Privacy-preserving Pedestrian Tracking using Distributed 3D LiDARs

Masakazu Ohno
*Osaka University**
Osaka, Japan
m-ohno@ist.osaka-u.ac.jp

Riki Ukyo
*Osaka University**
Osaka, Japan
r-ukyoh@ist.osaka-u.ac.jp

Tatsuya Amano
*Osaka University**
Osaka, Japan
t-amano@ist.osaka-u.ac.jp

Hamada Rizk
*Osaka University, Osaka, Japan**
Tanta University, Tanta, Egypt
hamada_rizk@f-eng.tanta.edu.eg

Hirozumi Yamaguchi
*Osaka University**
Osaka, Japan
h-yamagu@ist.osaka-u.ac.jp

***Abstract*—The growing demand for intelligent environments unleashes an extraordinary cycle of privacy-aware applications that makes individuals' life more comfortable and safe. Examples of these applications include pedestrian tracking systems in large areas. Although the ubiquity of camera-based systems, they are not a preferable solution due to the vulnerability of leaking the privacy of pedestrians. In this paper, we introduce a novel privacy-preserving system for pedestrian tracking in smart environments using multiple distributed LiDARs of non-overlapping views. The system is designed to leverage LiDAR devices to track pedestrians in partially covered areas due to practical constraints, e.g., occlusion or cost. Therefore, the system uses the point cloud captured by different LiDARs to extract discriminative features that are used to train a metric learning model for pedestrian matching purposes. To boost the system's robustness, we leverage a probabilistic approach to model and adapt the dynamic mobility patterns of individuals and thus connect their sub-trajectories. We deployed the system in a large-scale testbed with 70 colorless LiDARs and conducted three different experiments. The evaluation result at the entrance hall confirms the system's ability to accurately track the pedestrians with a 0.98 F-measure even with zero-covered areas. This result highlights the promise of the proposed system as the next generation of privacy-preserving tracking means in smart environments.**

***Index Terms*—Point cloud-based recognition, LiDAR, Privacy-preserving, Re-ID, Pedestrian tracking**

I. Introduction

Many systems have been proposed to detect and track people in large areas, especially using RGB cameras (the nowadays commodity devices for such a purpose). Those camera-based tracking systems usually use multiple RGB cameras to track people (referred to as Multi-Camera Tracking (MCT)) [1]. Since person Re-ID (re-identification) among multiple cameras is the primary issue in MCT, recent MCT approaches rely on the power of deep learning techniques to get features from RGB cameras. These features are used to identify each person who appears in different camera scenes to enable tracking of the target person. However, these features are often bio-metric features of the bystander, e.g., face, skin color, gender, age, and body shape. Deep neural networks can encode the image features into latent space, which is safer (*i.e.* unable to recover the original data) as the structure of the networks becomes complicated. Nevertheless, there have been many attacks that attempt to leak the privacy of users (along with their images) involved in such systems methods like membership inference attacks [2] and model inversion attacks [3].

Recently, 3D Light Detection and Ranging sensors (3D LiDARs, or simply LiDARs) have attracted more attention in terms of the balance between the privacy-preserving features and the capability of spatial sensing. LiDARs only acquire distances to the nearest objects in each 3D direction in FoV (Field of View), and the distance error is usually in the range of $10^{-2} \sim 10^{-1}$ meters with a coverage range of up to 100 meters. LiDARs generate colorless 3D point clouds, which is more privacy-reserving compared to camera-based systems. Moreover, 3D LiDARs enjoy accurate ranging characteristics making their pedestrian tracking accuracy generally higher than the camera-based counterparts [4], [5].

This gives rise to the development of large-scale pedestrian tracking systems with Multi-LiDARs (MLT) distributed over the target area. However, full coverage of large areas with LiDARs (or even cameras) may not always be possible due to the associated cost, the lack of power supplies, and/or occlusion (LoS constraints)[0]. Therefore, pedestrian tracking with multi-LiDARs in partially covered areas is a more challenging problem. In other words, the problem is to track pedestrian(s) captured with one LiDAR and re-identify them with a colorless

* Mobile Computing Lab., Osaka University, Japan

[0]one common issue in indoor tracking using cameras/LiDARs is how to obtain clear views to track pedestrians with a lot of obstacles like ad signs, plants, etc.

point cloud of other LiDARs given a non-overlapping field of views (FoVs) and a non-covered area in-between.

In this paper, we propose a privacy-preserving pedestrian tracking system using distributed 3D LiDARs of non-overlapping views. Specifically, the proposed system attempts to find the correspondence between multiple pedestrians' sub-trajectories obtained by each LiDAR to estimate the whole trajectory of each person in the target area. Towards this end, the proposed system identifies pedestrians and recognizes their sub-trajectories based on two criteria: the similarity of the point cloud signature of each pedestrian and the Spatio-temporal characteristics of the pedestrian sub-trajectories. The former is achieved by employing the Fisher Vector approach [6] to extract discriminative fixed-size features representing the shape and behavior of each person given her point cloud. The system then trains a deep-metric learning model to learn the dissimilarity between the features of different persons. The second criterion leverages sub-trajectory start/end points to learn possible point transitions of pedestrians. This is done by defining the probability distribution of traveling time and mobility patterns and updating these distributions using the Bayesian approach.

To demonstrate the usefulness of the proposed approach, we deployed the system in a large testbed of six floors building equipped with 70 LiDARs and conducted three different experiments. The evaluation results of the system on 32 pedestrians confirm its efficacy in achieving a consistently high matching accuracy of the pedestrian trajectories with 0.98 F-measure. This result is achieved with only colorless sparse 3D LiDARs that ensure the privacy of pedestrians.

To summarize, our contributions are three-fold. (i) *Uniqueness of the problem.* We tackle a new problem to obtain complete pedestrian trajectories from sub-trajectories which are captured by distributed 3D LiDARs. As far as we investigate, no other research has been done for this problem. (ii) *Novelty of the approach.* Unlike multi-camera multi-object tracking, the LiDAR-based sub-trajectories are more accurate, but we have less clue to connect those segmented ones in terms of person re-identification. To address the issue, we design a unique algorithm to find the most likely matching among them taking point cloud-specific features as input. The probability distribution functions are updated based on the Bayesian updating system. (iii) *Evaluation using the real data.* We conducted several experiments using the real LiDAR data obtained in our large-scale testbed. The testbed consists of 70 LiDARs installed over 6 floors of the 7-story building in our university campus. The performance of the method has been validated through the dataset.

II. Related Work

Multi-Object Tracking (MOT) has extensively studied to track persons or vehicles [1]. Solution for this problem can be, in general, categorized into *single camera approaches* and *multiple camera approaches*.

A. Multi-Object Tracking (MOT) with Single Camera / LiDAR

Tracking a person or vehicle using a single camera has been well investigated due to the availability of several public datasets, e.g., MOT Challenge [7] and KITTI dataset [8]. The MOT Challenge dataset consists of videos captured by surveillance cameras enabling tracking methods in various scenarios, such as tracking in congested areas. The KITTI dataset includes 3D point cloud data of pedestrians acquired using LiDARs on top of passing vehicles. DeepSORT [9] is a typical MOT system that uses the Yolo object detection method to detect target objects in a given frame. Then, it leverages Kalman filter to track the moving objects in consecutive frames. 3D vehicle detection and tracking from monocular videos has been proposed in [10]. It can estimate 3D bounding boxes surrounding each object in a sequence of 2D images, using a Deep Neural Network (DNN).

On the other hand, a lot of efforts have been dedicated to 3D tracking using 3D cameras or LiDARs. RGB images and 3D point clouds have been leveraged together for tracking pedestrians in [11]–[13], while 3D point clouds have been adopted alone in [14]–[18]. An advantage of using 3D point clouds alone is the resilience to varying colors and brightness, which usually affect cameras. Additionally, the privacy concerns associated with surveillance cameras do not apply to LiDARs. However, leveraging LiDARs makes MOT challenging as colors are the most critical signature to detect/identify persons. To cope with the issue, self-designed features are often incorporated to perform 3D MOT using 3D point clouds [19], [20]. We have also used our own tracking system using a single 3D LiDAR.

Different from these apporaches, the proposed system is designed to achieve accurate pedestrian tracking using multiple, distributed 3D LiDARs.

B. Multi-Camera Multi-Object Tracking with Person Re-Identification

Multi-Camera Multi-Object Tracking (Multi-Camera MOT) refers multiple object tacking with multiple distributed cameras which has been investigated for specific cases in [21], [22], [23]. While identifying a person detected by one camera and using a different camera at a different location and timing (called *person re-identification* in multi-camera MOT) has been proposed [24]. In [25], the similarity between pedestrians is calculated using the clothes' colors in RGB images and the traveling time and distance between the different cameras. Then, the matching is carried out using the Hungarian method [26], [27]. Additionally, different types of camera are fused for person re-identification [28], [29]. The system in [28] combines a RGB-D camera with a temperature sensor, while RGB and infrared cameras are used in [29]. In [28], RGB-D camera with depth data have been utilized to extract the pedestrians' skeletal information. Also, recent studies, e.g., [30], [31], tends to enhance the capability of RGB-D based recognition and person re-identification. On the other hand, the system in [32] estimates joints of human bodies from a 3D point cloud. However, this work cannot be directly adopted

for person re-identification as the human pose does not contain enough information to distinguish persons. Although the presence of datasets for camera-based person re-identification, e.g., Market-1501 [33] and Motion Analysis and Re-identification Set (MARS) [34], no similar datasets are available using multiple distributed 3D LiDARs.

To the best of our knowledge, this is the first work that leverages the privacy-preserving 3D point clouds captured by multiple distributed LiDARs for person re-identification. Additionally, the proposed system handle the challenges associated with processing 3D point cloud, such as unordered, unstructured, and varying size point clouds.

C. Trajectory Prediction

Trajectory prediction [35], [36] has been studied to extract patterns from moving trajectories. The method in [35] proposes a trajectory prediction model for traffic networks based on past movement patterns. By dividing trajectories into clusters and discovering frequently occurring trajectory patterns, future movement trajectories can be predicted with high accuracy. The authors of [36] also investigate trajectory prediction, but they use sparse coding [37] to represent each trajectory as a combination of predefined trajectory patterns, which enables the prediction of subsequent trajectories for each new trajectory. They also propose a similarity-based model fusion algorithm that allows agents to update their knowledge by communicating the data they have learned with each other.

On the contrary, this paper focuses on the problem of finding pedestrians trajectories from a given set of their sub-trajectories, the context of our work is entirely different.

III. Motivation for Using LiDARs

In this section, we motivate the adoption of LiDARs as the core technology for the proposed system. LiDAR is emerging as a powerful enabler of the next generation of smart and safe environments [38], [39]. LiDARs can provide long-range, real-time, centimeter-level distance measurements of surrounding objects in all lighting conditions.

Privacy: LiDAR provides a key advantage over camera-based systems – privacy protection. With increased concerns that facial recognition technology can be used for general surveillance, the U.S. Congress discusses legislation that seeks to ban the use of camera-based human identification and other biometric surveillance technology by federal law enforcement agencies. Thus, nowadays, industry has seen leading tech vendors stepping away from their own camera-based facial recognition technologies, as reported in Forbes [40][1]. In the U.S. Congress, there is legislation that seeks to ban the use of camera-based facial recognition and other biometric surveillance technology by federal law enforcement agencies. On the other hand, a lot of effort has been devoted to preventing/reducing camera's capabilities from obtaining detailed visual data (private information) by equipping cameras with additional hardware/software [41], [42]. In contrast, LiDAR, by definition, captures only point cloud representation of the scene, from which humans' biometric features, such as facial characteristics, hair and skin color, or even clothes, cannot be identified.

Cost: One strength commonly associated with cameras when compared to LiDARs is cost. However, when the system design necessitates optimal levels of privacy, coverage, and varying lighting conditions, the assumed advantage of camera-centric approaches diminishes greatly. A single LiDAR sensor can typically cover roughly four times the area of one camera, significantly decreasing the costs and logistics of installation. Moreover, nowadays, LiDARs are becoming as cheap as only 80$ [43] and reliable in different applications [44]–[48].

Setup: A LiDAR-based solution has setup efficiency and simplicity benefits over camera-based approaches. Using high-quality LiDAR, which generates dense point clouds at longer ranges, enables reliable tracking at scale with fewer devices. Additionally, LiDAR data is much faster and simpler to process, requiring less computing power within a system compared to cameras.

[1]IBM plans to leave the facial recognition business, Amazon is placing a one-year hold on police departments using its facial recognition technology, and Microsoft is waiting on federal legislation before the company starts selling its comparable technology to law enforcement.

IV. System Architecture and Problem Definition

A. Obtaining Sub-Trajectories by Each LiDAR

Each LiDAR can capture a part of the target 3D space, and in each frame (*i.e.* one scan of the space) from the data stream from the LiDAR, a 3D point cloud (or simply point cloud) is obtained. The 3D space where the point clouds are generated by LiDAR i is called *scan space* of LiDAR i and denoted as $\mathbb{S}_i^3$. Then a background subtraction method is applied to that point cloud to extract moving objects. The extracted point cloud is called the foreground point cloud, and segmentation is applied to find each person in the foreground point cloud. We use the Voxel Grid Filter [49] and apply downsampling to the foreground point cloud to convert each voxel grid cell into a single virtual point. Then, a clustering algorithm is applied to the foreground point cloud to segment it into *human segments*. We remove the Z-axis when the clustering is applied to reduce processing overhead and employ a DBSCAN-based clustering to obtain human segments.

A *sub-trajectory* refers to a two-dimensional trajectory obtained as a temporal sequence of the (x, y)-coordinates of human segments, which corresponds to one person's walking trajectory in a LiDAR's scan space $\mathbb{S}_i^3$ (Fig. 1). We note that the prefix "sub-" indicates that the sub-trajectory represents only a part of the whole trajectory of one person. For simplicity, the 2D area where sub-trajectories can be obtained by LiDAR i is called *trajectory area* of LiDAR i and denoted as $\mathbb{S}_i^2$. Generally, $\mathbb{S}_i^2$ is the projection of $\mathbb{S}_i^3$ onto the XY-plane.

We let $TR_i^{[t,t']}$ and H_i^t denote the set of sub-trajectories obtained in a time window $[t, t']$ and the set of human segments at time t by LiDAR i, respectively. We also let

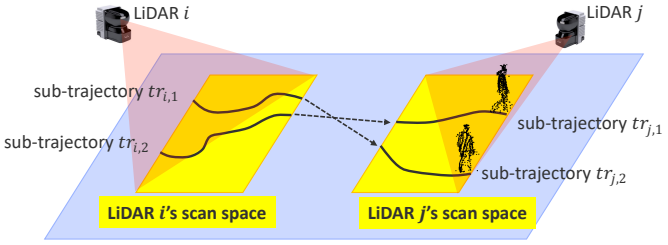


Fig. 1. Sub-trajectories by Distributed LiDARs

$tr_{i,j}^{[t_a,t_b]}$ denote a sub-trajectory j obtained by LiDAR i, which starts at time t_a and ends at t_b. We note that for any sub-trajectory $tr_{i,j}^{[t_a,t_b]} \in TR_i^{[t,t']}$, $t \leq t_a \leq t_b \leq t'$ holds. $tr_{i,j}^{[t_a,t_b]}$ is contained in $\mathbb{S}_i^2$ and their start and end points are on the boundary of $\mathbb{S}_i^2$.

At time t, we obtain from LiDAR i the set H_i^t of human segments. We assume that the set $TR_i^{[t-k,t-1]}$ of sub-trajectories for $k-1$ time window ($k > 1$) has been obtained. Then we find the correspondence between a sub-trajectory $tr \in TR_i^{[t-k,t-1]}$, and a human segment $h \in H_i^{t-1}$. We can easily find this correspondence by applying our Kalman-filter-based tracking method [50], and we obtain an updated $TR_i^{[t-k,t]}$. Finally, we define a temporal relation of two sub-trajectories, $tr_{i,u}^{[t_a,t_b]}$ and $tr_{j,v}^{[t_c,t_d]}$. We denote $tr_{i,u}^{[t_a,t_b]} < tr_{j,v}^{[t_c,t_d]}$ if and only if $t_b < t_c$ holds.

B. Problem Definition

We consider a time window, say $[t_s, t_e]$. We assume that a set L of LiDARs, their scan spaces and trajectory areas, and the map of the target area where those LiDARs are installed are given. We also assume that a sub-trajectory set $TR = \bigcup_{i \in L} TR_i^{[t_s,t_e]}$ is obtained.

The *trajectory estimation problem* in this paper is formulated as a problem to find a partition of TR, where each partition can form a *sub-trajectory sequence*. A sub-trajectory sequence is a temporal sequence of sub-trajectories satisfying the total order relation based on the temporal relation $<$. For example, a partition that contains sub-trajectories $tr_1 = tr_{i,u}^{[1,3]}$, $tr_2 = tr_{j,v}^{[4,6]}$ and $tr_3 = tr_{j,r}^{[10,12]}$, satisfies $tr_1 < tr_2 < tr_3$, and can form the sub-trajectory sequence $tr_1; tr_2; tr_3$.

Once a set of partitions is found, for each partition and the map, we may estimate the path between the subsequent sub-trajectories using the corridors or pathways information contained in the map. However, due to space limitations, this is not in the scope of this paper.

C. Algorithm for Trajectory Estimation

For a given $TR = \bigcup_{i \in L} TR_i^{[t_s,t_e]}$, our algorithm works as follows.

We prepare two sets V_1 and V_2 of sub-trajectories and a set E of sub-trajectory pairs, and all are initially empty. V_1 and V_2 correspond to the sets of those sub-trajectories whose end and start points are the connecting points, respectively. Then for every pair of sub-trajectories $tr_u, tr_v \in TR$, if $tr_u < tr_v$, we add tr_u, tr_v and (tr_u, tr_v) to V_1, V_2 and E, respectively. We also calculate the *affinity value* ($\in [0,1]$) of the pair, which is defined and explained in Section V, and make an weight function $W : E \to [0,1]$.

Finally, we obtain a weighted bipartite graph $G = (V_1 \cup V_2, E, W)$. since $|V_1| = |V_2|$ holds according to the way to build V_1 and V_2, the problem is induced to find the optimal one-to-one matching of V_1 and V_2 over E. This is equivalent to finding the subset E' of E, which maximizes the total sum of the affinity values, as indicated in Eq. (1).

$$E' = \arg\max_{E' \subseteq E} \ \Sigma_{e \in E'} W(e) \tag{1}$$

For this problem, we can employ the Hungarian algorithm with $O(N^3)$ to find the optimal matching [26].

It should be noted that the number of matching candidates, V_1 and V_2, can naturally be smaller if $t_s - t_e$ of TR is smaller. This leads to the design of an online version of the matching algorithm. Specifically, we keep monitoring the sub-trajectories and updating V_1 and V_2, and once $|V_1|$ ($= V_2$) reaches a sufficient number, we can calculate the optimal matching and continue the procedure. The choice of time window size depends on the target applications and services.

V. Sub-trajectory Affinity Calculation

In this section, we define an affinity value for each pair (tr_u, tr_v) of sub-trajectories. To do so, we exploit the following three features, (i) similarity of two human segments (point clouds) from tr_u and tr_v, respectively, (ii) statistical spatial feature (frequency of transitions) from the end point of tr_u to the start point of tr_v, and (iii) statistical temporal feature (traveling time) from the end point of tr_u to the start point of tr_v. The corresponding probabilities (likelihoods) are represented as P_1, P_2, and P_3, respectively, all of which range within $[0,1]$. The affinity value, denoted as $A(tr_1, tr_2)$, is a multiplication of the above probabilities.

$$A(tr_u, tr_v) = P_1 \cdot P_2 \cdot P_3 \tag{2}$$

In Sections V-A, V-B and V-C, we explain how P_1, P_2 and P_3 are calculated, respectively. Besides, we will incorporate the Bayesian system to update the likelihood distributions of P_2 and P_3 as they are based on statistics, *i.e.*, the prior distributions. We explain the update in Section V-D.

A. Similarity of Human Segments

We define the similarity of the two segments and calculate it to judge whether a pair of human segments (human point cloud segments) is from the same person or not. Straightforward adoption of any learning-based similarity scheme is generally inadequate since the point cloud data is usually unordered and unstructured, and the number of points in a segment differs. Accordingly, we design Fisher Vector-based feature extraction and deep metric learning-based similarity calculation to tackle the problem.

1) Feature Extraction: We employ the Fisher Vector (FV) method to extract fixed-size representations of the input human segments. Specifically, FV computes the deviation of a 3D point cloud from the Gaussian Mixture Model (GMM). The intuition behind using FV for feature extraction is its ability to capture the spatial formation of 3D points in space, yielding discriminative signatures of human segments. This can be done by calculating the gradients of the sample's log-likelihood with respect to the GMM model parameters (*i.e.*, Gaussian weight, mean, and covariance). The extracted feature representation of FV has a fixed-size independent of the number of points in a human segment. This advantage makes it easier to process variable-size human segments using a learning-based similarity technique.

Formally speaking, let $X_i = \{\boldsymbol{p}_t \in \mathbb{R}^3,\ t = 1, ..., T\}$ be the set of 3D points of a human segment i, where T denotes the number of points in a segment that dramatically varies depending on different factors, *e.g.*, the LiDAR resolution and range, the scene and the distance. Let us assume that each point comes from one of C different groups, representing body parts such as the head, arms, and legs, and the groups are the Gaussian distributions in a mixture (GMM). Then the set λ of parameters of C component GMM is defined as $\lambda = \{(w_c, \mu_c, \Sigma_c), c = 1, \dots C\}$, where w_c, μ_c, Σ_c are the weight in mixture, mean, and covariance matrix of c^{th} distribution, respectively. Different Gaussians are pre-defined and positioned on 3D grids with equal weights and standard deviations. The likelihood of a single 3D point belonging to the c^{th} Gaussian is:

$$u_c(\boldsymbol{p}) = \frac{1}{(2\pi)^{D/2} \left|\Sigma_c\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{p} - \mu_c\right)' \Sigma_c^{-1} \left(\boldsymbol{p} - \mu_c\right)\right\} \tag{3}$$

The likelihood of a point belonging to the GMM density is defined as:

$$u_\lambda(\boldsymbol{p}) = \sum_{c=1}^{C} w_c u_c(\boldsymbol{p}) \tag{4}$$

Given a specific GMM, and under the common independence assumption [51], the Fisher vector, G_λ^X, can be written as the sum of normalized gradient statistics, computed here for each point p_t:

$$G_\lambda^X = \sum_{t=1}^{T} L_\lambda \nabla_\lambda \log u_\lambda\left(\boldsymbol{p}_t\right) \tag{5}$$

where L_λ is the square root of the inverse Fisher Information Matrix [51]. We change the variables, from w_c to α_c, ensuring that $u_\lambda(x)$ is a valid distribution and simplifying the gradient calculation:

$$w_c = \frac{\exp\left(\alpha_c\right)}{\sum_{j=1}^{C} \exp\left(\alpha_j\right)} \tag{6}$$

Therefore, the normalized gradients can be written as:

$$\mathcal{G}_{\alpha_c}^X = \frac{1}{\sqrt{w_c}} \sum_{t=1}^{T} \left(\gamma_t(c) - w_c\right) \tag{7}$$

$$\mathcal{G}_{\mu_c}^X = \frac{1}{\sqrt{w_c}} \sum_{t=1}^{T} \gamma_t(c) \left(\frac{\boldsymbol{p}_t - \mu_c}{\sigma_c}\right) \tag{8}$$

$$\mathcal{G}_{\sigma_c}^X = \frac{1}{\sqrt{2w_c}} \sum_{t=1}^{T} \gamma_t(c) \left[\frac{\left(\boldsymbol{p}_t - \mu_c\right)^2}{\sigma_c^2} - 1\right] \tag{9}$$

The Fisher vector is formed by concatenating all of these components:

$$\mathcal{G}_{FV_\lambda}^X = \left(\mathcal{G}_{\alpha_1}^X, \dots, \mathcal{G}_{\alpha_c}^X, \mathcal{G}_{\mu_1}^{X'}, \dots, \mathcal{G}_{\mu_c}^{X'}, \mathcal{G}_{\sigma_1}^{X'}, \dots, \mathcal{G}_{\sigma_c}^{X'}\right) \tag{10}$$

To avoid the variation in the number of 3D points in each segment, the resulting FV is normalized by the sample size T:

$$\mathcal{G}_{FV_\lambda}^X \leftarrow \frac{1}{T} \mathcal{G}_{FV_\lambda}^X \tag{11}$$

Additionally, FV ensures that the extracted features are invariant to input permutation by leveraging symmetric functions. More specifically, FV calculates the summation of the gradients, which is a symmetric function. We extend the basic FV by computing additional symmetric functions including minimum and maximum, as inspired by the max-pooling in [52]. This yields a more descriptive and permutation-invariant representation. As a result, each human segment is mapped into a 20×54 feature matrix, where 54 is the number of Gaussians and 20 is the number of features as:

$$FV_\lambda^X = \begin{bmatrix} \sum_{t=1}^{T} L_\lambda \nabla_\lambda \log u_\lambda\left(p_t\right)\Big|_{\lambda=\alpha,\mu,\sigma} \\ \max_t \left(L_\lambda \nabla_\lambda \log u_\lambda\left(p_t\right)|_{\lambda=\alpha,\mu,\sigma}\right. \\ \min_t \left(L_\lambda \nabla_\lambda \log u_\lambda\left(p_t\right)\right)|_{\lambda=\mu,\sigma} \end{bmatrix} \tag{12}$$

2) Similarity Calculation: In order to classify the extracted features based on the similarity between human segments, we use deep metric learning to learn a transformation neural network to the embedding space. As a deep-metric learning method, we use Triplet loss [53]. Triplet loss increases the distance between samples of different classes and decreases it between those of the same class. Cosine similarity is used as the metric function for Triplet loss.

The overall flow is shown in Figure 2, where we use two human segments (input point clouds) to compute Fisher Vectors. The results are then fed into the trained neural network to obtain the "coordinates" in the embedding space. The output is the cosine similarity $cos(x, y)$ of the obtained coordinates x and y. Cosine similarity usually ranges in $[-1, 1]$, but we want to use it in the range of [0.1] for consistency with the other features explained later. Therefore, P_1 is defined as follows,

$$P_1 = \text{similarity}(h_i, h_j) = \frac{\cos(h_i, h_j) + 1}{2} \tag{13}$$

where h_1 and h_2 are human segments in two sub-trajectories of interest, respectively.

B. Spatial Feature

The spatial feature of two sub-trajectories represents how frequently similar transitions occurred in the past. For this purpose, we focus on the boundary of each LiDAR i's trajectory area, $\mathbb{S}_i^2$.

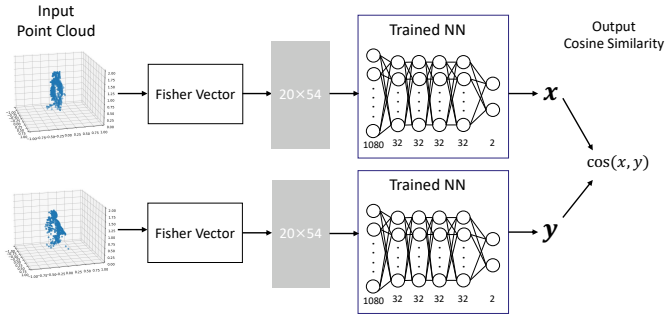


Fig. 2. Similarity Calculation of Two Human Segments

Leveraging the target area map, for each LiDAR i, we may find the parts of the boundary of $\mathbb{S}_i^2$ where pedestrians are likely to enter and leave, which are called *virtual gates*. A virtual gate may be a door, the boundary of $\mathbb{S}_i^2$ on hallways, and so on, and for each sub-trajectory, its start point (or end point) should belong to a virtual gate. Then we build a transition matrix Q where each element is a transition probability from one virtual gate to another.

Given a destination virtual gate, say g_2, to which the starting point of tr_v belongs, we define the spatial feature probability P_2 of sub-trajectories tr_u and tr_v as the ratio of the visit from the virtual gate, say g_1, to which the end point of tr_u belongs over the sum of all the probabilities from the other gates to g_2. This is defined as follows:

$$P_2 = \frac{Q(g_1, g_2)}{\sum_k Q(g_k, g_2)} \tag{14}$$

C. Temporal Feature

Similarly, the temporal feature of two sub-trajectories is defined to represent the likelihood of the traveling time from one virtual gate to another. Let $tr_{i,u}^{[t_a,t_b]}$ and $tr_{j,v}^{[t_c,t_d]}$ denote the two sub-trajectories. The traveling time is obtained as $\Delta t = t_c - t_b$ if $tr_{i,u}^{[t_a,t_b]} < tr_{j,v}^{[t_c,t_d]}$. Assuming prior probability density function $p_{time}(x)$, we can obtain the probability by

$$P_3 = p_{time}(\Delta t) \tag{15}$$

D. Spatial and Temporal Feature Distributions Update

Finally, we describe how to update the transition matrix Q (spatial feature) and probability density function $p_{time}(x)$ (temporal feature). The former is done by a histogram, and the latter is based on the Bayesian system.

These functions should be updated with high confidence during the operation, and one good phenomenon is to believe the case with only one pedestrian traveling from one end point to another starting point and no other pedestrian is observed. This phenomenon may happen in less crowded scenes (*e.g.*, early morning). The transition matrix can easily be updated by the recorded histogram of the past transitions with high confidence. For Bayesian updating of travel time probability distribution, the likelihood of travel time $P(E|H)$ in such a case with high confidence is similar to a normal distribution.

TABLE I
3D LiDAR SPECIFICATIONS

	Livox Avia	Hokuyo YVT-35LX
Maximum number of points (point/frame)	240,000	2,664
Frame rates (frame/s)	10	10
Maximum detection distance (m)	460	35
Horizontal field of view angle (°)	70.4	210
Vertical field of view angle (°)	77.2	40
Distance precision (1σ at 20m) (cm)	$\pm$ 2.0	$\pm$ 0.1
Angular precision (1σ) (°)	$\pm$ 0.05	$\pm$ 2.0

TABLE II
DATASET STATISTICS

	Experiment-1	Experiment-2	Experiment-3
# of observed persons	32 (max)	2,356	15,101
# of sub-trajectories	319 (max)	4,062	19,356
# of switches	287 (max)	1,706	4,255

Therefore, the prior distribution is updated using a Bayesian formula shown in Formula (16).

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{16}$$

Here, the prior distribution of travel time $P(H)$ and the posterior distribution $P(H|E)$ created are both calculated as inverse gamma distributions.

VI. EVALUATION

In this section, we evaluate the proposed system using our testbed. We have installed 70 LiDARs in our university's new campus building, covering from the 1st floor (= ground floor) to the 6th floor. It took almost two years for design, implementation, and installation, and we have just started collecting human trajectory data.

A. System Specification, Environment and Dataset

The specifications of LiDARs are summarized in Table 1.

In this paper, we conducted the following experiments at three on the campus to evaluate (i) basic performance under intended controlled scenarios (at the mid-size indoor square on 2F, **Experiment-1**), (ii) in-situ performance evaluation with high-density, narrow FoV LiDARs (similar to RGB camera) at the mid-size entrance hall on 1F for comparison with the RGB camera-based method, **Experiment-2**), and (iii) in-situ performance evaluation with wide FoV and relatively low-density LiDAR in a long corridor with a lot of lecture rooms (5F, **Experiment-3**). These experimental environments are shown in Fig. 3 and 4.

Comprehensive investigations in Experiment-1 include the accuracy variation according to congestion (number of persons) (**Scenario 1-(a)**), performance improvements with the proposed update method (**Scenario 1-(b)**), component-by-component performance measurements (**Scenario 1-(c)**).

B. Data Collection

The statistics of dataset obtained is described in Table. II

In Experiment-1, we recruited 32 general subjects with different distributions of genders and ages (20's–50's). Each

subject was asked to walk on a designated route among the four turning points. The rectangle by those points is $4m \times 7m$ as shown in Fig. 3(a), 4(a). For each subject, we collected approximately 1000 frames of 3D point clouds. In this experiment, the LiDAR beams are not occluded by other subjects. This means that the sub-trajectories are clearly obtained, and we can evaluate the pure matching performance with complete sub-trajectories.

In Experiment-2, we observed residents and visitors of the building at the entrance hall using 4 LiDARs (Livox Avia). The trajectories, eliminated space (the red rectangle) and trajectory areas are shown in Fig. 3(b), 4(b). In total, 2,356 complete trajectories (*i.e.*2,356 pedestrians) were observed from September 10 to September 13, 2022 (4 days). Most importantly, the comparison with the RGB camera-based method was conducted in Experiment-2.

In Experiment-3, similar to Experiment-2, we observed the residents and visitors of the building on the 5th floor with one corridor with a lot of rooms (Fig. 3(c), 4(c)). However, the different LiDARs (Hokuyo YVT-35LX) are installed on 5F as we need a more horizontal view (wide range). We have used 7 Hokuyo LiDARs to track pedestrians. The key signatures of this experiment are different mobility (longer trajectories), different LiDARs (lower point cloud density compared with the former experiments), and the wider area.

Although almost the entire floor in each experiment is captured by the installed LiDARs, we intentionally eliminate the point cloud in the center area represented by the red rectangle in Fig. 3, and evaluated our method using these area as LiDAR-blank regions. More detailed configurations are explained in the following subsections.

C. Experiment-1: Scenarios and Results

The recruited 32 subjects walked independently, following the same route. Then we synthesized the point clouds of multiple subjects to generate multiple scenarios with the different numbers of subjects with different timings.

1) Scenarios: In **Scenario 1-(a)**, we synthesized, we generated different numbers (2, 4, 8, 16, and 32) of subjects, where we delayed for 10 seconds the start time of the following subjects to make intervals between subsequent subjects. In **Scenario 1-(b)**, we changed the delay time (0, 5, 10, 15, and 20 seconds). This scenario mainly aims to assess the effect of travel time distribution. That is, the shorter the delay time is, the harder it is to distinguish travel time. In **Scenario 1-(c)**, to evaluate the contributions of each features (P_1, P_2 and P_3), the matching is performed only with one of the three features.

In all the scenarios, the matching performance is compared before and after the spatial and temporal features (transition matrix Q and travel time probability distribution $p_{time}(x)$) is updated. As their initial values, all the probabilities in Q and $p_{time}(x)$ are uniform (we assumed a certain range for $p_{time}(()x)$) and the observations obtained through all the scenarios are used to update the both.

2) Result in Scenario 1-(a): We show F-measure for each number of subjects in Figure 5(a), where before and after the updates of the transition matrix and travel time distribution are shown. As the initial distribution is fully uniform (zero knowledge), the accuracy with 32 subjects (this is an extreme (highly-crowded) case where 32 people in $28m^2$ [54])) is around 0.6, but after the update, it becomes much better. With the normal walking speed, 4-8 subjects generate appropriate densities. By looking at the values, the F-measure after the update is 0.8-0.9, which is sufficiently high. Based on the observation above, we chose the four-subject case for Scenario 1-(b).

3) Result in Scenario 1-(b): We also show F-measure for each interval time in Fig. 5(b). With a longer interval, we achieved higher accuracy till 10 sec., which is very natural. We also see a decrease with longer intervals, and this means that a 10-second interval was optimal for making distinguishable features in travel time and patterns. This entirely depends on the scenarios.

4) Result in Scenario 1-(c): We evaluated F-measure of the cases with only one feature of P_1, P_2 and P_3, and the result is shown in Table III. In all the one-feature cases, F-measure values are between 0.7 to 0.8, and with the three features, it is 0.89, which showed the effectiveness of the combination of the features.

5) Effect of Distributions Update: Table IV shows the accuracy before and after updates. The updated F-measure has improved to 0.89. The distribution of affinities for the matched pairs is shown in Fig. 6. Before the update, negative pairs exist in a wide range, converging to lower affinity cases after the update. The distribution of affinities for the pairs with high confidence is shown in Fig. 7(a). We can clearly see high affinities have few negative pairs. Finally, Fig. 7(b) shows how the distribution is updated in traveling from the right top corner to the left top corner in Fig. 3(a). The prior distribution is an orange curve with high variance and the likelihood is the purple curve with the distribution calculated from the travel with high confidence. The posterior distribution is the blue curve, which is the result of the Bayesian update.

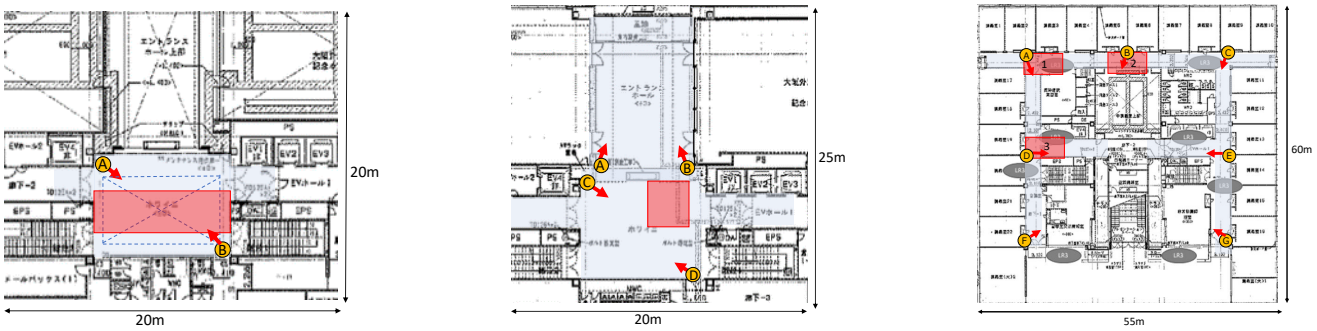
6) Deep Metric Learning Performance: We investigated the basic performance of deep metric learning over 32 subjects. We trained the model on 90% of the randomly selected data

TABLE III
CONTRIBUTIONS OF FEATURES TO ACCURACY (# SUBJECTS=4 AND INTERVAL=10SEC.)

Feature	F-measure (Post-update)
Point Cloud Feature (P_1)	0.74
Spatial Feature (P_2)	0.71
Temporal Feature (P_3)	0.80
Combination ($P_1 \cdot P_2 \cdot P_3$)	0.89

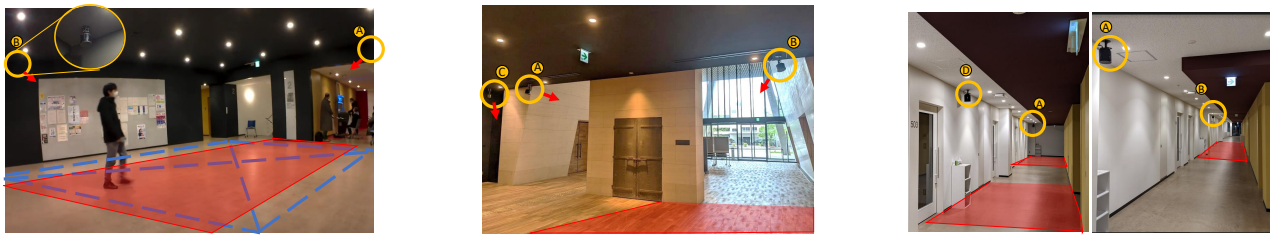
TABLE IV
ACCURACY (PRE- AND POST-UPDATES) (# SUBJECTS=4 AND INTERVAL=10SEC.)

	Precision	Recall	F-measure
Pre-update	0.85	0.84	0.84
Post-update	0.89	0.88	0.89



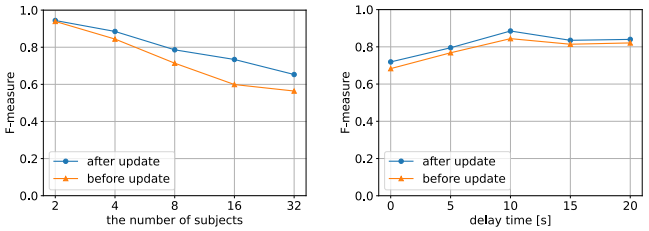
(a) 2nd Floor (Experiment-1) (b) 1st Floor (Experiment-2) (c) 5th Floor (Experiment-3)

Fig. 3. Maps of the experimental environments. Yellow circles and red arrows indicate LiDAR locations and directions, respectively. The total visible range of LiDARs' are filled with light blue, and the eliminated regions for evaluations are illustrated as red rectangles. Each side of the red rectangle corresponds to the "virtual gates". (b) Blue dotted lines represent the real trajectory.



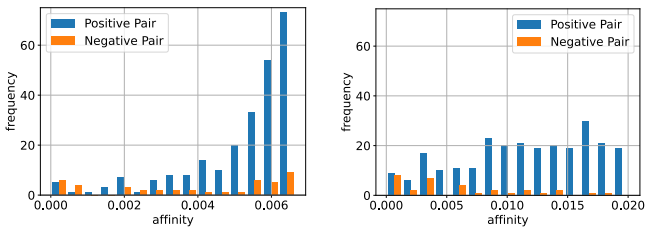
(a) 2nd Floor (Experiment-1) (b) 1st Floor (Experiment-2) (c) 5th Floor (Experiment-3)

Fig. 4. Experimental environments. Lines and regions correspond to the maps.



(a) in Scenario 1-(a) (with Different Number of Subjects, Interval=10sec.)d (b) in Scenario 1-(b) (with Different Time Intervals, # Subjects = 4)

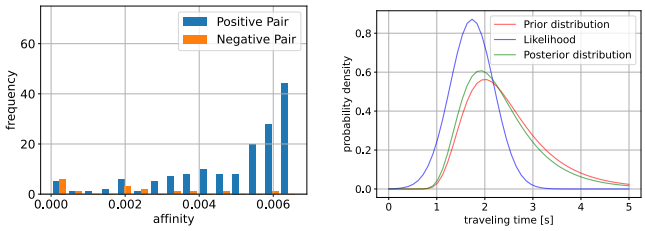
Fig. 5. F-measure variation



(a) Pre-update (b) Post-update

Fig. 6. Affinity Distribution

and tested it on the remaining 10%. The average cosine similarity value of two human segments contained in the test data was calculated, and the result is shown as the matrix in



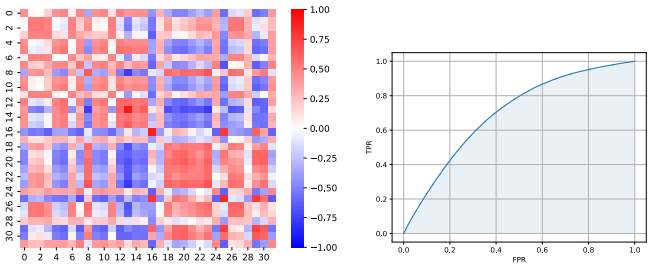
(a) Affinity Distribution of Reliable Data (b) Travel Time Distribution Update

Fig. 7. High Confidence Data for Update

TABLE V
MEAN AND STANDARD DEVIATION OF SIMILARITY VALUES

	mean	standard deviation
similarity between same person	0.57	0.14
similarity between different persons	0.02	0.41

Fig. 8(a) where red cells mean higher similarity values. We can see red cells are seen along the diagonal line from top-left to bottom-right (this means similarity tread is correct), but there are also red cells in different cells. To quantify the result, the means and standard deviations of the similarity between the same and different persons on the diagonal are shown in Table V. From the result, we can clearly see the larger deviation with different subjects. The ROC curve is also shown in Fig. 8(b) where AUC (Area Under the Curve) was 0.71. Generally, AUC above 0.7 means pretty high accuracy.



(a) Similarity Matrix of Subject Pairs (red (close to 1) and blue (close to -1) mean more and less similarity values, respectively)

(b) ROC Curve of Re-id

Fig. 8. Performance of deep metric learning

D. Experiment-2: Comparison with RGB Camera

In Experiment-2, which involved four days of tracking pedestrians in in-situ environment, the accuracy with the proposed method was fairly high (F-measure=0.98) due to fewer crowds as shown in Table. II.

We compared our method with a well-known person re-identification system using RGB cameras [55], using this dataset since the tracking situation in Experiment-2 is similar to when using distributed RGB cameras. LiDARs on the 1st floor are installed in the same position and orientation as the surveillance camera, taking into account the design of the building. Also, the installed LiDAR (Livox Avia) has a relatively similar FoV as the camera. Since it is not possible to implement the same system, we have used the re-id function in [55], which is much more accurate, instead of our point cloud-based re-identification. On the other hand, if we use the camera system, it is usually possible to obtain the transition pattern as the tracking in a single camera is less accurate than single LiDAR-based tracking. Consequently, our method can leverage point cloud features (P_1), transition (P_2) and travel time (P_3), while the camera-based method can use color-based features (P_1^+) and travel time (P_3). The result is shown in Table VI. The proposed method achieved sufficient accuracy with the help of reasonable P_1 and original P_2, while RGB can achieve higher with the power of image-based features. Our method achieved a very good trade-off between privacy and accuracy compared with the well-known camera-based re-id method.

E. Experiment-3: Long Corridor

In Experiment-3, we used Hokuyo LiDARs, which generates 3D point clouds with fewer densities. Therefore, our proposed FV-based re-id is not adequate. Instead, we have used a simpler feature as an alternate of P_1, *i.e.,* the heights of pedestrians. As shown in Fig. 3(c), 4(c), 7 Hokuyo LiDARs are installed. Since the scan areas of neighboring LiDARs are overlapped, similarly with Experiments-1/-2, we eliminated some areas (the three areas by blue rectangles) for this experiment.

We have collected the data from 11 am, Jan. 20th, 2022 till 1 pm, Jan. 24th, 2022. The total number of trajectories was 15,101, and break down is, Jan 20th (Thu.): 3,691, Jan 21st (Fri.): 3,901, Jan 22nd (Sat.): 437, Jan 23rd (Sun.): 5,864, and Jan 24th (Mon.):1,208.

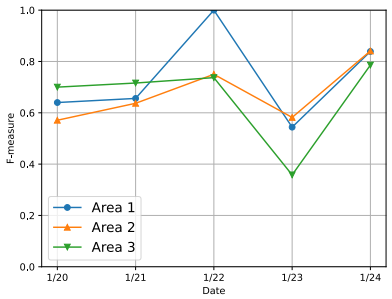


Fig. 9. Accuracy with Hourly Update

We have measured the matching accuracy with hourly updates of distributions. The change of F-measure is shown in Fig. VI-E. The accuracy on the 3rd day (*i.e.,* estimated with the previous two days' distributions) was high, and that in the 4th day (*i.e.,*, estimated with unusual trajectories on Saturday) decreased. Except for those cases, the accuracy increased with updated distributions; on the 5th day, it was 0.80.

VII. Conclusion

This paper proposed a privacy-preserving pedestrian tracking system using multiple distributed LiDARs of non-overlapping views. We deployed the system in a large-scale testbed with 70 colorless LiDARs and conducted three different experiments. The evaluation result on 32 participants confirms the system's ability to accurately track the pedestrians with a 0.98 F-measure even with zero-covered areas.

References

[1] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, 129(4):845–881, 2021.

[2] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proc. of IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[3] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proc. of 22nd Conference on Computer and Communications Security)*, pages 1322–1333, 2015.

[4] John Shackleton, Brian VanVoorst, and Joel Hesch. Tracking people with a 360-degree lidar. In *Proc. of 7th International Conference on Advanced Video and Signal Based Surveillance*, pages 420–426, 2010.

[5] Claudia Álvarez-Aparicio, Ángel Manuel Guerrero-Higueras, Francisco Javier Rodríguez-Lera, Jonatan Ginés Clavero, Francisco Martín Rico, and Vicente Matellán. People detection and tracking using lidar sensors. *Robotics*, 8(3):75, 2019.

[6] Yizhak Ben-Shabat, Michael Lindenbaum, and Anath Fischer. 3d point cloud classification and segmentation using 3d modified fisher vector representation for convolutional neural networks. *arXiv preprint arXiv:1711.08241*, 2017.

[7] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé. MOT20: A benchmark for multi object tracking in crowded scenes. *arXiv:2003.09003[cs]*, March 2020. arXiv: 2003.09003.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[9] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *Proc. of International Conference on Image Processing*, pages 3645–3649, 2017.

TABLE VI
COMPARISON WITH OTHER RE-IDENTIFICATION METHODS

Method	Features to use	F-measure (Pre-update)	F-measure (Post-update)
Ours	Point-cloud Re-ID (P_1) + Accurate Transition (P_2) + Travel Time (P_3)	0.844	0.885
RGB [55]	RGB-based Re-ID (accuracy++ / privacy--) + Travel Time (P_3)	0.851	0.943

[10] H. N. Hu, Q. Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu. Joint monocular 3d vehicle detection and tracking. In *IEEE/CVF ICCV)*, 2019.

[11] Davi Frossard and Raquel Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. In *Proc. of 18th International Conference on Robotics and Automation*, pages 635–642, 2018.

[12] A. Shenoi, M. Patel, J. Gwak, P. Goebel, A. Sadeghian, H. Rezatofighi, R. Martín-Martín, and S. Savarese. JRMOT: A Real-Time 3D Multi-Object Tracker and a New Large-Scale Dataset. In *Proc. of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10335–10342, 2020.

[13] A. Rangesh and M. M. Trivedi. No Blind Spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles Using Cameras and LiDARs. *IEEE Trans. on Intelligent Vehicles*, 4(4):588–599, 2019.

[14] H. Wu, W. Han, C. Wen, X. Li, and C. Wang. 3D Multi-Object Tracking in Point Clouds Based on Prediction Confidence-Guided Data Association. *IEEE Trans. on Intelligent Transportation Systems*, pages 1–10, 2021.

[15] M. Simon, K. Amende, A. Kraus, J. Honer, T. Samann, H. Kaulbersch, S. Milz, and H. M. Gross. Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds. In *CVPR Workshops*, 2019.

[16] X. Weng, J. Wang, D. Held, and K. Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In *Proc. of 2020 International Conference on Intelligent Robots and Systems*, pages 10359–10366, 2020.

[17] L. Spinello, K. Arras, R. Triebel, and R. Siegwart. A Layered Approach to People Detection in 3D Range Data. *Proc. of AAAI Conference on Artificial Intelligence*, 24(1):1625–1630, 2010.

[18] A. Carballo, A. Ohya, and S. Yuta. People Detection Using Range and Intensity Data from Multi-Layered Laser Range Finders. In *Proc. of 2010 International Conference on Intelligent Robots and Systems*, pages 5849–5854, 2010.

[19] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer. Multi-Target Tracking Using a 3D-Lidar Sensor for Autonomous Vehicles. In *Proc. of 16th International Conference on Intelligent Transportation Systems*, pages 881–886, 2013.

[20] S. Song, Z. Xiang, and J. Liu. Object Tracking with 3D LIDAR via Multi-Task Sparse Learning. In *ICMA*, pages 2603–2608, 2015.

[21] Ergys Ristani and Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proc. of 2018 IEEE conference on computer vision and pattern recognition*, pages 6036–6046, 2018.

[22] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *IEEE/CVF CVPR*, pages 8797–8806, 2019.

[23] Hung-Min Hsu, Tsung-Wei Huang, Gaoang Wang, Jiarui Cai, Zhichao Lei, and Jenq-Neng Hwang. Multi-camera tracking of vehicles based on deep features re-id and trajectory-based camera link models. In *CVPR Workshops*, pages 416–424, 2019.

[24] Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, et al. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *Advances in Neural Information Processing Systems*, 33:11309–11321, 2020.

[25] Yukihisa, Ikegame and Makoto, Hirano and Toru, Tamaki and Masanobu, Yamamoto. Probabilistic walking path estimation of a person using multiple cameras with no view duplication. *IEICE Technical Report PRMU2004–186*, 2005.

[26] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[27] Francois Bourgeois and Jean-Claude Lassalle. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804, 1971.

[28] Andreas Mogelmose, Chris Bahnsen, Thomas Moeslund, Albert Clapés, and Sergio Escalera. Tri-modal person re-identification with rgb, depth and thermal features. In *CVPR Workshops*, pages 301–307, 2013.

[29] Guan'an Wang, Tianzhu Zhang, Jian Cheng, Si Liu, Yang Yang, and Zengguang Hou. Rgb-infrared cross-modality person re-identification via joint pixel and feature alignment. In *ICCV*, pages 3623–3632, 2019.

[30] Ancong Wu, Wei-Shi Zheng, and Jian-Huang Lai. Robust depth-based person re-identification. *IEEE Trans. on Image Processing*, 26(6):2588–2603, 2017.

[31] Jingjing Wu, Jianguo Jiang, Meibin Qi, Cuiqun Chen, and Jingjing Zhang. An end-to-end heterogeneous restraint network for rgb-d cross-modal person re-identification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 18(4), 2022.

[32] Tianxu Xu, Dong An, Yuetong Jia, and Yang Yue. A review: Point cloud-based 3d human joints estimation. *Sensors*, 21(5):1684, 2021.

[33] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, pages 1116–1124, 2015.

[34] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *Proc. of 2016 European Conference on Computer Vision*, pages 868–884. Springer, 2016.

[35] Shaojie Qiao, Nan Han, William Zhu, and Louis Alberto Gutierrez. Traplan: an effective three-in-one trajectory-prediction model in transportation networks. *IEEE Trans. on Intelligent Transportation Systems*, 16(3):1188–1198, 2014.

[36] Golnaz Habibi and Jonathan P How. Human trajectory prediction using similarity-based multi-model fusion. *IEEE Robotics and Automation Letters*, 6(2):715–722, 2021.

[37] Yu Fan Chen, Miao Liu, and Jonathan P How. Augmented dictionary learning for motion prediction. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA 2016)*, pages 2527–2534, 2016.

[38] Hamada Rizk, Hirozumi Yamaguchi, Moustafa Youssef, and Teruo Higashino. Gain without pain: Enabling fingerprinting-based indoor localization using tracking scanners. In *SigSpatial*, page 550–559, 2020.

[39] Hamada Rizk, Hirozumi Yamaguchi, Moustafa Youssef, and Teruo Higashino. Laser range scanners for enabling zero-overhead wifi-based indoor localization system. *ACM Trans. Spatial Algorithms Syst.*, 2022.

[40] Glenn Gow. Why are technology companies quitting facial recognition? https://www.forbes.com/sites/glenngow/2020/06/23/why-are-technology-companies-quitting-facial-recognition/?sh=50f1fbb66994.

[41] Carlos Hinojosa, Juan Carlos Niebles, and Henry Arguello. Learning privacy-preserving optics for human pose estimation. In *Proc. of 2021 IEEE/CVF International Conference on Computer Vision*, pages 2573–2582, 2021.

[42] Francesco Pittaluga and Sanjeev Jagannatha Koppal. Pre-capture privacy for small vision sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2215–2226, 2017.

[43] Magikeye ilt lidar. https://www.switch-science.com/products/7281. Accessed: 2022-12-26.

[44] Shota Yamada, Hamada Rizk, and Hirozumi Yamaguchi. An accurate point cloud-based human identification using micro-size lidar. In *Percom Workshops*, pages 569–574, 2022.

[45] Yuma Okochi, Hamada Rizk, and Hirozumi Yamaguchi. On-the-fly spatio-temporal human segmentation of 3d point cloud data by micro-size lidar. In *IE*, pages 1–4, 2022.

[46] Hikaru Katayama, Teruhiro Mizomoto, Hamada Rizk, and Hirozumi Yamaguchi. You work we care: Sitting posture assessment based on point cloud data. In *PerCom Workshops*, pages 121–123, 2022.

[47] Hamada Rizk, Yuma Okochi, and Hirozumi Yamaguchi. Demonstrating hitonavi-: A novel wearable lidar for human activity recognition. In *MobiCom*, page 756–757. Association for Computing Machinery, 2022.

[48] Yuma Okochi, Hamada Rizk, Tatsuya Amano, and Hirozumi Yamaguchi. Object recognition from 3d point cloud on resource-constrained edge device. In *WiMob*, pages 369–374, 2022.

[49] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library. In *Proc. of IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13 2011. IEEE.

[50] Riki Ukyo, Tatsuya Amano, Akihito Hiromori, and Hirozumi Yamaguchi. Pedestrian tracking in public passageway by single 3d depth sensor. In *Proc. of 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events*, pages 581–586, 2022.

[51] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[52] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. of IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[53] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014.

[54] Marija Nikolić, Michel Bierlaire, Bilal Farooq, and Matthieu de Lapparent. Probabilistic speed–density relationship for pedestrian traffic. *Transportation Research Part B: Methodological*, 89:58–81, 2016.

[55] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.