electronics

MDPI

Article

Cooperative People Tracking by Distributed Cameras Network

Yi-Chang Wu [1], Ching-Han Chen [2,*], Yao-Te Chiu [2] and Pi-Wei Chen [3]

[1] Forensic Science Division, Investigation Bureau, Ministry of Justice, New Taipei 231209, Taiwan; shintenwu@gmail.com
[2] Department of Computer Science and Information Engineering, National Central University, Taoyuan 320317, Taiwan; ted24821@gmail.com
[3] Department of English, Wenzao-Ursuline University of Languages, Kaohsiung 807679, Taiwan; chenpiway@gmail.com
* Correspondence: pierre.miat@gmail.com

Abstract: In the application of video surveillance, reliable people detection and tracking are always challenging tasks. The conventional single-camera surveillance system may encounter difficulties such as narrow-angle of view and dead space. In this paper, we proposed multi-cameras network architecture with an inter-camera hand-off protocol for cooperative people tracking. We use the YOLO model to detect multiple people in the video scene and incorporate the particle swarm optimization algorithm to track the person movement. When a person leaves the area covered by a camera and enters an area covered by another camera, these cameras can exchange relevant information for uninterrupted tracking. The motion smoothness (MS) metrics is proposed for evaluating the tracking quality of multi-camera networking system. We used a three-camera system for two persons tracking in overlapping scene for experimental evaluation. Most tracking person offsets at different frames were lower than 30 pixels. Only 0.15% of the frames showed abrupt increases in offsets pixel. The experiment results reveal that our multi-camera system achieves robust, smooth tracking performance.

Keywords: multi-camera; cooperative tracking; PSO; gait recognition

check for updates

Citation: Wu, Y.-C.; Chen, C.-H.; Chiu, Y.-T.; Chen, P.-W. Cooperative People Tracking by Distributed Cameras Network. *Electronics* **2021**, *10*, 1780. https://doi.org/10.3390/electronics10151780

Academic Editors: Soon Ki Jung and Sajid Javed

Received: 30 June 2021
Accepted: 23 July 2021
Published: 25 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Video surveillance has attracted widespread attention in the field of biometrics in recent years. Because gait is a biometric feature that can be effectively recognized at a long distance and without attention. In real applications, appearance changes due to changes in viewing angles or walking directions are one of the main difficulties in gait recognition. This is because people can walk freely in any direction within the field of view of a single camera. When people walk out of one camera and then enter another camera, in addition to continuously capturing the same target, the angle of view will also change. When the angle of view is changed, the accuracy of character tracking and gait recognition will be greatly reduced.

This problem can be solved by integrating multiple views of information from multiple cameras [1]. However, in reality, the development and deployment of multiple views system is a challenging task.

When using a single fixed camera, the foreground person in the image is detected as the region-of-interest (ROI), then continue to track [2]. In some works, they also used YOLO to track the person in their camera vision [3]. For a multicamera system, person tracking is much more complex. Many issues may have to be considered, like as: multicamera spatial arrangement [4], FOV overlapping cameras [5,6] or non-overlapping cameras [7,8], whether to use pan-tilt-zoom (PTZ) cameras [9,10], how to perform cooperative multicamera tracking when the target leaves the coverage of a camera [11,12] and the method of passing tracked object feature between different camera [13,14].

In this paper, we present a distributed camera network for cooperative gait recognition. With this camera network, the progressive background modeling, robust foreground detection and for multi-target tracking by particle swarm optimization (PSO) algorithms are designed for real-time robust person identification.

2. Materials and Methods

2.1. Cameras Network

A robust gait recognition system by distributed cameras network relies on moving object detection and tracking. Firstly, the system detects persons by performing preprocessing on the newly received image to determine the number of moving persons. Then, the system tracks the moving persons by creating a template of these persons and running a tracking algorithm to determine the person's locations. Finally, the system performs gait recognition. Through cooperative tracking of distributive cameras, the system ensures continuous reliable person identification when a person in question leaves an area covered by a camera and enters an area covered by another camera.

A cameras network would model frame-to-frame correspondence, correspondence in overlapping or non-overlapping camera views and track the feature identity when a person leaves/re-enter camera view. The topology structure of the camera's network can be one of the following types in Figure 1 in real cameras network deployment.

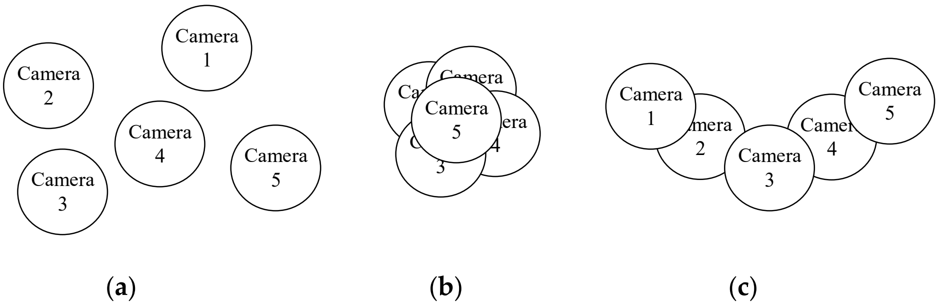
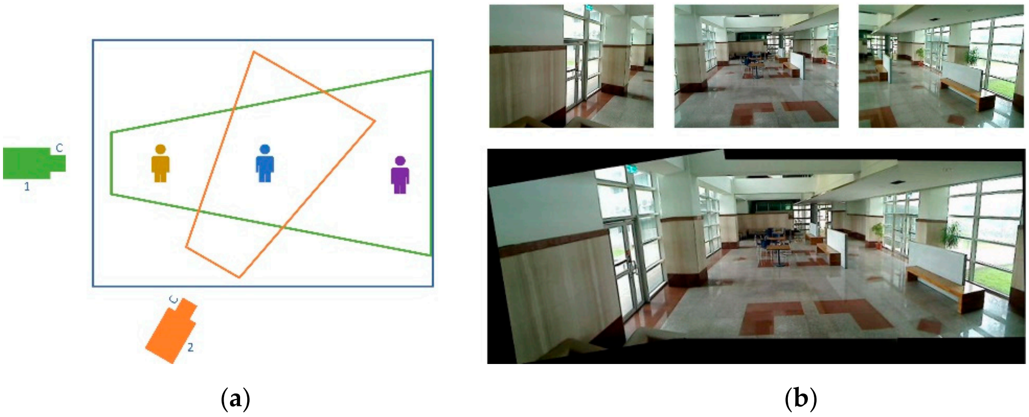


Figure 1. Different topology of cameras network (**a**) no region of overlap between any cameras (**b**) each camera overlapping with each other (**c**) each camera overlapping with one or two cameras.

Using a camera network to track the target object has two advantages. First, the depth information of images can be retrieved to solve the problem of the target being occluded. Another advantage is that the coverage of multiple cameras is wider than that of a single camera. The tracking mode of the camera network is roughly divided into three types: (1) Object tracking of images in different hours, (2) object tracking of overlapped FOVs and (3) object tracking of nonoverlapped FOVs.

Techniques such as graph generation [15] and person reidentification model [16] can help the system construct 3D models without manual calibration and the models can be used to facilitate multi-camera object tracking in overlapped FOVs (Figure 2). All the aforementioned approaches are to calculate the absolute position of a detected object and adopt the resultant absolute position as a feature to determine whether the detected object is the target.

In the aforementioned multicamera tracking methods, object tracking is performed using the overlapped images captured by fixed cameras. Benito-Picazo et al. [17] use fixed cameras and PTZ cameras to capture overlapped images for object tracking. The methods use a surveillance system that employs both static and active cameras with panning, tilting and zooming functions. The most noticeable difference between this type of multicamera tracking technique and other tracking techniques is that the multicamera system can rotate and enlarge surveillance images by using active cameras. Thus, the multicamera system can more effectively monitor the objects of interest in a surveillance environment and achieve comprehensive active surveillance.



(**a**) (**b**)

Figure 2. Multicameras scene model (**a**) FOV lines and projections (**b**) Graph Generation.

Because of the deployment complexity and resource constraints, the camera network-based tracking method with highly overlapping FOVs cannot be realized easily in reality. The cameras network we designed was aimed to expand view coverage instead of overlapping FOVs. Therefore, in the proposed camera network-based tracking method, we must strive to extract the information of the extracted object, namely time, velocity, motion path and object features, to obtain the relative positions of the monitored object captured by different cameras. This study also focused on developing an object tracking approach in which the FOVs of multiple cameras are partially overlapped to ensure continuous and robust tracking.

The setup of multiple cameras influences the extent to which the coverage areas are overlapped. For example, if the areas monitored by cameras are mostly overlapped, the collected depth information of the target object can be relatively accurate; however, data redundancy is also likely. By contrast, if the areas covered by cameras do not overlap with one another [18], the scenario is similar to that of using a single fixed camera. To find a balance between the two aforementioned setups, we use a setup with partially overlapping coverage areas (Figure 3).

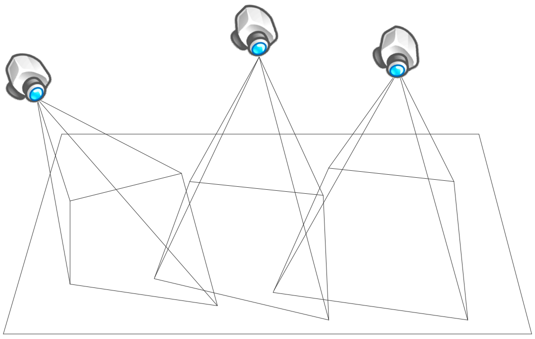


Figure 3. Cameras Network with partially overlapping coverage areas.

The setup we chose has two advantages. When coverage areas of cameras are overlapped, depth information can be obtained. Moreover, the FOVs for the coverage areas can be expanded. Differing from most multicamera systems that require high levels of overlap to accurately set parameters and construct 3D models, this study focused on expanding the view coverage of the designed system. Therefore, construction of 3D scenes and depth information were not required. The system only has to draw demarcations for coverage areas that were overlapped in a binary image. This indicated that the system obtains labeled regions of cameras by simply extracting corner features and matching those features. This study used corner features to acquire the information of overlapped images. To determine a corner feature, the difference between the brightness of the central detection pixel (x, y)

and that of the neighboring pixels was calculated. Figure 4 shows the corner detection in different camera views.

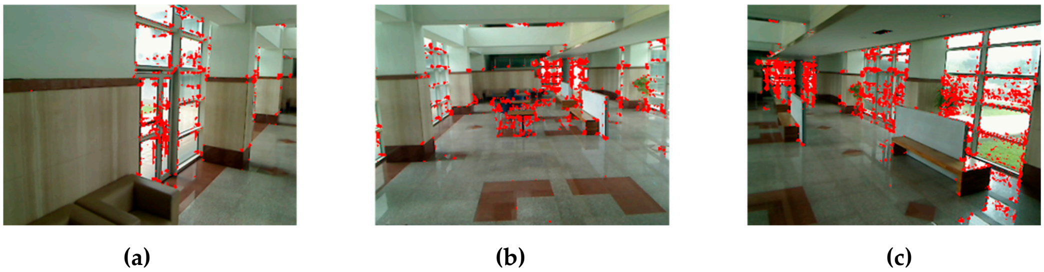


Figure 4. Extract Corner Features from background images. (**a**) Vision of the camera capturing left hand side. (**b**) Vision of the camera capturing the front. (**c**) Vision of the camera capturing right hand side.

2.1.1. Moving Person Detection

During the person detection part, besides some traditional methods like foreground object extraction to obtain the moving object, we can also use a neural network to do the same work. Because these devices will be applied to the edge, they will be constrained by the limited power and low performance. Therefore, we focus on some neural networks such as Mobilenet and Tinier-YOLO which can be performed well on the edge side. We take Tinier-YOLO [19] to introduce the method of feature extraction and Figure 5 shows the structure of Tinier-YOLO.

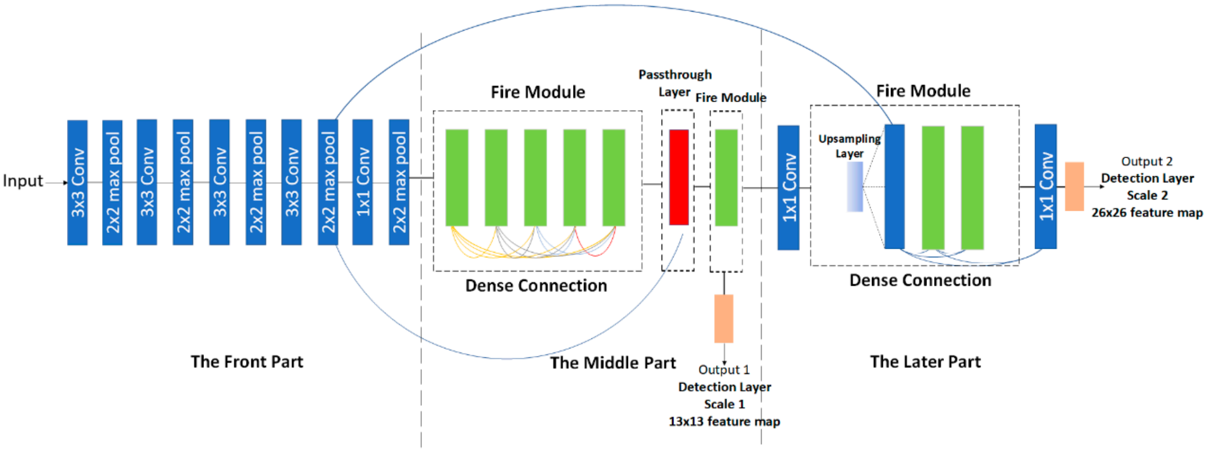


Figure 5. The structure of Tinier-YOLO.

Tinier-YOLO came from Tiny-YOLO-v3; when the image enter the network, it will go through three parts. The first part of Tinier-YOLO is the same as Tiny-YOLO-v3 and the remaining part will replace the bottleneck layer of the original network into the Fire Module. This is to solve the performance downgrading because of too many convolutional layer in Tiny-YOLO-v3 without lowering the accuracy. The Fire Module in the middle part will connect to each other to pass feature and process parameters compression. In addition, it will combine the feature map with the one obtained from last Fire Module in order to get a bounding box result at the detection layer of Output1. Furthermore, the Fire Module in the later part will combine the feature map obtained from middle part to get the result of fine-grained feature.

Fire Module is a convolutional network based on SqueezeNet; it is comprised of the squeeze part and the expand part to compress and expand the data, respectively. Squeeze part will replace the 3 × 3 convolutional layer with 1 × 1 one to improve performance without lowering too much accuracy. Moreover, the expand part will use 1 × 1 and 3 × 3 convolutional layer to compute the result from squeeze part and then combined them to be the final output of the Fire Module.

Fire Module uses dense connections between each other to construct a feed-forward network. This kind of structure can let the later part of the convolutional layer reuse the feature generated by the front part and reduce the resource we need. By applying this technique, we can deploy Tinier-YOLO to our edge devices. When the Fire Module in the middle part finishes its computation, it will go through a Passthrough Layer. In this part, we output the training result into a 13 × 13 feature map and then upsample it by 2× to feed the data to the later Fire Module. At last, we send the final feature map to the PSO module to process person tracking. The result will be like Figure 6.

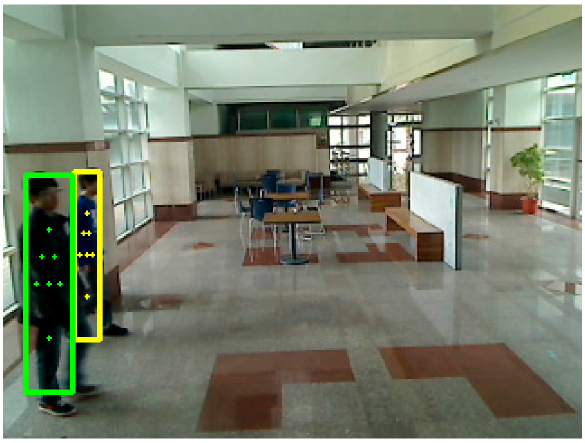


Figure 6. The Bounding Box of Person Detection.

2.1.2. Person Tracking

After completing object detection, the system collected the template features of the extracted moving object. The system searched for the target object and identified the target's position among a continuous series of images. We ran the aforementioned PSO algorithm [20] to detect and track the moving object. Figure 7 exhibits the process of object tracking.

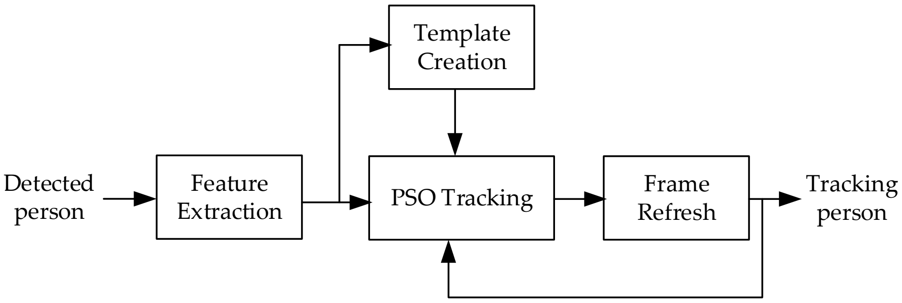


Figure 7. Person Tracking.

When performing PSO-based tracking, particles were randomly distributed in an image during initialization. The system then determined the optimal solution according to the positions of current particles and corrected the positions for subsequent iterations. Multiplying the number of particles by the frequency of iterations yields the number of comparisons. In this study, we performed 10 iterations on 50 particles to achieve particle convergence, which was equivalent to 500 comparisons. Compared with universal search, PSO can perform more efficient searches and comparisons.

To acquire efficient and accurate search results, particles were not distributed throughout the entire image once the position of the target was determined. Instead, the particles were distributed in a limited manner around the current target. By utilizing the objects movement continuity, the system distributed particles within the predicted target range to more efficiently track the target object.

2.2. Cooperative Tracking

When a moving object is about to leave the area covered by a camera, ensuring continuous and stable tracking is a crucial goal of a cooperative multicamera tracking system.

This paper presents a novel approach to transferring tracking tokens from camera to camera that entails a three-step process for achieving cooperative multicamera tracking and is detailed as follows.

Step 1: Entrance of a moving object into the demarcation area

We establish that the camera monitoring and tracking the moving object has the tracking token. When the object enters the demarcation area (i.e., the overlapping area between the camera with token and another camera), the camera with token calls another camera and informs it that the object is about to enter its FOV.

Step 2: Detection of the moving object

After the moving object enters the demarcation area (i.e., overlap area) and another camera is called, the system extracts the current foreground object and establishes a template model. Subsequently, the system determines whether this foreground object is the moving object tracked by the previous camera according to time features (i.e., the time that the object enters the demarcation region). The system determines this using the following formula:

$$C(\mathrm{cam1},\ \mathrm{cam2}) = \begin{cases} 1, T(obj,\ cam1) = T(obj,\ cam2) \\ 0,\ otherwise \end{cases} \tag{1}$$

$$L(\mathrm{D},\ \mathrm{C}) = \begin{cases} 1,\ \ C = 1\ and\ D < \varepsilon \\ 0,\ otherwise \end{cases} \tag{2}$$

We assume that *T(obj, cam no.)* is the time that the moving object enters the demarcation area and *C(cam no., cam no.)* is the result that depends on whether the times detected by two cameras are the same. This rule-of-time feature determination is incorporated with the difference of color feature vectors *D*. A difference lower than the threshold indicates high similarity between the detected objects. *L(*D, C) is used to verify whether the currently detected object is the previously detected moving object following the time feature *C* and color feature of the moving object.

Step 3: Transfer of tracking tokens

Following from Step2, if the system determines that the currently detected object is the moving object detected by the previous camera (Camera 1), the previous camera will transfer the tracking token to the current camera (Camera 2). The system can thus continue to track the object in question. Conversely, if the system finds that the two detected objects are different, it identifies the currently detected object as a new moving object. Camera 2 then begins to track this object and owns the tracking token for it. The cooperative tracking protocol between cameras is depicted in Figure 8.

2.3. Gateway

In order to manage all of distributed cameras and perform person tracking and detection, we design a system as Figure 9 to get the image data and run the tracing task on it.

The gateway is composed of gateway master node and AI hardware acceleration nodes. When the camera finished image capture, it will send the data to the gateway. The master node has two functions: distribute data to idle AI hardware acceleration nodes and establish connection between each camera to let them communicate with the gateway. Firstly, the master node receives data flow and check which deep learning worker node is in the idle state, then master node will forward data to the idle node. When the computation of deep learning worker nodes is completed, they send the result back to gateway and the gateway will keep this feature in order to track the same object between different cameras.

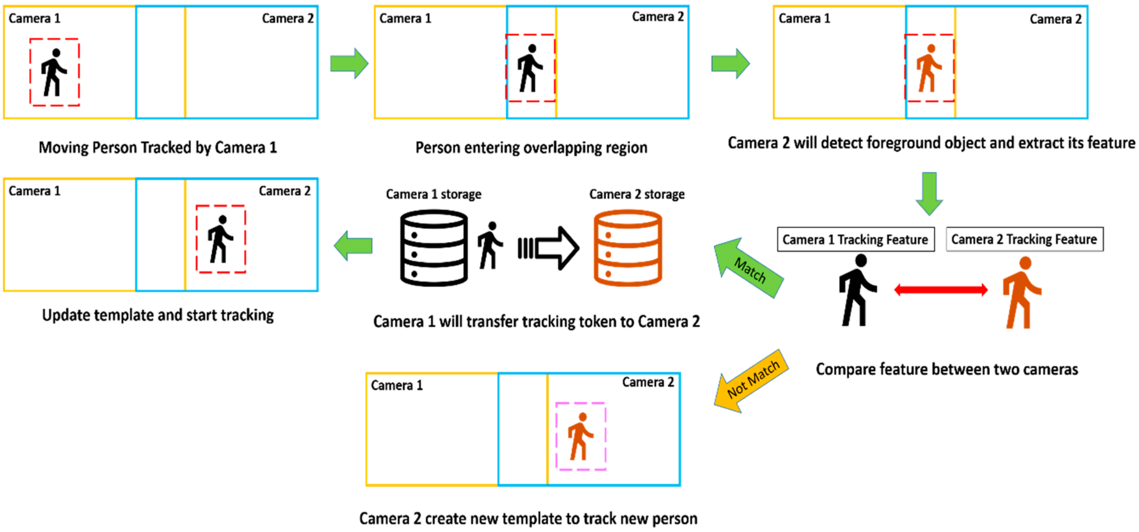

Camera 1
Camera 2
Moving Person Tracked by Camera 1
Person entering overlapping region
Camera 2 will detect foreground object and extract its feature
Camera 1 Tracking Feature
Camera 2 Tracking Feature
Compare feature between two cameras
Match
Not Match
Camera 1 storage
Camera 2 storage
Camera 1 will transfer tracking token to Camera 2
Update template and start tracking
Camera 2 create new template to track new person

Figure 8. Cooperative Tracking between cameras.

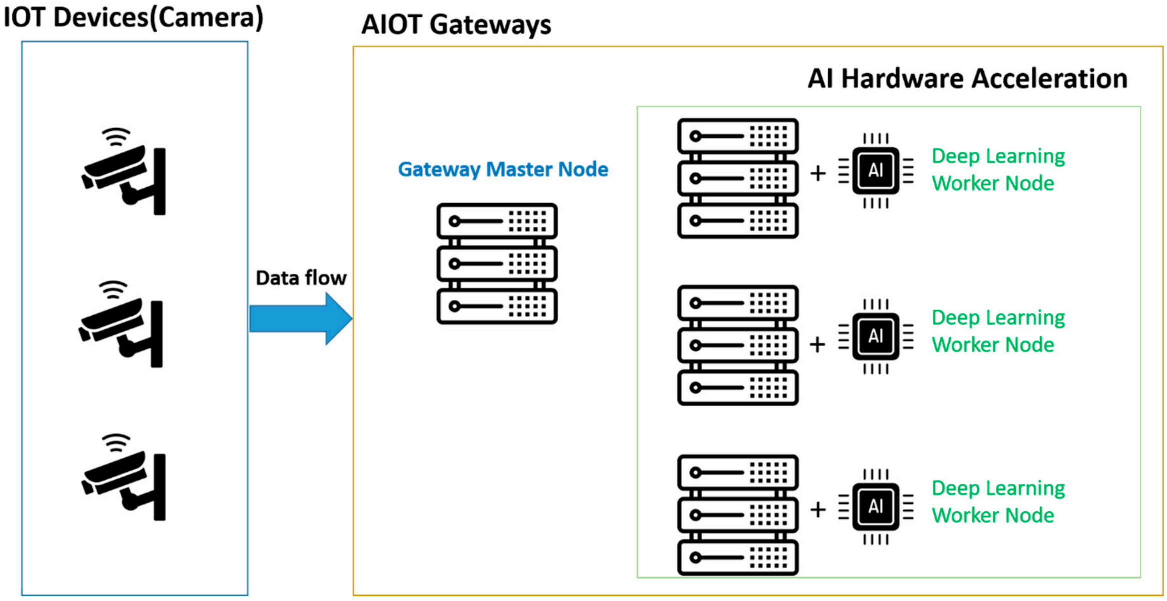


Figure 9. Structure of the system.

3. Results

During the experiment, we use raspberry pi camera module v2 as our IoT camera. In addition, we set these modules working at 720p resolution, to perform our real time image capturing task.

The analysis and accuracy test was conducted on the developed cooperative tracking algorithm. The non-ground-truth (NGT) measure motion smoothness (MS) [21] was employed to evaluate tracking accuracy. MS evaluates the tracking accuracy of a system by measuring the offset of the tracked object and the formula is as follows [22]:

$$\mathrm{MS(t)} = \mathrm{d_E(T(t),\ T(t-\Delta))} = \sqrt{\mathrm{(x(t)-x(t-\Delta))^2 + (y(t)-y(t-\Delta))^2}} \quad (3)$$

where d_E denotes the Euclidean distance between the positions of the moving object being tracked over two consecutive frames, T(t) is the position of the tracked object at t hour and represents the time interval. A high value of MS indicates that the moving distance of the tracked object has experienced great changes, whereas a low MS value indicates a small offset. The offset of a person is normally not large. Accordingly, when the MS curve abruptly lifts or drops, a tracking error is likely. Another explanation for an abrupt increase or decrease is that the object actually has a large offset. Conversely, a smooth MS curve indicates a small offset, revealing that the system is stably tracking the object.

In the resultant curve diagram, the x-axis represents the image frame, which can also be regarded as timeline and the y-axis represents the tracking quality. MS measurement was used to calculate the offsets of tracked objects.

A smooth curve without abrupt lift or drop indicates that the tracked object has a small offset and that the tracking is stably in progress. We assessed the quality of multicamera tracking by creating scenarios of one person, two persons and two persons with overlapped. When the moving object entered the demarcation area, another camera communicated with the previous camera and took over the tracking token through a protocol to continue tracking and analyzing the detected object. Additionally, when the object in question entered the demarcation area, we observed whether the mechanism of cooperative communication was initiated between cameras and whether the tracking token was transferred to achieve continuous tracking.

3.1. Tracking of One Person

The images in Figure 10 present the scenes where a person walked through the coverage of Camera 1, Camera 2 and Camera 3 and then returned to the coverage of Camera 2. This scenario was designed to assess the performance of the tracking system in a simple situation. Figure 11. presents the evaluation of the tracking results. The results in Figure 12 showed that the curve was mostly steady until the pixel values suddenly soared at Frame nos. 260~275 under the coverage of Camera 3. We extracted these frames for further analysis as Figure 11.

According to the results depicted in Figure 11, the system successfully tracked the person between Frame 260 and Frame 275, but the search boxes were not in optimal positions, which caused pixel values to abruptly increase. Regarding cooperative multicamera tracking, the pictures in 12 depict the target (Object 1) entering the demarcation area at Frame no. 57. Camera 2 detected the new moving object and labeled it as Object 2. However, according to the proposed cameras network protocol, the system determined that Object 2 was actually the object detected by Camera 1. Therefore, Camera 1 passed the tracking token to Camera 2 at Frame no. 58 and Camera 2 changed the label of the target from Object 2 to Object 1 and continued the tracking work.

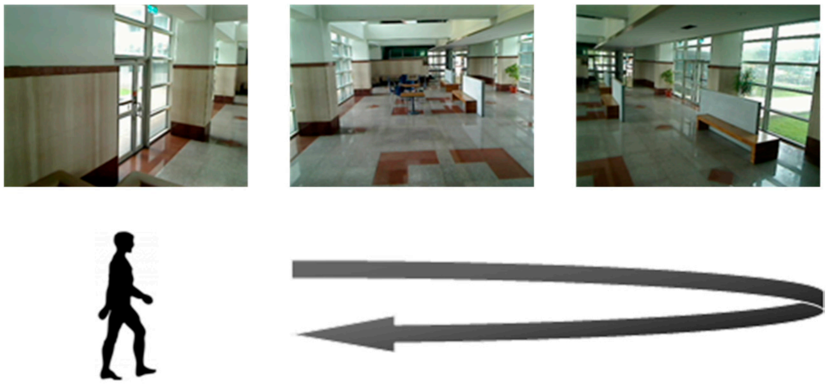


Figure 10. Three cameras view for one person tracking.

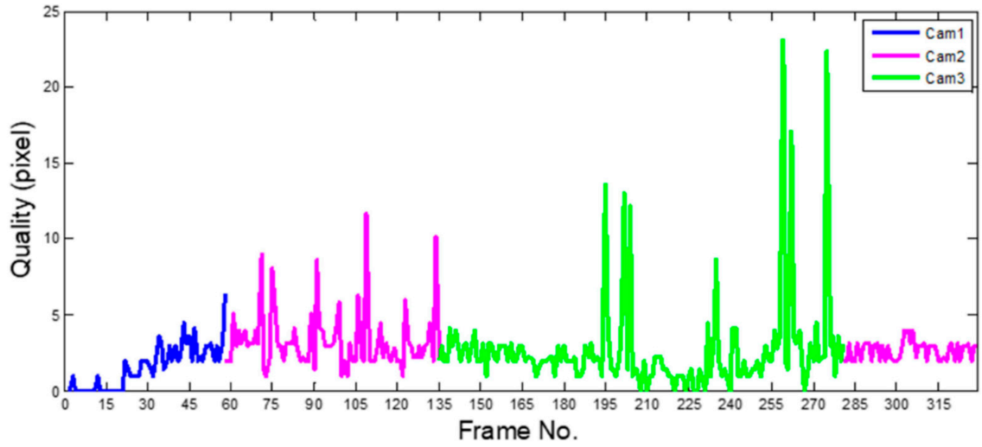
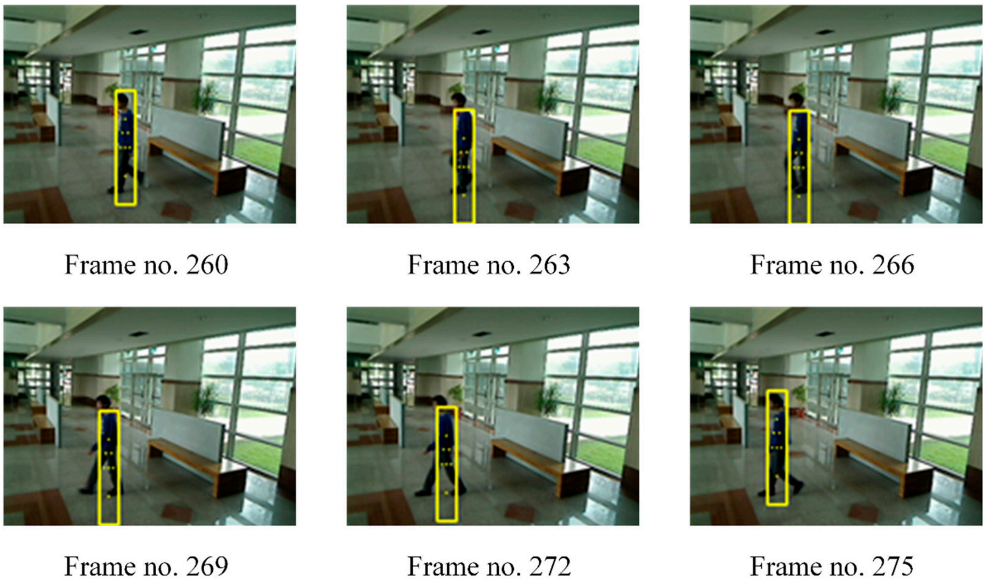


Figure 11. Evaluate single person tracking using MS metrics.



Frame no. 260 Frame no. 263 Frame no. 266

Frame no. 269 Frame no. 272 Frame no. 275

Figure 12. Frame no. 260~275 from Cam3 in single person tracking scene.

3.2. Tracking of Two People

Figure 13 presents the scenes where two peoples separately started their journeys from the scenes covered by Camera 1 and Camera 3 (Object 1 from Camera 1 and Object 2 from Camera 3). Object 1 was the tracked object being occluded (i.e., when the two peoples passed each other, Object 1 was occluded by Object 2). This scenario was created to assess whether the designed algorithm can accurately track an occluded object.

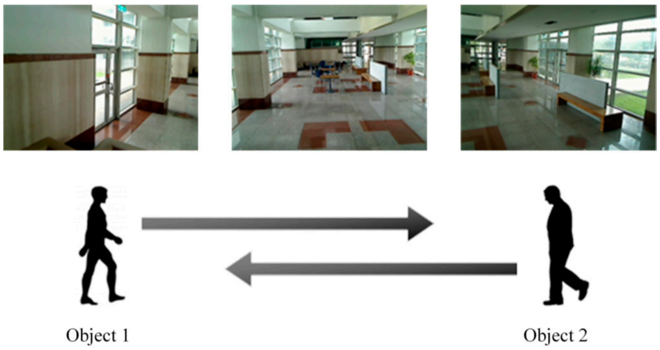
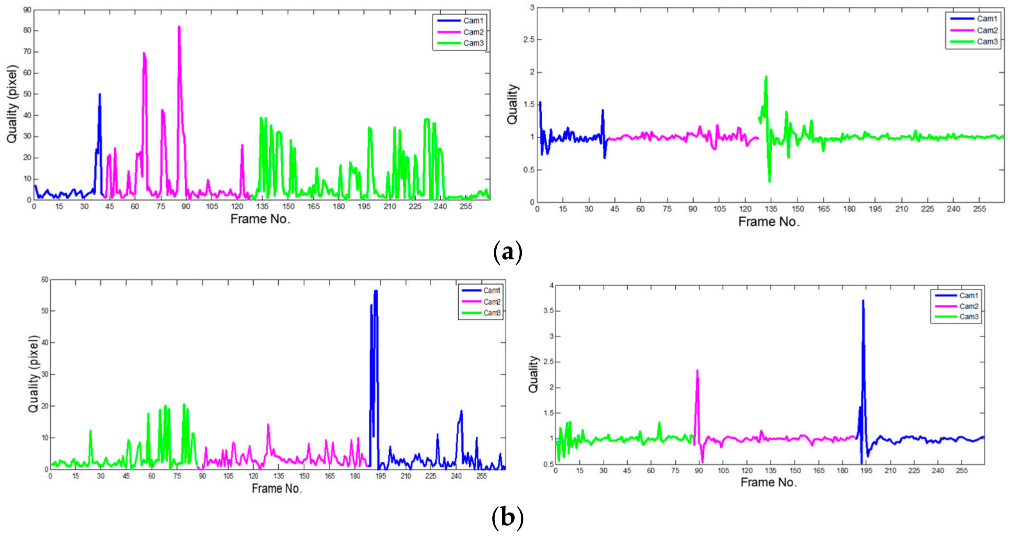


Figure 13. Three cameras view for two peoples tracking.

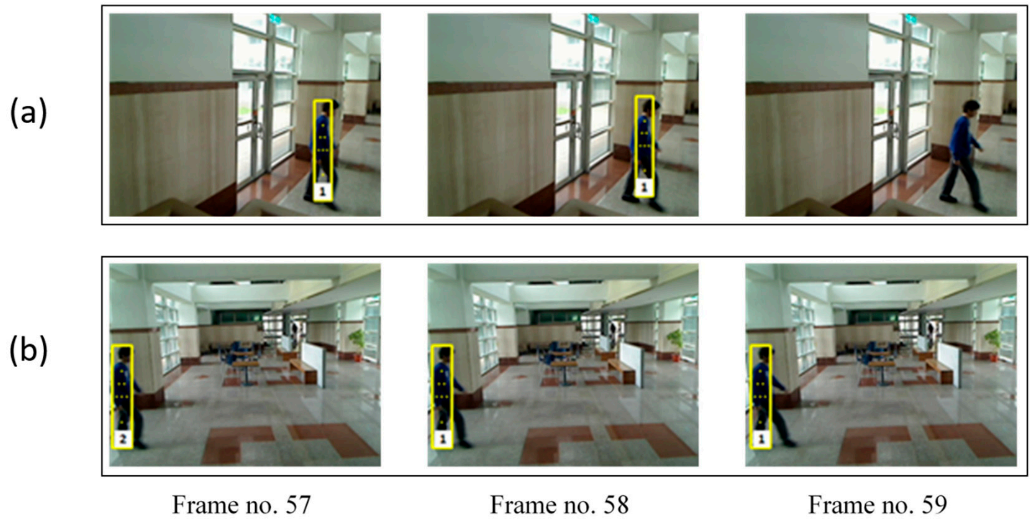
In the experiment, the system tracked two people. When the system tracked two people, the assessed algorithm could maintain its tracking accuracy compared with single people tracking. Figure 14a,b left picture exhibits the MS curves of Objects 1 and 2, respectively. Most object offsets at different frames were lower than 30 pixels. Only several frames showed abrupt increases in pixel values. For example, Figure 15a depicts an abrupt increase at Frame no. 61 and Frame no. 90.



(**a**)

(**b**)

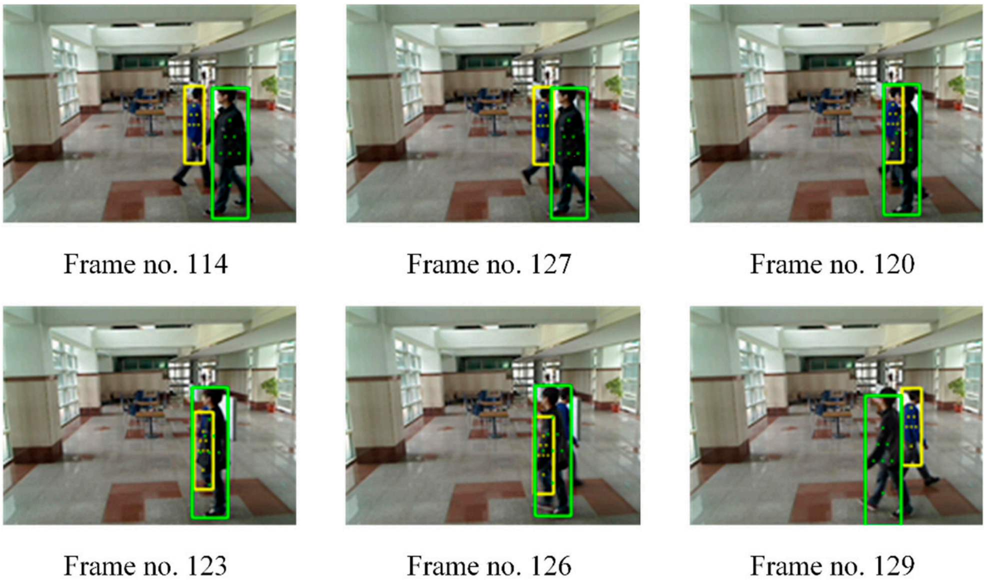
Figure 14. (**a**) MS metrics (**left**) and FIT metrics(**right**) of People 1 tracking result in two camera scenes; (**b**) MS metrics and FIT metrics of People 2 tracking result in two camera scenes.



Frame no. 57 Frame no. 58 Frame no. 59

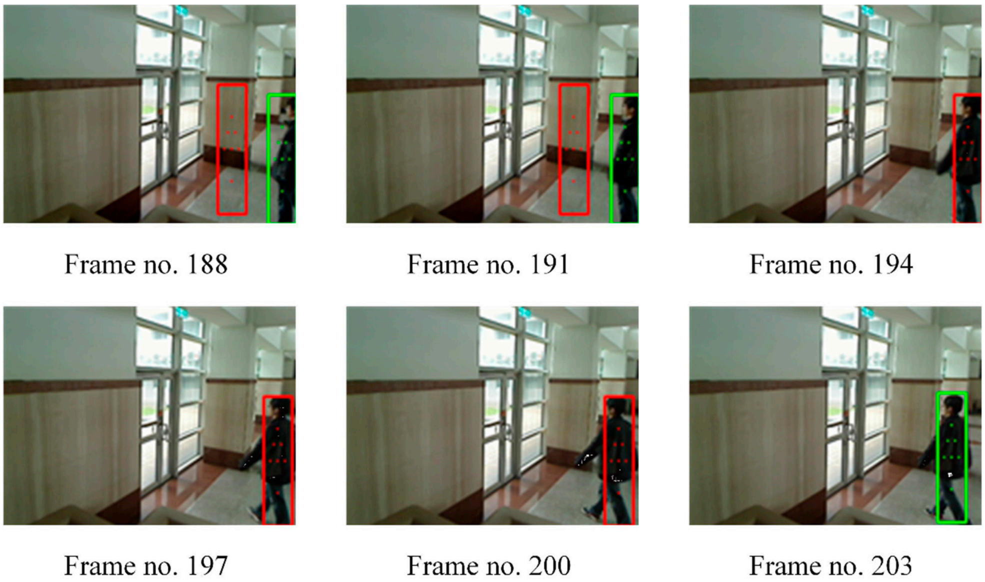
Figure 15. Tracking token transfer between Cam1 and Cam2 in single person tracking scene.

As shown in Figure 16, from Frame no. 114 to Frame no. 129, Object 1 (in the yellow box) and Object 2 (in the green box) passed each other under the coverage of Camera 2, during which time Object 1 was occluded by Object 2. After the people separated, the cameras again captured their targets. The MS curve of Object 2 (Figure 15b) abnormally rose at Frame no. 193. To examine this situation, we analyzed the images of relevant frames (Figure 17) and discovered that the abnormal increase occurred when the target had just entered the surveillance area (entering the area from the edge). Under such a circumstance, the image of the object was incomplete and caused a tracking error. The system then corrected the error, thus leading to an abrupt offset increase. Figure 18 shows the tracking privilege transfer between cameras. Regarding cooperative tracking of cameras network, the people detected by Camera 2 was labeled as Object 1, but Camera 3 labelled this people as Object 5 at Frame no. 128. At this moment, the privilege transfer protocol allowed Camera 3 to take over the tracking privilege from Camera 2, fix the error and change the label of this People back to Object 1 at Frame no. 129.


Frame no. 114 Frame no. 127 Frame no. 120

Frame no. 123 Frame no. 126 Frame no. 129

Figure 16. (Frame no. 114~129) from Cam2 in two People tracking scene.


Frame no. 188 Frame no. 191 Frame no. 194

Frame no. 197 Frame no. 200 Frame no. 203

Figure 17. Object2: Frame no. 188~203 from Cam1 in two people tracking scene.

3.3. *Tracking of Two Persons in Overlapped Scene*

Figure 19 presents the scenario where two persons jointly set off from the scene covered by Camera 1 and reached the scene of Camera 3 with different walking speeds. Because Object 1 was on the left of Object 2 (i.e., Object 1 was farther from the cameras compared with Object 2), it was sometimes occluded by Object 2. This scenario was created to assess the tracking performance of the designed algorithm in a unique situation.

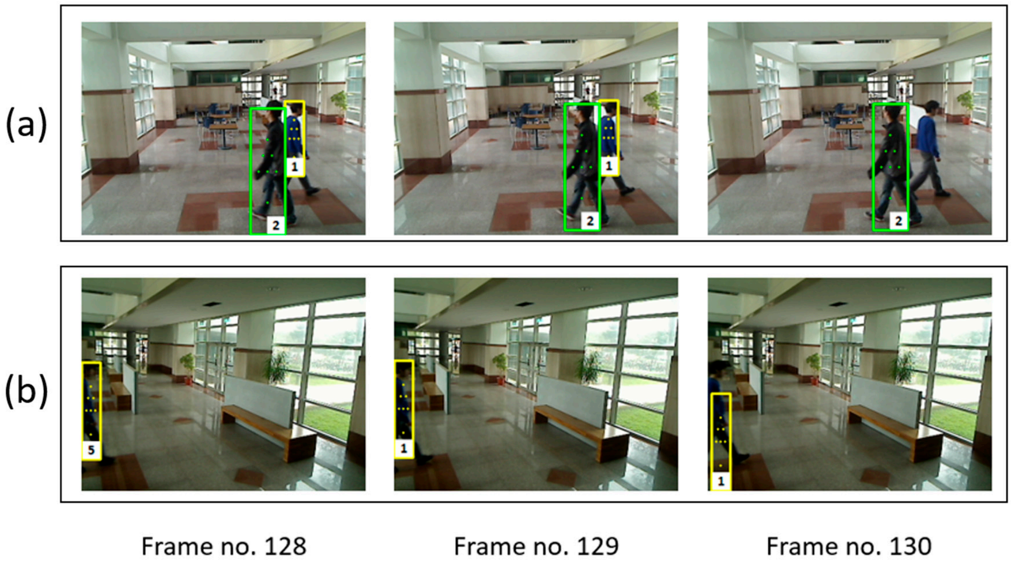


Figure 18. Two people privilege transfer between cameras. (**a**) Person 1 heading right and keep walking until enter different camera vision in (**b**). (**b**) The camera in (**a**) will relay information of person 1 to camera in (**b**), so the camera in (**b**) can keep tracking.

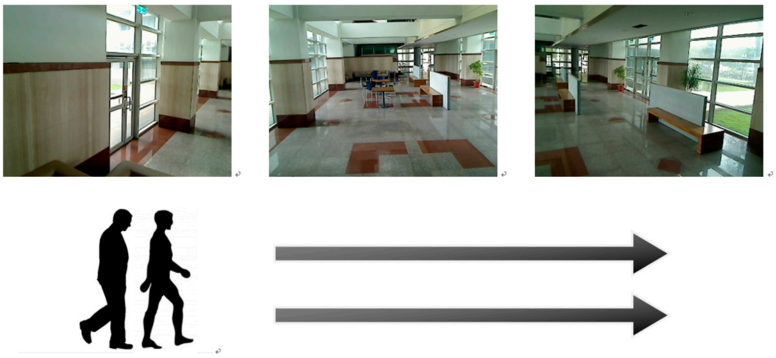


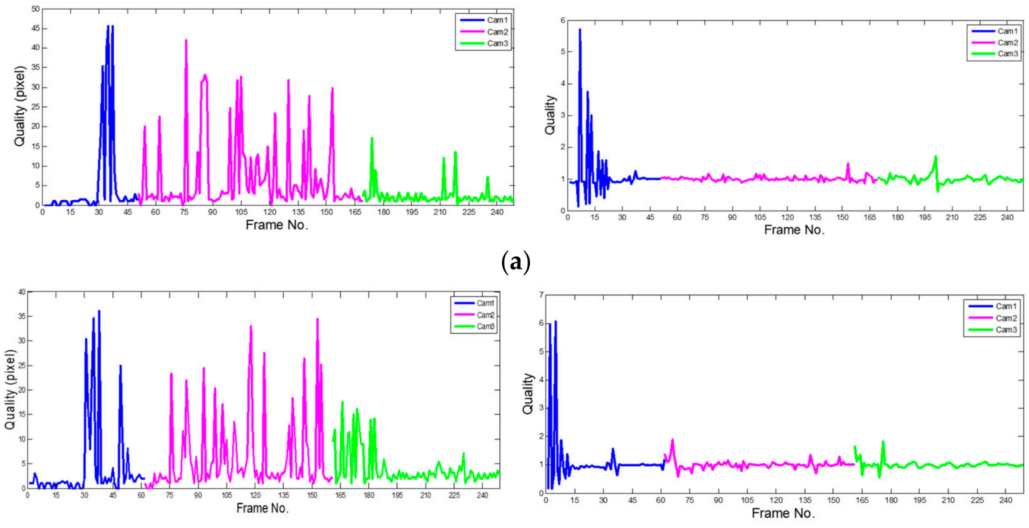
Figure 19. Two persons tracking in overlapping scene.

This test image shows two people starting from Cam1 at the same time and walking to Cam3 at different speeds and Object1 is a tracked object standing behind and obscured (Object2 standing in front).

Because of the occlusion effect, the occluded target exhibited greater MS curve fluctuations in the evaluation. Specifically, the curves in Figures 19 and 20 left exhibit more sudden increases compared with the MS curves in the aforementioned single person and two-persons situations.

In the experiment, the system tracked two persons. When the system tracked two persons, the assessed algorithm could maintain its tracking accuracy compared with single person tracking. Figure 20a,b exhibit the MS curves of Objects 1 and 2, respectively. Most object offsets at different frames were lower than 30 pixels. Only 0.15% of frames showed abrupt increases in offset pixel.

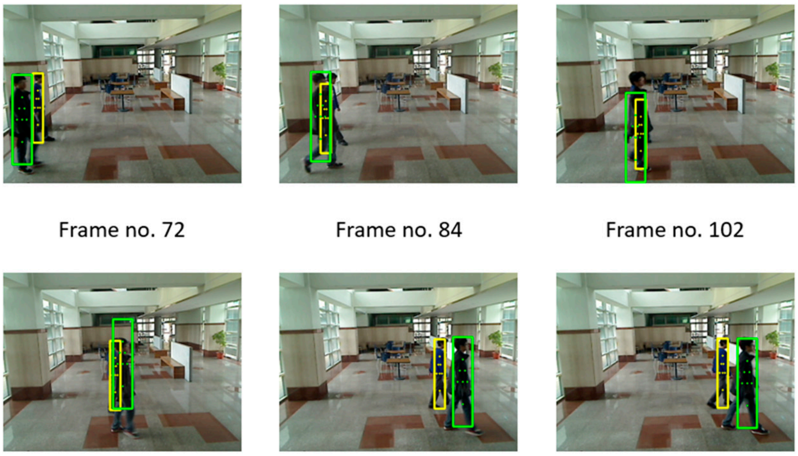
When the two persons were walking parallel to each other, the person occluded by his counterpart could not be accurately tracked because most of his body was occluded for a while. Nevertheless, once the occlusion was gone (i.e., the two persons separated), we could again detect the positions of these two persons. Although the system using the PSO-based algorithm could not accurately identify the position of an occluded target, it continued to identify and track the target when the occlusion disappeared. Even the human eye has difficulty conforming that an occluded object is the target. In this scenario, one can only recognize the target by using logical cognition to analyze the positions of an object in a continuous series of images. Figure 21 presents the results of a scenario where one person was occluded by another (Object 1 is in the yellow box and Object 2 is in the green box).



(**a**)

(**b**)

Figure 20. MS metrics (**left**) FIT metrics (**right**) for two persons tracking in overlapping scene (**a**) person 1; (**b**) person 2.



Frame no. 72 Frame no. 84 Frame no. 102

Frame no. 125 Frame no. 156 Frame no. 161

Figure 21. Two persons in overlapped transfer between cameras.

4. Conclusions

In this paper, we proposed a cameras network architecture and the hand-off protocol for cooperative people tracking and also designed a reliable and highly efficient algorithm that enables real-time people detection and tracking. YOLO object model is used to detect the peoples in the video scene, when a person leaves the FOV area covered by a camera and enters an FOV area covered by another camera, these cameras can exchange relevant information for uninterrupted tracking. By incorporating a particle swarm optimization algorithm, our system can perform robust inter-camera tracking using multiple cameras system. The motion smoothness (MS) metrics is proposed for evaluating the tracking quality of multi-camera networking system. For experience, we used a three-camera system for two persons tracking in overlapping scene. Most tracking person offsets at different frames were lower than 30 pixels. Only 0.15% of frames showed abrupt increases in offsets pixel. The experiment results reveal that the approach we adopted achieves superior tracking performance.

Author Contributions: Y.-C.W. works for dataset collection and method. C.-H.C. was involved in system design and experiment. Y.-T.C. and P.-W.C. were participated in paper writing and language editing. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the Investigation Bureau, Ministry of Justice, R.O.C., grants No. 109-1301-05-17-02.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Ethics Committee of National Taiwan University Behavior and Social Science Research Ethics Committee (protocol code 202007ES034, 18 August 2020).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, J.; Peng, K. A Multi-View Gait Recognition Method Using Deep Convolutional Neural Network and Channel Attention Mechanism. *Comput. Modeling Eng. Sci.* **2020**, *125*, 345–363. [CrossRef]
2. Ding, L.; Wang, Y.; Laganière, R.; Huang, D.; Luo, X.; Zhang, H. A robust and fast multispectral pedestrian detection deep network. *Knowl. Based Syst.* **2021**, *227*, 106990. [CrossRef]
3. Huang, Q.; Hao, K. Development of CNN-based visual recognition air conditioner for smart buildings. *J. Inf. Technol. Constr.* **2020**, *25*, 361–373. [CrossRef]
4. Zhang, X.; Izquierdo, E. Real-Time Multi-Target Multi-Camera Tracking with Spatial-Temporal Information. In Proceedings of the 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 1–4 December 2019; pp. 1–4.
5. Benrazek, A.-E.; Farou, B.; Seridi, H.; Kouahla, Z.; Kurulay, M. Ascending hierarchical classification for camera clustering based on FoV overlaps for WMSN. *IET Wirel. Sens. Syst.* **2019**, *9*, 382–388. [CrossRef]
6. Seok, H.; Lim, J. Rovo: Robust omnidirectional visual odometry for wide-baseline wide-fov camera systems. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6344–6350.
7. Sonbhadra, S.K.; Agarwal, S.; Syafrullah, M.; Adiyarta, K. Person tracking with non-overlapping multiple cameras. In Proceedings of the 2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI), Yogyakarta, Indonesia, 1–2 October 2020; pp. 137–143.
8. Narayan, N.; Sankaran, N.; Setlur, S.; Govindaraju, V. Learning deep features for online person tracking using non-overlapping cameras: A survey. *Image Vis. Comput.* **2019**, *89*, 222–235. [CrossRef]
9. Chapel, M.-N.; Bouwmans, T. Moving objects detection with a moving camera: A comprehensive review. *Comput. Sci. Rev.* **2020**, *38*, 100310. [CrossRef]
10. Bisagno, N.; Xamin, A.; De Natale, F.; Conci, N.; Rinner, B. Dynamic Camera Reconfiguration with Reinforcement Learning and Stochastic Methods for Crowd Surveillance. *Sensors* **2020**, *20*, 4691. [CrossRef] [PubMed]
11. He, L.; Liu, G.; Tian, G.; Zhang, J.; Ji, Z. Efficient Multi-View Multi-Target Tracking Using a Distributed Camera Network. *IEEE Sens. J.* **2020**, *20*, 2056–2063. [CrossRef]
12. Cai, Z.; Hu, S.; Shi, Y.; Wang, Q.; Zhang, D. Multiple Human Tracking Based on Distributed Collaborative Cameras. *Multimed. Tools Appl.* **2017**, *76*, 1941–1957. [CrossRef]
13. Huang, Q.; Rodriguez, K.; Whetstone, N.; Habel, S. Rapid Internet of Things (IoT) prototype for accurate people counting towards energy efficient buildings. *J. Inf. Technol. Constr.* **2019**, *24*, 1–13. [CrossRef]
14. Jain, S.; Zhang, X.; Zhou, Y.; Ananthanarayanan, G.; Jiang, J.; Shu, Y.; Bahl, V.; Gonzalez, J. Spatula: Efficient cross-camera video analytics on large camera networks. In Proceedings of the ACM/IEEE Symposium on Edge Computing (SEC 2020), San Jose, CA, USA, 12–14 November 2020.
15. Esterle, L.; Lewis, P.R.; Yao, X.; Rinner, B. Socio-Economic vision graph generation and handover in distributed smart camera networks. *ACM Trans. Sens. Netw. (TOSN)* **2014**, *10*, 1–24. [CrossRef]
16. Tang, X.; Sun, X.; Wang, Z.; Yu, P.; Cao, N.; Xu, Y. Research on the Pedestrian Re-Identification Method Based on Local Features and Gait Energy Images. *CMC-Comput. Mater. Contin.* **2020**, *64*, 1185–1198. [CrossRef]
17. Benito-Picazo, J.; Dominguez, E.; Palomo, E.J.; Lopez-Rubio, E.; Ortiz-de-Lazcano-Lobato, J.M. Motion detection with low cost hardware for PTZ cameras. *Integr. Comput. Aided Eng.* **2019**, *26*, 21–36. [CrossRef]
18. Zhao, D.; Wang, H.; Yin, H.; Yu, Z.; Li, H. Person re-identification by integrating metric learning and support vector machine. *Signal Process.* **2020**, *166*, 107277. [CrossRef]
19. Fang, W.; Wang, L.; Ren, P. Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments. *IEEE Access* **2020**, *8*, 1935–1944. [CrossRef]
20. Eberhart; Shi, Y. Particle swarm optimization: Developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Korea, 27–30 May 2001; Volume 1, pp. 81–86.

21. Yang, J.; Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.* **2015**, *42*, 181–213. [CrossRef]
22. SanMiguel, J.C.; Cavallaro, A.; Martinez, J.M. Evaluation of on-line quality estimators for object tracking. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; pp. 825–828.