sensors

MDPI

*Article*

# Lightweight Indoor Multi-Object Tracking in Overlapping FOV Multi-Camera Environments

**Jungik Jang, Minjae Seon and Jaehyuk Choi ***

School of Computing, Gachon University, 1342 Seongnam-daero, Sujeong-gu, Seongnam-si 13120, Korea; jji4449@gachon.ac.kr (J.J.); ddol0225@gachon.ac.kr (M.S.)
* Correspondence: jchoi@gachon.ac.kr

**Abstract:** Multi-Target Multi-Camera Tracking (MTMCT), which aims to track multiple targets within a multi-camera network, has recently attracted considerable attention due to its wide range of applications. The main challenge of MTMCT is to match local tracklets (i.e., sub-trajectories) obtained by different cameras and to combine them into global trajectories across the multi-camera network. This paper addresses the cross-camera tracklet matching problem in scenarios with partially overlapping fields of view (FOVs), such as indoor multi-camera environments. We present a new lightweight matching method for the MTMC task that employs similarity analysis for location features. The proposed approach comprises two steps: (i) extracting the motion information of targets based on a ground projection method and (ii) matching the tracklets using similarity analysis based on the Dynamic Time Warping (DTW) algorithm. We use a Kanade–Lucas–Tomasi (KLT) algorithm-based frame-skipping method to reduce the computational overhead in object detection and to produce a smooth estimate of the target's local tracklets. To improve matching accuracy, we also investigate three different location features to determine the most appropriate feature for similarity analysis. The effectiveness of the proposed method has been evaluated through real experiments, demonstrating its ability to accurately match local tracklets.

**Keywords:** multi-camera tracking; multi-target tracking; global tracklet matching; Dynamic Time Warping; KLT algorithm



**Citation:** Jang, J.; Seon, M.; Choi, J. Lightweight Indoor Multi-Object Tracking in Overlapping FOV Multi-Camera Environments. *Sensors* **2022**, *22*, 5267. https://doi.org/10.3390/s22145267

Academic Editors: Ran Liu, Pik Lik Billy Lau and Jianwen Huo

Received: 15 June 2022
Accepted: 11 July 2022
Published: 14 July 2022



## 1. Introduction

Multi-Target Multi-Camera Tracking (MTMCT) has recently received considerable attention due to the growing demand for intelligent monitoring and surveillance systems. It aims to track multiple interested targets and infer a complete trajectory for each target across a multiple-camera network [1]. MTMCT can be applied to various tasks such as video surveillance [1–3], city-scale traffic management [4], smart buildings [5,6], and in-store customer analysis [7].

Owing to the rapid development of object detection techniques [8–11], most state-of-the-art MTMCT methods employ a two-phase pipeline, namely detection-and-tracking [12], to focus on the tracking functionality. In the first phase, they detect targets using a modern object detector and generate sets of local trajectories for each detected target within a single camera. To track targets within the entire multi-camera network, the cross-camera tracklet matching is performed in the next phase, which matches local tracklets across all the cameras to generate a complete trajectory for each target [1].

However, it is a challenging task, as these methods are inherently prone to camera field-of-view (FOV) issues, such as occlusion, i.e., the blind areas of camera views, and/or a significant change in the visual appearance of moving targets. As a result, the trajectory of each target generated within each camera is easily divided into multiple local tracklets, i.e., sub-trajectories. In addition, the tracker can generate incorrect duplicate local tracklets for the same target, which makes the cross-camera tracklet matching problem even more

challenging. The cross-camera global tracklet matching task exhibits high computational complexity, and thus, it may not be available for real-time applications.

In this paper, we tackle the cross-camera tracklet matching problem in MTMCT, which focuses on the multi-camera association of local trajectories from each camera in overlapping FOV multi-camera environments. To this end, we present an MTMC framework that takes videos from multiple cameras and generates global trajectories for targets using light-weight similarity analysis based on the Dynamic Time Warping (DTW) algorithm. Our system consists of three major components: (1) Multi-Object Tracker (Section 3.2), (2) Ground Projector (Section 3.3), and (3) Global Trajectory Mapper (Section 3.4). As we will elaborate in Section 3, the first component, Multi-Object Tracker, considers an input video clip from each camera in the network and generates a set of local image information such as target ID, bounding boxes, and locations. Then, Ground Projector processes local tracklets to generate features suitable for similarity analysis by projecting the tracklets to the ground or bird's-eye views. Moreover, we alleviate the inborn position error caused by camera distortion by exploiting the moving direction information of the target, instead of the absolute position of the target, for similarity analysis. In order to effectively predict the moving direction of the target and reduce the computational cost, we utilize a Kanade–Lucas–Tomasi (KLT) algorithm-based [13,14] frame-skipping method that uses only one frame per n-frames instead of using all the frame information. In addition, we have created the attribute sequence of each target using three different features, namely, scalar, vector, and unit vector, to perform similarity analysis. Finally, Global Trajectory Mapper analyzes similarity based on the DTW algorithm and generates a complete global trajectory and ID for each target across the multiple cameras. We demonstrate the accuracy of our approach through extensive real experiments, where we constructed FOV environments by installing multiple cameras (up to four cameras) in various places, including classrooms and day-care centers. The evaluation results have shown that our approach is highly accurate in matching local tracklets and tolerant to noises with a low computational overhead.

We summarize the main contributions of this study as follows:

(1) Introduction of a new MTMC framework for overlapping FOV multi-camera networks (Section 3.1).
(2) Proposal of a lightweight global tracklet matching algorithm based on DTW similarity analysis (Section 3.4).
(3) Investigation of several movement features of targets to generate sequence sets for similarity analysis (Section 3.3).
(4) Implementation and evaluation of a prototype of the proposed framework with extensive real experiments (Section 4).

The rest of the paper is organized as follows: Section 2 discusses the related work and illustrates the preliminaries. Section 3 details the proposed framework and its main components. Section 4 demonstrates the evaluation results. Section 5 concludes the study.

## 2. Related Work

### *2.1. Multi-Target Multi-Camera Tracking (MTMCT)*

The MTMCT task aims to infer the perfect path for each target in multi-camera environments. It usually comprises two steps: (1) generating local tracklets of all targets within each camera, and (2) matching the local tracklets for the same global target across a multi-camera network. In recent years, studies on the MTMCT task based on various approaches have been actively conducted.

Fleuret et al. [15] proposed a solution that utilizes a probabilistic occupancy map (POM) to approximate the probabilities of occupancy and combines it with the usual color and motion attributes. Berclaz et al. [16] optimized the MTMCT task by employing the POM and K-Shortest Path (KSP) algorithm. Hu et al. [17] and Eshel and Moses [18] presented a matching algorithm for the same target across multiple cameras using the homography correspondence. Similarly, the proposed method employs homography; however, it uses the DTW algorithm to match global IDs. Hou et al. [19] presented a new approach focusing

on local neighboring data matching using a Locality Aware Appearance Metric (LAAM) composed of a metric network. Bredereck et al. [20] matched local tracklets of all cameras using the greedy matching association method. As an example of solving the MTMCT task using hierarchical clustering, Zhang et al. [21] proposed an approach that uses the distance matrix between averaged Re-ID features and applies re-ranking [22] to cluster local tracklets. Jiang et al. [23] solved the problem of trajectory association under orientation variations and occlusions, and they improved the matching efficiency using camera topology. Xu et al. [24] presented an approach using a hierarchical composition model for MTMCT. They re-formulated MTMCT as a composition structure optimization problem. He et al. [1] obtained the tracklet-TID assignment matrix with the Restricted Non-negative Matrix Factorization (RNMF) algorithm and used it to match the tracklet to the target ID (TRACTA). You et al. [25] used Optical-based Pose Association (OPA) for MTMCT and solved the occlusion problem using local pose matching. In addition, the distance problem caused by fast motion was reduced by applying optical flow. Wu et al. [26] proposed a three-step cooperative tracking method to track people in a multi-camera environment through tracking token transfer. Zhang et al. [27] proposed an online (real-time) tracking framework, and they improve the cross-camera person recall performance through appearance and spatial–temporal features. The MTMCT task has been widely used to target vehicles and humans. Hsu et al. [28] proposed the vehicle MTMCT framework, Trajectory-based Camera Link Model (TCLM), through which spatial–temporal information is obtained and MTMCT performance is improved by reducing the Re-ID candidate search process.

### *2.2. Multi-Object Tracking (MOT)*

Multi-Object Tracking (MOT), which is the first step among the process steps of MTMCT described above, can be viewed as a problem of tracking multiple objects in a single camera. MOT aims to estimate and associate a bounding box and ID for a number of objects appearing in an image. As our study utilizes the object information (tracklet) of local image data, which are generated by MOT, the accuracy of the proposed method relies on the MOT performance. The MOT task can be divided into two methods, namely detection-by-tracking and tracking-by-detection, depending on how detection results are used. The tracking-by-detection method has recently attracted attention with the advent of high-performance object detection models. Moreover, it can be categorized into two approaches: (1) a batch tracking method in which data are correlated using the entire data frame information and (2) an online tracking method in which data are correlated using past and current frame information. SORT was proposed by Bewley et al. [8]; it is a popular method that uses online tracking, predicts the tracklet position for a new frame using a Kalman filter [29] and correlates the data using a Hungarian algorithm [30]. DeepSORT [9] supplements the ID switching problem caused by occlusion, which is a problem in SORT, with Deep Appearance Descriptor, and it enables more accurate tracking with the cascated matching strategy. FastMOT [31] used in our study solves the bottleneck caused by the use of DeepSORT's two-stage tracker by running the detector and feature extractor every specific frame. In addition, motion compensation makes it possible to track objects with a moving camera. Most MOT-related studies deal with outdoor tracking such as video surveillance and autonomous driving; however, there is a clear difference from indoor environments. Therefore, Liu et al. [32] presented the depth-enhanced tracking-by-detection (DET) framework optimized for the indoor environment where occlusion frequently occurs. ByteTrack [11], which has recently achieved state-of-the-art results in the field of MOT, dramatically improves performance by associating a detection box with an object with a low detection score. ByteTrack achieves the highest performance, but we wanted to take advantage of FastMOT's skip function, which enhances the frame processing speed. In addition, there are various MOT methods [10,33–45]. In the experimental stage, the accuracy of global ID matching is measured by changing the skip parameter value. In the proposed framework, the MOT model can be replaced by other models.

## 3. System Design

This section presents an overview of the proposed system and its main components.

### *3.1. System Architecture*

Figure 1 illustrates the high-level overview of our proposed system. We consider a multi-camera network comprising $M$ cameras. The proposed system takes $M$ input video clips from each camera and generates a complete trajectory for each target across the $M$ camera network. It consists of three components: (1) a Multi-Object Tracker (Section 3.2), (2) Ground Projector (Section 3.3), and (3) Global Trajectory Mapper (Section 3.4); these components perform two steps: (i) local tracklet generation and (ii) complete global trajectory generation. As shown in Figure 1, the first two components handle the first step, and the latter one performs the second step. We explain each component and phase in detail in the following sub-sections.

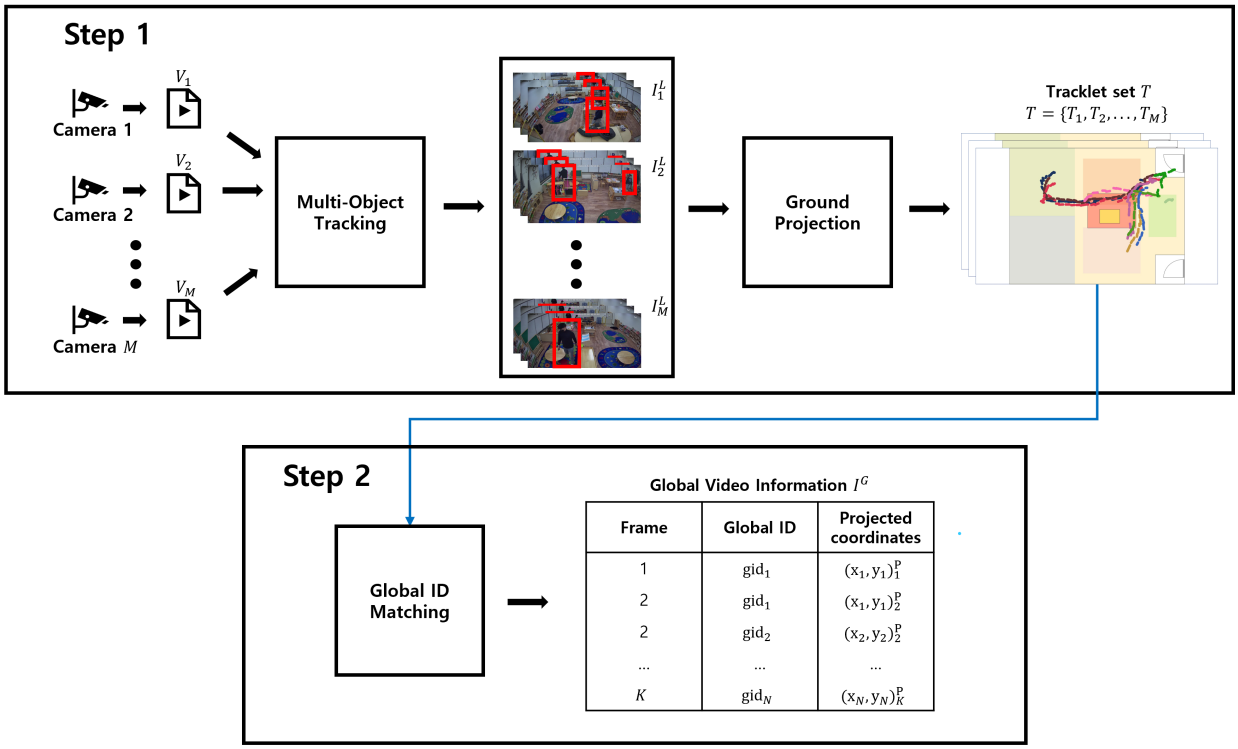



**Figure 1.** System Architecture.

### *3.2. Multi-Object Tracker*

Given input video $V_i$ from the $i$-th camera ($i = 1, \ldots, M$), *Multi-Object Tracker* detects a set of targets and generates their local tracklet information. As mentioned in the previous section, we rely on the state-of-the-art modern multiple object tracker for this task. To detect the targets, i.e., people, in the field-of-view (FOV) and their bounding boxes, we have employed FastMOT [31], which significantly accelerates the object tacking system to run in real time.

As the output of input video $V_i$ ($i = 1, \ldots, M$), Multi-Object Tracker generates a set of frame-by-frame local object detection information, $I_i^L$, which is given by

$$I_i^L = \{I_{i,1}^L, I_{i,2}^L, \ldots, I_{i,K}^L\},$$

where $K$ is the number of frames in video $V_i$ and $I_{i,k}^L$ denotes a set of local image tuples for each target detected in the $k$-th frame ($k = 1, \ldots, K$). Here, each video $V_i$ may have

a different value of $K$ and should be denoted as $K_i$. However, we omit the subscript $i$ to simplify the notation throughout the paper.

Let $\Pi_i$ denote a set of targets' local IDs (LIDs) observed in video $V_i$ and $\pi_{i,k}$ denote a set of LIDs observed in the $k$-th frame in $V_i$. Then, we can obtain

$$\Pi_i = \bigcup_{k=1}^{K} \pi_{i,k}. \quad (1)$$

A set of local image information $I_{i,k}^L$ for the $k$-th frame ($k = 1, \ldots, K$) is composed of a series of tuples $u_{i,k}(LID)$ with three attributes: (1) the frame number $k$, (2) the target's local ID, namely $LID$, and (3) the local foot coordinates (bottom center of the bounding box, which is often assumed to be on the ground):

$$I_{i,k}^L = \{u_{i,k}(LID) := (k, LID, (x,y)^L) | LID \in \pi_{i,k}\}, \quad (2)$$

where the coordinate $(x,y)^L$ is the reference location of target $ID^L$, which is the bottom center of its bounding box.

One of the challenges at this stage is that the object detection method demonstrates large temporal variations when it generates the bounding box of the target across the frame, where fluctuations may occur owing to motion blur, partial occlusion, change in poses, and other factors. This can cause short-term fluctuations in the derivation of local foot coordinates $(x,y)^L$ that act as short-term noise values.

To handle this issue, we use a simple yet effective strategy, known as Frame Skipping, which detects and tracks only the selected frames at a specific sampling period $s$. Frame Skipping predicts target positions using the KLT tracker without executing the detector and feature extractor for the frames between two selected frames. For a skipping period $s$ and a certain selected $k$-th frame, $I_{i,k+1}, I_{i,k+2}, \ldots, I_{i,k+s-1}$ are estimated by the KLT tracker. Frame Skipping alleviates the bottleneck of traditional MOT methods and enables real-time execution. In addition, we will show that the accuracy of the proposed matching algorithm is improved by removing noise in the calculation of the direction of each target (vector features), which will be described in Section 3.4.

### 3.3. Bird's Eye View (BEV) Ground Projector and Feature Extractor

*Ground Projector* takes the set of local image information in $I^L = \{I_1^L, I_2^L, \ldots, I_M^L\}$ and for each $I_i^L$ ($i = 1, \ldots, M$) produces the set of coordinates $(x,y)^P$ projected on the target coordination map by using a *homography matrix* $H$; homography is a transformation process (a 3 × 3 matrix) that maps the points in one image to the corresponding points in the other image [46].

To obtain the homography matrix $H_i (\in H)$ for $V_i$, eight point coordinates are needed, including four point coordinates in the image data and four corresponding points in the projection map. We have obtained reference points in the experiments by placing a rectangular grid carpet and by measuring edge positions offline.

In our experiments, we have observed a well-known camera distortion problem, i.e., barrel distortion [47], at the edges of frames (Figure 2). This distortion significantly affects on-the-ground projection, especially depending on the installation height and angle of the camera. This can cause a decrease in accuracy with regard to the performance of our similarity-based global ID-matching algorithm.

Therefore, camera calibration including distortion correction is required to obtain the accurate matrix $H$ and improve the matching accuracy. The parameters derived for camera calibration can be used continuously once acquired; however, if the camera's angle of view is changed, it must be derived again.

One key observation to address this challenge is that the error caused by camera distortion significantly affects the derivation of the targets' coordinates, but it negligibly impacts on the derivation of the targets' moving direction. Based on this observation,

author does not correct distortion by refi

we adopt the approach of using the targets' movement direction as the key feature for our global ID-matching algorithm instead of a cost calibration process. In particular, we investigate the vector-based features of moving targets for the global ID-matching algorithm, which will be described in Section 3.4.

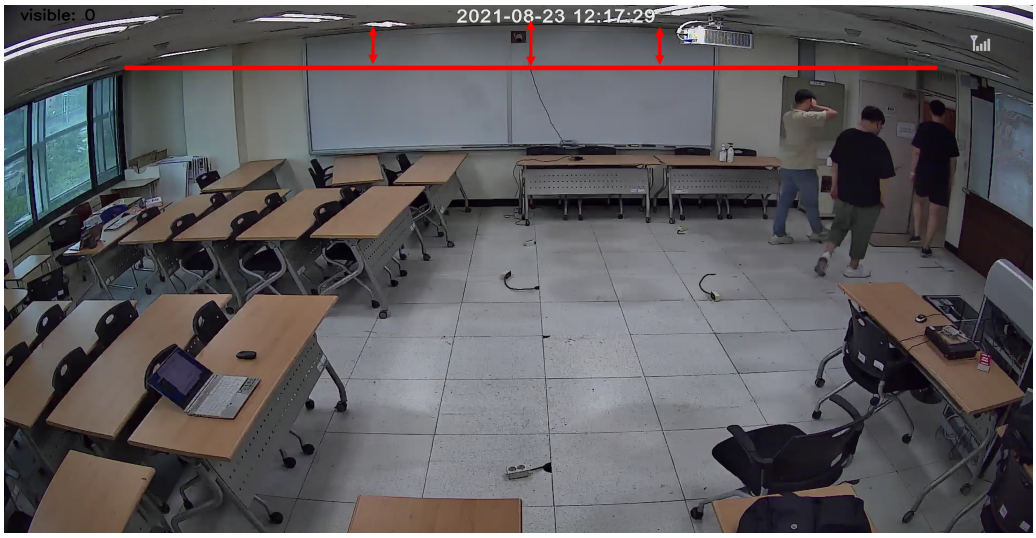



**Figure 2.** Barrel distortion commonly observed in commercial cameras.

Each local image information $I_{i,k}^{L}$ in Equation (2), which contains local coordinates $(x,y)^{L}$, is projected into the corresponding coordinates $(x,y)^{P}$ on the target map (Figure 3), which will be to generate a set of projected local image tuples $I_{i,k}^{P}$ for the $k$-th frame $(k = 1,\ldots,K)$:

$$I_{i,k}^{P} = \{u_{i,k}(LID) := (k, LID, (x,y)^{P}) | LID \in \pi_{i,k}\}. \tag{3}$$

Then, for each target with local ID x $\in \Pi_i$, we can construct the local tracklet $T_{i,x}^{L}$ as a set of tuples over the $k$-th frame $(k = 1,2,\ldots,K)$ in $V_i$:

$$T_{i,x}^{L} = \bigcup_{k=1}^{K} \{u_{i,k}(x) | x \in \pi_{k,i}\}. \tag{4}$$

Let $T_{i}^{L}$ denote a set of local trackets for $V_i$ extracted from the $i$-th camera, which is given by

$$T_{i}^{L} = \{T_{i,1}^{L}, T_{i,2}^{L}, \ldots, T_{i,N}^{L}\}, \tag{5}$$

where $N$ is the total number of targets observed across all the cameras, given by $N = |\bigcup_{i=1}^{M} \Pi_i|$. Here, the number of targets in $V_i$ may be less than $N$ because a few targets may never appear in camera $i$.

For example, when the $p$-th local ID is detected in the image data with the frame range $[1,4,\ldots,1024]$, $T_{i,p}$ for $V_i$ is given as:

$$T_{i,p} = \begin{pmatrix} \{frame_1, local - id_p^L, (x_{p,1}, y_{p,1})^P\} \\ \{frame_4, local - id_p^L, (x_{p,4}, y_{p,4})^P\} \\ \cdots \\ \{frame_{1024}, local - id_p^L, (x_{p,1024}, y_{p,1024})^P\} \end{pmatrix}.$$

Figure 3 shows the local tracklet set $T$ on the projection map. A set of $M$ local tracklets generated across the multi-camera network is used as an input for the global ID-matching module.

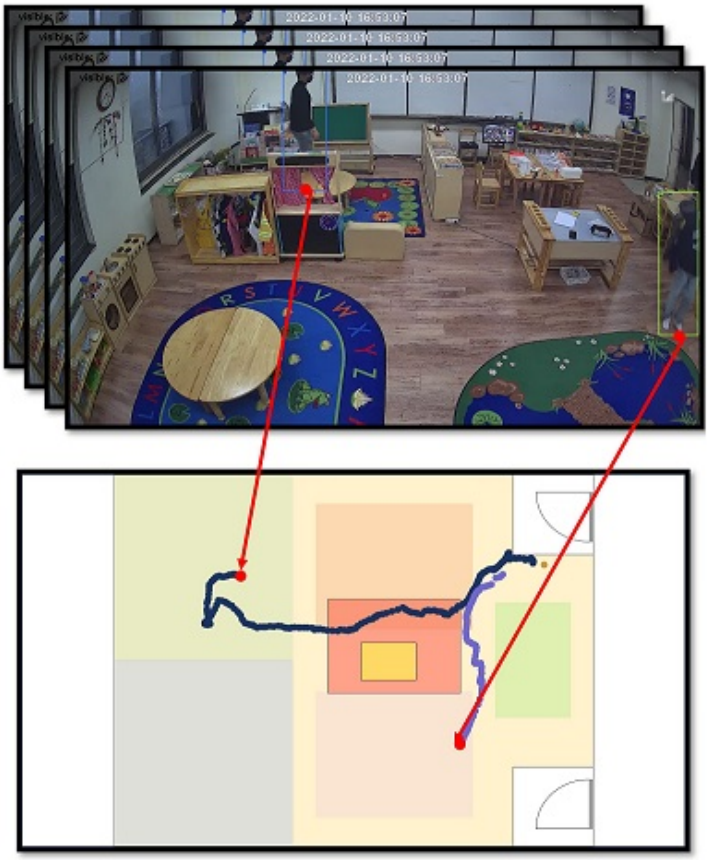


**Figure 3.** Local tracklet represented in the projection map (bird's eye view).

### *3.4. Global ID Matching*

Finally, given $M$ sets of local tracklets $T_i^L$ ($i = 1, 2, \ldots, M$) from $M$ cameras, the global ID-matching component uses the DTW (Dynamic Time Warping) algorithm to perform similarity analysis. The DTW algorithm can measure the similarity of two sequences, and it has the advantage of being able to measure even if the lengths of input sequences are different from each other. We used the two-dimensional sequence set $S_{i,a}$ ($\forall a \in \Pi_i$) generated from the coordinate information of $T_{i,a}$ to measure the similarity.

The features used for the input sequence generation are (i) scalar, (ii) vector, and (iii) unit vector of each target's movement information. In particular, to generate $S_{i,a}$ for target $a$, the coordinates $(x, y)_k^P(a)$ and $(x, y)_{k+1}^P(a)$ of two adjacent frames (i.e., $k$-th and $k+1$-th frames) in tuples $u_{i,k}(x)$ and $u_{i,k+1}(x) \in T_{i,x}$ are used. The $k$-th element values of $S_{i,a}$ generated using the (i) scalar, (ii) vector, and (iii) unit vector are $s_k$, $v_k$, and $w_k$, respectively, which are given by:

$$s_k = |(x, y)_{k+1}^P(a) - (x, y)_k^P(a)|, \tag{6}$$

$$v_k = (x, y)_{k+1}^P(a) - (x, y)_k^P(a), \tag{7}$$

$$w_k = v_k / |v_k|. \tag{8}$$

Algorithm 1 can generate sequences using the aforementioned features. To compare the similarity between synchronized trajectories, we compute the sequence of each target's local tracklet over the same period of time (frames).

**Algorithm 1:** Generation of Sequence Set $S_i$

**Input :**
Local tracklet set $T_i^L = \{T_{i,1}^L, T_{i,2}^L, \ldots, T_{i,a}^L, \ldots, T_{i,N}^L\}$
for each target $a(= 1, 2, \ldots, N = |\Pi_i|)$ generated from video $V_i$

**Output:**
Vector sequence set $S_i^L = \{S_{i,1}^L, S_{i,2}^L, \ldots, S_{i,N}^L\}$

**for all** $T_{i,a}^L \in T_i^L$ **do**
  **for all** $u_{i,k}(a) \in T_{i,a}^L$ **do**
    Generate $v_k$ (or $s_k$, $w_k$) using Equation (7) (or Equation (6), Equation (8), respectively)
    and add $v_k$ (or $s_k$, $w_k$) to $S_{i,a}$
  **end for**
**end for**

Given the sequence sets of each target, we use the DTW distance function $d_{DTW}(S_{i,a}, S_{j,b})$ for two different sequences from the $i$-th and $j$-cameras ($i$ and $j \in \{1, 2, \ldots, M\}$, $i \neq j$) to calculate the distance value $D$:

$$D(S_{i,a}, S_{j,b}) = d_{DTW}(S_{i,a}, S_{j,b}), \quad (9)$$

where the lower the $D$ value is, the higher the similarity will be.

We use Algorithm 2 to perform global ID matching by calculating a Tracklet Similarity Candidate List $R$. Under the overlapping FOV condition that a target assigned a local ID $p$ in camera $i$ has appeared in other cameras $j$ ($j \in 1, 2, \ldots, M, j \neq i$), we calculate $D(S_{i,p}, S_{j,q})$—the distance between $S_{i,p}$ and $S_{j,q}$ ($S_{j,q} \in S_j$)—by using Equation (9), which will be added to a tracklet similarity matrix $R_{i,j}$.

**Algorithm 2:** Calculation of Similarity List $R$ and Generation of a Global ID List $G_i$

**Input :**
Vector sequence set $S = \{S_1, S_2, \ldots, S_i, \ldots, S_M\}$

**Output:**
Similar local ID set $G_i = \{G_{i,1}, G_{i,2}, \ldots, G_{i,N}\}$

Generate empty ranking list $R$
**for all** $S_i (i = 1, 2, \ldots, M-1) \in S$ **do**
  **for all** $S_{i,p} \in S_i$ **do**
    **for all** $S_j (j = i+1, i+2, \ldots, M) \in S$ **do**
      **for all** $S_{j,q} \in S_c$ **do**
        Calculate DTW distance $D(S_{i,p}, S_{j,q})$ using Equation (9)
        Set the element of row $p$ and column $q$ in $A_{i,j}$ to $D(S_{i,p}, S_{j,q})$, i.e., $A_{i,j}(p,q) = D(S_{i,p}, S_{j,q})$
      **end for**
    **end for**
  **end for**
  Calculate matching matrix $A_{i,j}$ according to Equation (10)
  Add all $A_{i,j}(p,q) = 1$ to $G_i$ in the form of a tuple $(p,q)$.
**end for**

Given $R_{i,j}$, we calculate a matching matrix $A_{i,j} \in 0, 1^{N \times N}$, where each element $A_{i,j}(p,q)$ denotes a binary value of 1 or 0—which represents matching (i.e., $A_{i,j}(p,q) = 1$) and non-matching (i.e., $A_{i,j}(p,q) = 0$) operations. If tracklets $p$ and $q$ have a high similarity score $D(S_{i,p}, S_{j,q}) \to 1$, we should have $A(p,:)A(q,:)^T = 1$. On the contrary, for a small similarity

score, $A(p,:)A(q,:)^T = 0$, which implies that $p$ and $q$ are not the same target. This bipartite-graph-matching problem can be solved by the following optimization problem:

$$\begin{aligned} A^*_{i,j} = \arg\max_{G_{i,j}} ||R_{i,j} \odot G_{i,j}||_2, \\ s.t. \forall p, \sum G(p,:) \leq 1, \\ \forall q, \sum G(:,q) \leq 1, \end{aligned} \tag{10}$$

where $\odot$ denotes element-wise matrix multiplication, and $||\cdot||_2$ denotes the L2 norm of the input matrix. The constraints ensure the mutual exclusion of trajectories such that each detection occupies using at most one trajectory. Based on this, the optimization problem can be efficiently solved by the Hungarian algorithm [30].

Finally, given the sets $G = \{G_1, G_2, \ldots, G_M\}$ containing the local tracklet matching information, we assign the global ID to each local matching information and create a set $I^G$ for $V$. $I^G$ has the same configuration as $I^L$ except that the ID is global rather than local, and the coordinates possess the average coordinates of local IDs mapped to the global IDs. We assign a new global ID to the elements with the tuple $(p,:)$ and/or $(:,q)$ for $\forall(p,q) \in G_i$ ($\forall G_i \in G$). The overall pipeline is described in Algorithm 3.

**Algorithm 3:** Overall Pipeline to Generate Global Information

**Input :**
Video set $V = \{V_1, V_2, \ldots, V_M\}$ collected from $M$ cameras
**Output:**
Global video information $I^G$

**for all** $V_i \in V$ **do**
  Generate a local video information $I^L_i$ using a multi-object tracker [31]
**end for**
**for all** local video information $I^L_i \in I^L$ **do**
  Calculate the projected footprint coordinate $(x,y)^P$ using homography matrix $H$ and generate a projected video information $I^P$
**end for**
**for all** $I^P_i \in I^P$ **do**
  **for all** $p$ in $V_i$ **do**
    Generate local tracket $T_{i,p}$ and create vector sequence set $S_{i,p}$ using Algorithm 1 ($T_{i,p} \in T_i, S_{i,p} \in S_i$)
  **end for**
**end for**
Calculate similarity using DTW and generate similar local ID list set $G = \{G_1, G_2, \ldots, G_M\}$ using Algorithm 2
For $\forall(p,q) \in G_i$ ($\forall G_i \in G$), assign a new global ID to the elements with the tuple $(p,:)$ and/or $(:,q)$
Create an empty global video information $I^G$
**while** current frame < the end of frame **do**
  In $I^P$, find all existing ID on the current frame and update to global ID, calculate average coordinates
  Add the update to $I^G$
**end while**

## 4. Performance Evaluation

### *4.1. Experimental Setup*

To evaluate our method, we have conducted experiments using Wisenet's four-channel camera systems, SNK-B73047BW [48]. For experiments, we constructed overlapping FOV environments by installing up to four cameras in various places including classrooms

and daycare centers, as shown in Figure 4. Each video had three different configurations: 180,000, 135,000, and 90,000 frames.

In the first stage of MTMCT, wherein the Ground Projector generates the sets of local tracklet (Section 3.3), we consider different parameters for the KLT algorithm-based frame-skipping method with a fixed sampling interval, namely *Skip value*: 1 (no skip), 5, and 10. A larger skip value reduces the number of frames used for target tracking and speeds up the calculation. The frames between the two selected frames are inferred by applying the KLT tracker. To evaluate its effect and find the desired skip parameter, we compared the performance for the same dataset with different skip parameters. Moreover, as explained in Section 3.3, we considered the moving direction of the targets to filter out the errors in tracklet positions caused by camera distortion.

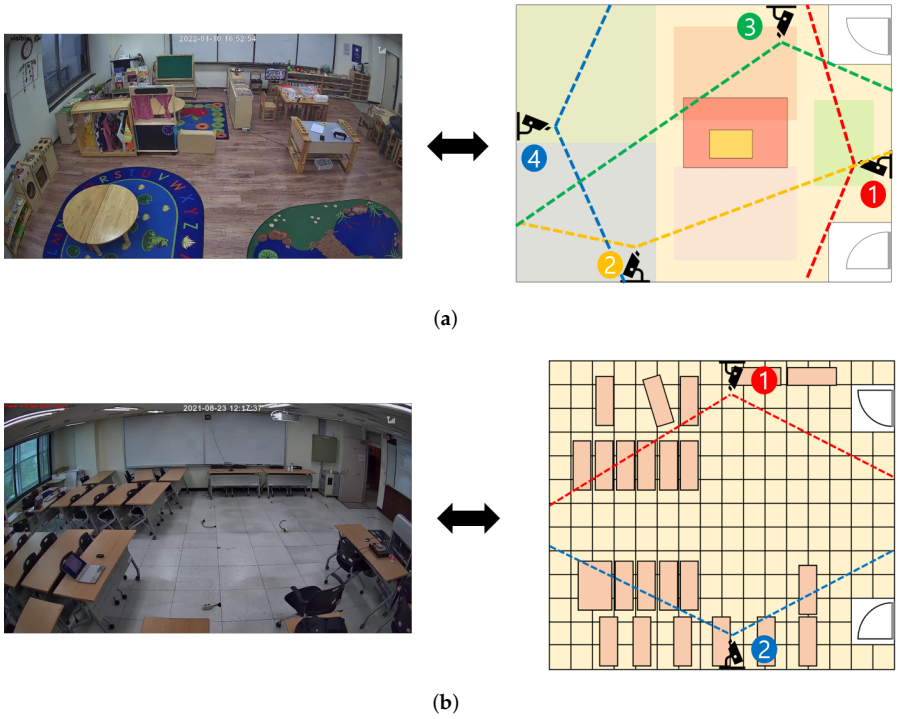



**Figure 4.** The testbeds and projected maps at two different locations (**a**) with four cameras and (**b**) two cameras, respectively.

In the second step, we performed global ID matching based on the similarity analysis between projected tracklets using the DTW algorithm and measured accuracy. For a given coordinate of each target, three different features—(i) scalar, (ii) vector, and (iii) unit vector based on Equations (6)–(8), respectively—were used to generate input sequences for DTW similarity analysis, and we compare their effect on the accuracy of global ID matching.

### *4.2. Evaluation Results*

The accuracy of the proposed global ID-matching method mainly relies on the tracklet data obtained from the output of the Multi-Object Tracker. This implies that the performance of the Multi-Object Tracker used has a significant impact. In particular, it is sensitive to the problems of ID switching and fragmentation. ID switching is a phenomenon in which an existing ID assigned to object X is incorrectly assigned to another object Y. It can be caused by several reasons, including occlusion, where other objects partially or totally obscure the identified object during a short period. For instance, Figure 5 shows a case of ID switching due to occlusion when two targets assigned with IDs 1 and 2 intersect for a short time.

In order to evaluate the performance of our proposal, therefore, it is important to understand the basic characteristics of the object tracker and obtain a ground-truth dataset

for performance evaluation. To this end, we collected ground-truth values by preprocessing the experimental image data to identify information on these ID-switching problems and measured the number of incorrect ID switching occurrences.

First, we examine the effect of the Frame Skipping method applied to alleviate the fundamental accuracy problem of the MOT.

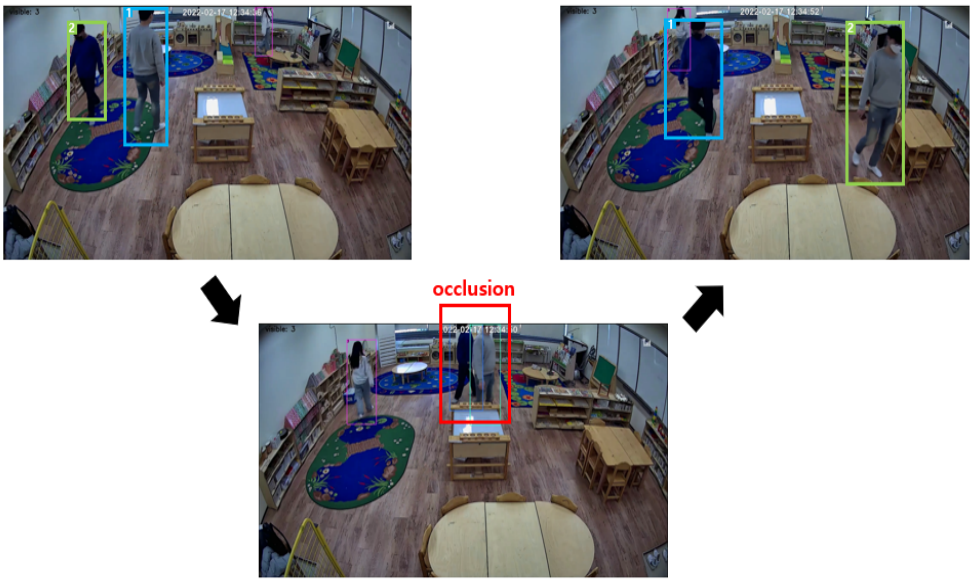



**Figure 5.** ID switching owing to short-term occlusion.

Table 1 lists the number of incorrect ID switching for each camera in a four-camera network with three different skip values. As ID switching occurs depending on the state of the video image, it varies depending on the installation location or angle of the camera. In our experiment, Re-ID failure occurred most frequently in camera 2, i.e., 48 out of a total of 150 data. This can be attributed to the fact that there are no obstacles near the wall where camera 2 was installed; thus, the target can move under camera 2. Only the upper body was recognized, which reduces the accuracy of Re-ID.

**Table 1.** Incorrect ID switching in a four-camera network with three different skip values.

| Camera ID | Camera 1 | Camera 2 | Camera 3 | Camera 4 |
|---|---|---|---|---|
| Original | 5 | 21 | 2 | 7 |
| Skip 5 | 2 | 13 | 3 | 5 |
| Skip 10 | 11 | 14 | 2 | 8 |
| Total | 18 | 48 | 7 | 20 |

Figure 6 shows the global ID-matching accuracy for three target persons in several FOV indoor environments covered by two cameras with three different target's movement features (i.e., scalar, vector, and unit vector) and three different skip values (i.e., no-skip, skip 5, and skip 10). Table 2 lists their F1 scores. In calculating the accuracy of the assignment results, an ID assignment is only considered successful if all global ID-matching results are correctly assigned to all three targets across the cameras. As shown in Figure 4, the overlapping area of cameras 1 and 4 was relatively wide, and camera 1 was deployed to observe the obstacles around the window easily. Thus, the results of experiments using cameras 1 and 4 and camera 1 and 3 in Figure 6a,b show very high accuracy for all the three motion features.

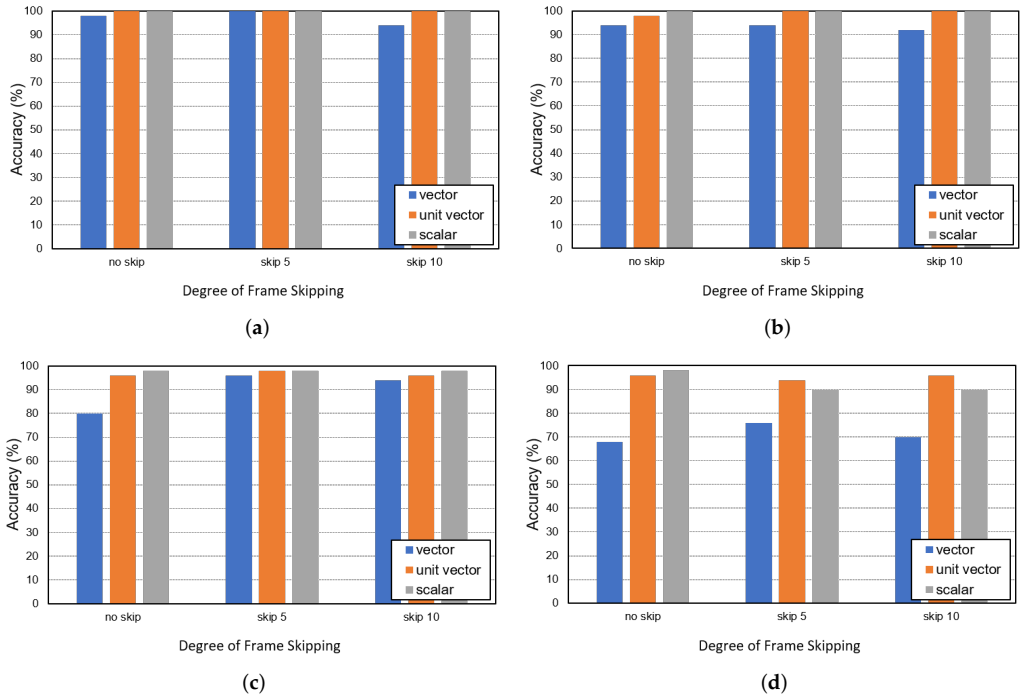



**Figure 6.** Global ID matching results for three targets in several overlapping FOV indoor scenarios with two cameras: (**a**) cameras #1 and #4, (**b**) cameras #1 and #3, (**c**) cameras #1 and #2, and (**d**) cameras #2 and #3.

**Table 2.** F1 score for global tracklet matching in cameras 2 and 3 FOV environments.

| **cameras #1 and #4** | **Original** | **Skip 5** | **Skip 10** |
|---|---|---|---|
| Vector | 0.98 | 1.0 | 0.94 |
| Unit vector | 1.0 | 1.0 | 1.0 |
| Scalar | 1.0 | 1.0 | 1.0 |
| **cameras #1 and #3** | **Original** | **Skip 5** | **Skip 10** |
| Vector | 0.96 | 0.96 | 0.9 |
| Unit vector | 0.98 | 1.0 | 1.0 |
| Scalar | 1.0 | 1.0 | 1.0 |
| **cameras #1 and #2** | **Original** | **Skip 5** | **Skip 10** |
| Vector | 0.96 | 0.98 | 0.87 |
| Unit vector | 0.98 | 0.98 | 0.98 |
| Scalar | 1.0 | 0.98 | 0.98 |
| **cameras #2 and #3** | **Original** | **Skip 5** | **Skip 10** |
| Vector | 0.77 | 0.83 | 0.80 |
| Unit vector | 0.98 | 0.97 | 0.98 |
| Scalar | 0.98 | 0.93 | 0.94 |

As shown in Figure 6c,d, the results including camera 2 exhibit relatively low performance. We analyzed the cause of matching failure and observed that there were mainly two reasons. First, the bounding box of the Multi-Object Tracker itself fluctuated severely in the videos from camera 2, and it had a greater impact on the similarity analysis between sequences obtained from vector motion features using direction information. Next, we observed high Re-ID errors in videos from camera 2 due to its underlying deployment environment. Table 3 shows the Re-ID failure rate for the results in Figure 6d for cameras 2 and 3, which yielded the lowest performance among the four combinations. In the cases of original (no skip) and sampling with a skip value of 5 (skip 5), global ID-matching failure

due to Re-ID failure accounts for more than half of all the failures. This implies that most errors occur not because of the proposed matching solution but due to the inherent Re-ID problem of Multi Object Tracker. After all, the Multi-Object Tracker's Re-ID problem (which is outside the scope of our paper) greatly affects the performance of the proposed technique.

**Table 3.** Ratio of errors due to (inherent) Re-ID to the total number of failures for global tracklet matching in cameras 2 and 3 FOV environments.

| Feature | Original | Skip 5 | Skip 10 |
|---|---|---|---|
| Vector | 10/16 | 6/12 | 5/15 |
| Unit vector | 2/2 | 3/3 | 1/2 |
| Scalar | 1/1 | 4/5 | 2/5 |

In the case of the combination of cameras 1 and 2 it can be seen that using frame skipping with a skip value of 5 (skip 5) significantly improved the performance of the unit vector feature, although the accuracy degrades again when using a higher skip value, i.e., 10. This is because 'skip 10' predicts the target's position over a longer frame, resulting in a loss of original information. Based on these results, we can observe the effect on the performance of the frame-skipping technique based on the KLT. That is, despite a small amount of frames used to generate sequence sets compared to the original data, the frame-skipping scheme improved the matching accuracy.

Next, we conducted experiments for additional people in the overlapping FOV indoor environment with two cameras. Figure 7 shows the experimental results for five and ten persons with three different frame skipping and motion features. Table 4 lists their F1 scores. In the five-person experiment, the proposed scheme achieved an accuracy of 100% with a frame skipping value of 5 and scalar motion feature. In the case of the ten-person experiments, the matching was counted as a failure even in one false matching among 10 matchings, so a significant decrease in matching accuracy was observed. Note that the ten-person experiment is a very congested scenario. Nevertheless, we observed that frame skipping improved the performance for all motion features in these experiments. When the vector feature was used, the accuracy was shown to be lower than the other two features. This is because the vector takes into account the stride length and directionality of the target. As previously explained, accurate stride measurements are not possible for reasons such as undulations in the bounding box or errors in the ground projection. Therefore, it can be confirmed that the scalar and the unit vector are suitable motion features for the proposed method.

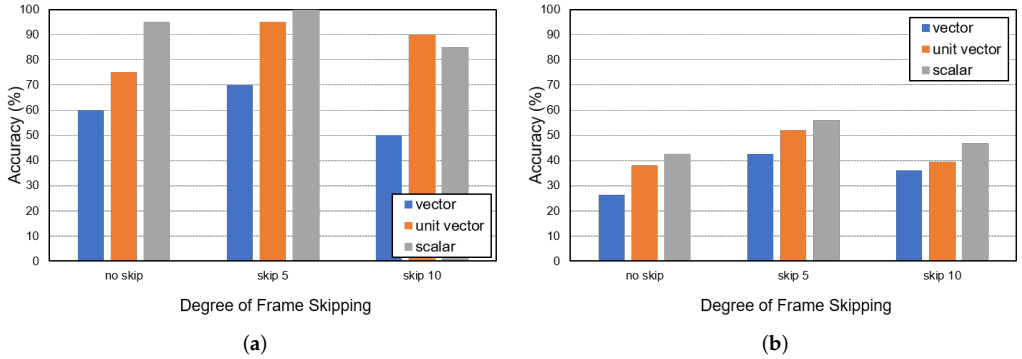



**Figure 7.** Global ID-matching accuracy with two cameras for (**a**) five persons and (**b**) ten persons.

Indoor environment have blind spots due to obstacles and other reasons, which decrease MOT accuracy. Therefore, it is necessary to intentionally construct an overlapping FOV environment using redundant cameras. In this regard, in the following experiment, we conducted measurements with more cameras to test whether our global ID-matching method will show stable performance for using two or more cameras.

**Table 4.** F1 score for Global Tracklet Matching with Two Cameras for Five Persons and Ten Persons.

| **cameras #1 and #4** | **Original** | **Skip 5** | **Skip 10** |
|---|---|---|---|
| Vector | 0.80 | 0.85 | 0.77 |
| Unit vector | 0.88 | 0.98 | 0.96 |
| Scalar | 0.98 | 1.0 | 0.96 |
| **cameras #1 and #3** | **Original** | **Skip 5** | **Skip 10** |
| Vector | 0.36 | 0.56 | 0.50 |
| Unit vector | 0.48 | 0.65 | 0.56 |
| Scalar | 0.52 | 0.68 | 0.61 |

Figure 8a,b show high accuracy using three and four cameras, respectively, for three persons. While calculating accuracy, it was considered a success only if the targets of all the cameras matched correctly. Despite the increase in the number of cameras, the proposed technology achieved high accuracy. It achieved the best performance with the skip values of 5 and 10 for vector and both scalar and unit vector features, respectively. We investigated and summarized the errors caused by Re-ID of Muilti Object Tracker among the causes of the failure of matching for four cameras in Table 5. The results indicate that most errors (in the experiments with the unit vector and scalar features) are not related to our algorithms but rather are due to the inherent noise involved in the sequence sets caused by Re-ID problems.

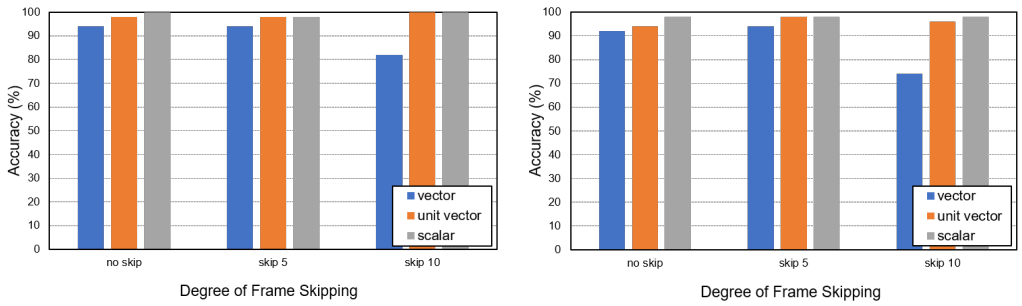



**Figure 8.** Global ID-matching accuracy with (**a**) 3 cameras (cameras 1, 2, and 3), and (**b**) 4 cameras.

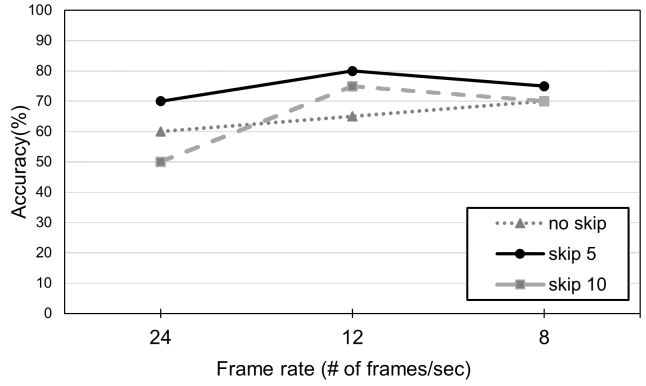
**Table 5.** Ratio of errors due to Re-ID to the total number of failures for global tracklet matching in the experiment with four cameras.

| **Feature** | **Original** | **Skip 5** | **Skip 10** |
|---|---|---|---|
| Vector | 2/4 | 3/3 | 7/13 |
| Unit vector | 3/3 | 1/1 | 1/2 |
| Scalar | 1/1 | 1/1 | 0/1 |

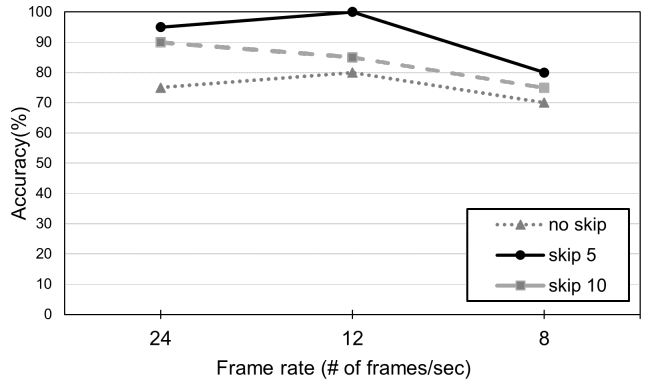
Finally, we studied the impact of our approach in reducing the effects of computational complexity on matching accuracy. As explained in Section 3.2, sequence sets were generated using Algorithm 1 and KLT-based frame skipping. In addition, we further reduced the amount of data required to generate sequence sets by selectively using frames from each video $V_i$. Given $V_i$ with the default frame rate of 24 frames/sec, we use interval sampling (also known as systematic sampling), which selects every $k$-th frame in the video, where we have tested for $k = 1$ (default/original), 2 (half frame rate), and 3 (one-third), which correspond to 24, 12, and 8 frames/sec, respectively.

Figure 9 shows the experimental performance of two cameras and five persons considering vector, unit vector, and scalar features. In the results, we have observed a clear effect on performance with frame rate for all the three movement features and a significant performance improvement using the half rate of 12 frames/s instead of using the original

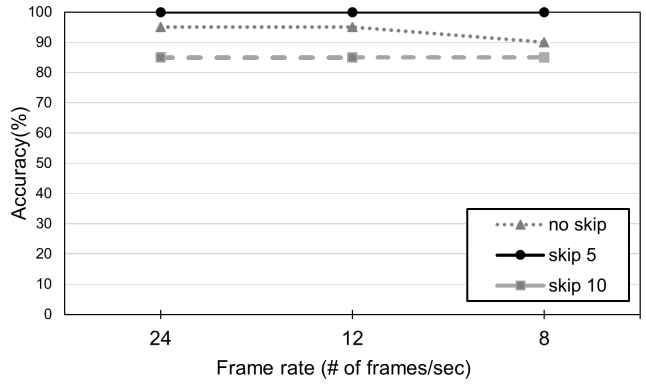
frame rate. If the frame rate is too low (e.g., 8 frames/s), the accuracy will decrease again. While using unit vectors for generating sequence sets, the original video without frame skipping at 24 frames/s had an accuracy of 90%. On the other hand, it is improved to an accuracy of 100% when applying frame skipping with a skip of 5 at a half frame rate, i.e., 12 frames/s, where the amount of data corresponds to 1/10 of the original data. In the case of scalar feature, sampling with a skip value of 5 achieved an accuracy of 100%, and there was no performance degradation even when the amount of information is reduced by 1/3 with 8 frames/s. These results imply that the proposed approach showed high accuracy even with a smaller amounts of data (1/10 of that of the original data), lightweight computational overhead for DTW similarity, and matching tasks.




(**a**)




(**b**)




(**c**)

**Figure 9.** Effect of frame sampling rate for three movement features: (**a**) vector, (**b**) unit vector and (**c**) scalar.

## 5. Conclusions

In this paper, we proposed a new lightweight matching method for MTMC tracking consisting of two steps: (i) extracting targets' motion information based on a ground projection method, and (ii) matching the tracklets using similarity analysis based on the Dynamic Time Warping (DTW) algorithm. We reduced the computational overhead by leveraging a KLT-based frame skipping and smoothing method to reduce computational costs in using targets' location information to generate input sequence sets for our matching algorithm. In addition, three different location features including scalar, vector, and unit vector have also been studied to derive the best input feature for similarity analysis to improve matching accuracy. Extensive experiments demonstrated the effectiveness of the proposed method, showing that our scheme achieved high accuracy in most overlapping FOV environments. In dense environments, performance degradation occurred, but it has been shown that many errors occur not because of the proposed matching solution but due to the inherent Re-ID problem of the Multi Object Tracker.

**Limitation and Future Work.** A significant limitation is that the accuracy of the proposed framework strongly relies on the performance of the Multi-Object Tracker, since the tracklet data are obtained from the output of the Multi-Object Tracker. Although we alleviate the inborn position error caused by camera distortion by exploiting the moving direction information of the target instead of the absolute position of the target, the matching based on the similarity analysis suffers from the errors even with the proposed features when the camera distortion is severe.

In the future, we would like to extend our work in the following three categories: (i) employing adaptive feature selection for different object's movement features, (ii) optimizing tracklet matching using graph models and/or Bayesian formulation, and (iii) reducing the computational complexity of DTW for obtaining similarity measures between feature sequences by applying fast DTW algorithms.

**Author Contributions:** Conceptualization, J.J. and J.C.; methodology, J.J. and J.C.; software, J.J. and M.S.; validation, J.J.; formal analysis, J.J.; writing–original draft preparation, J.J. and J.C; writing–review and editing, J.C.; visualization, J.J., M.S. and J.C.; supervision, J.C.; project administration, J.C.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2020R1A2C1013308) and in part by the Gachon University research fund of 2021(GCU-202110050001).

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

## References

1. He, Y.; Wei, X.; Hong, X.; Shi, W.; Gong, Y. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Trans. Image Process.* **2020**, *29*, 5191–5205. [CrossRef]
2. Li, D.; Wei, X.; Hong, X.; Gong, Y. Infrared-visible cross-modal person re-identification with an x modality. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 4610–4617.
3. Vijeikis, R.; Raudonis, V.; Dervinis, G. Efficient Violence Detection in Surveillance. *Sensors* **2022**, *22*, 2216. [CrossRef] [PubMed]
4. Shim, K.; Yoon, S.; Ko, K.; Kim, C. Multi-target multi-camera vehicle tracking for city-scale traffic management. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4193–4200.
5. Huang, Q.; Hao, K. Development of CNN-based visual recognition air conditioner for smart buildings. *J. Inf. Technol. Constr.* **2020**, *25*, 361–373. [CrossRef]
6. Choi, H.; Um, C.Y.; Kang, K.; Kim, H.; Kim, T. Review of vision-based occupant information sensing systems for occupant-centric control. *Build. Environ.* **2021**, *203*, 108064. [CrossRef]
7. Chen, M.; Burke, R.R.; Hui, S.K.; Leykin, A. Understanding lateral and vertical biases in consumer attention: An in-store ambulatory eye-tracking study. *J. Mark. Res.* **2021**, *58*, 1120–1141. [CrossRef]
8. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
9. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3645–3649.

10. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
11. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
12. He, Y.; Yu, W.; Han, J.; Wei, X.; Hong, X.; Gong, Y. Know Your Surroundings: Panoramic Multi-Object Tracking by Multimodality Collaboration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2969–2980.
13. Tomasi, C.; Kanade, T. Detection and tracking of point. *Int. J. Comput. Vis.* **1991**, *9*, 137–154. [CrossRef]
14. Shi, J. Good features to track. In Proceedings of the 1994 IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
15. Fleuret, F.; Berclaz, J.; Lengagne, R.; Fua, P. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 267–282. [CrossRef]
16. Berclaz, J.; Fleuret, F.; Turetken, E.; Fua, P. Multiple object tracking using k-shortest paths optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1806–1819. [CrossRef]
17. Hu, W.; Hu, M.; Zhou, X.; Tan, T.; Lou, J.; Maybank, S. Principal axis-based correspondence between multiple cameras for people tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 663–671. [PubMed]
18. Eshel, R.; Moses, Y. Homography based multiple camera detection and tracking of people in a dense crowd. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
19. Hou, Y.; Zheng, L.; Wang, Z.; Wang, S. Locality aware appearance metric for multi-target multi-camera tracking. *arXiv* **2019**, arXiv:1911.12037.
20. Bredereck, M.; Jiang, X.; Körner, M.; Denzler, J. Data association for multi-object tracking-by-detection in multi-camera networks. In Proceedings of the 2012 Sixth International Conference on Distributed Smart Cameras (ICDSC), Hong Kong, China, 30 October–2 November 2012; pp. 1–6.
21. Zhang, Z.; Wu, J.; Zhang, X.; Zhang, C. Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project. *arXiv* **2017**, arXiv:1712.09531.
22. Zhong, Z.; Zheng, L.; Cao, D.; Li, S. Re-ranking person re-identification with k-reciprocal encoding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1318–1327.
23. Jiang, N.; Bai, S.; Xu, Y.; Xing, C.; Zhou, Z.; Wu, W. Online inter-camera trajectory association exploiting person re-identification and camera topology. In Proceedings of the 26th ACM International Conference on Multimedia, Seoul, Korea, 22–26 October 2018; pp. 1457–1465.
24. Xu, Y.; Liu, X.; Liu, Y.; Zhu, S.C. Multi-view people tracking via hierarchical trajectory composition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4256–4265.
25. You, S.; Yao, H.; Xu, C. Multi-Target Multi-Camera Tracking with Optical-based Pose Association. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3105–3117. [CrossRef]
26. Wu, Y.C.; Chen, C.H.; Chiu, Y.T.; Chen, P.W. Cooperative People Tracking by Distributed Cameras Network. *Electronics* **2021**, *10*, 1780. [CrossRef]
27. 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), Munich, Germany, 5–8 December 2019; pp. 1–4.
28. Hsu, H.M.; Cai, J.; Wang, Y.; Hwang, J.N.; Kim, K.J. Multi-target multi-camera tracking of vehicles using metadata-aided re-id and trajectory-based camera link model. *IEEE Trans. Image Process.* **2021**, *30*, 5198–5210. [CrossRef]
29. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Fluids Eng.* **1960**, *82*, 35–45. [CrossRef]
30. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **1955**, *2*, 83–97. [CrossRef]
31. Yang, Y. FastMOT: High-Performance Multiple Object Tracking Based on Deep SORT and KLT. *J. Korea Multimed. Soc.* **2020**, *20*, 893–910. [CrossRef]
32. Liu, C.J.; Lin, T.N. DET: Depth-Enhanced Tracker to Mitigate Severe Occlusion and Homogeneous Appearance Problems for Indoor Multiple-Object Tracking. *IEEE Access* **2022**, *10*, 8287–8304. [CrossRef]
33. Du, Y.; Song, Y.; Yang, B.; Zhao, Y. StrongSORT: Make DeepSORT Great Again. *arXiv* **2022**, arXiv:2202.13514.
34. Chu, P.; Wang, J.; You, Q.; Ling, H.; Liu, Z. Transmot: Spatial-temporal graph transformer for multiple object tracking. *arXiv* **2021**, arXiv:2104.00194.
35. Zheng, L.; Tang, M.; Chen, Y.; Zhu, G.; Wang, J.; Lu, H. Improving multiple object tracking with single object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2453–2462.
36. Yang, F.; Chang, X.; Sakti, S.; Wu, Y.; Nakamura, S. ReMOT: A model-agnostic refinement for multiple object tracking. *Image Vis. Comput.* **2021**, *106*, 104091. [CrossRef]
37. Yu, E.; Li, Z.; Han, S.; Wang, H. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *IEEE Trans. Multimed.* **2022**. [CrossRef]
38. Wang, Q.; Zheng, Y.; Pan, P.; Xu, Y. Multiple object tracking with correlation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 3876–3886.

39. Wang, Y.; Kitani, K.; Weng, X. Joint object detection and multi-object tracking with graph neural networks. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 13708–13715.
40. Liang, C.; Zhang, Z.; Zhou, X.; Li, B.; Zhu, S.; Hu, W. Rethinking the competition between detection and ReID in multiobject tracking. *IEEE Trans. Image Process.* **2022**, *31*, 3182–3196. [CrossRef]
41. Li, W.; Xiong, Y.; Yang, S.; Xu, M.; Wang, Y.; Xia, W. Semi-tcl: Semi-supervised track contrastive representation learning. *arXiv* **2021**, arXiv:2107.02396.
42. Sun, P.; Jiang, Y.; Zhang, R.; Xie, E.; Cao, J.; Hu, X.; Kong, T.; Yuan, Z.; Wang, C.; Luo, P. Transtrack: Multiple-object tracking with transformer. *arXiv* **2020**, arXiv:2012.15460.
43. Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; Alameda-Pineda, X. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv* **2021**, arXiv:2103.15145.
44. Shan, C.; Wei, C.; Deng, B.; Huang, J.; Hua, X.S.; Cheng, X.; Liang, K. Tracklets predicting based adaptive graph tracking. *arXiv* **2020**, arXiv:2010.09015.
45. Tokmakov, P.; Li, J.; Burgard, W.; Gaidon, A. Learning to track with object permanence. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10860–10869.
46. Lubbes, N. Families of circles on surfaces. *arXiv* **2013**, arXiv:1302.6710.
47. Prescott, B.; McLean, G. Line-based correction of radial lens distortion. *Graph. Model. Image Process.* **1997**, *59*, 39–47. [CrossRef]
48. WISENET. 4 × 4 SNK-B73047BW 1080p NVR. Available online: https://www.wisenetlife.com/ko/product/All-in-One/SNK-B73047BW/feature/ (accessed on 3 January 2022).