can cope with the challenges of target loss, occlusion and illumination changes in complex scenes. Among them, in order to obtain more accurate depth information, a flexible calibration algorithm is developed based on the ultra-fast galvanometer camera. This method is suitable for stereo active camera systems, which can accurately obtain the spatial positions of multiple objects being tracked.

This research has the potential to advance the field of multi-object stereo tracking and improve existing applications. For example, it helps improve object recognition and tracking performance in autonomous driving systems, improving traffic safety and driving experience. It also helps the robot to obtain more detailed and accurate map information in autonomous navigation, improving mapping ability and navigation performance.

1.3 Outline of thesis

This thesis is organized as 7 Chapters, including this introduction.

Chapter 2 summarizes related work related to multi-object tracking, stereo correspondence for multi-object tracking, and high-speed vision.

Chapter 3 explains in detail the concept of each component of the multi-object tracking system based on ultra-fast active vision.

In the Chapter 4, based on the ultra-fast galvanometer camera, a multi-target fast tracking system based on time-division multiplexing is developed. In order to verify the effectiveness of our multi-target tracking, we set multiple moving or stationary targets

such as cars, faces, footballs, etc. Experiments have proved that we can track up to 20 slower moving objects or several faster moving objects at the same time.

In the Chapter 5, a flexible calibration method for high-precision calibration of a galvanometer-based reflective camera system is proposed. This method can be used for the calibration of stereo active vision systems, using triangulation to obtain the spatial position of multiple objects. The effectiveness and accuracy of the method are evaluated by the re-projection error of the control voltage and the spatial localization of the binocular system. Experiments show that the error of the control voltage after calibration is less than 0.2%. At an indoor distance of about 7 m, the Mean Squared Error (MSE) of spatial visual localization is less than 0.3 cm.

In the Chapter 6, the concept of using highly synchronized motion information instead of appearance information for stereo correspondence of multiple moving objects was proposed. To validate our approach, we conducted stereo correspondence experiments using markers attached to a metronome and natural hand movements to simulate simple and complex motion scenes. Furthermore, we use the motion information in the stereo vision system to carry out the correspondence of multiple objects. The experimental results demonstrate that our method achieved good performance in stereo correspondence.

In Chapter 7, it summarized the contributions of this study and discussed future work.

Chapter 2

Related works

2.1 Multiple object tracking

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 [6]. Research on multi-object tracking heavily relies on the study of object detection methods.

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 [7]. 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 [8]. 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 [9].

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 develop-

ment, convolutional neural network (CNN)-based object detection methods have evolved into a series of high-performance structural models such as AlexNet [10], VGG [11], GoogLeNet [12], ResNet [13], ResNeXt [14], CSPNet [15], and EfficientNet [16]. 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 [17]. 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 [18], SPP-Net [19], Fast R-CNN [20], Faster R-CNN [21], FPN [22], R-FCN [23], and DetectoRS [24]. 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 [25]. Among the one-stage object detection algorithms, the most famous are single shot multibox detector (SSD) [26], YOLO [27], RetinaNet [28], CenterNet [29], and Transformer [30]-based detectors [31]. YOLO was the earliest one-stage target detection algorithm applied to actual scenes, obtaining stable and high-speed detection results [32]. 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 [33], YOLOv3 [34], YOLOv4 [35], YOLOv5 [36], and YOLOv6 [37] have been proposed. Owing to its high stability and detection speed, in this study we use yolov4 as the AI detector.

Early classical object tracking methods, such as Meanshift [38], particle filtering [39], KCF [40], and MOSSE [41], 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 [42] and DeepSORT [43]. 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 [44], MOTDT [45], Tracktor++ [46], and FFT [47]. The attention mechanism-based MOT is inspired by the powerful processing ability of the Transformer model in natural language processing. Representative algorithms include TransTrack [48] and TrackFormer [49]. 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.

2.2 Stereo correspondence for multi-object tracking

Stereo correspondence of multiple moving objects with similar appearances in a stereoscopic video is closely related to research on image similarity measurement and trajectory similarity measurement.

The computation of image matching serves as the initial step in stereo correspondence, relying primarily on the similarity of target pixel blocks surrounding the stereo images. Over time, the measurement of similarity between image blocks has evolved from region-based approaches to feature-based approaches, and finally to deep learning techniques.

Region-based matching methods can be classified into two categories. The first approach minimizes differences in pixel information by using methods such as cross-correlation [50], mean square error (MSE) [51], and mutual information [52]. The second approach transforms images from the time domain to the frequency domain and performs similarity analysis in the transformed domain using techniques such as Fourier transform [53], Walsh transform [54], and wavelet transform [55]. However, region-based image matching methods require high-quality images because noise, lighting, and changes in shape can greatly affect the quality of the match. Feature-based methods can significantly reduce the impact of image quality on similarity and have been extensively researched to date [56]. These features are often manually designed, such as SURF [57], ORB [58], and LBP [59]. Feature-based methods require additional computational power to find matching points with similar features between image blocks. The Structural Similarity Index (SSIM) combines brightness, contrast, and structure to achieve matching results similar to human visual perception and has been widely used for comparing image similarity [60].

Recently, convolutional neural networks (CNNs) have replicated the huge success in image recognition and have become a research hotspot in image region matching. Based on CNNs, image matching can be mainly divided into two research directions: (1) using deep networks such as ResNet [13] and VGG [61] to extract image features and

then using similarity metrics such as Euclidean distance and cosine distance to measure the similarity of high-dimensional features; (2) using metric learning to directly output the similarity of two image blocks. In Ref. [62], the ResNet model was used to extract periocular features from different spectral bands, and cosine similarity was used for image verification, achieving high accuracy. In Ref. [63], the VGGNet was used to extract multi-scale features from segmented patches and achieved detection of forged images. Compared to manually extracted features, features extracted by CNNs are more effective in handling noise and morphological changes. In Ref. [64], MatchNet was proposed, which uses CNN for region feature extraction and then computes similarity through a three-layer fully connected network. The DeepCompare method was proposed in Ref. [65], which improved the performance of the Siamese network using the Center-Surround Two-Stream Network and Spatial Pyramid Pooling (SPP) [66]. DeepCD based on the Triplet network was proposed in Ref. [67]. This method describes image patches as complementary descriptors and improves the performance in various applications. Currently, methods based on deep learning are difficult to output calculation results in extreme time and are not suitable for high-speed vision systems. However, the matching performance they provide is unmatched by traditional algorithms.

When objects are well tracked under good conditions of a single camera, their motion information is less affected by lighting, shape changes, and noise. Motion-based matching has been widely used in cross-camera multi-object matching, such as in smart traffic [68], user behavior analysis [69], and motion pose estimation [70].

There are various ways to represent motion information, such as trajectories, angles, and velocities. Trajectories, as an easily obtainable form of motion information, have been widely used in multi-object tracking. Trajectories can be classified into two types: sequence-only trajectories and spatiotemporal trajectories, depending on whether the temporal property is considered [71].

Different methods have been developed for measuring the similarity between dif-

ferent target trajectories, which are mainly divided into three directions: distance-based, feature-based, and deep learning-based trajectory similarity calculation methods. Distance-based trajectory similarity calculation methods mainly measure the similarity between trajectories by calculating the distance between trajectory points. Some classic methods include Dynamic Time Warping (DTW) [72], Edit Distance on Real sequence (EDR) [73], and Longest Common Subsequence (LCSS) [74]. For instance, LCSS is used to calculate the similarity of the 3D GPS trajectories of the trucks in Ref. [75] to identify the movement patterns of the trucks. In Ref. [76], a trajectory evaluation method based on Dynamic Time Warping was proposed to evaluate the discrepancy between robot trajectories and human motion. However, these methods have limitations in dealing with data noise and missing values.

Feature-based trajectory similarity calculation methods extract features from trajectories and then calculate the similarity between features to measure the similarity between trajectories. Some classic methods include Shape Context [77], Histogram of Oriented Gradients (HOG) [78], and Global Alignment Kernel (GAK) [79]. For example, a skeleton-based action recognition method is proposed in Ref. [80], which combines trajectory images and visual features to simulate human actions. Based on the Fréchet distance, a shape-based local spatial association metric is proposed in Ref. [81] for detecting anomalous activities of moving ships. However, these methods are more complex in feature extraction and computation, and require a larger amount of computation.

Deep learning-based trajectory similarity calculation methods use machine learning to model and learn trajectory data, and then calculate the similarity between trajectories. Some classic methods include neural network-based methods, decision tree-based methods [82], and support vector machine-based methods [83]. For instance, an RNN-based Seq2Seq autoencoder model is proposed in Ref. [84], which improves the calculation of similarity. In Ref. [85], an attention-based robust autoencoder model is proposed, which learns low-dimensional representations of noisy ship trajectories. An unsupervised learn-

ing method is proposed in Ref. [86], which can automatically extract low-dimensional data features through convolutional autoencoders. The similarity between trajectories can be obtained from the similarity between low-dimensional data, which ensures high-quality trajectory clustering performance. However, these methods require a large amount of training data and computation resources, but they offer higher accuracy and robustness in trajectory similarity calculation.

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 [87], security monitoring [88], and industrial automation [89].

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 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 [90], 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 pro-

vides high performance and low latency, making it suitable for real-time applications that require high-speed image processing. In [91], 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 [92].

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 [93]. 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 [94], 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 [95]. 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.

Chapter 3

Concept

3.1 Multi-object tracking based on ultra-fast active vision

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 3.1. 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 4.1, 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

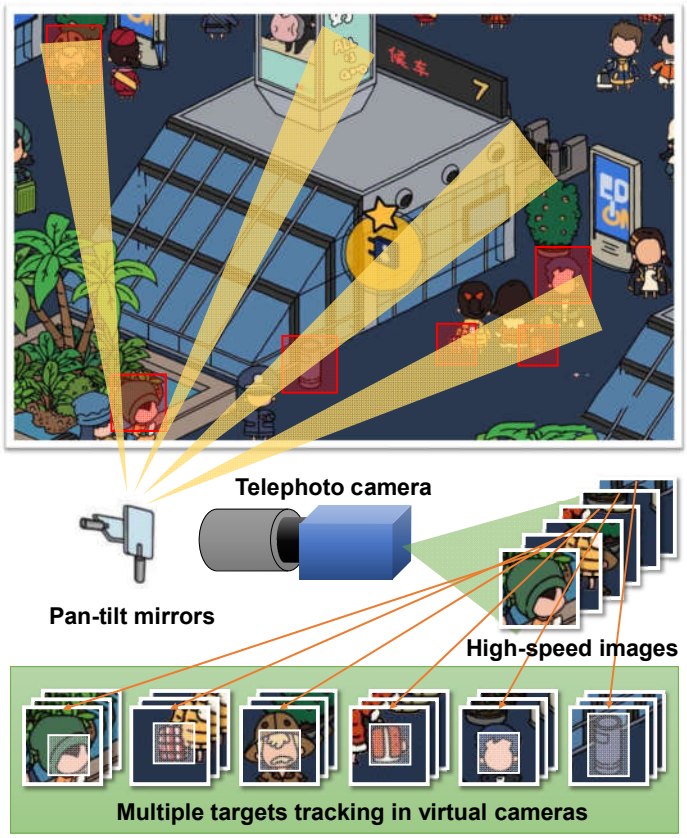


Figure 3.1: Wide field of view registration and multi-object tracking by virtual cameras using a galvanometer-based reflective PTZ camera.

such as robotics, surveillance, and autonomous navigation.

As shown in Figure 3.2, 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 ultra-fast 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.

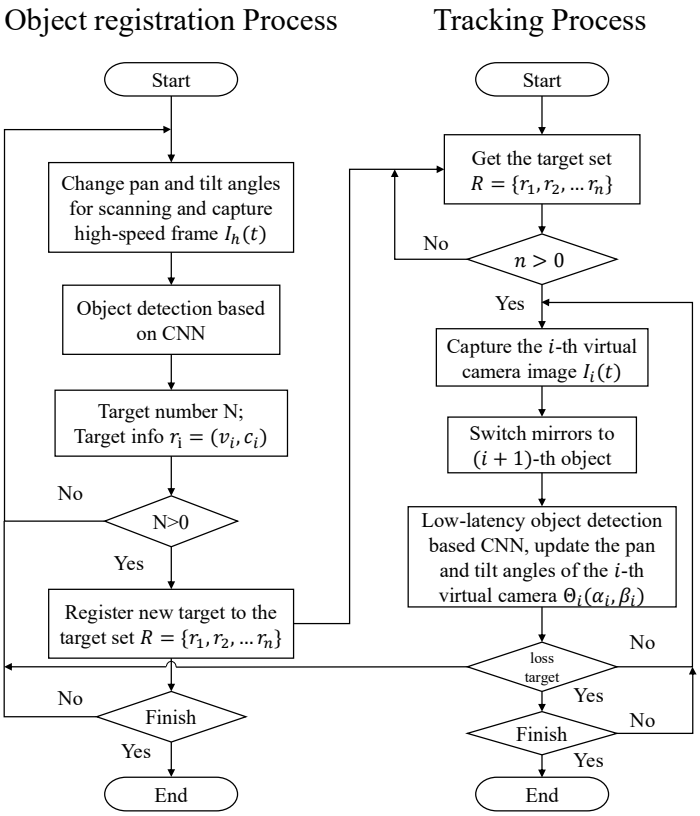


Figure 3.2: Flowchart of object registration process and multi-object tracking process.

3.2 Spatial localization based on active stereo vision

The galvanometer is an optical component capable of tiny vibrations under precise control. Its high-precision control, fast response and precise position detection capabilities make it one of the key components to achieve high-precision measurement. As shown in Figure 3.3, we propose the concept of spatial localization based on a stereo active vision system. The stereo active vision system responds extremely quickly, rotating the galvanometer with an ultra-short delay, so that the object is always in the center of the camera's field of view.

In the context of three-dimensional reconstruction based on stereo cameras, spatial points are mapped to pixels in the images through projection transformations. Similarly, in stereo active vision systems, spatial points are mapped to control voltages of the mirrors

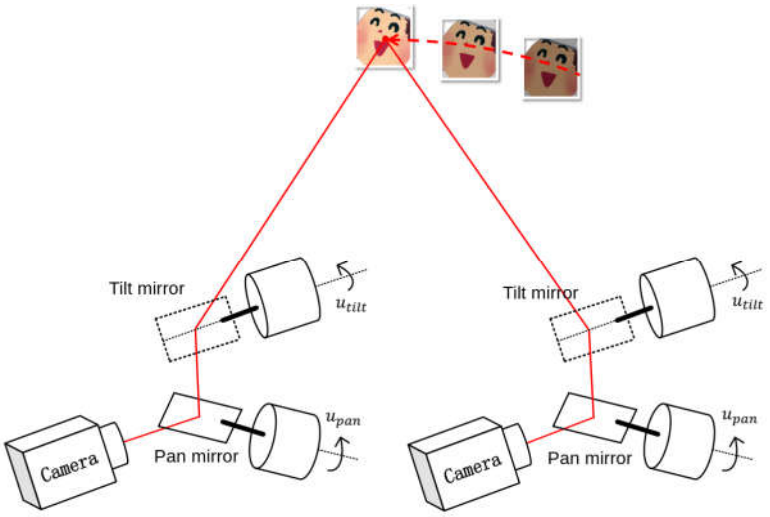


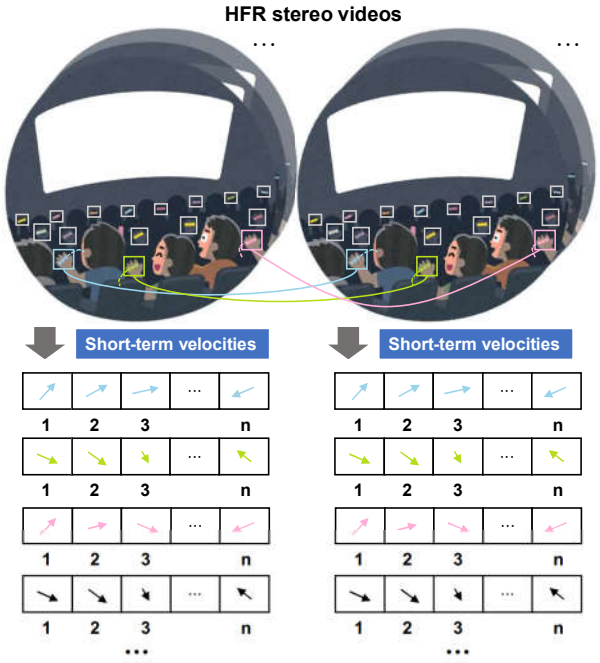
Figure 3.3: Concept of spatial localization based on active stereo vision.

through projection transformations. Each set of control voltages (of pan and tilt mirror) uniquely determines a spatial line. Leveraging the low latency of high-speed vision systems, it becomes effortless to acquire the position information of an object when it is simultaneously observed by the stereo active cameras.

In Figure 3.3, the red line is the line of sight of the camera center in the stereo active vision system. By determining the precise positions of two sightlines in space, we can accurately ascertain the location of an object in three-dimensional space.

3.3 Stereo correspondence based on high-synchronous motion information

Active camera systems have gained significant attention in computer vision due to their ability to actively control camera viewpoint and illumination, providing rich visual information for various applications. As mentioned in previous chapter, matching multiple moving objects with similar appearances in stereoscopic video is a significant challenge. To address this issue, we propose a method for correspondence based on motion information suitable for stereo vision systems (fixed cameras or active cameras), as



(a) Motion features composed of short-term velocities.

(b) Correspondence based on short-term velocities.

Figure 3.4: Concept of stereo correspondence based on high synchronous short-term velocities.

depicted in Figure 3.4. In fixed camera, we use the change of the object on the pixel as the motion, and as for the active camera, we use the angle change of the active camera, control voltage, etc. as the motion. The entire process of stereo correspondence for multiple objects is divided into two steps: independent multiple-object tracking and stereo correspondence based on high synchronous short-term velocities.

In the independent multiple-object tracking step, firstly, we need to complete high-speed real-time detection of multiple targets. Higher video frame rates provide greater synchronization. Then, we define the pixel-scale movement of an object between HFR

frames as its velocity, which comprises horizontal and vertical components. As shown in Figure 3.4(a), we utilize n velocities over a period of time before the current time as the motion feature of the objects, referred to as short-term velocities. In this way, we transform the similarity measure between object image blocks into the similarity measure between image block motions. In the object stereo correspondence step, we analyze the similarity between the high synchronous short-term velocities of multiple objects frame-by-frame to establish correspondences among different objects, as illustrated in Figure 3.4(b). We compare the similarity between multiple vectors in real time to distinguish different moving objects.

Chapter 4

An active multi-object ultrafast tracking system with CNN-based hybrid object detection

4.1 Introduction

Multi-target tracking and high-definition image acquisition are important issues in the field of computer vision [96]. 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 [97], security monitoring [98], intelligent transportation systems [99], robot navigation [100], auto pilot [101], and video surveillance [102].

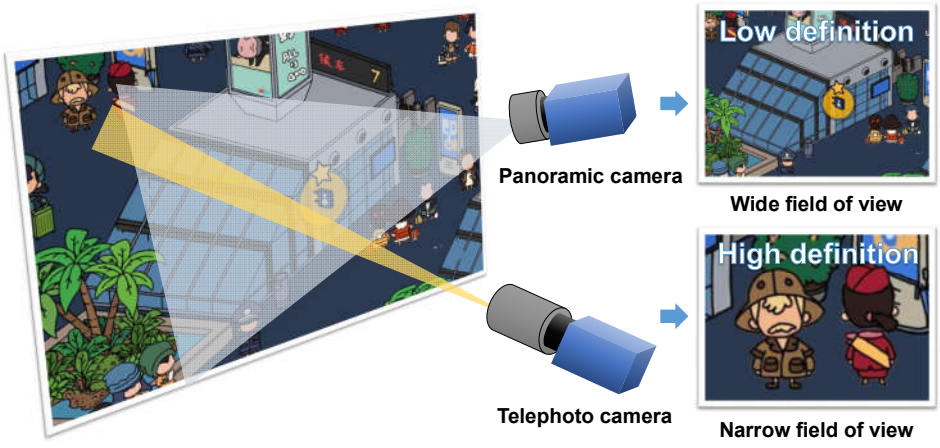


Figure 4.1: Contradiction between wide field of view and high-definition images.

However, there is a contradiction between wide field of view and high-definition resolution, as shown in Figure 4.1. 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 [103]. 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 [104]. 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 [105].

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 [106]. 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 [107]. 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 [108]. Compared to traditional camera systems operating at 30 or 60 fps, high-speed vision systems can work at 1000 fps or more [109]. The high-speed vision system acquires and processes image information with extremely low latency and interacts with the environment through visual feedback control [110]. 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 [111]. This reflective PTZ camera system is able to virtualize multiple virtual cameras in a time-division multiplexing manner in order to observe multiple objects [112]. Compared with traditional gimbal-based and panoramic cameras, galvanometer-based reflective PTZ cameras have the advantages of low cost, high speed, and high stability [113], and are suitable for multi-target tracking and high-definition capture.

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 [114]. 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 [115]. 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 chapter 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 [95]. Compared with the previous system, this system 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.

4.2 Proposed galvo-based multi-target tracking system

4.2.1 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 $v(t) = \left\{u_{pan}(t), u_{tilt}(t)\right\}$, 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:

$$D\left(I_h(t)\right) = \left\{d^1\left(t\right), d^2\left(t\right), \ldots, d^j(t), \ldots, d^J\left(t\right)\right\}, \tag{4.1}$$

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

$$d^j\left(t\right) = \left\{o_x^j\left(t\right), o_y^j\left(t\right), w^j\left(t\right), h^j\left(t\right), p^j\left(t\right), c^j\left(t\right)\right\}, \tag{4.2}$$

where $o^j\left(t\right) = \left(o_x^j\left(t\right), o_y^j\left(t\right), w^j\left(t\right), h^j\left(t\right)\right)$ denotes the bounding box of j-th detection object. In addition, $p^j\left(t\right)$ and $c^j\left(t\right)$ 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:

$$v_i = \left\{u_{pan}^i(t) + \varepsilon_{pan} d_{pan}, u_{tilt}^i(t) + \varepsilon_{tilt} d_{tilt}\right\}, \tag{4.3}$$

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.

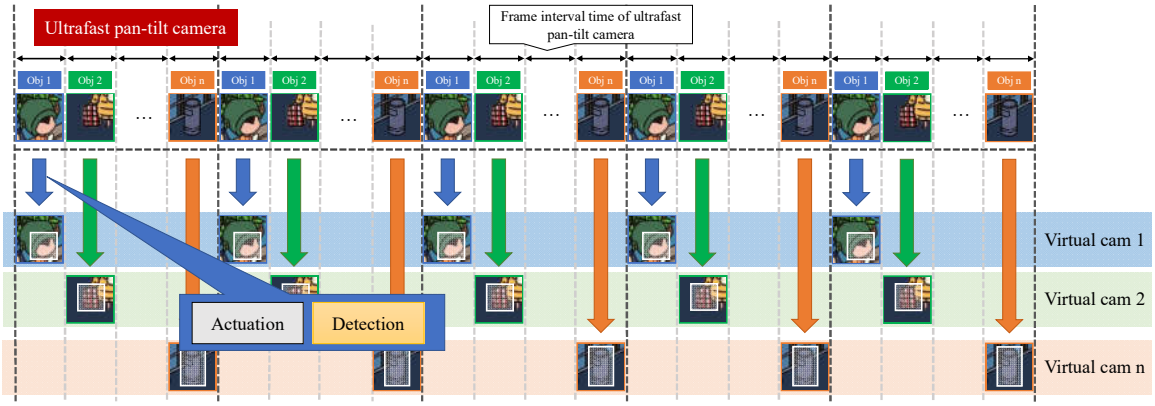
Then, the detected object info $r_i = \{v_i, c_i\}$ is registered to the target set $R = \{r_1, r_2, \cdots, r_n\}$ for high-speed multi-object tracking, while c_i denotes the label of the i-th object.

4.2.2 Multi-object tracking process

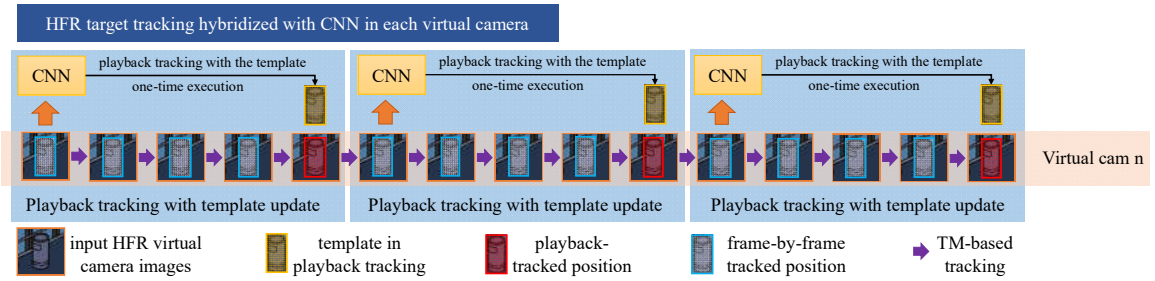
The multi-object tracking process is initiated when the target set size R is greater than 0. In the image acquisition process of the galvanometer-based reflective PTZ camera, as depicted in Figure 4.2(a), the galvano-mirror movement and camera exposure represent the two primary stages. To cope with high-speed 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 4.2(b): (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



(a) Time-division threaded gaze control process for simultaneous multi-object observation.



(b) HFR target tracking hybridized with CNN in each virtual camera.

Figure 4.2: Time-division threaded gaze control process for multiple target tracking based on HFR object detection hybridized with CNN.

frame during playback tracking is used to determine the template of the TM-based tracker and the initial position of frame-by-frame tracking.

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 nth virtual camera. The objects detected in an input image $I_n(t_d)$ from the nth virtual camera at time t_d can be described as follows:

$$D\left(I_n\left(t_d\right)\right) = \left\{d_n^1(t_d), d_n^2(t_d), \ldots, d_n^J(t_d)\right\}. \tag{4.4}$$

The definition of the detection results $d_n^j(t_d)$ is the same as that in Eq. (4.2). Results that differ from the target labels c_n tracked by the nth virtual camera are initially eliminated. To update the template I_n of the nth virtual camera for the S detection results for which the class is the same as c_n, we use the following method:

$$T_n = \begin{cases} \underset{T_s}{\text{argmax}} NCC\left(T_s, T_n'\right)(S > 0) \\ don't\ update, (S = 0), \end{cases} \tag{4.5}$$

$$NCC(T_s, T_n') = \frac{\text{Cov}(T_s, T_n')}{\sqrt{\text{Var}(T_s)\,\text{Var}(T_n')}}. \tag{4.6}$$

Here, T_n and $T_n^{'}$ are the current and last template of the nth virtual camera, respectively. The NCC [116] (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:

$$p\left(t_n\right) = p'\left(t_n\right) + \underset{|x|\leq Ran, |y|\leq Ran}{\arg\min} E\left(x, y\right), \tag{4.7}$$

$$E(x, y) = \frac{\sum_{x', y'}\left(T_n\left(x', y'\right) - I_n\left(x_n' + x + x', y_n' + y + y'\right)\right)^2}{\sqrt{\sum_{x', y'} T_n\left(x', y'\right)^2 \cdot \sum_{x', y'} I_n\left(x_n' + x + x', y_n' + y + y'\right)^2}}. \tag{4.8}$$

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*

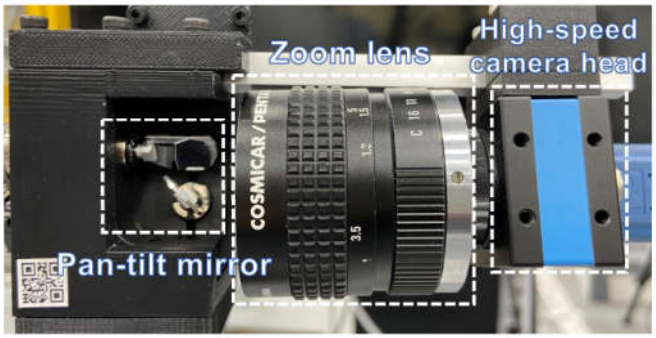
for object detection in the image to a few pixels. Here, (x'_n, y'_n) represents the top-left coordinate of the target region in the image from the previous time step in the nth virtual camera, while I_n represents the new image from the nth virtual camera. Because the algorithm is applied to a real-time 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.

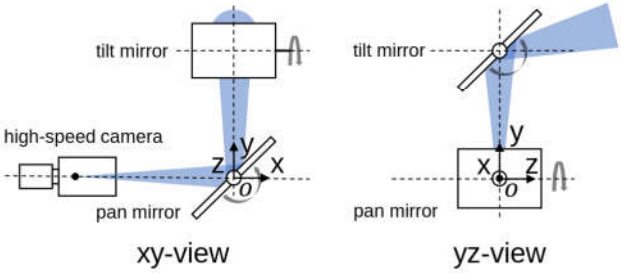
$$p\left(t'_n + (k+1)\tau_n\right) = p\left(t'_n + k\tau_n\right) + \underset{|x|\leq Ran, |y|\leq Ran}{\arg\min} E\left(x, y\right), (0 <= k\tau_n <= t_d). \tag{4.9}$$

Here, τ_n represents the time interval between frames for the nth virtual camera. As shown in Figure 4.2b, 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.

4.3 Experiments



(a) Overview of the galva-based camera system.



(b) Geometry of the pan-tilt camera system.

Figure 4.3: Overview and geometry of galva-based multi-object tracking system.

4.3.1 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 4.3. 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.

4.3.2 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 obtained by the new object registration process.

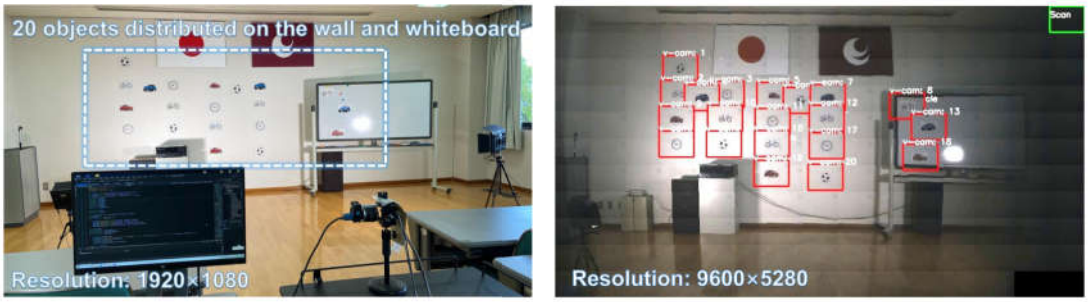
Using YOLOv4 for CNN detection, we can execute object detection with a delay of about $\tau_d = 33$ ($m = 16$) ms using multiple high-speed PTZ virtual cameras. In the implemented YOLOv4, eighty object categories (car, bicycle, sports ball, apple, mouse, etc.) were pre-trained using the COCO dataset, 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

Table 4.1: 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)

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 4.1 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.



(a) Overview of the experimental scene.

(b) Panoramic stitched image from the PTZ camera (targets are pasted on the panoramic image in the form of a red frame texture).

Figure 4.4: The 1920 × 1080 input images from the digital camera and the panoramic stitched 9600 × 5280 images from the PTZ camera.

4.3.3 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 4.4(a) 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 4.4(b) 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 4.5 presents high-definition images of all tracked objects continuously updated within the image frames at 30 fps. Figure 4.6 shows the pan and tilt angles of the

Figure 4.5: HD images of twenty objects tracked simultaneously.

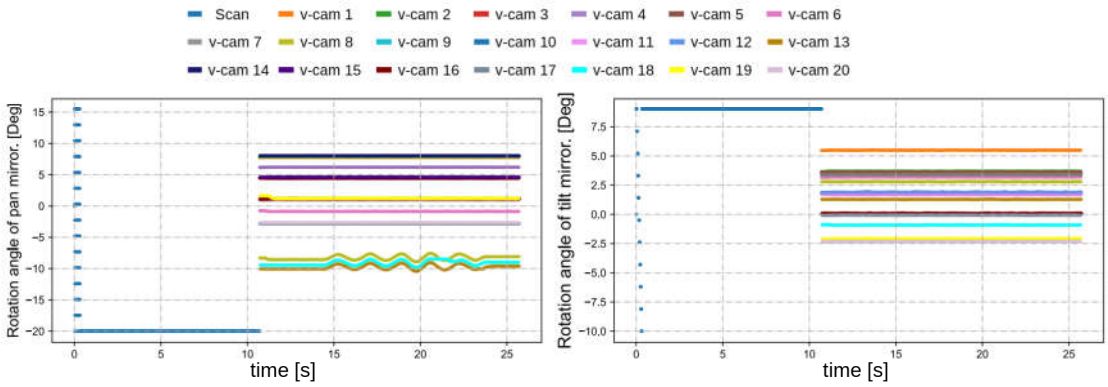


Figure 4.6: Pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking twenty different targets.

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.

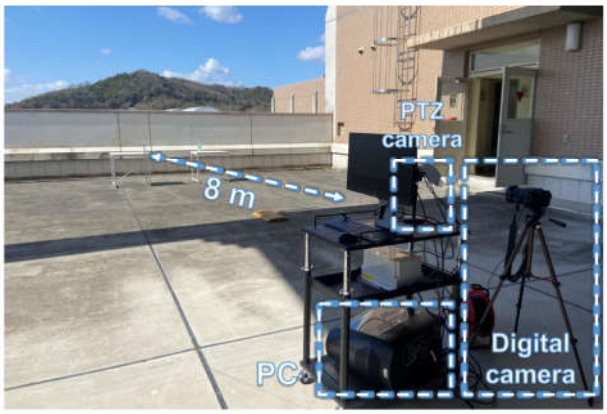


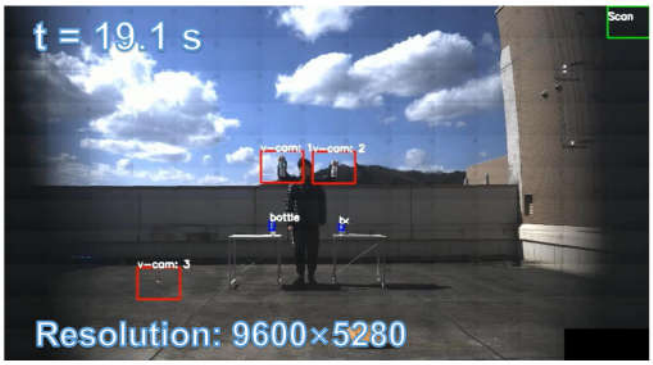
Figure 4.7: Experimental environment used for tracking multiple moving objects in an outdoor scene.

4.3.4 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 4.7. 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 4.8 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 surveil-

(a) Input image of digital camera.



(b) Panoramic stitched image from the PTZ camera.

Figure 4.8: The 1920 × 1080 input images from the digital camera and panoramic stitched 9600 × 5280 images from the PTZ camera.

lance 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 4.9 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 4.10 shows the pan and tilt angles of the galvanometer-based reflective PTZ

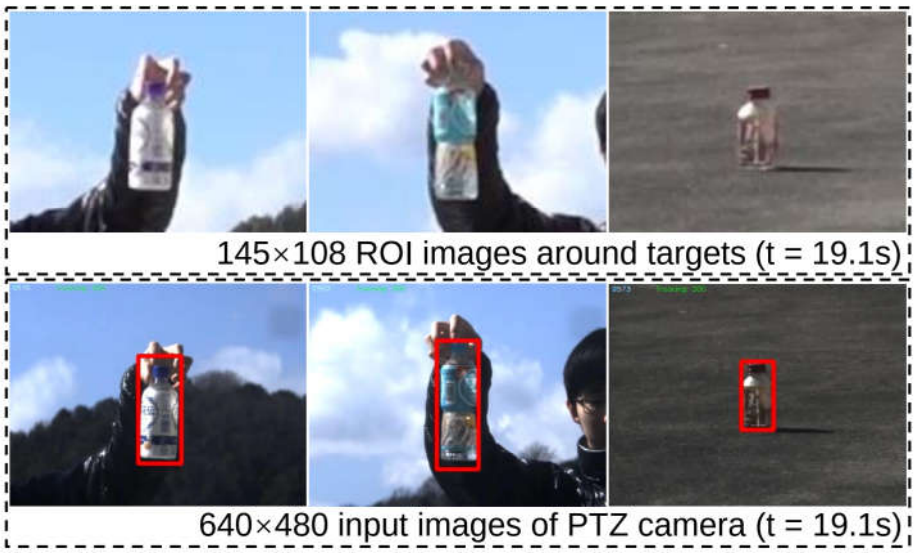


Figure 4.9: The 145 × 108 ROI images around targets from the digital wide-view camera and 640 × 480 input images from the virtual PTZ cameras (red boxs are the test results).

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 4.11 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 4.12 depicts the tracking status during free fall of bottle 1 when tracking three bottles simultaneously. Figure 4.12a,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 4.13 shows the relationship between the velocity and distance

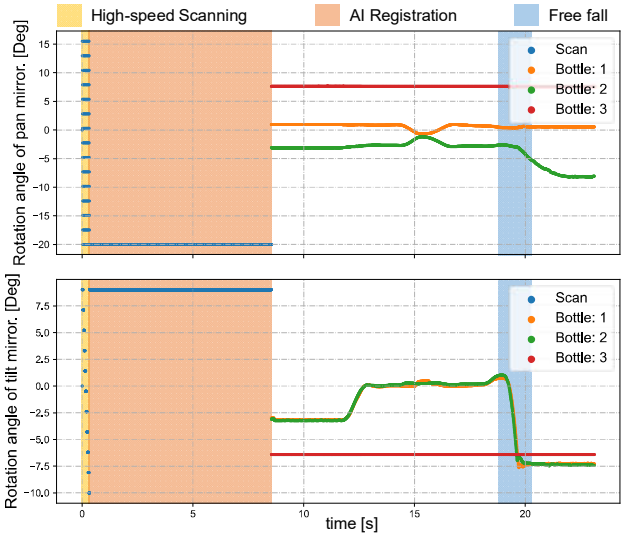


Figure 4.10: Pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking multiple bottles.

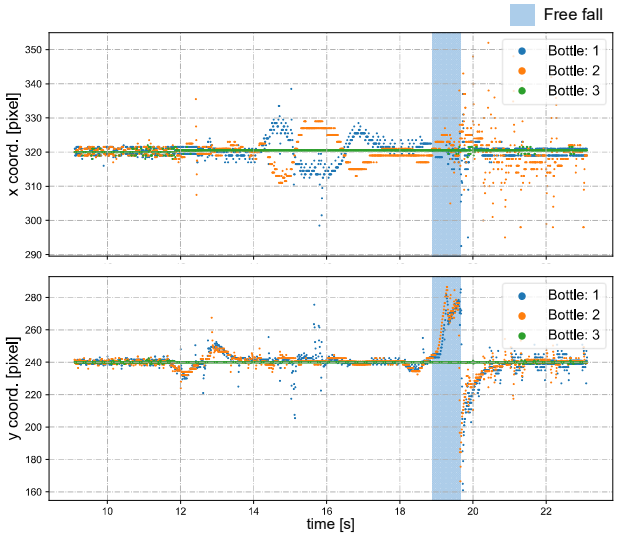
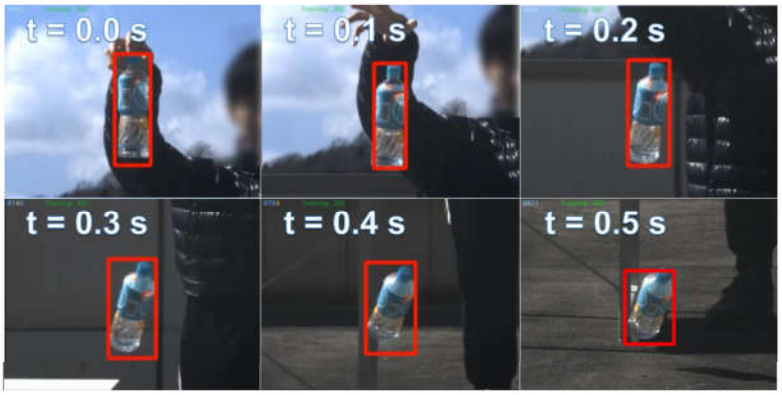
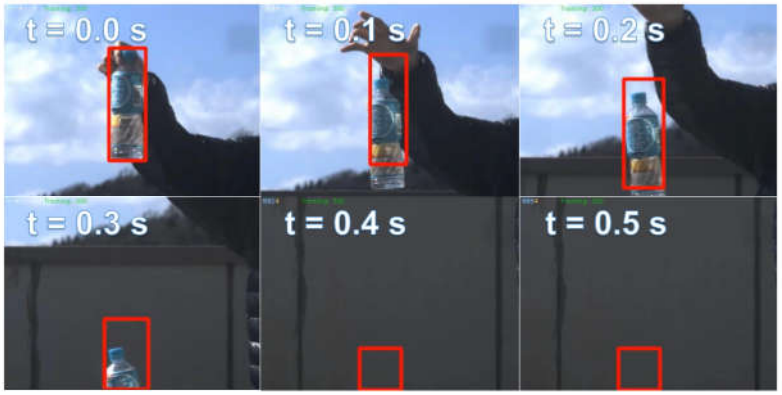


Figure 4.11: The x and y centroids of tracked bottle regions.

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



(a) Free-fall of bottle 1 based on CNN-based hybrid tracking.



(b) Free-fall of bottle 1 based on YOLOv4 tracking.

Figure 4.12: Tracking status of the free-fall of bottle 1 when tracking three bottles simultaneously.

8 m simultaneously. Finally, we conducted experiments to track three free-falling bottles simultaneously using different tracking methods.

Figure 4.14 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

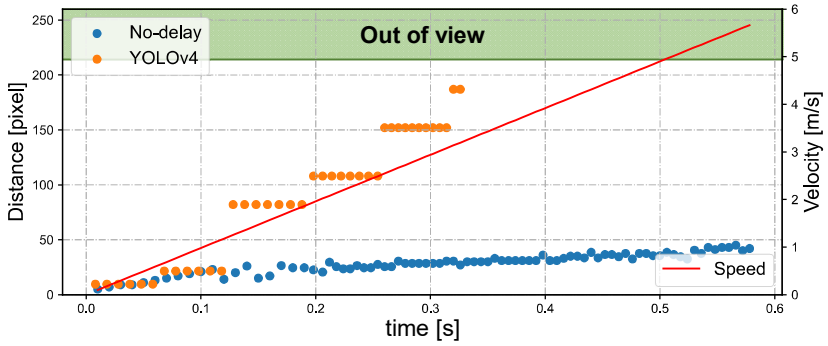


Figure 4.13: Relationship between velocity and distance from the detection ROI to the image center during free-fall bottle tracking.

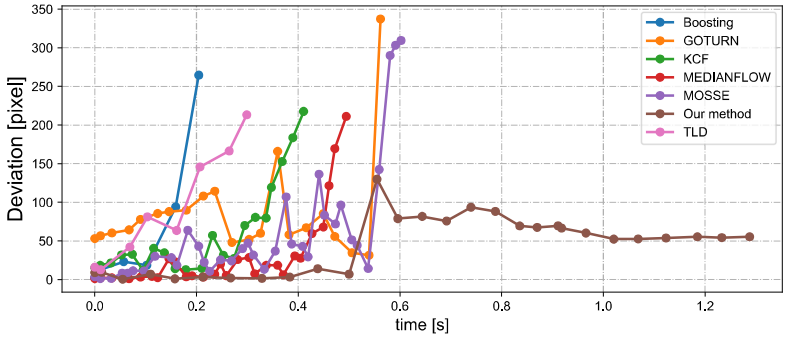


Figure 4.14: Pixel deviation value between the object position calculated by different algorithms and the object's real position during free-falling.

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.

4.3.5 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 4.15(a) 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 4.15(b) depicts the scanning and detection outcome at the start of the experiment.

(a) Input image of the digital camera.

(b) Panoramic stitched image from the PTZ camera.

Figure 4.15: The 1920 × 1080 input images from the digital camera and panoramic stitched 9600 × 5280 images from the PTZ camera.

Figure 4.16 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 [117] 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

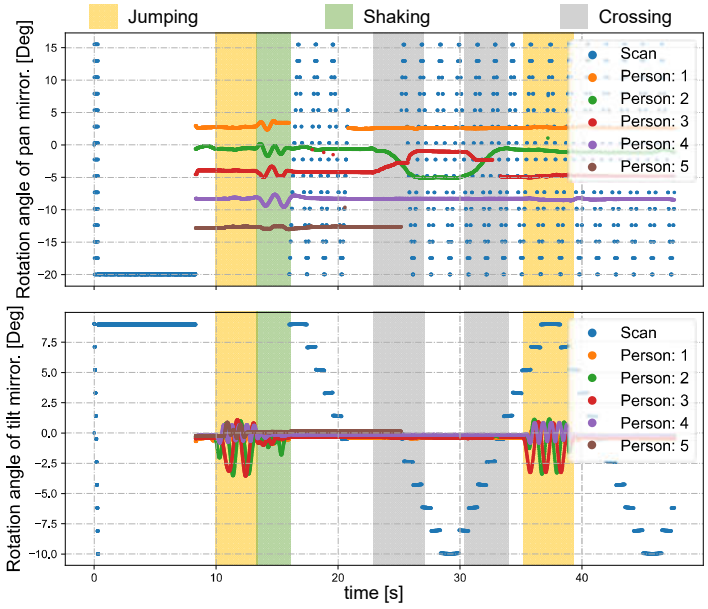


Figure 4.16: Pan and tilt angles of the galvanometer-based reflective PTZ camera when scanning and tracking multiple persons.

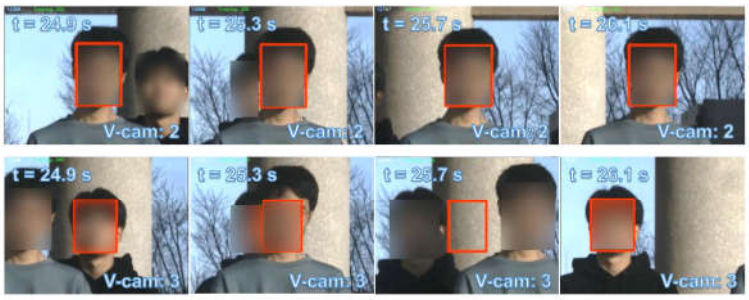


Figure 4.17: Cross-tracking of person 2 and person 3 between 24.9 and 26.1 s.

camera always followed a single person.

As shown in Figure 4.17, 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 4.18. 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

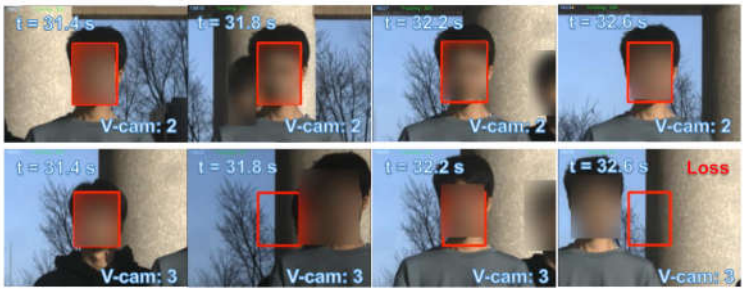


Figure 4.18: Cross-tracking of person 2 and person 3 between 31.4 and 32.6 s.

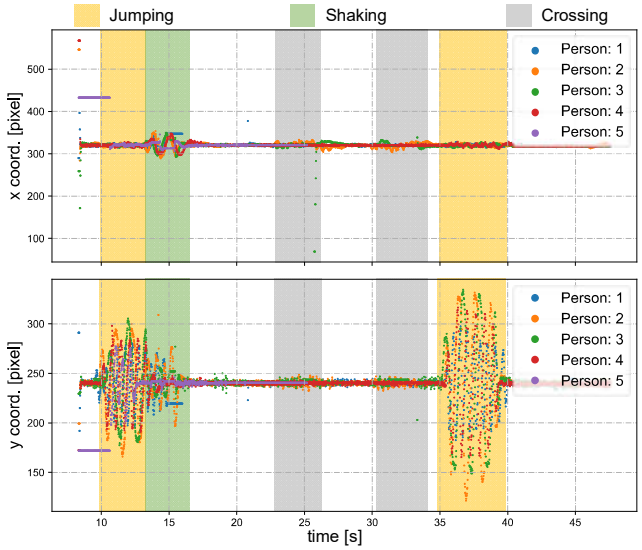


Figure 4.19: The x and y centroids of the regions with tracked people.

in Figure 4.19. 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.

4.4 Concluding remarks

In this chapter, 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 indi-

viduals. 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.

Chapter 5

Spatial localization based on stereo active vision

5.1 Introduction

Stereo active vision system serves as a crucial component of intelligent robots, mimicking the ability of humans or other animals to perceive the environment through their eyes. The servo-based active vision system enables multiple cameras to simultaneously focus on the same visual target and utilizes the control information from the active vision system to determine the spatial position of the target relative to the intelligent robot. Intelligent robots equipped with stereo active vision system have enormous potential in perception-based applications such as intelligent driving, digital twin, visual tracking, and visual localization [118].

In stereoscopic active vision, the system employs multiple cameras or sensors to emulate binocular vision. The eyeball of an animal is typically a spherical shape with a center of rotation that is not fixed, resulting in an extremely complex movement pattern. In contrast, the human-like binocular structure developed by the MIT Artificial Intelligence Laboratory in 1998 has two degrees of freedom - horizontal and vertical - and is equipped with two CCD cameras, each with high and low resolution. While this pan-tilt-based two-axis mechanical system can produce clear images, it is difficult to capture images accurately. As a result, the development of three-degree-of-freedom visual devices has become a research hotspot in recent years. Wang X.y et al. proposed a novel

humanoid robot eye that rotates at 3-DOF using six pneumatic artificial muscles [119]. However, this robot eye requires compressed air. Gosselin and his colleagues developed an agile eye using six sets of links. Y. B. Bang et al. proposed a 3-DOF human-like eye movement system that reproduces realistic human eye movements for human-sized human-like applications [120]. Most of the aforementioned active vision systems utilize gimbal structures with motion motors, which can result in motion blur and slow motion speeds due to camera movements.

Galvanometers are optical scanners with high precision, reliability, and speed. The mirror-based reflective active vision system separates the camera from complex mechanical structures, enabling high-quality imaging and fast motion speeds. This approach provides a new way of studying active vision. Early galvanometer-based active vision was generally used in object detection and visual tracking. Jiang et al. [114] obtained images using galvanometer reflection, detected object positions at 500 fps, and controlled the galvanometer to achieve high-speed object tracking. Hu et al. [94] combined a panoramic camera and a galvanometer camera. The panoramic camera used deep learning to detect targets, and the galvanometer camera quickly switched between multiple targets at 500 fps. In recent years, some researchers have engaged in stereo vision based on galvanometer cameras. Hu et al. [121] proposed a novel catadioptric stereo tracking concept. The galvanometer camera was virtualized into two tracking cameras with different viewing angles through multiple mirrors, and 3D measurements were completed during the tracking process. However, precision is currently not a primary focus when using galvanometers, and spatial position calculations are completed solely based on angle relationships.

Before a galvanometer-based active vision system can perform precise measurement or localization tasks, it requires calibration. However, there are currently few methods available for calibrating such a system. As galvanometers are primarily used in the field of lasers, most research focuses on calibrating laser galvanometers. Manakov [122] proposed a calibration method for a dual-mirror galvanoscopic laser scanner, but this

method is challenging to optimize the mathematical model of the system. Wissel [123] suggested a data-driven learning calibration method that requires large amounts of data collection. Wagner [124] uses statistical learning methods such as artificial neural networks (ANN) and linear regression to calibrate the system. However, this method is prone to overfitting problems, and the computational cost is often high. Yu [125] designed a novel single-mirror galvanoscopic laser scanner. However, the calibration process is complicated, and the objective function has 11 independent unknown parameters that need optimization. Due to the relationship between the two reflection imaging, the galvanometer-type active vision system has higher complexity than the traditional gimbal-based active vision system.

Inspired by Dr. Zhang Zhengyou [126] in camera calibration algorithm, it is very important to develop a flexible, robust and low-cost galvanometer calibration algorithm in the galvanometer system. In this chapter, a calibration method of the reflective galvanometer-based active vision system is proposed. In this method, the galvanometer is virtualized as a camera model, and the three-dimensional spatial points are projected to the control voltage parameter space of the galvanometer. The calibration only needs to be observed by the galvanometer camera at several plane patterns of different angles, and the calibration result has high accuracy.

5.2 Error analysis of galvanometer-based camera

In this section, aiming at the accuracy of galvanometer-based active camera scanning, we analyze the main sources of the deviation between the voltage signal and the actual voltage during the 3D galvanometer scanning process. Fig. 5.1 shows the structure of galvanometer-based active camera, in which the rotation axes of the pan mirror and the tilt mirror are orthogonal to each other, and they are controlled by servo motors respectively. The camera is arranged parallel to the tilt axis, so that the optical axis of

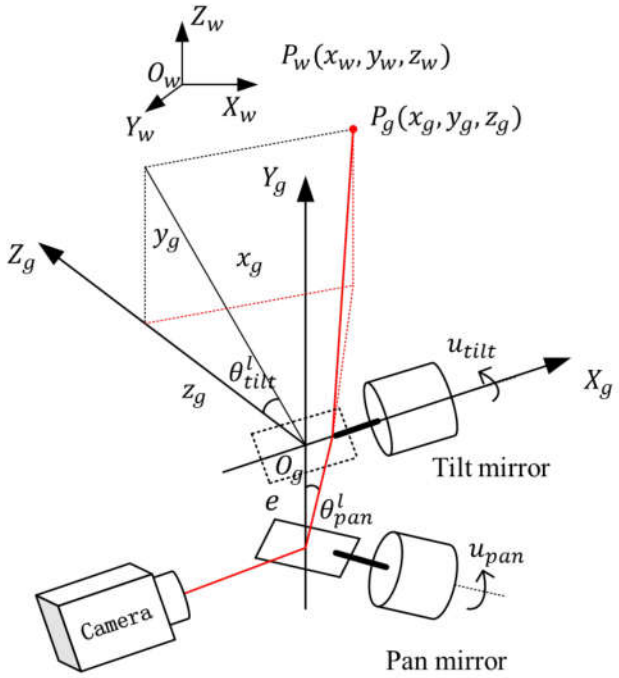


Figure 5.1: Structure of active camera based on galvanometer.

the camera center and the rotation axis of the tilt mirror are parallel to each other, and the red line is the optical path of the camera optical center. When the galvanometer is in the initial state, the left-handed coordinate system $O_gX_gY_gZ_g$ is established. The intersection of the optical axis and the rotation axis of the tilt mirror is the origin O_g, the rotation axis of the tilt mirror is the X_g axis, and the rotation axis of the pan mirror is the Z_g axis. The common errors in the galvanometer model mainly include: voltage error, pincushion error, and nonlinear error.

5.2.1 Voltage error

During the movement of the galvanometer, as the driving voltage of the servo motor changes, the rotation angle of the flat mirror mounted on the servo motor also changes. The computer outputs a digital signal, and the D/A converter outputs a high-precision analog voltage. There is a very high-precision mapping relationship between the driving

voltage and the rotation angle. But they maintain a linear relationship, $u' = k_u u$, where k_u is a linear change coefficient and a constant.

For general scenarios, when a low-precision D/A converter is used to output an analog voltage to control the galvanometer, there is a deviation between the output voltage u' and the input voltage u. At the same time, the deflection angle θ_m of the galvanometer is proportional to the control voltage u', $\theta_m = k_\theta u'$, k_θ is the linear coefficient of the control voltage and the angle, which is also a constant. The relationship with the rotation angle θ_m of the galvanometer and the input digital voltage u is,

$$\theta_m = k_\theta k_u u. \tag{5.1}$$

Let $k = k_\theta k_u$, where k is the linear coefficient of the input digital voltage u and the rotation angle of the galvanometer θ_m. In Fig 5.1, the pan mirror and the tilt mirror are driven by two servo motors respectively. The rotation angle of the pan mirror and the tilt mirror θ_p, θ_t and the control voltage u_p, u_t of the two mirrors are,

$$\begin{cases} \theta_\mathrm{p} = k_\mathrm{p} u_\mathrm{p} \\ \theta_\mathrm{t} = k_\mathrm{t} u_\mathrm{t}. \end{cases} \tag{5.2}$$

Among them, k_p and k_t are the linear coefficients between the control voltage of the pan mirror and the tilt mirror and the deflection angle, respectively.

As shown in Figure 5.2, the deflection angle ${}^l\theta_\mathrm{p}$, ${}^l\theta_\mathrm{t}$ of the light path after the reflection of the pan mirror and the tilt mirror will be twice the deflection angle of the mirror, which is

$$\begin{cases} {}^l\theta_\mathrm{p} = 2k_\mathrm{p} u_\mathrm{p} \\ {}^l\theta_\mathrm{t} = 2k_\mathrm{t} u_\mathrm{t}. \end{cases} \tag{5.3}$$

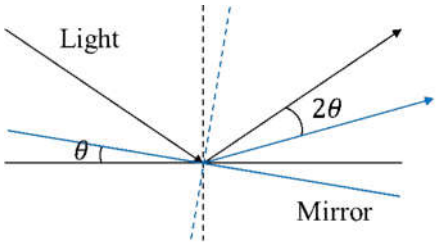


Figure 5.2: The relationship between the galvanometer angle and the deflection angle of the optical path.

5.2.2 Pincushion error

It can be seen from Figure 5.1 that the control voltages u_{p}, u_{t} of the pan and tilt deflection mirror are adjusted respectively, so that the spatial point $P_g\left(x_g, y_g, z_g\right)$ falls at the center of the camera screen, and the red broken line is the optical path from P_g to the camera. The corresponding swing angles of the pan mirror and the tilt mirror are expressed as,

$$\begin{cases} \theta_{\mathrm{t}} = \frac{1}{2}\arctan\frac{y_g}{z_g} \\ \theta_{\mathrm{p}} = \frac{1}{2}\arctan\frac{x_g}{z_g \sec 2\theta_{\mathrm{t}} + e}. \end{cases} \tag{5.4}$$

Assuming that the spatial point P_g moves in the plane, that is z_g does not change. At the same time, fix the angle of the pan mirror and only change the angle of the tilt mirror. Combining the above formula, we can know that,

$$\frac{\left(x_g - e\tan\left(2k_p u_p\right)\right)^2}{\left(z_g \tan\left(2k_p u_p\right)\right)^2} - \frac{y_g^2}{z_g^2} = 1. \tag{5.5}$$

Since z_g, k_{p}, and u_{p} are all constants, the trajectory of the camera's field of view in the $z = z_g$ plane is a hyperbola. As the control voltage of the pan mirror increases, the curvature of the hyperbola will increase, which is the pillow cause of shape distortion. The physical cause of pincushion distortion is the distance e between the two galvanometer-mirrors, which cannot be eliminated.

5.2.3 Non-linear error

In the general galvanometer scanning process, the scanning position of the galvanometer is usually determined by the deflection angle of the pan and tilt mirrors. Combine the above formula and rewrite the formula,

$$\begin{cases} y_g = z_g \tan\left(2k_{\mathrm{t}}u_{\mathrm{t}}\right) \\ x_g = \tan\left(2k_{\mathrm{p}}u_{\mathrm{p}}\right)\left(z_g \sec\left(2k_{\mathrm{t}}u_{\mathrm{t}}\right) + e\right). \end{cases} \tag{5.6}$$

It is also assumed that the spatial point P_g moves in the plane, and z_g does not change. Since $\tan\theta > \theta$, nonlinear errors will appear in the scanning process. The greater the deflection angle of the galvanometer, the greater the error. The same applies to pan mirrors, with greater nonlinearity.

5.3 Mathematical model of galvanometer-based camera

Similar to the camera, the camera establishes a mapping relationship between pixels and spatial points, and the galvanometer-based active camera also establishes a mapping relationship between the control voltages of the pan and tilt mirror and the spatial points. Therefore, referring to the calibration process of the camera, we have developed a calibration method for the active vision system based on the galvanometer. Different from the calibration process of the traditional camera model, the internal parameter matrix and the external parameter matrix of the camera model are both linear transformation processes, while the galvanometer model is mixed with the nonlinear transformation process. So we first solve the approximate solution of the galvanometer model by linear approximation, and then optimize these parameters by nonlinear optimization.

5.3.1 Linear approximation to obtain the initial value

The relationship between spatial point and galvanometer control voltage: a 2D point is denoted by $p = [u, v]^T$. A 3D point is denoted by $P = [x, y, z]^T$. We use $\widetilde{a}$ to denote the augmented vector by adding 1 as the last element: $\widetilde{p} = [u, v, 1]^T$ and $\widetilde{P} = [x, y, z, 1]^T$. Then the homogeneous coordinate $\widetilde{P_w}$ of the spatial point P in the world coordinate system is $\widetilde{P_w} = [x_w, y_w, z_w, 1]^T$. The homogeneous form $\widetilde{u}$ of the control voltage u_p, u_t of the pan and tilt mirror is $\widetilde{u} = [u_p, u_t, 1]^T$. The voltage control of the galvanometer needs to be calculated in the galvanometer coordinate system, then the spatial point P in the galvanometer coordinate system is $P_g = \left[x_g, y_g, z_g\right]^T$,

$$P_g = [R\ t]\widetilde{P_w}. \tag{5.7}$$

$[R\ t]$ is the extrinsic matrix connecting the world coordinate system and the galvanometer coordinate system. The dimension of $[R\ t]$ is 3×4, and it is composed of two vectors of rotation and translation.

It can be seen from Eq. (5.4) that in the galvanometer coordinate system, there is a complicated nonlinear mapping relationship between the spatial point P_g and the control voltage u_t and u_p of the galvanometer. Since when the value of θ is small, $\tan\theta \approx \theta$ and $\cos\theta \approx \theta$, we use θ to replace $\tan\theta$ and $\cos\theta$. And the distance e between the pan mirror and the tilt mirror is much smaller than the depth component z_g of the spatial point P_g, which can be approximately zero. So the Eq. (5.4) can be rewritten to,

$$\begin{cases} u_t = \frac{1}{2k_t}\arctan\frac{y_g}{z_g} \approx \frac{1}{2k_t}\frac{y_g}{z_g} \\ u_p = \frac{1}{2k_p}\arctan\frac{x_g}{z_g\sec 2\theta_t + e} \approx \frac{1}{2k_p}\frac{x_g}{z_g}. \end{cases} \tag{5.8}$$

We can rewrite Eq. (5.8) in matrix form,

$$s\begin{bmatrix} u_p \\ u_t \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2k_p} & 0 & 0 \\ 0 & \frac{1}{2k_t} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = A \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix}. \tag{5.9}$$

s is an arbitrary scale factor. We use A to replace the left half of the matrix, and call it the intrinsic matrix. Finally, combining Eq. (5.7) and Eq. (5.9), we can get the relationship between the spatial point P_w and the control voltage u_p and u_t of the galvanometer while the spatial point falling in the center of the camera's field of view,

$$s\begin{bmatrix} u_p \\ u_t \\ 1 \end{bmatrix} = A\,[R\ t] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \tag{5.10}$$

Homology matrix between world coordinates and control voltage: since $[R\ t]$ is composed of three rotation vectors and one translation vector, which is a 3×4 matrix. We use R_i to represent the i-th column of the rotation matrix,

$$s\begin{bmatrix} u_p \\ u_t \\ 1 \end{bmatrix} = A \begin{bmatrix} R_1 & R_2 & R_3 & t \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \tag{5.11}$$

Therefore, a spatial point P_w and the control voltage u can be connected by a homography matrix H,

$$s\tilde{u} = H\tilde{P_w}, \text{ with } H = A \begin{bmatrix} R_1 & R_2 & R_3 & t \end{bmatrix}. \tag{5.12}$$

As shown above, the 3×4 matrix H is defined as a scale factor. By collecting the control voltage and multiple sets of points in the space, we can solve the H matrix by means of SVD or least squares.

Separate Intrinsic and Extrinsic Matrix: after we get the H matrix, we use R_1, R_2, R_3 as the three columns of the rotation matrix R, there is a unit orthogonal relationship, which is

$$\begin{cases} R_1^T R_2 = R_1^T R_3 = R_2^T R_3 = 0 \\ R_1^T R_1 = R_2^T R_2 = R_3^T R_3 = 1. \end{cases} \tag{5.13}$$

Expressing R_1, R_2, R_3 with A and H, we can get,

$$\begin{cases} R_1 = A^{-1} H_1 \\ R_2 = A^{-1} H_2 \\ R_3 = A^{-1} H_3. \end{cases} \tag{5.14}$$

$H_i = [h_{1i}, h_{2i}, h_{3i}]^T$ represents the i-th column of the matrix H. Substituting Eq. (5.14) into Eq. (5.13), we can get

$$\begin{cases} H_1^T A^{-T} A^{-1} H_2 = 0 \\ H_1^T A^{-T} A^{-1} H_3 = 0 \\ H_2^T A^{-T} A^{-1} H_3 = 0 \\ H_1^T A^{-T} A^{-1} H_1 = 1 \\ H_2^T A^{-T} A^{-1} H_2 = 1 \\ H_3^T A^{-T} A^{-1} H_3 = 1. \end{cases} \tag{5.15}$$

Let $A^{-T}A^{-1} = B$, then B is,

$$B = \begin{bmatrix} 4k_p^2 & 0 & 0 \\ 0 & 4k_t^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix}. \tag{5.16}$$

So the parameter variables of B can form a vector $\mathbf{b}$,

$$\mathbf{b} = [b_{11}, b_{22}, b_{33}]^T. \tag{5.17}$$

Combining Eq. (5.16), Eq. (5.17) and writing Eq. (5.15) in a general form, we have,

$$H_i^T B H_j = v_{ij}^T \mathbf{b} = \begin{bmatrix} h_{i1}h_{j1} & h_{i2}h_{j2} & h_{i3}h_{j3} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{22} \\ b_{33} \end{bmatrix}. \tag{5.18}$$

Therefore, Eq. (5.15) can be transformed into an overdetermined equation with 6 equations and 3 unknowns, and $\mathbf{b}$ can be solved by SVD through the H matrix. Then the internal parameter matrix A can be obtained from the B matrix,

$$A = \begin{bmatrix} \frac{1}{2k_p} & 0 & 0 \\ 0 & \frac{1}{2k_t} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{b_{33}}}{\sqrt{b_{11}}} & 0 & 0 \\ 0 & \frac{\sqrt{b_{33}}}{\sqrt{b_{22}}} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.19}$$

Among them, $k_p = \frac{\sqrt{b_{11}}}{2\sqrt{b_{33}}}$, $k_t = \frac{\sqrt{b_{22}}}{2\sqrt{b_{33}}}$. It can be obtained by A matrix, and the external parameter matrix $[R\ t]$ can be solved as,

$$[R\ t] = [R_1\ R_2\ R_3\ t] = A^{-1}H. \tag{5.20}$$

So far, we have completed the initialization of the approximate values of the intrinsic ma-

trix A and extrinsic matrix $[R\ t]$ of the galvanometer by means of a linear approximation.

5.3.2 Non-linear optimization

By deriving and solving the above formula, we have obtained the parameters other than the distance e between the pan mirror and the tilt mirror. Generally, we can obtain rough length e by measuring with a ruler. So for all the parameters k_p, k_t, e, R, t needed to be optimized, we can obtain the optimized value by minimizing the control voltage errors of the two mirrors.

In a model optimization process, we give n images of different poses. Each image has m two-dimensional code labels generated by OpenCV. We assume that these two-dimensional code labels are interfered by independent and identically distributed noise. Then the minimized error function can be written in the following form,

$$\begin{cases} \varepsilon_p = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| {}^{ij}u_t - \frac{1}{2k_t} \arctan \frac{{}^{ij}y_g}{{}^{ij}z_g} \right\|^2 \\ \varepsilon_t = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| {}^{ij}u_p - \frac{1}{2k_p} \arctan \frac{{}^{ij}x_g}{{}^{ij}d_g + e} \right\|^2 \end{cases}, \tag{5.21}$$

$$s.t. \quad \begin{bmatrix} x_g & y_g & z_g \end{bmatrix}^T = \exp(\xi^\wedge) \widetilde{P_w}, \tag{5.22}$$

$${}^{ij}d_g = \frac{{}^{ij}z_g}{\cos(\arctan \frac{{}^{ij}y_g}{{}^{ij}z_g})}. \tag{5.23}$$

In order to facilitate the calculation, we use the Lie algebraic form ξ of $[R\ t]$ to represent the rotation and translation relations. Minimizing residuals ε_p and ε_t is a nonlinear minimization problem, which is solved by the LM algorithm implemented in minpack. The initial values of the parameters k_p, k_t, e, R, t, in the LM algorithm can be calculated from the previous section.

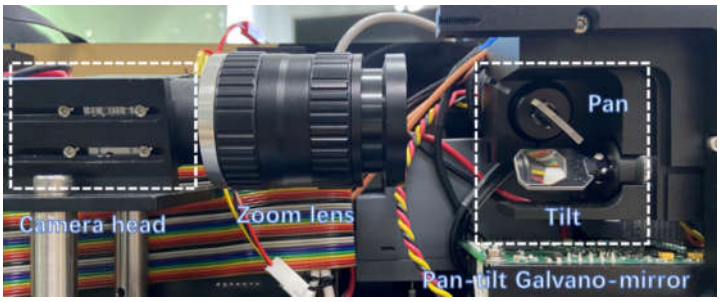


Figure 5.3: Overview of galvanometer-based active vision system.

5.4 Experiment

5.4.1 Hardware configuration

As shown in Figure 5.3, the galvanometer-based active vision system consists of a high-speed CMOS camera head with a resolution of 720×540 (MV-CA004-10UC, Hikvision, China), and a two-axis pan-tilt galvanometer (TSH8130A, Sunny Technology , China). The camera is equipped with a 55 mm telephoto lens, and the experimental scene is set indoors at a distance of 7 m. A 720×540 images corresponded to a 0.68×0.51-m-area and one pixel corresponds to 0.94 mm. In addition, there is a 16-bit precision D/A control board (AX301B, ALINX, China) to accept the digital signal of the computer and convert it into an analog voltage to drive the galvanometer. The input 16-bit digital voltage of the D/A control board is -5V~5V, and there is a certain deviation between the output analog control voltage and the digital voltage. For example, the digital voltage is set to 5V, and the output analog voltage is about 4.7V.

5.4.2 Calibration process

As shown in Figure 5.4, the whole process is mainly divided into two processes, active detection process and solution process. During active detection, we need to place planar targets in different areas covering the field of view of the active camera. We use OpenCV's dictionary toolkit to generate and detect QR codes. The active camera detects the position of the QR code through an active detection program, adjusts the QR code to

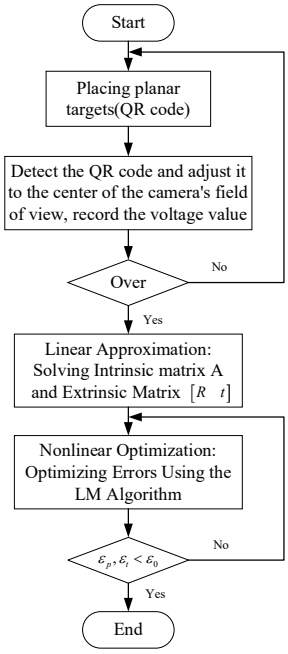


Figure 5.4: Calibration flow chart based on planar target

the center of the camera's field of view, and finally records the voltage. In the solution process, the intrinsic matrix A and the extrinsic matrix $[R\ t]$ are firstly solved by linear approximation. After that, the LM algorithm is used to optimize the variables k_p, k_t, e, R, t which are needed to be optimized. When the error $\varepsilon_p, \varepsilon_t < \varepsilon_0$ ($\varepsilon_0 = 10^{-5}$), the calibration is ended. It takes more time to change the position of the calibration plate and detect it in the whole calibration process, about 20 minutes. The acquisition and optimization of model parameters is very fast and can be completed within one minute.

5.4.3 Indoor calibration based on calibration board

In a close-range scene, due to the small field of view, the overall linearity is high and the error is small. We tested the effect of the algorithm within 7 m in an indoor scene. As shown in Fig 5.5, we select the ground point O_w about 7 m away from the galvanometer as the world coordinate origin, and the vertical direction is the Y_w axis to establish the left-hand coordinate system $O_wX_wY_wZ_w$. Similarly, the QR code calibration

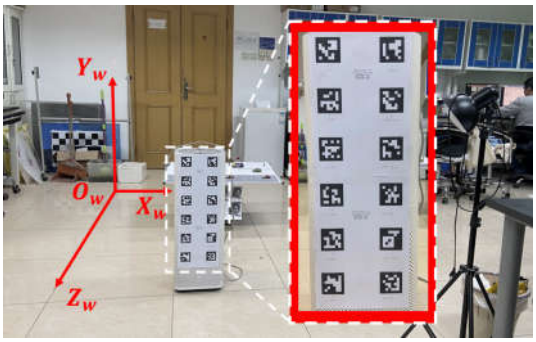


Figure 5.5: Overview of indoor calibration environment.

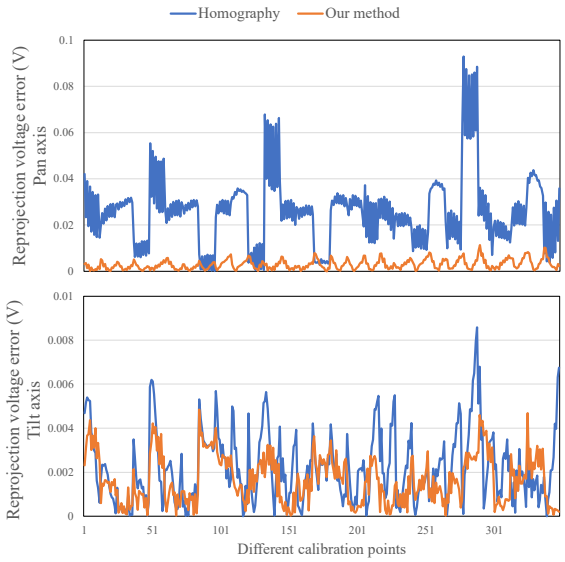


Figure 5.6: Reprojected control voltage error for indoor calibration.

board generated by OpenCV contains a total of 6×2 QR code blocks, and each QR code has this unique number, ranging from 0 to 11. The horizontal interval between each two-dimensional code is 130 mm, and the vertical interval is 100 mm. By constantly moving the air purifier and adjusting the position of the spatial two-dimensional code, try to fill the pan axis scanning space of the entire galvanometer to obtain the relationship between multiple sets of spatial points and the control voltage.

Finally, our calibration algorithm is applied to multiple sets of data in indoor scenarios, and the results are shown in Fig. 5.6. Obviously, in the linear-based homography matrix prediction method in Fig. 5.6, the reprojection voltage error varies regularly. Be-

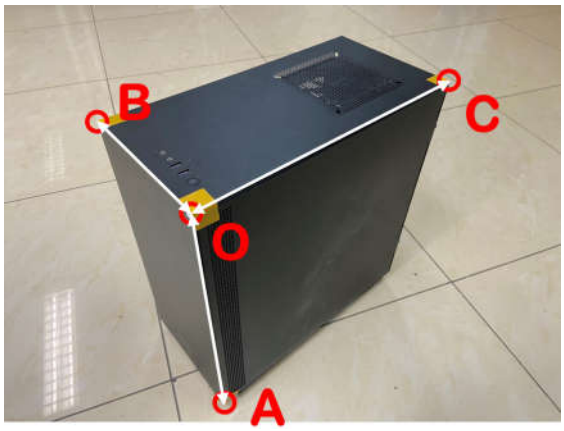


Figure 5.7: Chassis placed at a distance.

cause the pan-axis mirror experienced a change from the maximum negative angle to the maximum positive angle in the process of moving the air purifier. When the deflection angle is close to 0°, because $\tan\theta \approx \theta$, the nonlinear error is small. When the scanning angle is large, the nonlinear error increases, resulting in a larger error in the predicted voltage. But the reprojection error calculated by our proposed algorithm is always within 0.01 V, which has high performance. At the same time, since the non-linearity of the tilt mirror is only provided by the tan function, the overall error is smaller than that of the pan mirror.

5.4.4 Spatial localization based on dual galvanometer-based stereo active vision

Visual localization is an important application of the robot stereo active vision system. Dual galvanometer-based active vision systems are fixedly placed to form a stereotaxic system, and the triangulation principle is used to complete the spatial localization. The dual galvanometer-based active vision systems are about 1.9 m apart, completing the same calibration process as in the previous subsection. As shown in the Fig 5.7, the chassis is placed about 7 m away from the stereo active vision system. The four vertices of the chassis are O, A, B, and C respectively. We use colored tape to stick the four corners

Table 5.1: The localization results

		Spatial point coordinate (X_w, Y_w, Z_w) (cm)	Length (cm)	Error (cm)
1	O	(131.24, 25.91, 175.87)		
	A	(131.57, -17.99, 174.51)	43.92 (OA)	0.32
	B	(116.61, 25.99, 160.61)	21.13 (OB)	0.13
	C	(161.98, 26.05, 145.98)	42.88 (OC)	0.18
2	O	(65.15, 25.86, 119.82)		
	A	(65.37, -18.18, 118.49)	44.06 (OA)	0.46
	B	(52.22, 25.98, 103.64)	20.72 (OB)	0.28
	C	(98.99, 26.06, 93.64)	42.79 (OC)	0.09
3	O	(176.71, 25.91, 129.39)		
	A	(177.28, -18.09, 127.88)	44.03 (OA)	0.43
	B	(167.42, 25.99, 110.71)	20.86 (OB)	0.14
	C	(215.24, 25.98, 110.32)	42.99 (OC)	0.29

of the chassis for later use. The color extraction algorithm performs corner detection. Among them, the three-dimensional length of the chassis is OA = 43.60 cm, OB = 21.00 cm, OC = 42.70 cm. By placing the chassis in different positions in the room, through the color detection method, the corners of the chassis are detected, and the spatial position of each corner is calculated. The results are shown in Table 5.1.

Table 5.1 lists the measurement results of the chassis in three different positions. The measurement errors are distributed from 0.09 to 0.46 cm and the RMSE value is 0.28 cm, which proves that the precision of our calibration method is sufficient in indoor scenes.

5.4.5 Real-time spatial positioning of moving objects

As shown in Figure 5.8, we simultaneously constructed a camera-based stereo vision system and a galvanometer-based stereo active vision system. The camera-based stereo vision system consists of 2 CMOS camera heads with a resolution of 1920×1200 (A5201CU150, Hikvision, China) and 6 mm lens. The system is capable of continuously

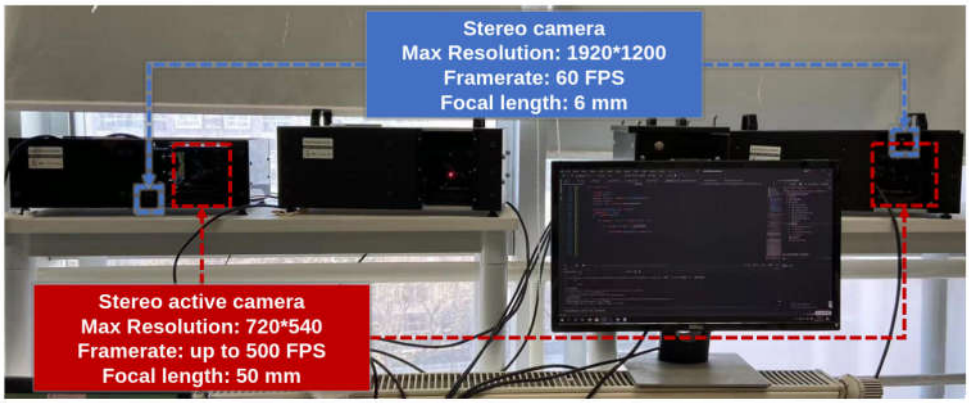


Figure 5.8: Overview of the stereo active vision system.

Figure 5.9: High-speed stereo tracking and real-time display.

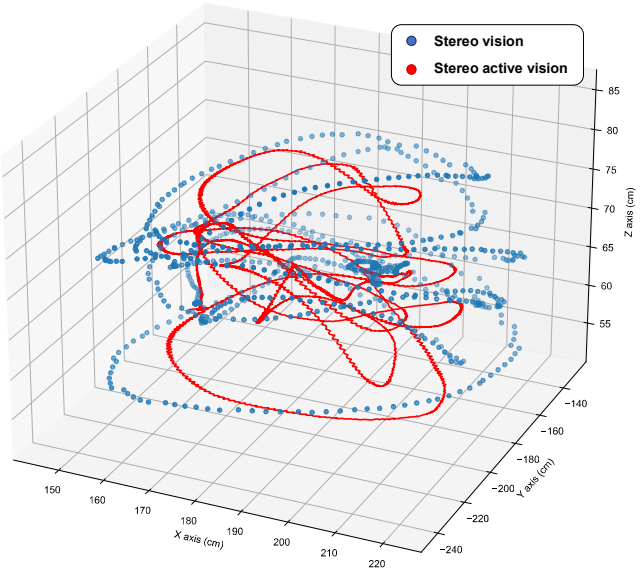


Figure 5.10: The spatial trajectory of the ball under stereo vision and stereo active vision.

capturing stereoscopic panoramic images at a speed of 60 fps. The stereo active vision system consists of two active cameras as illustrated in Figure 5.3. The two sets of stereo vision systems are connected to the control computer via USB. The control computer is equipped with an Intel i7-8700K processor (3.7 GHz), 32-GB DDR4 RAM, and Windows 10 Home (64-bit).

A high-speed stereo tracking and real-time display system utilizing a high-speed stereo active vision system was developed. As shown in Figure 5.9, a rapidly moving ping pong ball is positioned approximately 7 meters away from the stereo vision system. The stereo active vision system tracks the ping pong ball at a speed of 500 fps, ensuring that the ball lands in the center of the stereo active cameras' field of view. Simultaneously, the stereo active vision system calculates the position of the ping pong ball in space at a rate of 500 frames per second (fps) and updates the position of the virtual ball in the OpenGL window at a rate of 60 fps. In Figure 5.10, we present the spatial trajectory of the ping pong ball in both the traditional stereo vision system and the stereo active vision system.

5.5 Concluding remarks

In this chapter, a calibration method of reflective active vision system based on galvanometer is proposed. The method is flexible and suitable for some low-accuracy control galvanometer-based active vision systems. Based on the physical structure of the galvanometer, the mathematical model of the active vision system is established, and then the specific calibration process is given. We measured the reprojection error of the whole system and completed the spatial localization of the corner points of the chassis to evaluate the feasibility and accuracy of the calibration method. The experimental results show that the proposed calibration method has high accuracy indoors and is feasible.

However, at present, we only study the central optical path of the active camera,

which cannot be used for stereo vision. In the later stage, cameras can be added for joint calibration to complete more visual tasks such as stereo vision.

Chapter 6

HFR-video-based stereo correspondence using high synchronous short-term velocities

6.1 Introduction

Stereo vision offers a straightforward way for computers to comprehend the world and can reconstruct the three-dimensional geometric information of scenes [127]. It is widely used in various fields such as autonomous navigation systems for mobile robots [128], aerial and remote sensing measurements [129], medical imaging [130], SLAM [131], and more. Stereo correspondence is a crucial element of stereo vision that plays a vital role in finding corresponding point pairs between two images to calculate the depth information of the stereo image [132].

The goal of this study is to achieve stereo correspondence for multiple moving objects with similar appearances. Over the past few decades, extensive research has been dedicated to stereo correspondence. Traditional stereo correspondence algorithms can be categorized into local, global, and semi-global methods. These methods use manually extracted features, such as sum of absolute difference (SAD) [133], normalized cross-correlation (NCC) [134], SIFT (Scale-Invariant Feature Transform) [135], and ORB (Oriented FAST and Rotated Brief) [136], to provide similarity measures between left and right image patches. However, the performance of traditional stereo correspondence

methods is severely limited by the handcrafted features used in the cost function. In Ref. [137], convolutional neural networks (CNNs) were first introduced for stereo correspondence, demonstrating advantages in both speed and accuracy over traditional methods. Currently, deep learning-based image similarity measurement methods mainly rely on feature extraction from deep networks [138] and similarity comparison through metric learning [139].

However, appearance-based correspondence methods face significant challenges due to variations in camera viewpoints, lighting conditions, and pose changes [140]. Motion information, on the other hand, is independent of object appearance and exhibits excellent performance in scenes with similar appearances and drastic changes in appearance. Currently, a significant amount of research has been devoted to cross-camera multi-object correspondence based on motion information [141, 142]. Existing motion similarity measurement methods can be divided into two categories: spatial similarity and spatio-temporal similarity [143]. Spatial similarity only considers the same geometric shape and ignores the temporal dimension, which is not suitable for real-time stereo correspondence systems. The update of motion information is delayed due to the limited speed of traditional visual image input (30 or 60 fps) [144], making trajectory synchronization of high-speed moving objects difficult. However, high-speed vision sensors operate at hundreds or even higher frequencies, enabling them to observe moving objects and capture phase differences with extremely low latency [110]. Additionally, viewing angles significantly affect trajectory matching performance. First-order motion velocity and second-order acceleration directions are relatively insensitive to viewing angles.

6.2 Proposed algorithm

6.2.1 Independent multi-object tracking in HFR stereoscopic video

In this study, we conducted offline experiments using the HFR stereoscopic videos to validate the effectiveness of our algorithm. The first step involves the fast tracking of multiple objects using the HFR stereo camera, which enables the real-time update of the motion positions and velocities of the objects. However, HFR stereoscopic videos not only provide more image information but also impose a higher computational burden on multiple object tracking. HFR stereoscopic videos usually run at 200 frames per second or higher, leaving us with only 5 milliseconds or less for computation. However, detectors that yield good detection performance usually require longer running times. For instance, in this study, the hand detection using MediaPipe takes approximately 30 milliseconds, while the marker detector takes about 10 milliseconds. Therefore, we proposed a hybrid tracking approach that combines object detection with template matching to enable the tracking of multiple objects with very low processing time. This approach exhibits good tracking performance for objects with drastic appearance changes due to the constantly updated object templates. Due to the low latency of high frame rate (HFR) videos, the motion speed of objects between frames is relatively low. To quickly locate objects near their image blocks, template matching can be utilized. Figure 6.1 illustrates the hybrid detection method based on template matching and object detection. The time interval between input HFR images is denoted as τ milliseconds. The detector continuously performs object detection with a time interval of δ milliseconds, where $\delta(\delta > \tau)$ represents the processing time of the detector. The detection results $D(I_t)$ obtained from the detector in the input image I_t at time $t = k \times \delta$ $(k = 0, 1, 2, \dots)$ can be expressed as follows:

$$D(I_t) = \{d_t^1, d_t^2, \dots, d_t^l, \dots, d_t^L\}(l = 1, 2, \dots, L). \tag{6.1}$$

Each detection result d_t^l comprises six parameters:

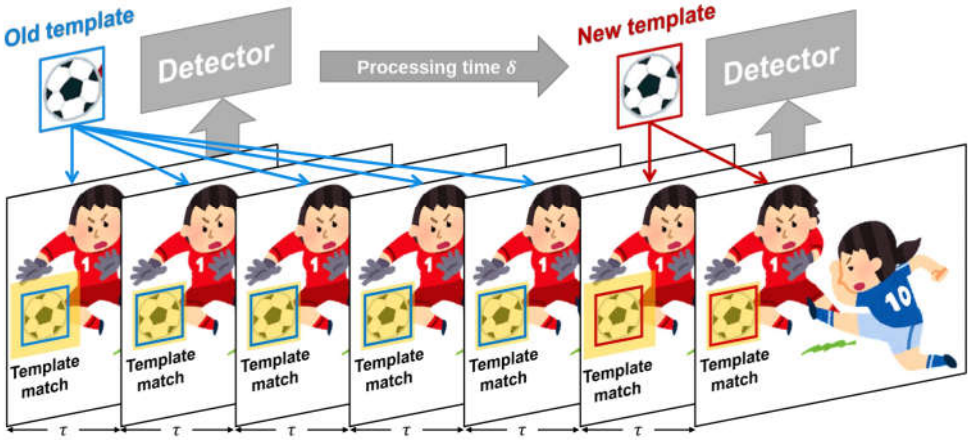


Figure 6.1: Hybrid detection method based on template matching and object detector.

$$d_t^l = \{x_l, y_l, w_l, h_l, p_l, c_l\}. \tag{6.2}$$

x_l, y_l, w_l, and h_l represent the starting image coordinates, width, and height of the l-th object image block, respectively. p_l and c_l represent the confidence score and category of the detection result, respectively. As indicated in Figure 6.1, we obtained object templates T_t updated at time intervals of δ. Simultaneously, we perform template matching using the most recently updated templates to detect objects at time intervals of τ. In high-speed visual systems where the system's operational speed is a priority, a trade-off between speed and accuracy is often necessary. Therefore, we employ the sum of absolute differences (SAD) as the similarity metric for image-template matching. The detection process for objects between adjacent HFR frames is as follows:

$$P_l(t) = P_l(t-\tau) + \underset{|x|\leq R,|y|\leq R}{\arg\min} E(x,y), \tag{6.3}$$

$$E(x,y) = \sum_{x',y'} (T_l(x',y') - I_t(x'_t + x + x', y'_t + y + y')). \tag{6.4}$$

$P_l(t-\tau)$ and $P_l(t)$ represent the coordinates of the center of the l-th object in the previous and current frames, respectively. T_l is the template image of the l-th object that is most recently updated. I_t represents the region of interest (ROI) being searched in the current image, as highlighted in yellow in Figure 6.1. $(x't, y't)$ represents the top-left point coordinate of the ROI region in the current image. R is the search range of the template matching. To mitigate the impact of object appearance changes on tracking, we perform template updates by searching in a larger region each time, as depicted in the yellow area in the figure.

In this work, we employ a distance matrix Φ between I objects in the previous frame and J objects in the current frame as a replacement for the Intersection over Union (IOU) method for object tracking.

$$\Phi = \begin{bmatrix} \psi(1,1) & \psi(1,2) & \cdots & \psi(1,J) \\ \psi(2,1) & \psi(1,2) & \cdots & \psi(2,J) \\ \cdots & \cdots & \cdots & \cdots \\ \psi(I,1) & S(I,2) & \cdots & \psi(I,J) \end{bmatrix}. \tag{6.5}$$

$\psi(i, j)$ represents the Euclidean distance between the i-th object in the previous frame and the j-th object in the current frame, measured in pixels. We employ the Hungarian matching algorithm to obtain tracking results quickly and efficiently.

In high-speed imaging, where object motion is relatively slow and motion between adjacent frames is approximately uniform, we use a Kalman filter for optimal estimation of motion. The Kalman filter can also be used for short-term motion prediction when object detection is temporarily lost.

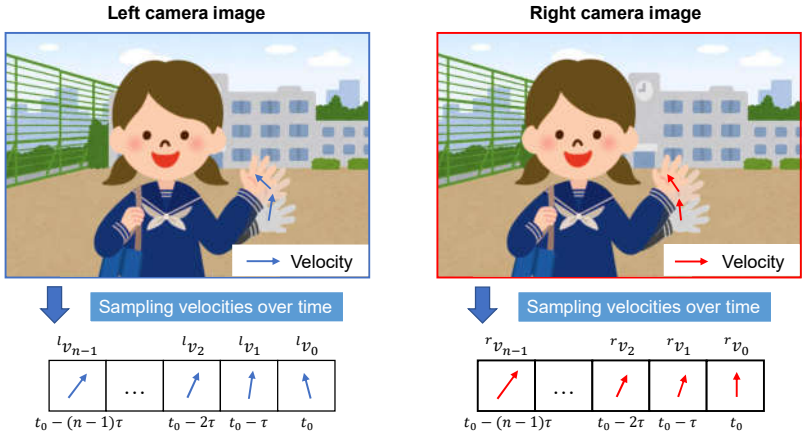


Figure 6.2: Sampling velocities over time in HFR stereoscopic video.

6.2.2 Correspondence based on highly synchronous motion

Velocity-Based Correspondence: Once the optimal tracking state of the object is obtained, we can obtain highly synchronized spatiotemporal velocities (STVs). As shown in Figure 6.2, we sampled the velocities of the object at the pixel scale within N high-speed frames to extract the motion feature of the object. The STVs V of the object were then obtained as follows:

$$V = \{v_{N-1}, \ldots, v_n, \ldots, v_1, v_0\}, (n = 0, 1, \ldots, N-1), \tag{6.6}$$

where $v_n = [dx_n, dy_n]$ is the velocity vector at the pixel scale in the n-th frame before the current frame.

In this study, we propose the concept of the scale cosine distance. While the calculation of the cosine distance yields the cosine of the angle between both vectors, which is close to 1 when the angle is small, the cosine distance does not consider the length of the vector. This means that two parallel vectors with different lengths would have a cosine distance of 1, even though their similarity is very low. To overcome this limitation, we introduce the scale cosine distance s between vectors A and B, which takes into account the length of the vector, as expressed below:

$$s = \frac{A \cdot B}{\max(|A|, |B|)^2}, \tag{6.7}$$

where $|A|$ and $|B|$ are the modulo lengths of vectors A and B, respectively. When the lengths of both vectors are similar and the included angle is small, the scale cosine distance is larger, with a higher similarity close to 1.

Hence, for the N-dimensional high-synchronization STVs ${}^{l}V_i$ and ${}^{r}V_j$ extracted from the left and right HFR stereo cameras, we calculated the scale cosine similarity $S_v(i, j)$ between them as follows:

$$S_v(i, j) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{{}^{l}v_k \cdot {}^{r}v_k}{\max\left(|{}^{l}v_k|, |{}^{r}v_k|\right)^2}. \tag{6.8}$$

Direction-Based Correspondence: The correlation of velocity decreases in the presence of a large viewing angle in the HFR stereo camera. The correlation between the direction of velocity change and the change in camera viewing angle is relatively small. We extract the cosine values of the angle changes between velocities to form a short-term angle for measuring the similarity A of direction changes.

$$A = \{a_{N-2}, \ldots, a_n, ..., a_1, a_0\}, (n = 0, 1, \ldots, N-2). \tag{6.9}$$

a_n is the cosine value between adjacent velocity angles,

$$a_n = \frac{v_{n+1} \cdot v_n}{|v_{n+1}| \cdot |v_n|}, (n = 0, 1, \ldots, N-2). \tag{6.10}$$

Hence, for the $(N-1)$-dimensional high-synchronization STVs ${}^{l}A_i$ and ${}^{r}A_j$ extracted from the left and right HFR stereo cameras, we calculated the direction similarity $S_a(i, j)$ between them as follows:

$$S_a(i, j) = 1 - \frac{1}{2(N-1)} \sum_{i=0}^{N-2} |{}^{l}a_k - {}^{r}a_k|. \tag{6.11}$$

Mixed Correspondence: The similarity measure of object motion is contributed by both the similarity of velocities and the similarity of velocity change directions. We define the mixed similarity $S(i, j)$ between the short-term velocities of the i-th target in the left camera and the j-th target in the right camera as follows:

$$S(i, j) = \omega_v S_v(i, j) + \omega_a S_a(i, j), \tag{6.12}$$

$$\text{s.t. } \omega_v + \omega_a = 1. \tag{6.13}$$

where ω_v and ω_a are scale factors that reflect the contribution of velocity and direction to the similarity metric in different camera perspectives. Generally, when the HFR stereo camera has a large field of view, the direction similarity $S_a(i, j)$ should contribute a larger proportion. Finally, based on the mixed similarity of short-term velocities, a bipartite graph S can be reconstructed for I targets in the left camera and J targets in the right camera,

$$S = \begin{bmatrix} S(1,1) & S(1,2) & \cdots & S(1,J) \\ S(2,1) & S(1,2) & \cdots & S(2,J) \\ \cdots & \cdots & \cdots & \cdots \\ S(I,1) & S(I,2) & \cdots & S(I,J) \end{bmatrix}. \tag{6.14}$$

Using the Hungarian matching algorithm, we can easily obtain the correspondence relationship based on motion information.

6.3 Experiment

The proposed stereo correspondence algorithm was implemented offline using an HFR stereo camera system that operated at a speed of 200 fps. The system was com-

posed of two high-speed USB 3.0 camera heads from Imaging Source Corp. (DFK 37BUX273, Germany) and a personal computer. The cameras were compact, measuring $36 \times 36 \times 25$ mm in size, weighing 70 g, and had no mounted lens. They were capable of capturing and transferring 10-bit color images of 1440×1080 pixels to RAM at a rate of 238 fps via a USB 3.0 interface. We used a PC with the following hardware specifications to record the HFR stereoscopic video: Intel Core i9-9900K @ 3.2 GHz CPU, 64 GB RAM, and an NVIDIA GeForce RTX 2080 Ti GPU.

To evaluate the performance of our stereo correspondence algorithm, we analyzed HFR stereo offline videos that were captured at a rate of 200 fps ($\tau = 5$ ms) with a 2-ms exposure time. In this study, we chose the hand as the detection target because it had a high similarity in texture and color across different people, and moved at a high speed relative to other body parts, making it difficult to use appearance-based methods for correspondence. We conducted three experiments to evaluate our algorithm: stereo correspondence evaluation, correspondence of fast-moving hands, and correspondence in a meeting room scene. For the hand detection task, we used Google's MediaPipe toolkit, which provided accurate and rapid hand detection.

6.3.1 Stereo correspondence evaluation

We conducted an evaluation of the correspondence performance of our HFR stereo correspondence algorithm when implemented offline in our system. Figure 6.3 illustrates the experimental setup for the stereo correspondence evaluation, where two metronomes were fixed 800 mm away from the HFR stereo camera. The small metronomes operated at frequencies of 3.0 and 2.6 Hz, respectively. OpenCV-generated markers were attached to different positions on the pointers of both metronomes. As a result, markers on a similar pointer exhibited similar movements when shaking, but with different magnitudes of movement. During the operation of the metronomes, we captured a 200-fps HFR stereoscopic video using 12-mm lens fixed cameras. The positions of the individual markers

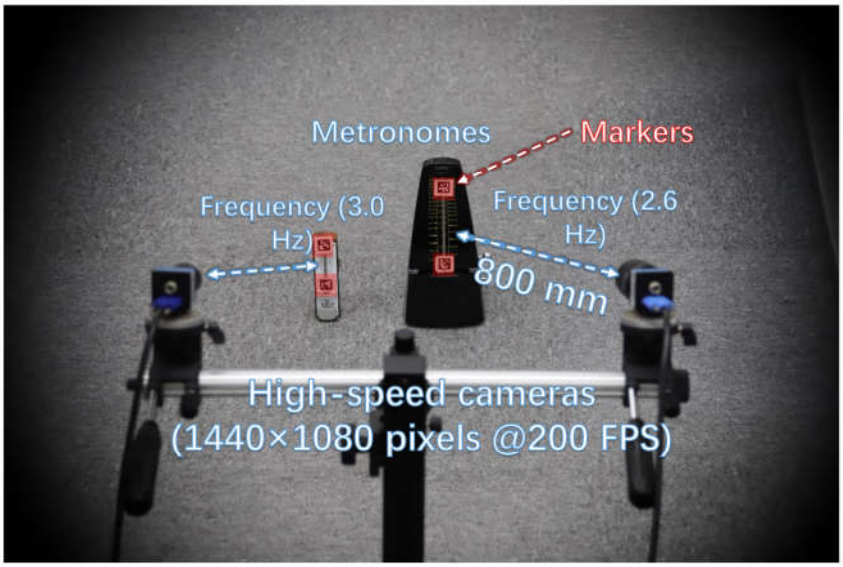
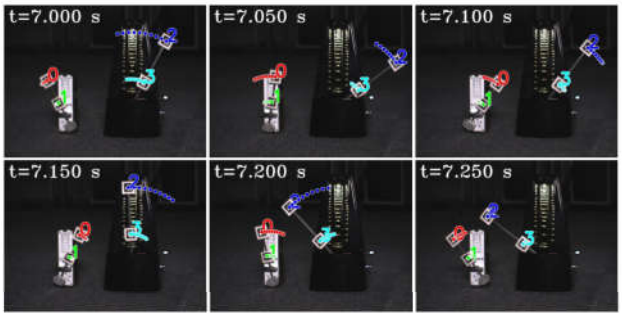


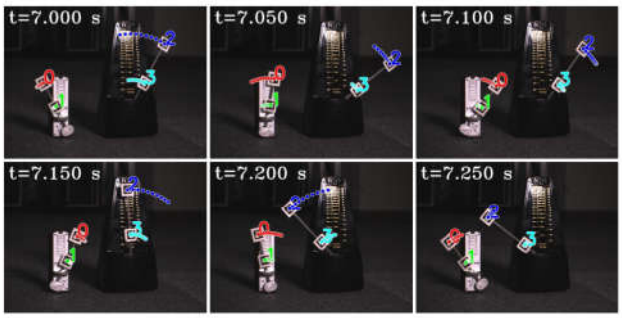
Figure 6.3: Experiment setup for similar motion correspondence.

were easily detected using OpenCV.

In Figure 6.4, we show the input stereo images of size 1440 × 1080 pixels, with the correspondence results at intervals of 0.06 s for t = 7.00~7.25 s. After applying the stereo correspondence algorithm, the same marker in the HFR stereoscopic video was marked with numerical symbols of a similar color. The xy coordinate values of the image centroids of the markers in the left HFR stereoscopic video are presented in Figure 6.5. From the image, markers 0 and 1 exhibited similar movement with different magnitudes than markers 2 and 3. The mixed similarities of the moving markers' STVs over time are shown in Figure 6.6. Figure 6.6(a), 6.6(b), 6.6(c), and 6.6(d) depict the mixed similarities between markers 0, 1, 2, and 3 in the left HFR stereo image and those in the right HFR stereo images, respectively. The graph indicates that similar markers in the HFR stereo images have a high degree of similarity, which is almost greater than 0.8. Markers 0 and 1 on a similar pointer have a similar angular velocity, but different linear velocities. However, our scale cosine distance includes a scale factor that can easily distinguish between markers 0 and 1. The same applies to markers 2 and 3. We also considered the effect of the duration of STVs on multi-object stereo correspondence. Figure 6.7 presents the



(a) Left HFR stereoscopic video.



(b) Right HFR stereoscopic video.

Figure 6.4: Input images and correspondence result in evaluation.

results of stereo correspondence using a 30 fps stereo camera in the same scene. It is evident that marker 0 and marker 3 do not match in the correspondence. Figure 6.8 shows the short-term velocity features of marker 0 within a 0.3-second interval in the stereo camera at $t = 7.710$ s. It is evident that traditional low-speed cameras have synchronization issues when tracking fast-moving objects. Velocity information is delayed by approximately 30 milliseconds, which significantly affects the correspondence results, especially when the object changes direction frequently. Figure 6.9 shows the short-term velocity features of marker 0 within a 0.3-second interval in the HFR stereo camera. In contrast, the HFR camera not only provides more motion information in a short time but also has much higher synchronization.

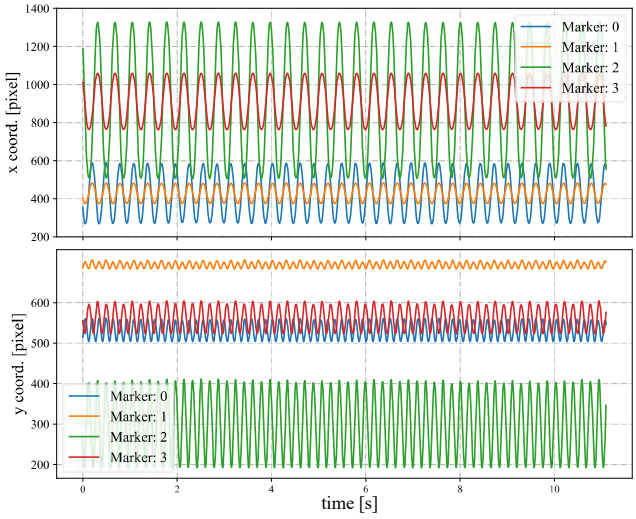
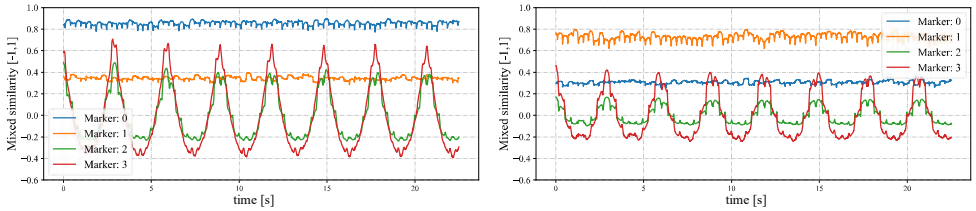
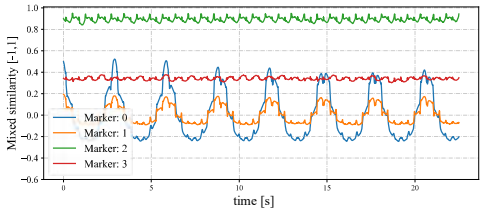


Figure 6.5: Image centroids of the markers in the stereo correspondence evaluation.

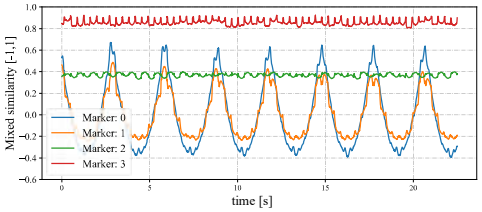


(a) Marker 0 in the left video.

(b) Marker 1 in the left video.



(c) Marker 2 in the left video.



(d) Marker 3 in the left video.

Figure 6.6: Mixed similarities of different markers in the HFR stereoscopic video when the STVs length is 64.

6.3.2 Stereo correspondence of hands with complex movements

We present the stereo correspondence results of hand movements during complex actions such as overlap and reappearance. The experimental setup is illustrated in Figure 6.10. Two individuals waved their hands approximately 8 m away from the HFR stereo cameras. Similar to the previous experiment, we captured a 200-fps HFR stereoscopic video using 12-mm fixed lens cameras. The hand movements in the video included mutual oc-

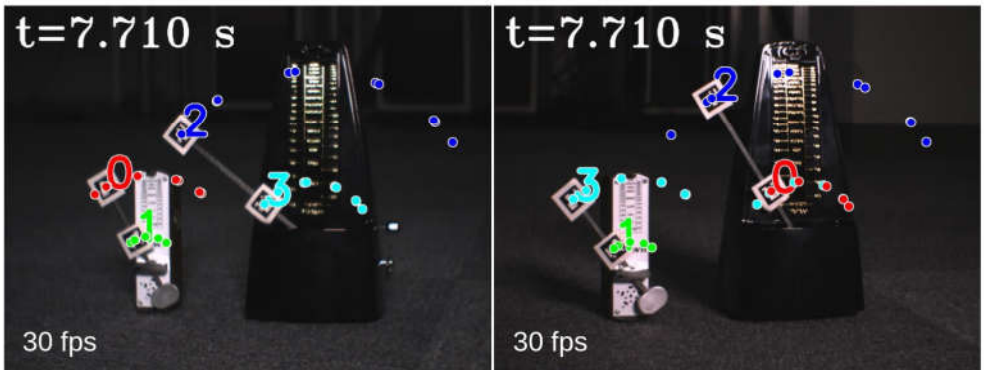


Figure 6.7: Correspondence results using a stereo camera at 30 fps (t = 7.710 s).

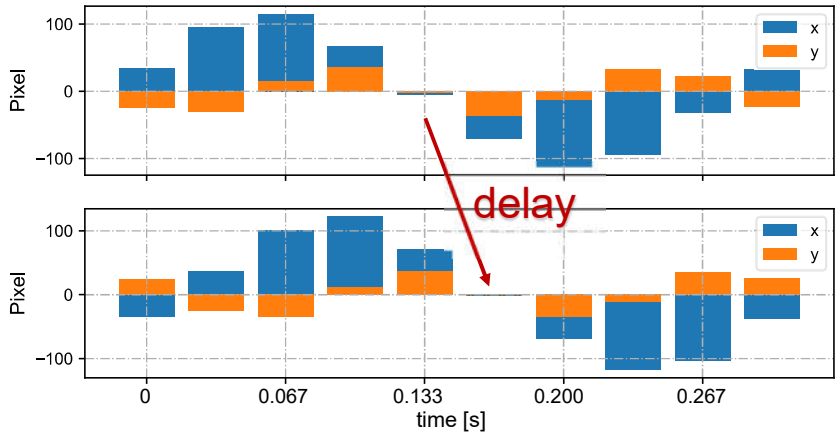


Figure 6.8: Short-term velocities of marker 0 in the stereo video in 0.3 s at 30 fps (t = 7.710 s).

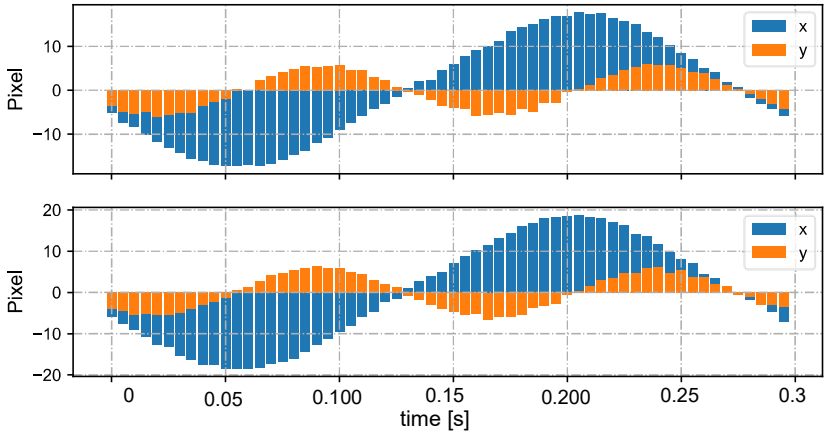


Figure 6.9: Short-term velocities of marker 0 in the stereo video in 0.3 s at 200 fps.

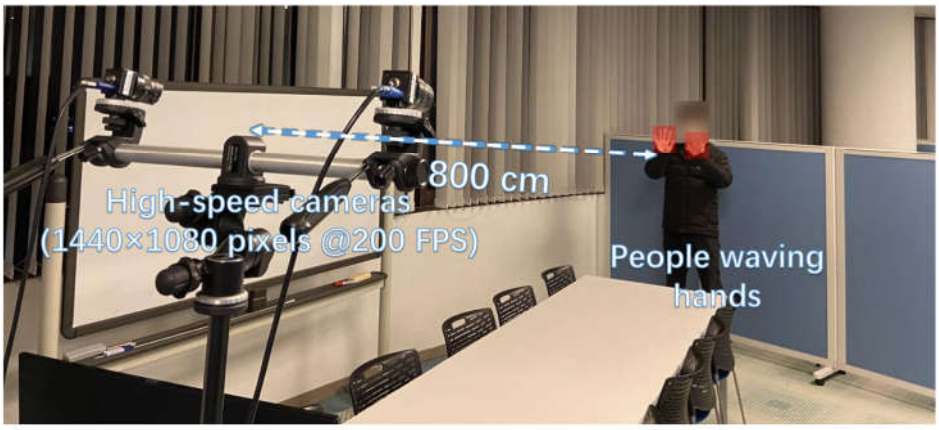
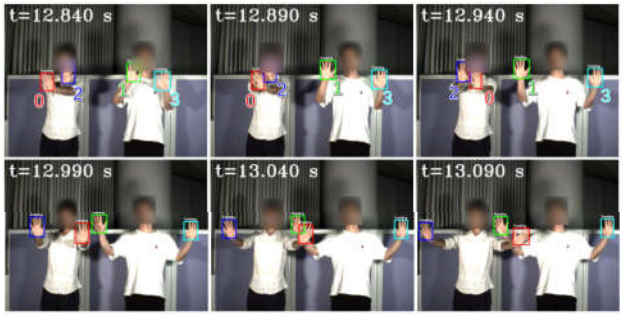


Figure 6.10: Experiment setup for hand stereo correspondence.

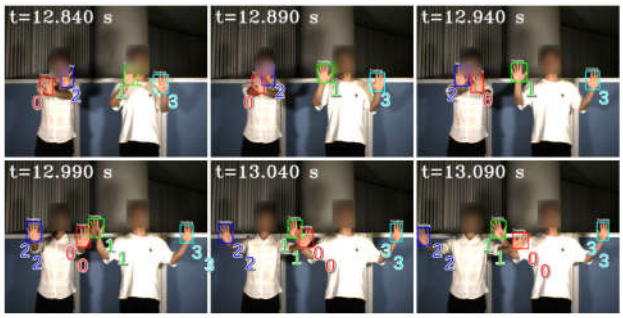
clusion, static states, disappearance, and reappearance. There were four hands in the HFR stereoscopic video, represented by hand 2, hand 0, hand 1, and hand 3 from left to right. In the offline detection process, we utilized MediaPipe to detect the hands.

In Figure 6.11, we depict the input HFR stereo images with a resolution of 1440×1080 pixels and the correspondence results at intervals of 0.05 s for t = 12.84~13.09 s. In the HFR stereoscopic video, similar hands are marked with similar colors from left to right. As shown in the graph, there is an overlap between hands 2 and 0, which belong to the person on the left. Hands 0 and 1, belonging to different people, also overlap. Our method correctly predicts the position of the hands and completes the hand correspondence even in the case of missing objects. The xy coordinate values of the image centroids of the hands in the left HFR stereoscopic video are shown in Figure 6.12. By analyzing the trajectories of the four hands, we can decompose the entire motion process into multiple actions. From 1.8 to 9.0 s, the hands belonging to the same person crossed each other and moved. From 23.0 to 28.0 s, hands 2 and 3 disappeared and reappeared. For the rest of the time, the four hands were stationary. The mixed similarities of different hands' STVs over time are shown in Figure 6.13.

Figure 6.13(a), 6.13(b), 6.13(c) and 6.13(d) show the mixed similarities of STVs between hands 0, 1, 2, and 3 in the left HFR stereo image and those in the right HFR stereo



(a) Left HFR stereo images.



(b) Right HFR stereo images.

Figure 6.11: Input images and hand correspondence result.

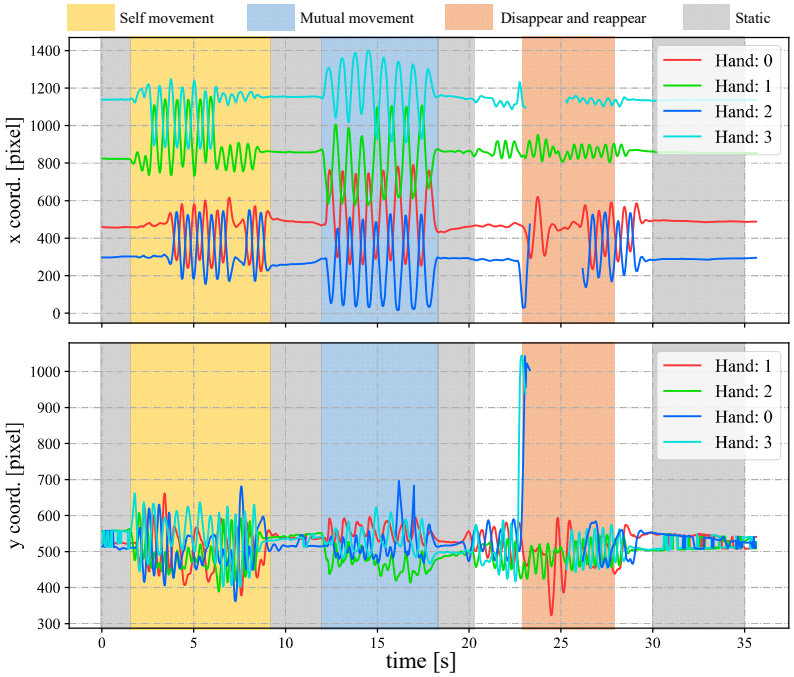
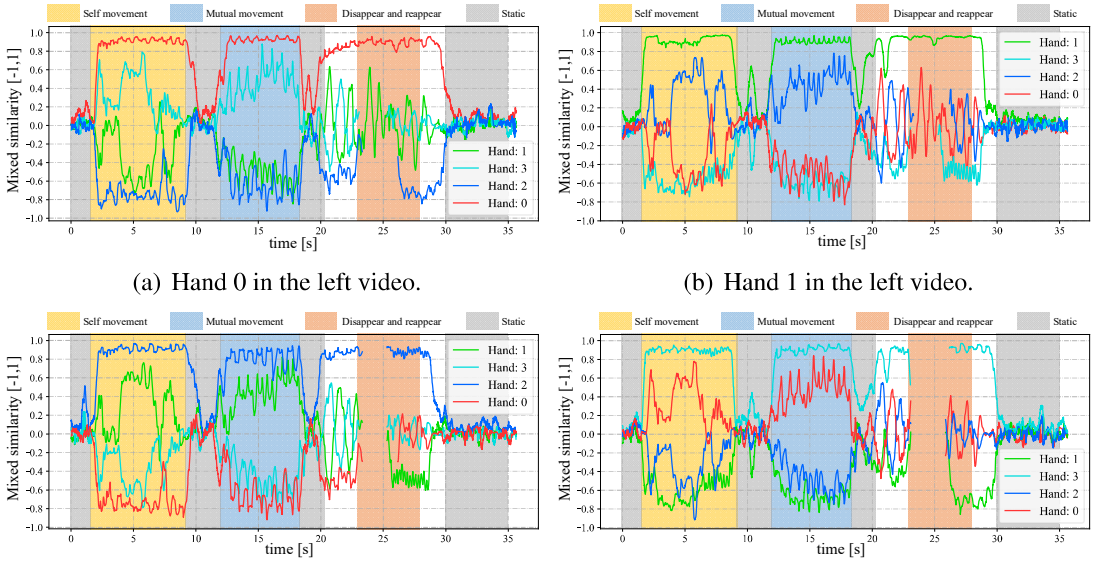


Figure 6.12: Image centroids of the hands in the left HFR stereoscopic video.



(a) Hand 0 in the left video.

(b) Hand 1 in the left video.

(c) Hand 2 in the left video.

(d) Hand 3 in the left video.

Figure 6.13: Mixed similarities between different hands in the HFR stereoscopic video when the STVs length is 64.

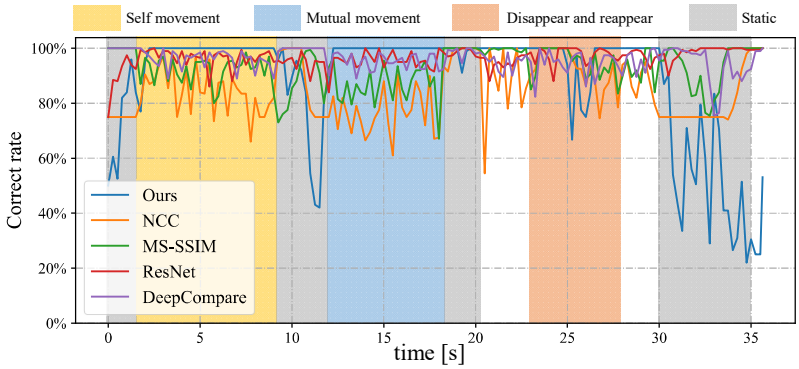


Figure 6.14: Correct rate of different stereo correspondence methods updated every 0.25 s.

images, respectively. Similar to the metronome correspondence, the motion features of a similar hand in the HFR stereoscopic video have a higher similarity. Since our features are motion-based, it can be seen from the figure that missing motion features introduced more uncertainty when the hand was stationary. Furthermore, we added appearance-based correspondence methods and calculated the accuracy of each method for hand correspondence every 0.25 s, as shown in Figure 6.14. The deep learning methods, ResNet

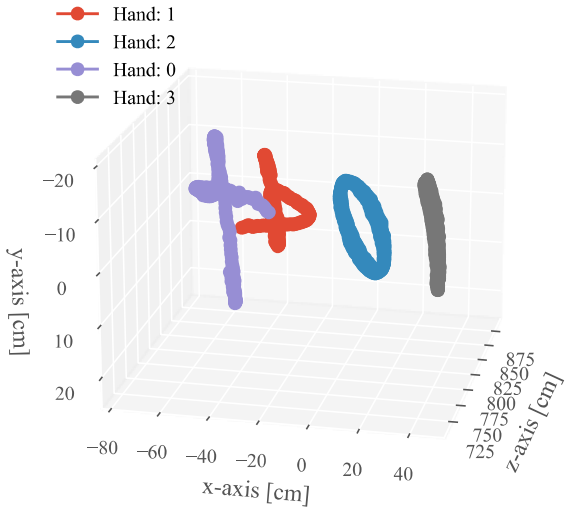


Figure 6.15: 3D trajectory of each hand with 7~8 s.

and DeepCompare, achieved significantly better results throughout the process and were clearly superior to traditional methods. Our method maintained an accuracy of almost 100% during hand movement. The accuracy rate was lower than that of the appearance-based methods only when the hand was stationary. In calibrated stereo cameras, when similar objects are found in the stereo camera, their spatial positions can be calculated. In Figure 6.15, we plotted the 3D trajectories of hands 0, 1, 2, and 3 over 7 to 8 s. From the image, we can see that the four hands moved up and down at a distance of approximately 8 m from the camera. The acquisition of spatial information helped us to better analyze the movement of objects.

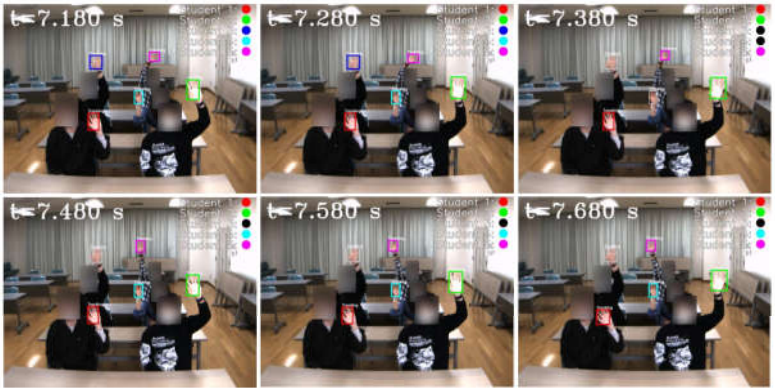
6.3.3 Stereo correspondence in the meeting room

Finally, we present the experimental results for stereo correspondences when the stereo cameras operate at 200 fps in a meeting room. To obtain a larger field of view, the stereo cameras are equipped with 6-mm lenses. The experimental setup is illustrated in Figure 6.16. In the meeting room, several students were more than 2 m away from the stereo cameras. Due to factors such as privacy and occlusion, it was difficult to detect and identify different students by their faces. Obtaining the spatial position using stereo

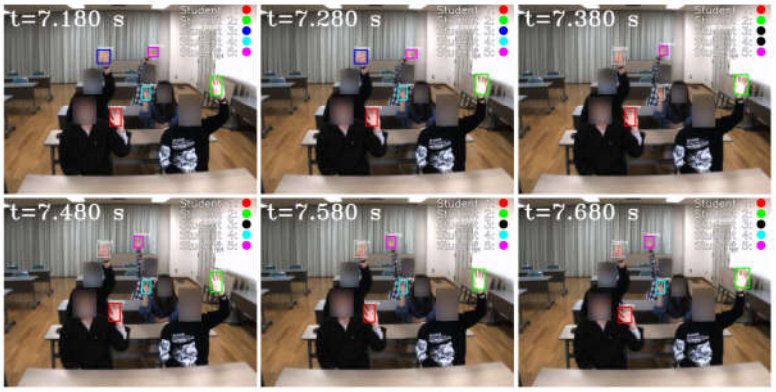
Figure 6.16: Experimental environment for stereo correspondence in the meeting room.

correspondence of the hands is a feasible solution to identify different students.

In Figure 6.17, we depict the input HFR stereo images of 1440 × 1080 pixels with correspondence results at intervals of 0.1 s for t = 7.180~7.680 s. We numbered the students from 1 to 5, from the nearest to the farthest. Similar hands in the stereo HFR video were marked with similar colors, as in the previous experiment. When the students raised their hands, we performed stereo correspondence using hand movements. Furthermore, we calculated the 3D positions of the different hands. In this experiment, we knew the seating distribution of each student in advance, and we could identify who raised their hand through the position of the hand. In Figure 6.17, we marked the hand-raising action of classmates in the upper right corner. When the hands were raised, circles belonging to different students were filled with different colors; otherwise, they were filled with black. The xy coordinate values of the hand images in the left HFR stereoscopic video are shown in Figure 6.18 at t = 0–16 s. Simultaneously, Figure 6.19 shows the time variation of the mixed similarity between similar hands at t = 0–16 s. From the graph, the hands of students 1 to 5 appeared individually in the HFR stereoscopic video. The students' hands moved at 0–7 and 10.5–16 s. During motion, the same hand in the HFR stereoscopic video had a high mixed similarity of approximately 0.8. We stopped the hand from mov-



(a) Left HFR stereo images.



(b) Right HFR stereo images.

Figure 6.17: Input images and stereo correspondence result.

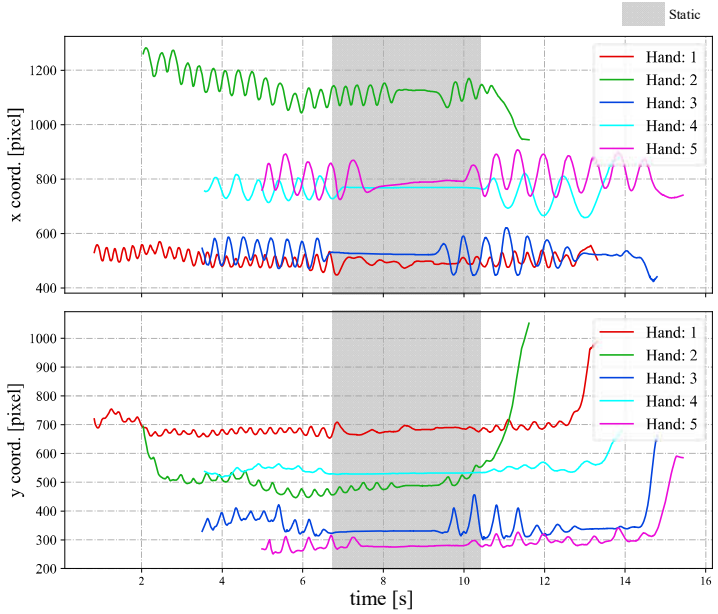


Figure 6.18: Image centroids of hands in the left HFR stereoscopic video.

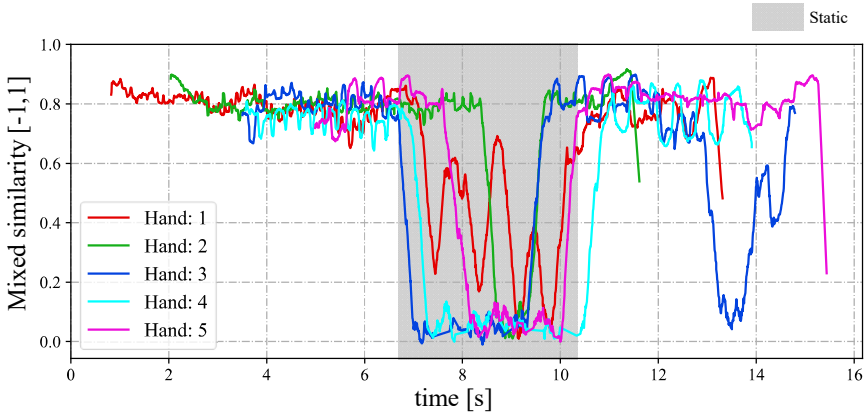


Figure 6.19: Mixed similarities between a similar hand in the HFR stereoscopic video when the STVs length is 64.

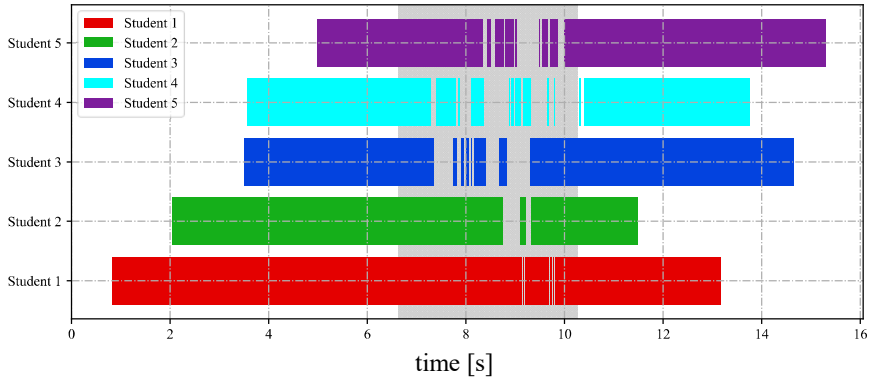


Figure 6.20: Statistical analysis of raised hands.

ing at 6.8–10.2 s. As seen in Figure 6.19, the mixed similarities of the same hand dropped rapidly, greatly reducing the accuracy of the correspondence. Figure 6.20 shows the Gantt chart of the detected students' hands raised over time. When the hand stopped moving, we could not accurately complete the correspondence. Our algorithm is currently limited regarding stereo correspondence in the static state. These results show that our method can accurately match objects in a stereoscopic video in moving scenes and use spatial information to complete certain applications.

Figure 6.21: Overview of stereo active vision systems.

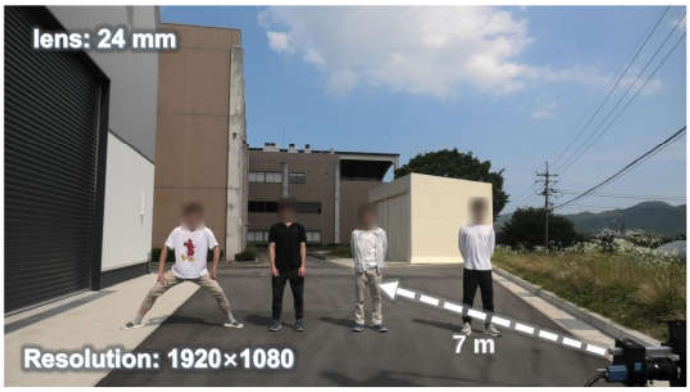


Figure 6.22: Experimental scene (taken by a digital camera).

6.3.4 Motion-based stereo correspondence in the stereo active vision system

The forms of motion are diverse, encompassing not only pixel-level movements within an image but also variations in control signals within an active servo system. The purpose of this experiment is to verify the stereoscopic matching of multiple targets in high-speed active stereo vision by utilizing two-dimensional variations in control voltages through mirror oscillation as motion signals. Figure 6.21 presents the high-speed active stereo vision system, which consists of two identical configured high-speed galvano-based cameras. The galvano-based camera consists of a high-speed CMOS camera head from Image Source, Bremen, Germany (DFK37BUX287) and a two-axis pan-tilt galvano-mirror from Cambridge Technology, Kansas City, MO, USA (6210H). The high-speed camera is equipped with a 75 mm lens. The stereo vision system is connected with the

Figure 6.23: Panoramic stitched images from stereo active camera.

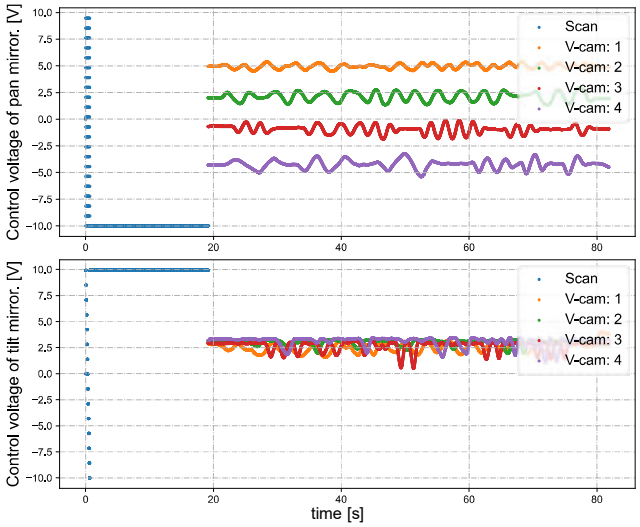


Figure 6.24: Trajectories of the control voltages for the pan and tilt mirrors of multiple virtual cameras.

control computer through USB, which is equipped with an Intel i9-10900X processor (3.6 GHz), 64-GB DDR4 RAM, and Window 10 Pro (64-bit). Control signals are sent to the stereo active vision system via a D/A board (PEX-340416) from Interface Corporation, Hiroshima, Japan.

Figure 6.22 shows the scene of the whole experiment taken by the digital camera, 4 experimenters are standing about 7 meters away from the active vision system. At the beginning of the experiment, the 4 experimenters remained still, and when the scanning initialization was completed, they would move individually. In Figure 6.24, trajectories of the control voltages for the pan and tilt mirrors of multiple virtual cameras in the left active vision system are shown. At the 20th second, the system completes the initializa-

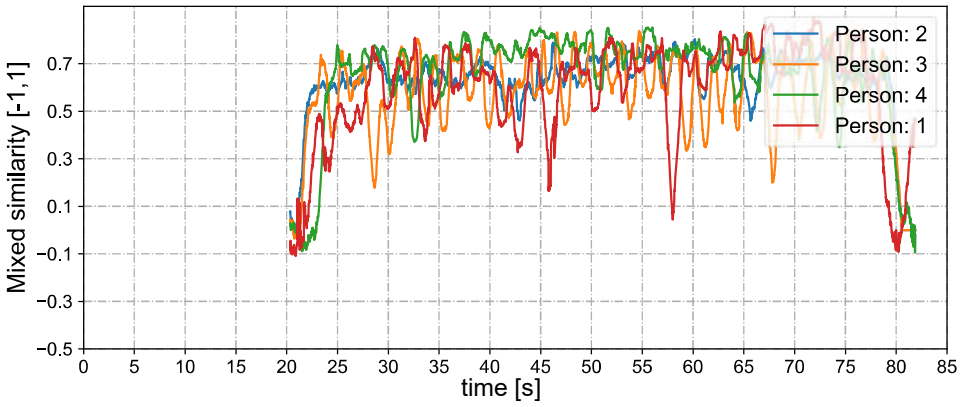


Figure 6.25: Motion-based mixed similarities of identical objects in stereo active vision systems.

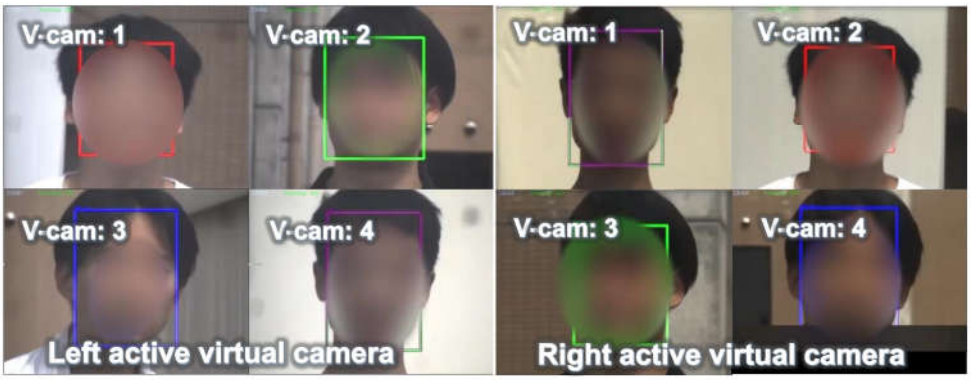


Figure 6.26: Virtual cameras from stereo active vision.

tion of scanning and detection, and four experiment participants initiate their movements. From the 20th second to the 80th second, the participants engage in independent movements for approximately one minute. As shown in Figure 6.25, it shows motion-based similarities of identical objects in stereo active vision systems. It is evident that starting from the 20th second, as different experiment participants initiate their movements, the similarity of motion among multiple identical targets in the active stereo vision system exceeds 0.7. This demonstrates our ability to achieve a high level of accuracy in performing stereoscopic correspondences for multiple moving objects in the active stereo vision system. Figure 6.26 shows multiple virtual cameras from stereo active vision. Rectangles of the same color are used to mark identical characters. In Figure 6.27, we plot the spatial

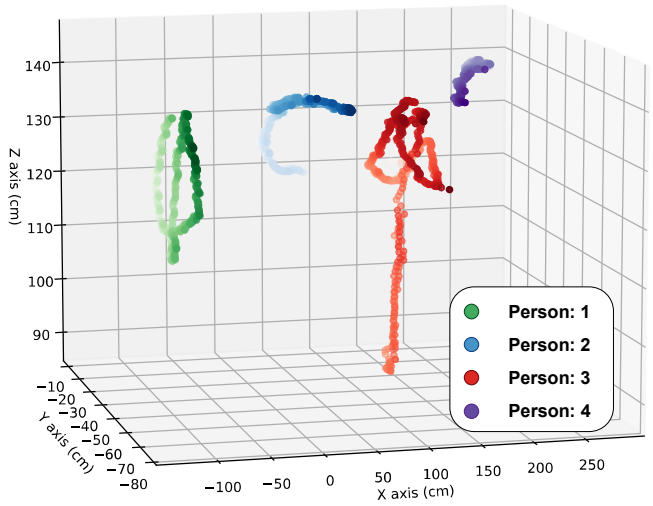


Figure 6.27: Spatial trajectories of multiple moving figures from 44.6 s to 49.8 s.

trajectories of the four experimenters between 44.6 s and 49.8 s. It's pretty obvious if they're squatting or shaking.

6.4 Concluding remarks

In this chapter, we addressed the problem of stereo correspondence of objects with similar appearances. Traditional appearance-based algorithms do not provide effective performance, so we proposed a method that uses highly synchronized motion information to overcome this limitation. Our approach involves using high-synchronous short-term velocities acquired by high-speed vision systems as features for stereo correspondence of moving objects. We demonstrated the effectiveness of our method through experiments on (1) the correspondence of markers for regular motion on a metronome and (2) motion tracking and correspondence of multiple hands in indoor scenes. These experiments confirmed the potential of high-speed vision technology to improve the stereo correspondence of objects with similar appearances. However, our current method cannot provide accurate results when objects are static.

Although our experiments were conducted in fixed camera settings, it is worth not-

ing that the proposed method holds potential for application in active camera systems. Future work could explore the adaptation and optimization of our method specifically for active camera systems and investigate its performance in real-world scenarios.

Chapter 7

Conclusion

This research aims to achieve real-time tracking of multiple targets by using an ultra-high-speed active vision system and combining the motion information of the objects for stereo correspondence. At the same time, to further improve the accuracy of the system, this study also established a mathematical model and developed a set of flexible calibration algorithms for the precise measurement of the spatial position of the ultra-high-speed galvanometer camera system. In this chapter, the main research content of the thesis is reviewed and summarized, and the future research direction is prospected.

First, this study uses an ultra-high-speed active vision system to achieve real-time tracking of multiple targets. Through the multiplexing and fast control algorithm of the active camera, the system can quickly respond to the movement of multiple targets and update the position and trajectory information of the targets in real-time. Furthermore, to deal with the problems of visual occlusion and object re-identification, this paper proposes a strategy based on template matching and appearance model updating to improve the robustness and accuracy of the system. Experiments have proved that we can simultaneously track up to 20 moving objects at a speed of 25 frames per second, or track multiple objects with a moving speed of up to 30m/s. Secondly, this paper also establishes a mathematical model and develops a set of flexible calibration algorithms for the spatial position measurement problem of the ultra-high-speed galvanometer camera sys-

tem. This method is suitable for stereo active vision systems, and the calibrated system can perform more precise spatial positioning. The algorithm can accurately measure the position and angle of the galvanometer camera system according to the actual application requirements, thereby improving the accuracy and reliability of the system measurement. Experiments have proved that the ultra-high-speed active stereo vision system through the calibration algorithm can complete the measurement with an error of fewer than 0.3 centimetres within a range of 7 meters. Last, this paper uses the motion information of the object for stereo correspondence and realizes the 3D position reconstruction of the target through stereo vision technology. This method converts the appearance matching of objects in stereo images into the matching of motion information in stereo images, and the success rate of stereo correspondence for moving objects is as high as 100%. Especially for some similar-looking objects, such as human faces, hands, etc., it is difficult for traditional appearance-based matching algorithms to obtain high accuracy. Our method of using highly synchronized motion information of objects for matching is a general method that can be applied not only to object pixel displacement in fixed cameras but also to voltage displacement in stereo active cameras.

Through extensive experimental verification and performance evaluation, the method in this study achieves remarkable results in real-time object tracking and stereo correspondence. Compared with traditional methods, the algorithm proposed in this paper shows obvious advantages in accuracy, real-time and precise measurement.

There are still some problems to be solved in the future. Our stereo correspondence method based on motion information cannot perform excellent performance when the object is stationary, and future work can consider developing a stereo correspondence method that is easy for appearance and motion information. At the same time, the current modelling of our galvanometer-based ultra-high-speed active camera system can only complete the measurement of spatial points, and it can be expanded later to complete ultra-high-speed stereo reconstruction of multiple objects. In addition, more applica-

tion fields can be explored, such as robot navigation, intelligent transportation, etc., the method can be popularized and applied, and new calibration algorithms and technologies can be further studied to meet the needs of different fields.

Bibliography

[1] I. Ahmed, S. Din, G. Jeon, F. Piccialli, and G. Fortino, "Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1253–1270, 2021.

[2] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: A survey," *Ieee Access*, vol. 8, pp. 2847–2868, 2019.

[3] H. Van Nguyen, H. Rezatofighi, B.-N. Vo, and D. C. Ranasinghe, "Online uav path planning for joint detection and tracking of multiple radio-tagged objects," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5365–5379, 2019.

[4] L. Leal-Taixé, A. Milan, K. Schindler, D. Cremers, I. Reid, and S. Roth, "Tracking the trackers: an analysis of the state of the art in multiple object tracking," *arXiv preprint arXiv:1704.02781*, 2017.

[5] W. Jung, S.-H. Kim, S.-P. Hong, and J. Seo, "An aiot monitoring system for multi-object tracking and alerting," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 337–348, 2021.

[6] Z. Soleimanitaleb, M. A. Keyvanrad, and A. Jafari, "Object tracking methods:a review," in *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2019, pp. 282–288.

[7] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[8] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[9] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," *Computer Vision: A Reference Guide*, pp. 1–9, 2020.

[10] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[13] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," *arXiv preprint arXiv:1603.08029*, 2016.

[14] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5987–5995.

[15] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "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*, 2020, pp. 390–391.

[16] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[17] T. Kattenborn, J. Leitloff, F. Schiefer, and S. Hinz, "Review on convolutional neural networks (cnn) in vegetation remote sensing," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, pp. 24–49, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620303488

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[19] P. Purkait, C. Zhao, and C. Zach, "Spp-net: Deep absolute pose regression with synthetic views," *arXiv preprint arXiv:1712.03452*, 2017.

[20] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[23] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, 2016.

[24] S. Qiao, L.-C. Chen, and A. Yuille, "Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 213–10 224.

[25] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[29] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6568–6577.

[30] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 778–10 787.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[32] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[33] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, "An improved yolov2 for vehicle detection," *Sensors*, vol. 18, no. 12, p. 4272, 2018.

[34] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[35] J. Yu and W. Zhang, "Face mask wearing detection algorithm based on improved yolo-v4," *Sensors*, vol. 21, no. 9, p. 3263, 2021.

[36] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 2778–2788.

[37] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[38] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[39] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, 1999.

[40] Y. Li, J. Zhu *et al.*, "A scale adaptive kernel correlation filter tracker with feature integration." in *ECCV workshops (2)*, vol. 8926. Citeseer, 2014, pp. 254–265.

[41] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2544–2550.

[42] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[43] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[44] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*. Springer, 2020, pp. 107–122.

[45] L. Chen, H. Ai, Z. Zhuang, and C. Shang, "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE international conference on multimedia and expo (ICME)*. IEEE, 2018, pp. 1–6.

[46] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 941–951.

[47] J. Zhang, S. Zhou, X. Chang, F. Wan, J. Wang, Y. Wu, and D. Huang, "Multiple object tracking by flowing and fusing," *arXiv preprint arXiv:2001.11180*, 2020.

[48] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and

P. Luo, "Transtrack: Multiple object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[49] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8844–8854.

[50] L. d. F. Costa, "Comparing cross correlation-based similarities," *arXiv preprint arXiv:2111.08513*, 2021.

[51] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through fsim, ssim, mse and psnr—a comparative study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.

[52] S. Zhao, Y. Wang, Z. Yang, and D. Cai, "Region mutual information loss for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[53] Y. Ye, L. Bruzzone, J. Shan, F. Bovolo, and Q. Zhu, "Fast and robust matching for multimodal remote sensing image registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9059–9070, 2019.

[54] N. Zermi, A. Khaldi, R. Kafi, F. Kahlessenane, and S. Euschi, "A dwt-svd based robust digital watermarking for medical image security," *Forensic science international*, vol. 320, p. 110691, 2021.

[55] L. Yang, H. Su, C. Zhong, Z. Meng, H. Luo, X. Li, Y. Y. Tang, and Y. Lu, "Hyperspectral image classification using wavelet transform-based smooth ordering," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 17, no. 06, p. 1950050, 2019.

[56] R. Pautrat, V. Larsson, M. R. Oswald, and M. Pollefeys, "Online invariance selection for local feature descriptors," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 707–724.

[57] S. Gupta, K. Thakur, and M. Kumar, "2d-human face recognition using sift and surf descriptors of face's feature regions," *The Visual Computer*, vol. 37, pp. 447–456, 2021.

[58] Y. Pang and A. Li, "An improved orb feature point image matching method based on pso," in *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)*, vol. 11069. SPIE, 2019, pp. 224–232.

[59] C. Chengtao and L. Mengqun, "Tire pattern similarity detection based on template matching and lbp," in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*. IEEE, 2019, pp. 419–423.

[60] A. K. Venkataramanan, C. Wu, A. C. Bovik, I. Katsavounidis, and Z. Shahid, "A hitchhiker's guide to structural similarity," *IEEE Access*, vol. 9, pp. 28 872–28 896, 2021.

[61] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, p. 95, 2019.

[62] K. Hernandez-Diaz, F. Alonso-Fernandez, and J. Bigun, "Cross spectral periocular matching using resnet features," in *2019 International Conference on Biometrics (ICB)*, 2019, pp. 1–7.

[63] R. Agarwal and O. P. Verma, "An efficient copy move forgery detection using deep

learning feature extraction and matching algorithm," *Multimedia Tools and Applications*, vol. 79, no. 11-12, pp. 7355–7376, 2020.

[64] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3279–3286.

[65] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4353–4361.

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[67] T.-Y. Yang, J.-H. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "Deepcd: Learning deep complementary descriptors for patch representations," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3334–3342.

[68] H.-M. Hsu, J. Cai, Y. Wang, J.-N. Hwang, and K.-J. Kim, "Multi-target multi-camera tracking of vehicles using metadata-aided re-id and trajectory-based camera link model," *IEEE Transactions on Image Processing*, vol. 30, pp. 5198–5210, 2021.

[69] M. Gou, S. Karanam, W. Liu, O. Camps, and R. J. Radke, "Dukemtmc4reid: A large-scale multi-camera person re-identification dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 10–19.

[70] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu, "Pose flow: Efficient online pose tracking," *arXiv preprint arXiv:1802.00977*, 2018.

[71] H. Su, S. Liu, B. Zheng, X. Zhou, and K. Zheng, "A survey of trajectory distance measures and performance evaluation," *The VLDB Journal*, vol. 29, pp. 3–32, 2020.

[72] L. Zhao and G. Shi, "A novel similarity measure for clustering vessel trajectories based on dynamic time warping," *The Journal of Navigation*, vol. 72, no. 2, pp. 290–306, 2019.

[73] P. Maergner, V. Pondenkandath, M. Alberti, M. Liwicki, K. Riesen, R. Ingold, and A. Fischer, "Combining graph edit distance and triplet networks for offline signature verification," *Pattern Recognition Letters*, vol. 125, pp. 527–533, 2019.

[74] A. Rubinstein and Z. Song, "Reducing approximate longest common subsequence to approximate edit distance," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 1591–1600.

[75] L. Ying, Z. Li, Z. Xiang-mo, and C. Ke, "Effectiveness of trajectory similarity measures based on truck gps data," *China Journal of Highway and Transport*, vol. 33, no. 2, p. 146, 2020.

[76] L. Gong, B. Chen, W. Xu, C. Liu, X. Li, Z. Zhao, and L. Zhao, "Motion similarity evaluation between human and a tri-co robot during real-time imitation with a trajectory dynamic time warping model," *Sensors*, vol. 22, no. 5, p. 1968, 2022.

[77] C. Zhu, J. Yang, Z. Shao, and C. Liu, "Vision based hand gesture recognition using 3d shape context," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 9, pp. 1600–1613, 2019.

[78] C. I. Patel, D. Labana, S. Pandya, K. Modi, H. Ghayvat, and M. Awais, "Histogram of oriented gradient-based fusion of features for human action recognition in action video sequences," *Sensors*, vol. 20, no. 24, p. 7299, 2020.

[79] X. Zhao, Y. Rao, J. Cai, and W. Ma, "Abnormal trajectory detection based on a sparse subgraph," *IEEE Access*, vol. 8, pp. 29 987–30 000, 2020.

[80] X. Liang, H.-B. Zhang, Y.-X. Zhang, and J.-L. Huang, "Jtcr: Joint trajectory character recognition for human action recognition," in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 2019, pp. 350–353.

[81] J. Cao, M. Liang, Y. Li, J. Chen, H. Li, R. W. Liu, and J. Liu, "Pca-based hierarchical clustering of ais trajectories with automatic extraction of clusters," in *2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)*. IEEE, 2018, pp. 448–452.

[82] Z. Xiao, Y. Wang, K. Fu, and F. Wu, "Identifying different transportation modes from trajectory data using tree-based ensemble classifiers," *ISPRS International Journal of Geo-Information*, vol. 6, no. 2, 2017. [Online]. Available: https://www.mdpi.com/2220-9964/6/2/57

[83] M. A. Bagheri, Q. Gao, and S. Escalera, "Support vector machines with time series distance kernels for action classification," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–7.

[84] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 3880–3887.

[85] R. Zhang, P. Xie, H. Jiang, Z. Xiao, C. Wang, and L. Liu, "Clustering noisy trajectories via robust deep attention auto-encoders," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2019, pp. 63–71.

[86] M. Liang, R. W. Liu, S. Li, Z. Xiao, X. Liu, and F. Lu, "An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity

computation," *Ocean Engineering*, vol. 225, p. 108803, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821002389

[87] G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, pp. 77–84, 2010.

[88] G. Loianno, D. Scaramuzza, and V. Kumar, "Special issue on high-speed vision-based autonomous navigation of uavs," *Journal of Field Robotics*, vol. 1, no. 1, pp. 1–3, 2018.

[89] S. Huang, N. Bergström, Y. Yamakawa, T. Senoo, and M. Ishikawa, "Applying high-speed vision sensing to an industrial robot for high-performance position regulation under uncertainties," *Sensors*, vol. 16, no. 8, p. 1195, 2016.

[90] I. Ishii, T. Taniguchi, R. Sukenobe, and K. Yamamoto, "Development of high-speed and real-time vision platform, h3 vision," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3671–3678.

[91] A. Sharma, K. Shimasaki, Q. Gu, J. Chen, T. Aoyama, T. Takaki, I. Ishii, K. Tamura, and K. Tajima, "Super high-speed vision platform for processing 1024×1024 images in real time at 12500 fps," in *2016 IEEE/SICE International Symposium on System Integration (SII)*, 2016, pp. 544–549.

[92] Q. Gu, N. Nakamura, T. Aoyama, T. Takaki, and I. Ishii, "A full-pixel optical flow system using a gpu-based high-frame-rate vision," in *Proceedings of the 2015 Conference on Advances In Robotics*, ser. AIR '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2783449.2783501

[93] K. Kobayashi-Kirschvink and H. Oku, "Design principles of a high-speed omni-

scannable gaze controller," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 836–843, 2016.

[94] S. Hu, K. Shimasaki, M. Jiang, T. Senoo, and I. Ishii, "A simultaneous multi-object zooming system using an ultrafast pan-tilt camera," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9436–9448, 2021.

[95] M. Jiang, K. Shimasaki, S. Hu, T. Senoo, and I. Ishii, "A 500-fps pan-tilt tracking system with deep-learning-based object detection," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 691–698, 2021.

[96] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[97] I. C. Mwamba, M. Morshedi, S. Padhye, A. Davatgari, S. Yoon, S. Labi, M. Hastak *et al.*, "Synthesis study of best practices for mapping and coordinating detours for maintenance of traffic (mot) and risk assessment for duration of traffic control activities," Purdue University. Joint Transportation Research Program, Tech. Rep., 2021.

[98] I. Ghafir, V. Prenosil, J. Svoboda, and M. Hammoudeh, "A survey on network security monitoring systems," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE, 2016, pp. 77–82.

[99] K. N. Qureshi and A. H. Abdullah, "A survey on intelligent transportation systems," *Middle-East Journal of Scientific Research*, vol. 15, no. 5, pp. 629–642, 2013.

[100] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 359–10 366.

[101] K. Rzadca, P. Findeisen, J. Swiderski, P. Zych, P. Broniek, J. Kusmierek, P. Nowak, B. Strack, P. Witusowski, S. Hand *et al.*, "Autopilot: workload autoscaling at google," in *Proceedings of the Fifteenth European Conference on Computer Systems*, 2020, pp. 1–16.

[102] R. Bohush and I. Zakharava, "Robust person tracking algorithm based on convolutional neural network for indoor video surveillance systems," in *Pattern Recognition and Information Processing: 14th International Conference, PRIP 2019, Minsk, Belarus, May 21–23, 2019, Revised Selected Papers 14*. Springer, 2019, pp. 289–300.

[103] R. Kaiser, M. Thaler, A. Kriechbaum, H. Fassold, W. Bailer, and J. Rosner, "Real-time person tracking in high-resolution panoramic video for automated broadcast production," in *2011 Conference for Visual Media Production*, 2011, pp. 21–29.

[104] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics*, vol. 870, pp. 106–120, 2019.

[105] S. Anwar, S. Khan, and N. Barnes, "A deep journey into super-resolution: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020.

[106] G. Carles, J. Downing, and A. R. Harvey, "Super-resolution imaging using a camera array," *Opt. Lett.*, vol. 39, no. 7, pp. 1889–1892, Apr 2014. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?URI=ol-39-7-1889

[107] S. Kang, J.-K. Paik, A. Koschan, B. R. Abidi, and M. A. Abidi, "Real-time video tracking using ptz cameras," in *Sixth International Conference on Quality Control by Artificial Vision*, vol. 5132. SPIE, 2003, pp. 103–111.

[108] C. Ding, B. Song, A. Morye, J. A. Farrell, and A. K. Roy-Chowdhury, "Collaborative sensing in a distributed ptz camera network," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3282–3295, 2012.

[109] Q. Li, M. Chen, Q. Gu, and I. Ishii, "A flexible calibration algorithm for high-speed bionic vision system based on galvanometer," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4222–4227.

[110] Q.-Y. Gu and I. Ishii, "Review of some advances and applications in real-time high-speed vision: Our views and experiences," *International Journal of Automation and Computing*, vol. 13, pp. 305–318, 2016.

[111] K. Okumura, H. Oku, and M. Ishikawa, "High-speed gaze controller for millisecond-order pan/tilt camera," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 6186–6191.

[112] T. Aoyama, L. Li, M. Jiang, K. Inoue, T. Takaki, I. Ishii, H. Yang, C. Umemoto, H. Matsuda, M. Chikaraishi *et al.*, "Vibration sensing of a bridge model using a multithread active vision system," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 179–189, 2017.

[113] S. Hu, K. Shimasaki, M. Jiang, T. Takaki, and I. Ishii, "A dual-camera-based ultrafast tracking system for simultaneous multi-target zooming," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 521–526.

[114] M. Jiang, K. Shimasaki, S. Hu, T. Senoo, and I. Ishii, "A 500-fps pan-tilt tracking system with deep-learning-based object detection," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 691–698, 2021.

[115] L. O. Chua and T. Roska, "The cnn paradigm," *IEEE Transactions on Circuits*

and Systems I: Fundamental Theory and Applications, vol. 40, no. 3, pp. 147–156, 1993.

[116] J.-C. Yoo and T. H. Han, "Fast normalized cross-correlation," *Circuits, systems and signal processing*, vol. 28, pp. 819–843, 2009.

[117] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li, and D. Wen, "Sface: Sigmoid-constrained hypersphere loss for robust face recognition," *IEEE Transactions on Image Processing*, vol. 30, pp. 2587–2598, 2021.

[118] J. M. Ong and L. da Cruz, "The bionic eye: a review," *Clinical & experimental ophthalmology*, vol. 40, no. 1, pp. 6–17, 2012.

[119] X.-y. Wang, Y. Zhang, X.-j. Fu, and G.-s. Xiang, "Design and kinematic analysis of a novel humanoid robot eye using pneumatic artificial muscles," *Journal of Bionic Engineering*, vol. 5, no. 3, pp. 264–270, 2008.

[120] Y.-B. Bang, J. K. Paik, B.-H. Shin, and C.-K. Lee, "A three-degree-of-freedom anthropomorphic oculomotor simulator," *International Journal of Control, Automation, and Systems*, vol. 4, no. 2, pp. 227–235, 2006.

[121] S. Hu, Y. Matsumoto, T. Takaki, and I. Ishii, "Monocular stereo measurement using high-speed catadioptric tracking," *Sensors*, vol. 17, no. 8, p. 1839, 2017.

[122] P. Eisert, K. Polthier, and J. Hornegger, "A mathematical model and calibration procedure for galvanometric laser scanning systems," in *Vision, Modeling, and Visualization*, 2011, pp. 207–214.

[123] T. Wissel, B. Wagner, P. Stüber, A. Schweikard, and F. Ernst, "Data-driven learning for calibrating galvanometric laser scanners," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5709–5717, 2015.

[124] B. Wagner, P. Stüber, T. Wissel, R. Bruder, A. Schweikard, and F. Ernst, "Accuracy analysis for triangulation and tracking based on time-multiplexed structured light," *Medical Physics*, vol. 41, no. 8Part1, p. 082701, 2014.

[125] C. Yu, X. Chen, and J. Xi, "Modeling and calibration of a novel one-mirror galvanometric laser scanner," *Sensors*, vol. 17, no. 1, p. 164, 2017.

[126] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[127] M. S. Hamid, N. Abd Manap, R. A. Hamzah, and A. F. Kadmin, "Stereo matching algorithm based on deep learning: A survey," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1663–1673, 2022.

[128] J. A. Oroko and G. Nyakoe, "Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: a review," in *Proceedings of the Sustainable Research and Innovation Conference*, 2022, pp. 314–318.

[129] C. Liu, C. Xing, Q. Hu, S. Wang, S. Zhao, and M. Gao, "Stereoscopic hyperspectral remote sensing of the atmospheric environment: Innovation and prospects," *Earth-Science Reviews*, vol. 226, p. 103958, 2022.

[130] N. Schlinkmann, R. Khakhar, T. Picht, S. K. Piper, L. S. Fekonja, P. Vajkoczy, and G. Acker, "Does stereoscopic imaging improve the memorization of medical imaging by neurosurgeons? experience of a single institution," *Neurosurgical Review*, vol. 45, no. 2, pp. 1371–1381, 2022.

[131] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual slam algorithms," *Robotics*, vol. 11, no. 1, p. 24, 2022.

[132] H. Shabanian and M. Balasubramanian, "A novel factor graph-based optimization

technique for stereo correspondence estimation," *Scientific Reports*, vol. 12, no. 1, p. 15613, 2022.

[133] R. A. Hamzah, M. N. Z. Azali, Z. M. Noh, M. Zahari, and A. I. Herman, "Development of depth map from stereo images using sum of absolute differences and edge filters," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 875–883, 2022.

[134] Q. Chang, A. Zha, W. Wang, X. Liu, M. Onishi, L. Lei, M. J. Er, and T. Maruyama, "Efficient stereo matching on embedded gpus with zero-means cross correlation," *Journal of Systems Architecture*, vol. 123, p. 102366, 2022.

[135] F. Wang and L. Ding, "Object recognition and localization based on binocular stereo vision," in *Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics*, 2022, pp. 196–201.

[136] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[137] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1592–1599.

[138] H. Laga, L. V. Jospin, F. Boussaid, and M. Bennamoun, "A survey on deep learning techniques for stereo-based depth estimation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 4, pp. 1738–1764, 2020.

[139] M. Kaya and H. Ş. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, no. 9, p. 1066, 2019.

[140] P. Köhl, A. Specker, A. Schumann, and J. Beyerer, "The mta dataset for multi target multi camera pedestrian tracking by weighted distance aggregation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 4489–4498.

[141] P. Li, J. Zhang, Z. Zhu, Y. Li, L. Jiang, and G. Huang, "State-aware re-identification feature for multi-target multi-camera tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[142] Y. He, X. Wei, X. Hong, W. Shi, and Y. Gong, "Multi-target multi-camera tracking by tracklet-to-target assignment," *IEEE Transactions on Image Processing*, vol. 29, pp. 5191–5205, 2020.

[143] N. Magdy, M. A. Sakr, T. Mostafa, and K. El-Bahnasy, "Review on trajectory similarity measures." Institute of Electrical and Electronics Engineers Inc., 2 2016, pp. 613–619.

[144] Q. Li, M. Chen, Q. Gu, and I. Ishii, "A flexible calibration algorithm for high-speed bionic vision system based on galvanometer," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4222–4227.

Acknowledgment

First of all, I would like to thank my advisor, Prof. **Idaku Ishii**. Thank you for your guidance and support throughout your doctoral studies. Your professional knowledge, deep academic insight and selfless dedication have played a vital role in my research. I have benefited a lot from your careful guidance on my personal and academic development.

Second, I would like to thank Prof. **Qingyi Gu**. Thanks for your recommendation, I just had the opportunity to study abroad for my Ph.D. Due to the impact of COVID-19, I cannot enter Japan to study. You gave me the opportunity to continue to stay in the Institute of Automation, Chinese Academy of Sciences to conduct doctoral research and guide my research. **Dr. Mengjuan Chen**, **Dr. Jianquan Li**, **Dr. Xianlei Long**, and **Mr. He Jiang** from the Institute of Automation also gave me a lot of help in academic and life during this period.

Additionally, I would like to express my gratitude to **Dr. Kohei Shimasaki** and **Dr. Shaopeng Hu**. They helped me overcome the unfamiliarity with a new experimental environment when I joined our laboratory. I would like to express my heartfelt gratitude to **Ms. Yukari Kaneyuki** and **Ms. Michiko Kanzaki** (educational administrator). They were my most reliable staff in our institution; I received thoughtful attention both in my study.

Here, I would like to express my sincere thanks to my classmates and lab members, **Dr. Feiyue Wang**, **Dr. Wenxiang Qin**, **Mr. Kotaro Fujita**, **Mr. Shun Yozima**, **Mr.**

Junhao Li, **Mr. Ziyuan Meng**, **Mr. Jiahua Wang**, **Mr. Kohei Masatsugu**, and **Mr. Takuto Ogata** . Thank you for your help and cooperation in experimental design, data acquisition and analysis. Your professional knowledge and team spirit make my research richer and more meaningful. I would also like to thank the **China Scholarship Council**, without your support, my research would not have been possible.

Finally, I want to express my profound gratitude to my family and girlfriend **Ms. Huimin Han** for their support and warm encouragement until now throughout my life.

May, 2023

Qing Li