Interpersonal Distance Tracking with mmWave Radar and IMUs

Yimin Dai
School of Computer Science and Engineering
Nanyang Technological University
Singapore

Xian Shuai
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong SAR, China

Rui Tan
School of Computer Science and Engineering
Nanyang Technological University
Singapore

Guoliang Xing
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong SAR, China

ABSTRACT

Tracking interpersonal distances is essential for real-time social distancing management and *ex-post* contact tracing to prevent spreads of contagious diseases. Bluetooth neighbor discovery has been employed for such purposes in combating COVID-19, but does not provide satisfactory spatiotemporal resolutions. This paper presents ImmTrack, a system that uses a millimeter wave radar and exploits the inertial measurement data from user-carried smartphones or wearables to track interpersonal distances. By matching the movement traces reconstructed from the radar and inertial data, the pseudo identities of the inertial data can be transferred to the radar sensing results in the global coordinate system. The re-identified, radar-sensed movement trajectories are then used to track interpersonal distances. In a broader sense, ImmTrack is the first system that fuses data from millimeter wave radar and inertial measurement units for simultaneous user tracking and re-identification. Evaluation with up to 27 people in various indoor/outdoor environments shows ImmTrack's decimeters-seconds spatiotemporal accuracy in contact tracing, which is similar to that of the privacy-intrusive camera surveillance and significantly outperforms the Bluetooth neighbor discovery approach.

KEYWORDS

mmWave radar, IMU, association, tracking

1 INTRODUCTION

Retrospective studies have shown that infectious control measures including wearing masks, hand hygiene, and interpersonal distancing contribute to the prevention of COVID-19 and also to the decline of influenza, enterovirus, and all-cause pneumonia [6]. When the mask-on requirement is gradually lifted during the current stage of the COVID-19 pandemic, interpersonal distancing is important to reducing personal health risks and societal costs in healthcare.

This paper aims to design a system for interpersonal distance tracking for moving people in relatively enclosed environments that require extra attention to airborne transmissions of pathogens via respiratory droplets. The tracking results can be used to detect unsafe contacts and generate real-time or *ex-post* alerts to the engaged individuals. COVID-19 contact tracing often adopts a spatiotemporal definition of contact, i.e., whether a questioned person spent more than τ seconds within x meters from an infectious source, where the thresholds τ and x can be updated according to the evolving understanding on virus transmissions. It has been commonly accepted that risk of transmission is greatest within one meter distance. In addition, SARS-CoV-2 has been found transmissible via a fleeting encounter [24]. The above suggest that effective contact tracing requires decimeters-seconds spatiotemporal accuracy.

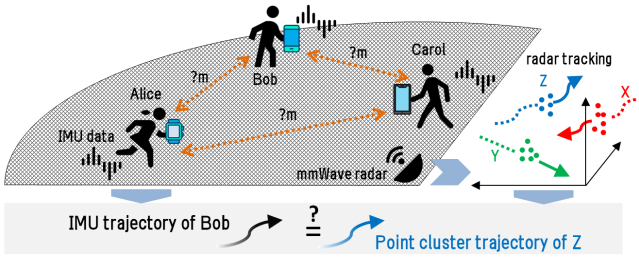


Fig. 1: ImmTrack for interpersonal distance tracking.

Bluetooth neighbor discovery (BND) is the prevailing solution for smartphone- or wearable-based contact tracing [16, 30]. However, as analyzed in [16], BND suffers 1) poor temporal resolution due to long discovery latency and 2) inaccurate distance estimation due to multipath and attenuation effects. As such, the BND-based Google/Apple Exposure Notifications System [1] cannot reliably detect contacts shorter than five minutes [16]. The existing indoor localization techniques are in general incompetent for contact tracing. As summarized in [43], device-free approaches, in which the user does not carry a device, face the *anonymity* problem in tracking multiple users, i.e., the approaches cannot identify individual users. Without (pseudo) identities, the tracking results cannot be used for contact tracing. On the other hand, smartphone-based approaches have respective limitations, e.g., requiring dense Bluetooth beacons, privacy-intrusive due to visual sensing [44], and insufficient accuracy of WiFi- or geomagnetism-based localization [12, 17].

Recently, millimeter wave (mmWave) radars emerged as a low-cost sensing modality and have been adopted for human detection and tracking [38, 42]. The following features of mmWave radars form a good basis for achieving accurate interpersonal distance tracking. First, mmWave radars directly provide the velocity and depth information of the targets, which facilitate tracking the targets' absolute positions. Second, an mmWave radar can cover a large area with good sensitivity. For instance, the Texas Instruments AWR1843 mmWave radar gives a 0.23 m sensing resolution within a 118° circular sector area with a radius of 40 m, covering a total area of more than 1,600 m^2. Third, compared with cameras, mmWave radars output coarse-grained point clouds, which are less privacy-sensitive, making the deployment less intrusive.

However, mmWave radars also face the anonymity problem. Although research has attempted to apply supervised learning to identify the human subjects from mmWave radar data based on gaits [46], training data from each user is needed, incurring undesirable deployment overhead. The key idea of this paper, which is illustrated in Fig. 1, is to exploit the inertial measurement units (IMUs) carried by the users to address the mmWave radar sensing's anonymity problem. This is based on the observations that (i) IMU data from the users inherently carry pseudo identities (PIDs), and (ii) both IMU and mmWave radar data contain rich information about the users' movements. While using IMU data only is insufficient for accurate tracking due to the error accumulation problem, by matching IMU data and mmWave radar data in terms of the consistency between their captured velocities and movements, the IMUs' PIDs can be transferred to mmWave radar's accurate tracking results. The re-identified, radar-sensed user trajectories can then be used for interpersonal distance tracking. Since IMUs are pervasively available on portable and wearable smart devices, the only requirement to enable the tracking is to share a summary of the IMU data regarding the user's movements.

Based on the above idea, we design a system called ImmTrack that employs one or more mmWave radar(s) and exploits the IMUs on the user-carried smartphones or wearables to achieve accurate interpersonal distance tracking. The design of ImmTrack needs to address the following two challenges. First, the point cloud from the radar is usually sparse and noisy [38], making it difficult to separate and track multiple users during their close contacts. Second, as radar and IMU capture different aspects of movements, the cross-modality matching is non-trivial. Specifically, the radar's point cloud indicates user's space occupancy and radial movement of torso, while the IMU time series data captures linear acceleration and angular speed of the IMU-carrying limb. As a result, a common representation of the movement inferred from the two modalities is needed for robust cross-modality matching.

To address the first challenge, ImmTrack clusters the point cloud in a single frame from the radar with initial centroids predicted by Kalman filters that capture the users' motions. The motion-aware clustering effectively prevents the wrong merge of the clusters of two users when they move close to each other. Moreover, we design a deep neural network called *mmClusterNet* to extract each cluster's feature capturing both shape and motion information. Then, the Hungarian algorithm associates the same user's clusters in two consecutive frames based on feature similarity, achieving multi-user inter-frame tracking. To address the second challenge, we employ a novel representation of the user's movement, called *trace map*, which is inferred from either the radar's tracking or the IMU's dead reckoning. We devise a Siamese neural network to extract a comparative feature from the trace map, such that the cosine similarity between two comparative features from the two modalities indicates whether they are from the same user.

We conduct experiments with up to 27 people to evaluate ImmTrack in various environments, including sports hall, lab space, and playground. Compared with the camera-based system, ImmTrack achieves similar user tracking accuracy but only incurs 1/4 to 1/2 computation overhead to process sensor data. For interpersonal distance estimation, ImmTrack achieves an average error of 22 cm. For pinpointing contacts within one meter over two seconds or more, ImmTrack achieves 90% precision and 94% recall. Compared with BND, ImmTrack reduces detection latency by up to 80 seconds. In sum, ImmTrack is suitable for hotspot venues that require extra care in preventing virus transmissions over close contacts.

	AWR1843 radar	A2M8 lidar
Dimension	3D	3D
Range	40m	12m
Resolution	5cm	1cm
Noise	3.2db	15db

Table 1: Comparison between AWR1843 mmWave radar and A2M8 lidar.

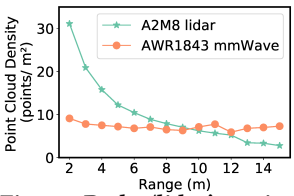


Fig. 2: Radar/lidar's point cloud density versus target range.

The contributions of this paper are summarized as follows.

- We design a motion-aware mmWave radar point cloud clustering algorithm and mmClusterNet neural network for extracting cluster feature, which work together to achieve robust multi-user inter-frame tracking.
- We propose trace map, a new modality-agnostic representation of human movement, and devise a Siamese neural network to extract feature from the trace map for effective mmWave-IMU matching.
- The above two designs make ImmTrack the first system that fuses data from mmWave radar and IMUs for simultaneous user tracking and re-identification. From extensive evaluation with up to 27 people in various environments, ImmTrack achieves decimeters-seconds spatiotemporal accuracy in contact tracing.

Paper organization: §2 presents the background and related work. §3 states the problem. §4 and §5 present the designs of mmWave tracking and cross-modality matching, respectively. §6 presents the evaluation results. §7 concludes this paper.

2 BACKGROUND AND RELATED WORK

2.1 mmWave Radar & Comparison with Lidar

An mmWave radar can output a three-dimensional (3D) Cartesian point cloud of all the targets in the field of view (FoV), where each point is associated with a radial velocity. Lidar and mmWave radar are often competing technologies in various applications. Lidars' higher susceptibility to occlusion, due to their short wavelengths, makes them less suitable for moving people tracking. In addition, we provide a brief comparison between the AWR1843 mmWave radar used in this paper and the A2M8 360° lidar used on a robot to gain more insights. The list prices of these two devices are similar. Table 1 shows their sensing dimensions, ranges, resolutions, and acoustic noise levels during operation. The AWR1843 radar outperforms the A2M8 lidar except on resolution. However, the radar's 5 cm resolution is satisfactory for interpersonal distance tracking. We also measure the two devices' point cloud densities when the target's radial range varies. Fig. 2 shows the result. As radar is mainly based on specular reflection, its point cloud density is insensitive to the target range. Differently, as lidar is mainly based on diffuse reflection, its point cloud density decreases with

the target range. The attenuation may create a challenge in system design. Thus, although lidar-based interpersonal distance tracking is possible, we design ImmTrack based on mmWave radar.

2.2 Related Work

2.2.1 Wireless localization, target identification, and IMU tracking. Wireless indoor localization has received extensive research in the last two decades. Except camera-based surveillance, the device-free approaches in general suffer the anonymity problem in the multi-target setting [43]. Recently, wireless signals are used for designing device-free systems that perform both target tracking and identification. For instance, the studies [40, 48] design human subject identification systems using Wi-Fi signal. However, the used channel state information is unstable in various environments. In addition, the studies [31, 46, 49] train a deep learning model for human subject identification using mmWave radar signal and achieve accuracy above 92%. However, the required extensive training data collection introduces high overheads in practice. Moreover, these systems [31, 46, 49] do not perform well with increased number of human subjects. This is because the distinctiveness among the radar data features is weakened when the number of human subjects increases. The device-based approaches, in which each target carries a signal transmitter/receiver, are free of the anonymity issue. However, as discussed in §1, the device-based approaches using various modalities have respective limitations.

Besides Bluetooth, acoustic sensing is another candidate for neighbor discovery and ranging. The BeepBeep system [32] performs ranging between two smartphones with audible acoustic signals. However, the beeps in continued use is annoying. When adapting to the near-inaudible frequency band, the operational resolution becomes unsatisfactory as the inter-device distance increases, because the smartphone audio systems are not designed to work in the inaudible band. The studies [47] and [10] that use the near-inaudible band manage to evaluate their systems when the inter-device distance is up to 0.4 m and 1.2 m, respectively. Thus, inaudible acoustic ranging is limited to near-field scenarios.

IMU can be used to track user's movements by dead reckoning. Embedding the resulting trajectory into the global coordinate system requires either the global coordinates of at least one point on the trajectory or certain prior knowledge like the spatial constraints expressed in the global coordinate system that any trajectory is subjected to. Dead reckoning suffers from the error accumulation problem. A recent study [45] applies machine learning to improve the accuracy of dead reckoning, which, however, requires massive training data and suffers domain shifts [26]. Therefore, IMU is better for complementing other sensing modalities that can perform localization in the global coordinate system. ImmTrack uses IMUs to re-identify the mmWave radar sensing results. As ImmTrack only requires IMU's short-term dead reckoning result, it is not sensitive to the IMU dead reckoning's error accumulation problem.

2.2.2 Multi-modality data processing. The existing works can be classified into the following three broad categories.

Cross-modality data translation generates synthetic data in the target modality from real data in the source modality. The studies [18, 21, 35] generate synthetic IMU data from videos of human activities. The work [2] generates mmWave radar data from videos. Since computer vision techniques can be employed to recognize the human activities from the videos, the synthetic IMU or mmWave radar data can be automatically labeled and used to train human activity recognition (HAR) models.

Multi-modal data fusion fuses data from complementary modalities at the feature level or score/decision level to improve the robustness of sensing. The work [38] fuses camera and mmWave radar to manage their respective limitations for robust object detection. Fusing camera, lidar, and radar data has been studied in the context of autonomous driving. The milliEgo system [23] improves the accuracy of trajectory reconstruction by fusing mmWave radar data and IMU data in the single-user setting.

Cross-modality data association associates the sensing results in different modalities to increase information about the monitored process. The work [15] matches body-worn IMU data traces with the body joints recognized by a camera. The work [36] applies the same approach to re-identify the body-worn IMUs from the video. The studies [3, 8, 27] associate camera data with Wi-Fi data for various purposes of augmenting the camera with depth information [3] or simultaneous human subject identification and tracking [8]. The work [20] associates users' smartphone Wi-Fi fine timing measurements and IMU data with a camera footage.

ImmTrack belongs to the cross-modality data association category. Different from the existing studies [3, 8, 20, 27, 36] that use camera as an association source, we employ mmWave radar that is less privacy-intrusive. Moreover, technically, mmWave radar directly provides 3D locations and velocities of the human subjects, which facilitate the association.

3 OVERVIEW OF IMMTRACK

3.1 Problem Description and Challenges

We consider an enclosed space that requires extra attention to interpersonal distances, due to say the risk of airborne transmissions of pathogens via respiratory droplets. One or more mmWave radars are deployed to fully cover the space such that any human subject therein can be sensed by the radar(s). The objective of ImmTrack is to track the interpersonal distances among the users in the space. The tracking results can be sent back to the users and/or fed into downstream applications (e.g., contact tracing). When a user is about to enter the space, the user needs to enrol in ImmTrack, e.g., by quick response (QR) code scanning. Certain user PID generation scheme can be used for ImmTrack, depending on the detailed privacy policy. For instance, the ImmTrack mobile app may generate a universally unique identifier (UUID) that takes effect throughout the lifetime of the app and is used as the PID across all ImmTrack-instrumented spaces; or the app may communicate with the ImmTrack server to generate a temporary PID that is unique in the enrolled space. The design of ImmTrack is agnostic to the PID generation scheme. When the user is in the ImmTrack-instrumented space, the ImmTrack mobile app runs in the background and collects IMU data. When the user exits the monitored space, the user needs to sign out. Thus, ImmTrack works in a nearly unobtrusive manner, except the little overhead of signing in and out incurred to the user. Such little overhead is acceptable for specific spaces that require close interpersonal distance monitoring.

The presentation of this paper focuses on a given time period, during which there are N users in the monitored space. Due to the mandatory enrolment, the value of N, although may vary with time, is known by the system at all times. To simplify exposition, the design presentation of ImmTrack focuses on the case that a single mmWave radar is deployed. When a single radar is insufficient to cover the entire space, multiple radars can be deployed. The existing planning algorithms to minimize the number of cameras while achieving visual coverage [11, 13] can be applied to plan the radars' deployment. §6.1.4 and §6.1.5 will present the details of merging the point clouds from multiple radars and the evaluation of multi-radar ImmTrack, respectively. Although the deployment of radar(s) involves a cost, it enables the demanded close interpersonal distance monitoring. Moreover, it is a one-time cost that brings sustained benefits to the users' health and safety.

The design of ImmTrack faces the following two main challenges.

First, robust tracking of multiple users with mmWave radar is challenging. Reflections from unrelated objects may cause excessive noise points in the radar's output point cloud. Moreover, as mmWave reflections are mostly specular, the radar's point clouds are generally sparse. As such, the state-of-the-art object detection and tracking algorithms developed for processing dense point clouds yielded by high-profile lidars are ill-suited for mmWave radars. The mmWave-based multi-user tracking also needs to deal with the users' close encounters and crossings in FoV. The DBSCAN algorithm that is widely adopted for point cloud clustering often mistakenly merges multiple users in proximity into a single cluster. As such, the clustering accuracy decreases drastically with the number of people (45% [22] and 65% [14] for five people). To address this issue, the clustering algorithm should maintain and incorporate the understanding of all users' movements.

Second, robust cross-modality association of the mmWave and IMU tracking results is non-trivial. The two modalities differ in the following two aspects. First, their sensing results are in different coordinate systems. Second, they capture different aspects of the user's movement. The mmWave radar captures the user torso location and velocity with lower frame rates, while the IMU captures the acceleration and angular speed of the user limb with higher frame rates. To achieve robust association, a common feature of the user's movement needs to be derived from both the mmWave data and IMU data. Moreover, the association algorithm needs to accommodate each modality's error in deriving the common feature.

3.2 System Overview

Fig. 3 overviews the design of ImmTrack. It consists of two components to address the above two challenges.

IMU-assisted mmWave tracking: This component consists of three steps. First, it clusters the points in each frame's point cloud into human bodies and associates the clusters corresponding to the same user across frames. ImmTrack maintains a recurive Kalman filter [9] to track each user's movement and uses its predicted user location as the initial centroid of the user's cluster for the clustering algorithm. This motion-aware clustering remains robust when the users encounter each other. Compared with the simplistic mmWave-based target tracking techniques such as that included in the radar vendor's application note [22], our algorithm avoids using

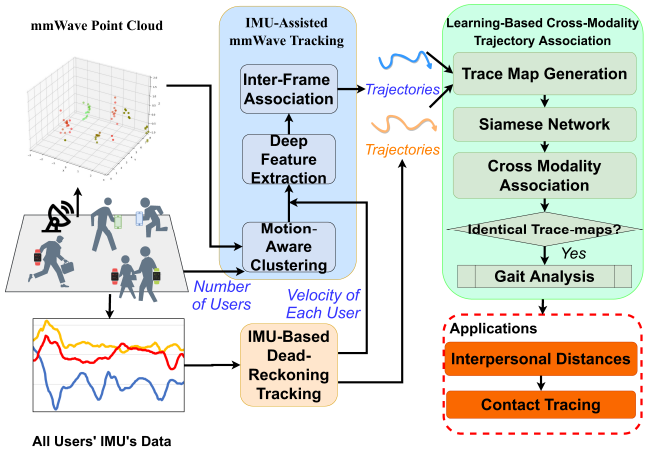


Fig. 3: Overview of ImmTrack. It processes data from one or more mmWave radars and users' IMUs with two components: *IMU-assisted mmWave tracking* and *learning-based cross-modality trajectory association*. The association results are fed to downstream applications.

heuristic object detectors such as constant false alarm rate (CFAR) detector, which easily result in detection errors. Second, to perform the cross-frame cluster association for user tracking, a deep neural network called mmClusterNet extracts the feature of each cluster produced by the clustering algorithm. The feature incorporates the shape and motion information of the point cluster, as well as the PID of a pre-matched IMU in terms of movement velocity. Such multidimensional information improves the robustness of the cross-frame cluster association. Third, the Hungarian algorithm associates the clusters across frames in terms of their features extracted by the mmClusterNet to achieve multi-user tracking. The details are presented in §4.

Learning-based cross-modality trajectory association: ImmTrack adopts the trajectory incorporated with velocity information as the common feature of the user's movement sensed by mmWave radar and IMU. Reasons are two-fold. First, velocity-incorporated trajectory is high-level information that summarizes the user movement and generally remains consistent between the two modalities. Second, trajectory includes both temporal and spatial information. With the temporal continuity embedded in adjacent frames, the noise flickering in single frame can be largely suppressed. After the users' trajectories are reconstructed from the mmWave and IMU tracking, ImmTrack computes an imagery representation of each trajectory, which is called *trace map*. Then, ImmTrack applies a Siamese neural network [37] with convolutional layers to extract comparative features from the trace maps, which are insensitive to the relative relationship between the radar's global coordinate system and the IMU's local coordinate system. Finally, a bipartite graph matching algorithm associates the mmWave and IMU tracking results in terms of the cosine similarity among the comparative features. For users with nearly identical trace maps due to say side-by-side walks or simple straight walks, gait analysis will be performed on the involved mmWave clusters and IMU traces to generate gait features for mmWave-IMU association. The details are presented in §5.

Note that except the IMU trace map generation running on each user's smartphone, all other processing tasks of ImmTrack run on an edge server or a cloud server. The smartphone transmits the periodically generated trace maps to the ImmTrack server.

4 IMU-ASSISTED MMWAVE TRACKING

4.1 Motion-Aware Intra-Frame Clustering

4.1.1 Design. The radar yields a point cloud per frame. For each frame, ImmTrack removes the static points that normally correspond to the background. Specifically, ImmTrack compares each point's velocity with an adaptive velocity threshold updated by the triangle histogram algorithm [19] to decide whether the point is static. ImmTrack adopts the k-means algorithm to divide the point cloud into N clusters by setting $k = N$. Notably, the initial centroids often affect the performance of k-means. ImmTrack uses the recursive Kalman filters (RKFs) [9] to predict the initial centroids.

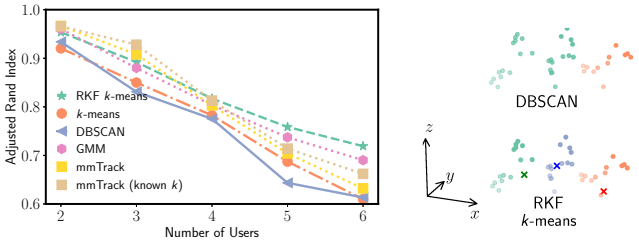
ImmTrack maintains an RKF for each user's volumetric centroid. The human body's kinetic model used by RKF is as follows. Let $\mathbf{x}_{i,j}$ denote the state of the i^{th} user's centroid in the j^{th} frame, where $i \in [1, N]$ is the internal PID in the domain of RKF. Note that this PID is different from the PID of the IMU. We define $\mathbf{x}_{i,j} = [r_{i,j}, \dot{r}_{i,j}, \theta_{i,j}, \dot{\theta}_{i,j}, \phi_{i,j}, \dot{\phi}_{i,j}]^\top$, where $r_{i,j}$, $\theta_{i,j}$, and $\phi_{i,j}$ are the radial range, azimuthal and polar angles, and the overhead dot denotes the velocity. Denote by $\mathbf{c}_{i,j} = [\hat{r}_{i,j}, \hat{\theta}_{i,j}, \hat{\phi}_{i,j}]^\top$ the i^{th} user's observed centroid, where the k-means algorithm fed with the point cloud is viewed as the observation process. By assuming that the user's velocity is constant in a frame duration (denoted by Δt), the state transition and observation models are

$$\mathbf{x}_{i,j} = \mathbf{F}\mathbf{x}_{i,j-1} + \mathbf{w}_{i,j}, \quad \mathbf{c}_{i,j} = \mathbf{H}\mathbf{x}_{i,j} + \mathbf{z}_{i,j}, \tag{1}$$

where $\mathbf{F}$ is the state-transition matrix capturing the movement kinetics, $\mathbf{w}_{i,j}$ is the stationary process noise capturing the uncertainty of the movement, $\mathbf{H}$ is the observation matrix, and $\mathbf{z}_{i,j}$ is the non-stationary observation noise capturing the uncertainties caused by the radar's sensing noises and inaccuracy of the k-means algorithm. Specifically, $\mathbf{F} = \text{diag}(\mathbf{A}, \mathbf{A}, \mathbf{A}) \in \mathbb{R}^{6\times6}$, where $\mathbf{A} = [1, \Delta t; 0, 1]$ and $\mathbf{H}$ is a binary matrix that selects $r_{i,j}$, $\theta_{i,j}$, and $\phi_{i,j}$ from $\mathbf{x}_{i,j}$.

Before processing the j^{th} frame, ImmTrack uses the RKF to predict the i^{th} user's centroid $\widetilde{\mathbf{c}}_{i,j}$ by $\widetilde{\mathbf{c}}_{i,j} = \mathbf{H}\mathbf{F}\mathbf{x}_{i,j-1}$, where $\mathbf{x}_{i,j-1}$ was obtained in the previous frame. When RKF is bootstrapped (i.e., $j = 0$), ImmTrack uses the DBSCAN algorithm to obtain $\widetilde{\mathbf{c}}_{i,0}$. Then, ImmTrack uses $\{\widetilde{\mathbf{c}}_{i,j} | i \in [1, N]\}$ as the initial centroids for the k-means algorithm with $k = N$ to process the point cloud in the j^{th} frame. We sequentially assign the PID of each initial centroid to the closest centroid of a cluster exclusively, forming the pseudo-identified clustering result $\{\mathbf{c}_{i,j} | i \in [1, N]\}$. Finally, ImmTrack uses a policy derived in [9] to update $\mathbf{x}_{i,j}$ and the covariance matrix of $\mathbf{z}_{i,j}$, i.e., $\mathbf{x}_{i,j} = \mathbf{x}_{i,j-1} + \mathbf{K}_{i,j}\left(\mathbf{c}_{i,j} - \mathbf{H}\mathbf{x}_{i,j-1}\right)$ and $\text{cov}(\mathbf{z}_{i,j}) = \text{cov}(\mathbf{M}\mathbf{c}_{i,j} - \mathbf{F}\mathbf{M}\mathbf{c}_{i,j-1}) - \text{cov}(\mathbf{w}_{i,j})$, where $\mathbf{K}_{i,j}$ is the constant Kalman gain and $\mathbf{M} = \left(\mathbf{H}^\top\mathbf{H}\right)^{-1}\mathbf{H}^\top$. We follow the approach described in [4] to estimate $\text{cov}(\mathbf{w}_{i,j})$ used in the above update. The update of $\text{cov}(\mathbf{w}_{i,j})$ enables ImmTrack to adapt to dynamic sensing performance of the radar due to the position variations of users.

Note that the distance-based heuristic rule of transferring the PID of the initial centroids to the resulting centroids of the k-means clustering may have errors when the trajectories of two users cross



(a) Performance of intra-frame clustering algorithms. Baselines: k-means, DBSCAN, GMM, mmTrack and its variant.

(b) Results of RKF-assisted k-means and DBSCAN when $N = 3$.

Fig. 4: Intra-frame clustering. The proposed RKF-assisted k-means clustering algorithm outperforms k-means, DBSCAN, and GMM. It also outperforms mmTrack [42] when $N \geq 4$. In (b), color represents cluster ID, cross represents centroid, and DBSCAN yields 2 clusters for 3 users.

in the radar's FoV. However, since the RKF is mainly used to assist better choosing the initial centroids rather than track the users, the swap of PIDs does not have long-lasting negative effect after the crossing because the models in Eq. (1) are Markovian. Note that tracking the users is the subject of §4.2.

4.1.2 Evaluation. We compare our RKF-assisted k-means algorithm with a variant without RKF and several other clustering approaches including DBSCAN and Gaussian mixture model (GMM) built with the expectation-maximization (EM) algorithm. We also implement the clustering algorithm proposed in mmTrack [42]. The mmTrack applies the k-means algorithm with random initial centroids to cluster the point cloud. During the k-means iterations, mmTrack uses the medoids of the clusters obtained in the previous iteration as the initial centroids of the next iteration. The mmTrack determines the value of k using the silhouette analysis. In addition, we implement a variant of mmTrack's clustering algorithm by removing the silhouette analysis and directly setting $k = N$. All the above baseline approaches do not consider motion.

We use the Adjusted Rand Index (ARI) to measure the quality of clustering. Zero ARI indicates random guessing-like clustering, whereas ARI of one suggests perfect clustering. We compute per-frame ARIs and report the average ARI. During the experiment, the users follow pre-defined trajectories, so that we can obtain the ground truth. More details of the experiment setup are presented in §6. From Fig. 4a, our RKF-assisted k-means outperforms k-means, DBSCAN, and GMM. When $N \leq 3$, the mmTrack and its variant with known k slightly outperform our RKF-assisted k-means in terms of ARI. However, the advantage of our RFK-assisted k-means over mmTrack and its variant increases with N when $N \geq 4$. The explanations for the above results are as follows. When the occlusion cases increase due to the increase of users, our RKF-assisted k-means algorithm outperforms mmTrack. When there are no or limited occlusions, mmTrack's clustering algorithm performs well. However, with our RKF-assisted k-means algorithm, some of the points corresponding to users in the point cloud are excluded in the phase of static points removal, leading to lower ARI. Fig. 4b shows the clustering results of the DBSCAN and RKF-assisted k-means algorithms when $N = 3$, respectively. DBSCAN mistakenly

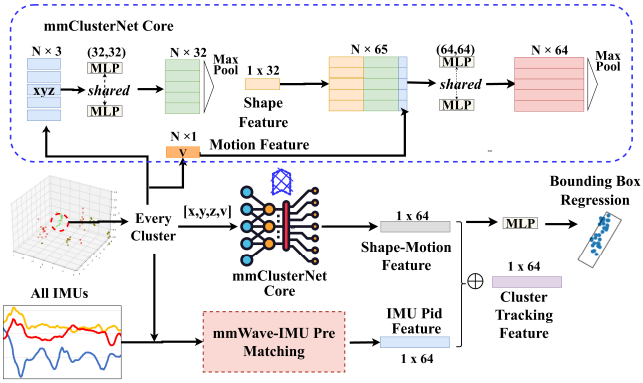


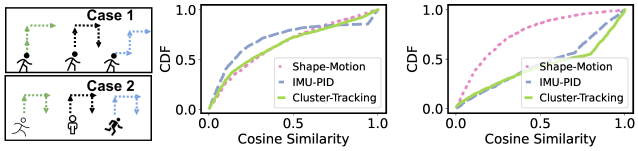
Fig. 5: mmClusterNet for fusing shape, motion, IMU PID features. The distance matrix of fused feature of clusters is used as the metric for inter-frame association and tracking.

combines two users into a single cluster. The above results suggest that the consideration of motion improves clustering performance.

4.2 IMU-Assisted Inter-Frame Cluster Tracking

4.2.1 Design. The association of the clusters in the consecutive frames that correspond to the same user is based on *space coherence* and *motion coherence*. The former means that the shape of a moving object at close locations are similar from the radar's perspective; the latter means that the object's motions in consecutive frames are similar. We design a new deep learning-based feature extractor called *mmClusterNet* that fuses *shape*, *motion*, and *IMU PID* features of a cluster into a single *cluster tracking feature* for each frame.

Fig. 5 shows mmClusterNet's architecture. For each frame, it takes each of the clusters produced by §4.1 as input. The mmClusterNet is designed to process a cluster with n 3D points, where n is fixed at the design phase. When processing a smaller cluster, ImmTrack firstly applies interpolation to generate an n-point cluster. For the AWR1843 mmWave radar, $n = 24$ is a good setting because it is an empirical upper bound of human cluster size. As shown in the upper branch in Fig. 5, each of the n points is processed by a shared multilayer perceptron (MLP) with two 32-neuron hidden layers. The results of the n shared MLPs are max-pooled to generate a 1×32 shape feature, which is copied vertically n times, concatenated with the shared MLPs' outputs and the cluster's radial velocity vector (as the motion feature) to form an $n \times 65$ tensor. Then, each row of the tensor is processed by an MLP with two 64-neuron hidden layers and max-pooling to produce a 1×64 shape-motion feature. Finally, the shape-motion feature is fused with the IMU PID feature, which is detailed shortly, by element-wise addition to produce the cluster tracking feature. To train mmClusterNet, we append a regression MLP as the downstream task that produces a bounding box of the cluster from the shape-motion feature. Then, we use manually labeled bounding boxes as ground truth to train the mmClusterNet core. In §6.3, we will evaluate the impact of different choices of the downstream task on mmClusterNet's performance. Note that the training data for the mmClusterNet core is unnecessary to be *in situ* data. The training can be based on a public dataset such as ShapeNet.



(a) Setup. **(b) Case 1 result.** **(c) Case 2 result.**

Fig. 6: Cosine similarity between features of clusters of same user in two consecutive frames.

The shape-motion feature is directly affected by the radar's sensing noises. Thus, we supplement user-specific static information (i.e., the IMU PID feature) to assist the cluster tracking. Specifically, we perform a *pre-matching* between the clusters generated by the radar and the IMUs, and then use the matched IMU's PID as the user-specific static information. The pre-matching is as follows. First, we compute $\overline{\mathbf{v}}_i$, which is the weighted average 3D velocity of all points in the i^{th} cluster, by $\overline{\mathbf{v}}_i = \sum_{s=1}^{n_i} \frac{projection_{\mathbf{v}_{\mathbf{c}_i}} \mathbf{v}_{i,s}}{\ln n_i \cdot rank(d_s, \{d_1,\ldots,d_{n_i}\})}$, where n_i is the number of points in the cluster, $\mathbf{c}_i$ is the cluster centroid, $\mathbf{v}_{\mathbf{c}_i}$ is $\mathbf{c}_i$'s 3D velocity, $\mathbf{v}_{i,s}$ is the 3D velocity of the s^{th} point of the cluster, d_s is the Euclidean distance between the s^{th} point and the centroid, the operator $projection_{\mathbf{a}}\mathbf{b}$ returns the projection of $\mathbf{b}$ in the direction of $\mathbf{a}$, and the operator $rank(a, A) \in \{1,\ldots,|A|\}$ returns the rank of a in the set A with elements in ascending order. With the reciprocal of rank as the weight, a point closer to the centroid receives a larger weight in the averaging. Using the rank instead of distance as weight for velocity avoids the issue of physical unit conciliation. We apply the coefficient $\frac{1}{\ln n_i}$ to make the sum of the weights to be approximately one, i.e., $\sum_{s=1}^{n_i} \frac{1}{\ln n_i \cdot rank(d_s, \{d_1,\ldots,d_{n_i}\})} \approx 1$. Second, with all clusters' average velocity magnitudes $\{|\overline{\mathbf{v}}_1|,\ldots,|\overline{\mathbf{v}}_N|\}$ and all IMUs' velocity magnitudes denoted by $\{|\mathbf{u}_1|,\ldots,|\mathbf{u}_N|\}$, we apply the Hungarian algorithm to find the one-to-one pre-match between the clusters and IMUs based on Euclidean distance. Let $\text{PID}_i \in \{1,\ldots,N\}$ denote the PID of the IMU pre-matched with the i^{th} cluster. We apply the position encoding [39] to generate the i^{th} cluster's 1×64 IMU PID feature as $[g_1, h_1, g_2, h_2, \ldots, g_{32}, h_{32}]$, where $g_m = \sin\left(\left(\frac{\text{PID}_i}{1000}\right)^{\frac{m}{32}}\right)$ and $h_m = \cos\left(\left(\frac{\text{PID}_i}{1000}\right)^{\frac{m}{32}}\right)$. As presented earlier, the IMU PID feature is added to the shape-motion feature to form the cluster tracking feature.

Given the cluster tracking features obtained in two consecutive frames, the Hungarian algorithm is used to associate one feature in the former frame and one feature in the latter, exclusively, based on cosine similarity. The associated clusters are considered from the same user. In addition, their centroids over time form the trajectory of the user. All trajectories will be input to the trajectory-based association module presented in §5.

4.2.2 Evaluation. We evaluate the advantage of the cluster tracking feature, compared with solely using either shape-motion feature or IMU PID feature. We consider two cases as illustrated in Fig. 6: (1) all users walk at the same speed but follow different paths of different shapes; (2) all users walk at different speeds and follow different paths of the same shape. We measure the cosine similarity between the features of the clusters corresponding to the same user

in two consecutive frames. Fig. 6 shows the cumulative distribution functions (CDFs) of the measured cosine similarities in the two cases. In case (1), the performance of shape-motion feature is similar to cluster tracking feature. In case (2), the performance of IMU PID feature is similar to cluster tracking feature, because the velocity-based mmWave-IMU pre-matching is accurate when users' speeds are different and the matched IMU PID contributes more information than the shape-motion feature. The above results show that the cluster tracking feature takes both the advantages of shape-motion feature and IMU PID feature.

5 LEARNING-BASED CROSS-MODALITY TRAJECTORY ASSOCIATION

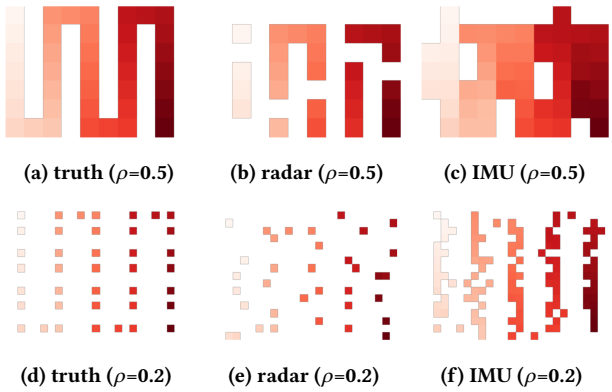
5.1 Design Principle

This module identifies the correspondence among the trajectories reconstructed by the radar in §4 and IMUs via dead reckoning, to re-identify radar's sensing results. Essentially, it is a weighted bipartite matching problem with trajectory similarity as the weight. For association, we use the 2D trajectory (without including the altitude dimension), as it is a common feature that can be derived from both the radar's and IMUs' results and is agnostic to modality-dependent details. For either a radar cluster or an IMU, a trajectory over an *association time window* $[t_0, t_1]$ is denoted by $\mathcal{T}(t) = \{x(t), y(t) | t \in [t_0, t_1]\}$. To compute the similarity between a radar cluster's trajectory $\mathcal{T}_r(t)$ and an IMU's trajectory $\mathcal{T}_i(t)$, the radar's and IMU's 2D coordinate systems need to be registered. A potential method to register the two coordinate systems, both originating at the start points of $\mathcal{T}_r(t)$ and $\mathcal{T}_i(t)$, is to exhaustively search a relative angle between them such that the similarity between $\mathcal{T}_r(t)$ and $\mathcal{T}_i(t)$ under the candidate registration is maximized. However, this registration incurs high compute overhead.

We design a learning-based, registration-free association approach. The main idea is that, instead of considering the distance between two registered trajectories in the same Euclidean space, we take advantage of the feature extraction capability of neural networks to transform trajectories into high-dimensional features, and perform the association based on the distance in the high-dimensional space. Specifically, we first encode the trajectory into an imagery representation, called *trace map*. This is a preparation step that restructures data to a uniform and compact form. Then, we feed *trace maps* from the two modalities into a Siamese neural network for feature extraction, based on whose outputs, the distance matrix can be calculated. Finally, in association, we introduce a soft voting mechanism which aggregates the information of multiple association time windows and thus mitigates the short-time interference. To train the Siamese network, we do not require the ground-truth trajectories. Instead, we extensively construct positive pairs and negative pairs of trajectories, and use a triplet loss to push negative pairs away while bringing together positive pairs, where the only labels required are the matching relationships of the trajectories from the two modalities.

5.2 Trace Map Generation

Let $\mathcal{M} = \{M(x, y) | \forall (x, y)\}$ denote a trace map converted from a trajectory $\mathcal{T}(t)$, where the pixel value $\mathcal{M}(x, y)$ encodes all the times elapsed from when the trajectory crosses the location (x, y).



(a) truth (ρ=0.5) **(b) radar (ρ=0.5)** **(c) IMU (ρ=0.5)**

(d) truth (ρ=0.2) **(e) radar (ρ=0.2)** **(f) IMU (ρ=0.2)**

Fig. 7: Trace maps of ground truth, radar, IMU trajectories.

Let f_s denote the sampling rate in frames per second (fps) of the sensor. Let $T(x, y)$ denote the set of the time instants at which the trajectory crosses (x, y). If $T(x, y) \neq \emptyset$, the map pixel value is given by $\mathcal{M}(x, y) = \sum_{t \in T(x,y)} f_s \cdot (t - t_0)$, where t_0 denotes the time instant that the trajectory starts; otherwise, $\mathcal{M}(x, y) = 0$. Intuitively, $\mathcal{M}(x, y)$ encodes the number of frames passed when the user's trajectory crossed (x, y) since the trajectory begins. Then, ImmTrack converts the obtained trace map into an image with three 8-bit channels of RGB data. We use $\mathcal{M}_r$ and $\mathcal{M}_i$ to denote the color trace maps converted from $\mathcal{T}_r(t)$ and $\mathcal{T}_i(t)$, respectively.

Furthermore, in order to mitigate the impact of noises, we adopt specific spatial grid size ρ for the trace maps. Fig. 7 shows the trace maps of the ground truth, radar, and IMU trajectories under two ρ settings, where a user follows a square zig-zag path to move. A darker red pixel indicates that the trajectory crosses the position more recently. We can see that, due to the inherent uncertainty of sensing, the radar's and IMU's trace maps have deviations from the ground truth. Moreover, under a certain ρ setting, the IMU's trace map has more colored pixels on the trace than the radar's because of IMU's higher sampling rate. As a result, for IMU, setting a smaller ρ can better reduce the crosstalks among different segments of the trajectory, while a larger ρ can make the trace for the radar more continuous. In the rest of this paper, we adopt $\rho = 0.2$ m and $\rho = 0.5$ m for IMU and radar, respectively. Finally, we crop the trace map in an area of 20m $\times$ 20m and resize it to 193×193, which will be fed into the Siamese neural network presented in §5.3.

5.3 Comparative Features Extraction

We design a Siamese neural network to extract comparative features from $\mathcal{M}_r$ and $\mathcal{M}_i$, whose cosine similarity characterizes how close the $\mathcal{T}_r(t)$ and $\mathcal{T}_i(t)$ are. Typically, a Siamese network contains two or more identical sub-networks that extract features from their respective input. During training, any parameter updates are mirrored across all sub-networks. As illustrated in Fig. 8, the Siamese network used by ImmTrack employs a convolutional neural network (CNN) as the feature extractor. The CNN consists of three convolutional layers with rectified linear unit (ReLU) activation followed by max-pooling and a final fully-connected layer producing a 1×1024 feature vector. During training, three such identical CNNs are used to process three inputs, i.e., anchor, positive, and negative inputs.

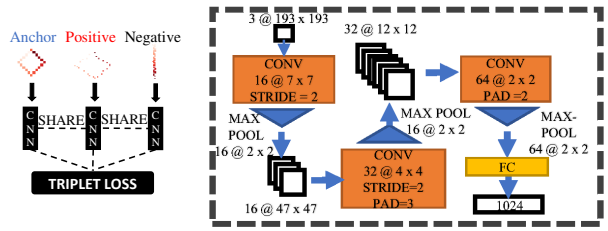


Fig. 8: Left: Siamese network using three identical CNNs with shared weights during training. Right: Architecture of CNN that extracts comparative feature from trace map.

The anchor and positive inputs are two trace maps generated from the radar and IMU for the same user at the same time, while the negative input is an unrelated trace map from either the radar or IMU. Denoting by $\mathbf{f}_a$, $\mathbf{f}_p$, and $\mathbf{f}_n$ the feature vectors produced by the CNN for the anchor, positive, and negative inputs, we use the triplet loss function for training: $\mathcal{L} = \max(\|\mathbf{f}_a - \mathbf{f}_p\|_{\ell_2} - \|\mathbf{f}_a - \mathbf{f}_n\|_{\ell_2} + \text{margin}, 0)$. We also generate simulated trajectories to augment the training data collected in a real environment. Specifically, we use a random walk stochastic process to generate the anchor, and obtain the positive input by scaling up or down the anchor and shifting 10% of the anchor positions to their neighbors. Note that the training data needed by the Siamese neural network is unnecessary to be *in situ* data, because the network only learns extracting environment-agnostic comparative features. At ImmTrack's run time, the trained CNN is used to extract the comparative feature from any given trace map $\mathcal{M}$.

5.4 Cross-Modality Association

For the w^{th} time step in an association time window, ImmTrack constructs a similarity matrix $\mathbf{S}_w \in \mathbb{R}^{N\times N}$, where its $(i,j)^{\text{th}}$ element is the cosine similarity between the comparative feature vectors extracted by the Siamese network from the trace maps of the i^{th} radar cluster and j^{th} IMU, respectively. ImmTrack generates an average similarity matrix, denoted by $\mathbf{S}$, over a total of W consecutive association time windows, i.e., $\mathbf{S} = \frac{1}{W}\sum_{w=1}^{W}\mathbf{S}_w$. Hungarian algorithm is applied to propose an association between the radar clusters and IMUs. If the proposal is accepted, the IMUs' PIDs are transferred to the radar clusters for re-identification. §6.1.2 will show via evaluation that the multi-window similarity averaging improves the robustness of the association, compared with using a single window only.

In addition, ImmTrack applies two criteria to accept an association proposal. If either criterion is not met, ImmTrack excludes the oldest window from the W windows, waits for a new window becoming available, and checks the two criteria again. The two criteria are as follows. **Criterion 1:** For each pair of associated radar cluster and IMU, the similarity between their comparative features needs to be higher than a pre-defined threshold α. This criterion sets a lower bound for the association quality. The α can be set according to the data used to train the Siamese network by $\alpha = \max\{\min_{\forall(\mathbf{a},\mathbf{p})\in\mathcal{P}} S_c(\mathbf{a},\mathbf{p}), \max_{\forall(\mathbf{a},\mathbf{n})\in\mathcal{N}} S_c(\mathbf{a},\mathbf{n})\}$, where $\mathcal{P}$ and $\mathcal{N}$ are the positive and negative pair sets, $S_c(\cdot,\cdot)$ denotes cosine similarity. Our training data gives $\alpha = 0.23$. **Criterion 2:** Any IMU cannot produce the highest cosine similarity with two or more radar clusters among all IMUs. Formally, $\forall i \in [1,N]$, if the $(i,j)^{\text{th}}$ element of $\mathbf{S}$ (denoted by $\mathbf{S}_{i,j}$) is the maximum value within the i^{th} row of $\mathbf{S}$, then $\nexists k \in [1,N]$ such that $\mathbf{S}_{k,j}$ is the maximum value within the k^{th} row of $\mathbf{S}$. This criterion makes sure that the IMU most similar with every radar cluster is unique.

5.5 Handling Users with Identical Trace Maps

Multiple users may generate nearly identical trace maps in certain cases, e.g., when they walk side by side or follow simple straight paths. Within a certain modality, such nearly identical trace maps can be detected by checking their pair-wise similarities. Based on a dataset collected from six human subjects in controlled experiments with pairs of human subjects walking side by side, the detection rates of identifying the side-by-side walk are 92.5% and 77.5% using mmWave radar data and IMU data, respectively, by adopting a threshold of 0.92 on the normalized similarity for the detection. After removing the entries of the $\mathbf{S}_w$ corresponding to the detected identical trace maps, the remaining entries are processed by the cross-modality association presented in §5.4. This section presents a separate cross-modality association approach for the nearly identical trace maps based on gait analysis. ImmTrack initializes the gait analysis if it detects users with nearly identical trace maps from the mmWave radar. The gait analysis for an mmWave cluster is as follows. First, we compute the measured spectrogram $\mathbf{X}_m(v_k,t_l)$ from the Doppler Fourier transform corresponding to the points belonging to the cluster, where v_k and t_l represent the velocity and time bins, respectively. Second, we use the Boulic model [5] to generate the simulated spectrogram $\mathbf{X}_s(v_k,t_k|f_c,l_c,\varphi_c)$, where the parameters f_c, l_c, and φ_c are the specified step frequency, step length, and start phase, respectively. By solving $\arg\min_{f_c,l_c,\varphi_c}\sum_{\forall v_k,t_l}\left\|\mathbf{X}_{\text{m}}^{\log}(v_k,t_l) - \mathbf{X}_{\text{s}}^{\log}(v_k,t_l|f_c,l_c,\varphi_c)\right\|_{\ell_2}^2$, where the superscript "log" means element-wise log normalization, the gait feature (f_c,l_c) is estimated from mmWave radar data. For IMU data, we employ the IMU-based gait analysis [25] to estimate the gait feature (f_c,l_c). Lastly, Hungarian algorithm is applied to associate the mmWave clusters and IMU traces that respectively produce nearly identical trace maps, in terms of the cosine similarity between the mmWave-based and IMU-based gait features. The effectiveness of the mechanism presented in this section will be evaluated in §6.1.5.

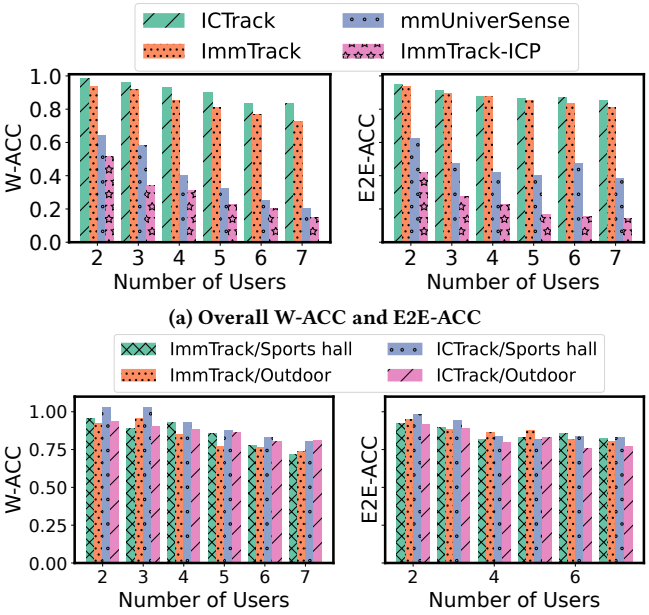
6 IMPLEMENTATION AND EVALUATION

We have implemented ImmTrack using a Texas Instrument AWR1843 mmWave radar hosted by a laptop computer. The users use their own smartphones of various models to participate in the evaluation.[1] We collect IMU data using the MATLAB Mobile app running on the users' smartphones. The sampling rates of the radar and IMU are 8 fps and 100 fps, respectively. The association time window is 12 seconds, with 2-second overlap between two consecutive windows. For cross-modality association, we set $W = 3$, i.e., the similarity matrices in three consecutive association windows are averaged. We primarily conduct experiments in an indoor sports hall and an

[1] Volunteers' participation is under NTU IRB protocol with reference no. IRB-2022-309.

(a) Sports hall setup (b) Subject detection (c) Outdoor setup

Fig. 9: The sports hall and outdoor experiment setups.



(a) Overall W-ACC and E2E-ACC

(b) Sports hall and outdoor W-ACC and E2E-ACC

Fig. 10: Cross-modality association accuracy.

outdoor space as shown in Fig. 9. We also conduct experiments in a lab space as shown in Fig. 13a with up to 27 people.

6.1 Cross-Modality Association Performance

6.1.1 Baselines and evaluation metrics. We employ the following three baseline systems.

■ *ICTrack* is the variant of ImmTrack with mmWave radar replaced by camera. Camera provides much higher resolution than mmWave radar, but causes privacy concerns. ICTrack employs YOLO [34] to detect objects and Deep SORT [41] to associate the bounding boxes of the same object in adjacent image frames. In our implementation, the feature dimension used in Deep SORT for each bounding box is 416. However, Deep SORT does not exploit the prior information of the total number of users (i.e., N). As a result, it often mistakenly creates a new tracking identity for a previously seen user. For fair comparison, we explicitly correct a wrongly created tracking identity by the nearest bounding box in the previous frame. ICTrack generates the 2D trajectory of each detected user from the video stream and executes the cross-modality trajectory association module presented in §5.

■ *ImmTrack-ICP* is the variant of ImmTrack with the Siamese network replaced by *colored-ICP* [29], a colored point cloud registration algorithm. ImmTrack-ICP applies colored-ICP to find the optimal transformation matrix from each trace map of the radar

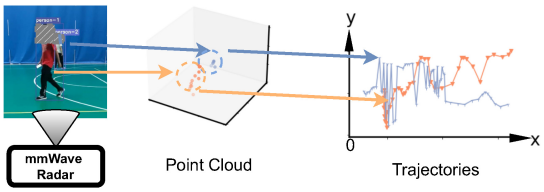


Fig. 11: ImmTrack can track the trajectory of a partially occluded user (marked in blue) correctly with help of IMU.

cluster to each trace map of IMU. ImmTrack-ICP adopts the optimization objective function value of the transformation as the similarity between the trace maps of the radar cluster and IMU.

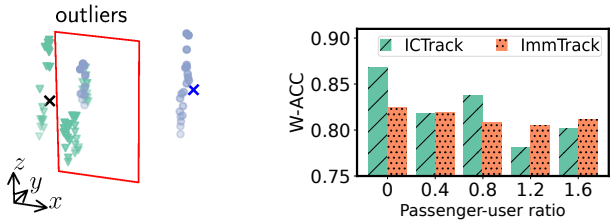
■ *mmUniverSense* is a variant of UniverSense [28] that associates the user's limb movement detected by camera with IMU data based on movement acceleration. We compare UniverSense's single metric-based association with ImmTrack's high-dimensional comparative feature-based association. For fair comparison, we adapt UniverSense to mmWave radar by replacing the acceleration metric with velocity metric, as mmWave radar directly provides velocity data. This adapted version is called mmUniverSense.

Evaluation metrics: We adopt the ratio of correctly associated pairs to all users to characterize the association accuracy. This accuracy in each association time window is denoted by W-ACC, while the accuracy of the association achieved by the average similarity matrix over W windows is called end-to-end accuracy (E2E-ACC).

6.1.2 Association performance in sports hall and outdoor spaces. Fig. 10a presents W-ACC and E2E-ACC of ImmTrack, ImmTrack-ICP, ICTrack, and mmUniverSense on the data collected in the sports hall and outdoor spaces. For each setting of N, the experiment lasts for half an hour. Overall, ImmTrack achieves comparable performance with ICTrack on cross-modality association, while remaining less privacy-intrusive. Specifically, ImmTrack achieves E2E-ACC from 81.4% to 93.6%, while ICTrack achieves 85.4% to 95.1%. On W-ACC and E2E-ACC, ICTrack outperforms ImmTrack by around 7% and 3%, respectively. The accuracy of mmUniverSense is inferior, because when users walk at similar speeds, the association merely based on velocity is prone to be erroneous. ImmTrack-ICP gives the lowest accuracy, which is close to random guessing. For each pair of trace maps from mmWave radar cluster and IMU, the colored-ICP algorithm finds a transformation with small error even if the cluster and IMU are from different users. As a result, all values in the similarity matrix are high and the association process is close to random guessing.

As shown in Fig. 10b, camera-based ICTrack yields higher accuracy indoors than outdoors. Essentially, the performance of ICTrack may degrade in certain environments with dimmed illumination, e.g., in museums with low illumination for protecting ancient artifacts. Differently, ImmTrack yields consistent accuracy, as mmWave radar is robust to different illumination condition.

By analyzing the results of ICTrack, YOLO in ICTrack performs well in detecting humans (as shown in Fig. 9b), while Deep SORT has difficulties in associating bounding box across frames due to the non-coherent visual features of the same user in different frames. Differently, ImmTrack employs extensive features including shape, motion and IMU PID to achieve robust inter-frame cluster tracking. Note that the experiments include cases of inter-person occlusions.



(a) ImmTrack's clustering with passenger causing outliers **(b) W-ACC vs. the ratio of passengers to users**

Fig. 12: Impact of passengers on cross-modality association.

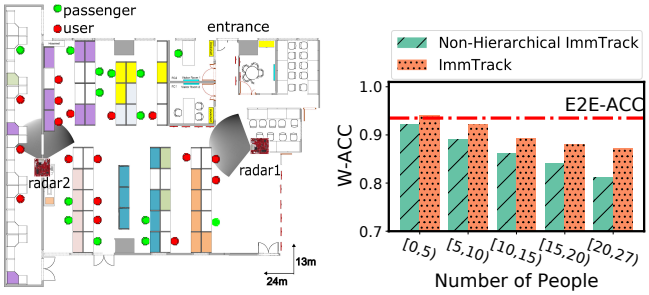
Table 2: Performance improvement by one more radar.

Number of users	2	3	4	5	6	7
W-ACC improvement	2.3%	2.1%	4.0%	2.5%	4.2%	3.8%
E2E-ACC improvement	1.1%	1.2%	1.1%	1.0%	1.7%	1.3%

In Fig. 11, we show that the mmWave radar can still yield some points on the visually occluded user, though with a lower density. This, together with our IMU-assisted design, makes ImmTrack work well in the transient occlusion cases.

6.1.3 Dealing with passengers entering the monitored space. A passenger refers to a person who is within the monitored space but does not participate in the monitoring. For instance, a person whose smartphone is not installed with the ImmTrack app is a passenger. In the presence of passengers, there are outlier points corresponding to the passengers away from the new centroids after the RKF-assisted k-means clustering. To address this problem, ImmTrack views all the points out of the new centroids' bounding boxes as outliers and removes them, where the bounding box size is set to be commensurate to human body dimension. This design is motivated by the fact that the enhanced RKF-assisted k-means algorithm can keep tracking the users even if passengers enter the space, as long as ImmTrack is bootstrapped from a situation with no passenger. Fig. 12a shows ImmTrack's clustering when one out of three people is a passenger. The outlier points away from the centroids represented by crosses are excluded from the clustering result. For fair comparison, we also augment ICTrack to deal with passenger. In specific, we use an asymmetric auction algorithm to perform the M-to-N bipartite cross-modality matching, where M is the total number of people detected by YOLO, and N is the number of users. We measure W-ACC when a certain number (0 to 8) of passengers enter the monitored space, while fixing the number of users at 5. From Fig. 12b, ImmTrack achieves similar or even better W-ACC than ICTrack when there are passengers; the W-ACC of ImmTrack is not sensitive to the passenger-user ratio.

6.1.4 Combining point clouds from multiple radars. Properly combining the point clouds from multiple radars may increase the spatial coverage of a space as well as the point density of a human target seen by multiple radars. In this set of experiments, we deploy two radars with their FOVs' axes of symmetry perpendicular. To accurately combine the two point clouds, we first apply a linear transform including a 90° rotation and origin shift to one point cloud, such that the two point clouds are roughly aligned. Then, we apply the iterative closest point (ICP) algorithm to perform a fine registration of the two point clouds. Table 2 presents the W-ACC and E2E-ACC improvement over varying number of users when two radars are used. With one more radar, there are about 4% and 1% absolute improvements in W-ACC and E2E-ACC, respectively, due to higher point cloud density.



(a) Floor plan; angular coverages of two radars; a snapshot of distribution of human subjects. **(b) W-ACC vs. number of people. Horizontal line indicates mean E2E-ACC over four days.**

Fig. 13: Cross-modality association in a live lab space.

6.1.5 Evaluation in a live lab space. Fig. 13a shows the floor plan. The total area of the space is about 300 m^2. We deploy two mmWave radars to fully cover the corridors and occupied workspaces, while accounting for the blockages caused by internal concrete structures. A total of 17 lab residents voluntarily participate in our evaluation by installing the IMU data collection program on their smartphones. Other lab residents are passengers to our system. During the timespan, the numbers of users and passengers in the lab change. Fig. 13a also shows a snapshot distribution of the users and passengers. We collect data for four consecutive days. In this setup, we observe the users may walk side by side in the corridor. Thus, we particularly evaluate the effectiveness of the mechanism presented in §5.5 for handling identical trace maps. The ImmTrack variant that does not apply the mechanism to separately process the nearly identical trace maps is called *non-hierarchical ImmTrack*. Note that stationary users, who can be detected in both the radar and IMU modalities, are excluded from the processing pipeline, because the workspaces in this lab conform to safe distancing requirement. However, the stationary users' locations and PIDs are maintained in the system. Fig. 13b shows the W-ACC of ImmTrack and the non-hierarchical ImmTrack, versus the total number of people in the monitored area. The x-axis is the number of people in the lab during different testing periods. ImmTrack achieves up to 5.6% higher W-ACC compared with the non-hierarchical ImmTrack. The horizontal line in Fig. 13b shows the mean E2E-ACC of ImmTrack over the entire evaluation period, which is 94.1%.

6.2 Distance Tracking and Contact Tracing

We compare the interpersonal distance tracking performance of ImmTrack with the performance of mmTrack [42]. In addition, we evaluate ImmTrack's performance for contact tracing. We collect a 47-minute trace with mmWave and camera data recorded, where seven users move in the sports hall shown in Fig. 9. We apply ICTrack and manually rectify ICTrack's tracking identities to generate de-anonymized groundtruth trajectories of all the users. In addition,

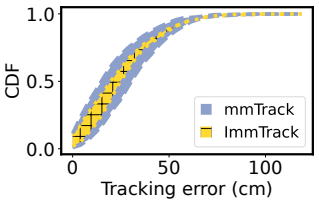


Fig. 14: Statistical analysis of ImmTrack's and mmTrack's tracking errors. The highlighted part of each color represents the area covered by the CDF curve of different users in a system.

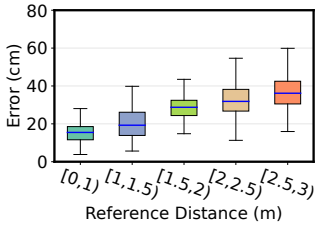


Fig. 15: ImmTrack's tracking error when reference distance is different. The horizontal line in the middle represents the average value of the error.

we project the trajectories to the world coordinate system based on the camera's setup geometry and calculate the interpersonal distance in the global coordinate system as the *reference* to evaluate the accuracy of ImmTrack's interpersonal distance tracking and contact tracing results.

■ **Spatial accuracy of interpersonal distance tracking.** Fig. 14 shows the CDF of ImmTrack's and mmTrack's tracking errors in centimeters with respect to the reference trajectory. For ImmTrack, most tracking errors are within 50 cm. The average tracking error is 22 cm, showing that ImmTrack can achieve re-identified human tracking with decimeters spatial accuracy. Compared with mmTrack, ImmTrack yields more stable tracking accuracy.

For contact tracing, the tracking accuracy is important especially when the actual interpersonal distances are small. Fig. 15 shows ImmTrack's interpersonal distance tracking errors when the reference distance is in different ranges. When the reference distance is within one meter, the tracking errors are within 28 cm and the mean error is 14 cm. The mean error remains under 40 cm when the reference distance is up to 3 m. These results show that ImmTrack can accurately track interpersonal distances in close contacts.

■ **Contact tracing performance.** We consider two definitions of contact: (1) By following a prevailing definition, a *close contact* is a contact with less than 2 m interpersonal distance; (2) An *infectious contact* is a contact with less than 1 m interpersonal distance over τ seconds or more, where we set τ from 2 to 16 seconds. Fig. 16a shows the accumulative close contact time for each pair of users during the 47-minute experiment. It shows that ImmTrack's result and the reference. We can see that ImmTrack gives satisfactory close contact monitoring accuracy. Then, we evaluate ImmTrack's performance in pinpointing infectious contact. We slide a time window of τ+2 seconds with two seconds overlapping and check whether an infectious contact occurs between any two users in the window. By checking against the reference result in each time window, ImmTrack's detection result is among the true/false positive/negative. We measure the *precision* and *recall* by precision = $\frac{\#\text{ of true positives}}{\#\text{ of all ImmTrack's positives}}$ and recall = $\frac{\#\text{ of true positives}}{\#\text{ of all reference's positives}}$. Fig. 16b shows the precision and recall for $\tau = 6$ s when N varies. Note that for each N setting, we conduct a separate experiment that lasts for about 47 minutes. ImmTrack achieves about 90% precision and 91%-96% recall in pinpointing infectious contacts. The opposite trend of recall and precision is due to the increase in the proportion of false negatives in all reference contacts.

Table 3: BND detection delay (s) vs. inter-user distance (m).

Min inter-user distance	[0,1)	[1,2)	[2,3)
Two users	3.2±2.1	4.8±1.7	6.9±3.7
Five users	11.0±4.0	23.9±9.8	42.9±14.1

Table 4: Summary of training datasets & downstream tasks.

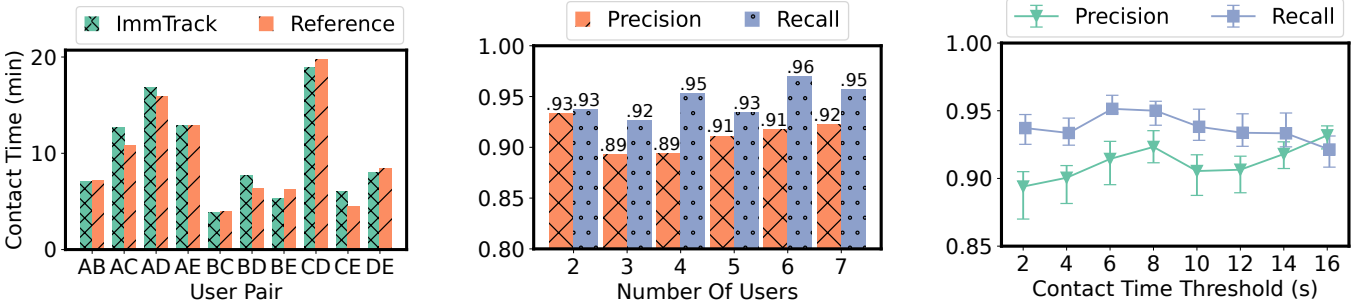
Model	Input	Training dataset	Downstream task
mmClusterNet	Point cloud with velocity	Self-collected	PC
			BBR
			NBBR
PointNet	Point cloud w/o velocity	ShapeNet	OC
			PC

■ **Temporal resolution of contact tracing.** We vary the setting of τ to investigate the temporal resolution of ImmTrack in contact tracing. Fig. 16c shows the precision and recall in pinpointing infectious contact versus the τ setting. While the recall remains stable at around 94%, the precision increases from about 90% to 93% when τ is from 2 to 16 seconds. This shows that ImmTrack can achieve satisfactory temporal resolution fine to 2 seconds with a little contact detection accuracy drop. For comparison, we measure the BND detection delays using two or five Android phones. When using five phones, we place them at vertexes of a pentagon. Table 3 shows the time for a phone to discover all other phones versus the distance between the two phones or side length of the pentagon. The discovery delay increases with the distance and the number of phones. When the distance is one and three meters, the measured worst-case delay is more than 30 and 80 seconds, respectively.

6.3 Training and Efficacy of mmClusterNet

The MLPs used by mmClusterNet to extract the shape-motion feature of a point cloud cluster needs to be trained before use. The training requires a downstream task that utilizes the shape-motion feature. This set of experiments evaluates the impact of various downstream tasks on the training of mmClusterNet. We also compare the cluster tracking feature extracted by mmClusterNet and the feature extracted by PointNet [33], a widely adopted point cloud feature extractor. PointNet takes a point cloud without velocity as input and also needs a downstream task to drive training.

Table 4 summarizes the input data, training datasets, and downstream tasks used to train mmClusterNet and PointNet. Beside the widely adopted point cloud completion (PC), bounding box regression (BBR), and object classification (OC) tasks, we devise a new task called *next-frame bounding box regression* (NBBR), which predicts the 2D bounding box with orientation in the next frame based on the feature extracted from the current frame. The loss functions used by the downstream tasks are as follows: PC uses chamber distance [7]; BBR and NBBR use intersection over union (IoU); OC uses negative log likelihood. We employ the multiple object tracking error (MOTE) and ratio of mismatches (RoM) to jointly measure the inter-frame cluster tracking performance. A mismatch refers to the case that a cluster is associated with another cluster in the previous frame that corresponds to a different user.



(a) Accumulative close contact time estimation between any two users in the 47-minute experiment ($N = 5$).

(b) ImmTrack's precision and recall in pinpointing infectious contacts versus N (τ = 6 s).

(c) ImmTrack's precision and recall in pinpointing infectious contacts under different contact time threshold τ.

Fig. 16: ImmTrack's performance on contact time estimation and pinpointing infectious contacts.

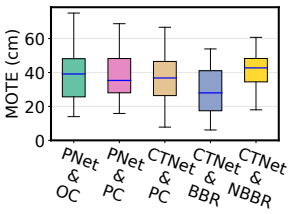


Fig. 17: Multi-object tracking error of inter-frame cluster tracking.

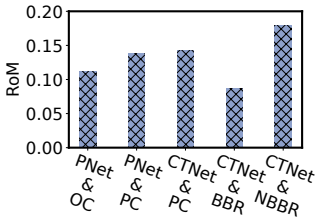


Fig. 18: Ratio of mismatches during inter-frame cluster tracking.

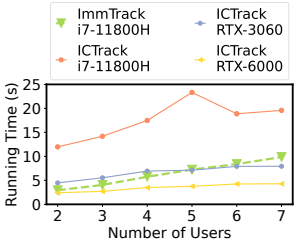


Fig. 19: Runtime latency of ImmTrack and ICTrack with different hardwares.

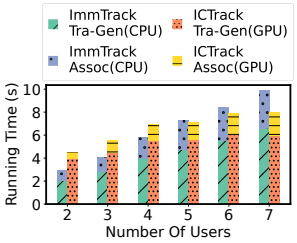


Fig. 20: Time for trace map generation (Tra-Gen) and cross-modality association (Assoc).

The results in Figs. 17 and 18 show that: (1) mmClusterNet outperforms the off-the-shelf PointNet in achieving inter-cluster tracking; (2) BBR is an appropriate downstream task for training mmClusterNet. BBR enforces the model to simultaneously capture cluster contour and enforces utilization of the velocity information of the shape-motion feature. Thus, BBR helps mmClusterNet better learn the shape-motion feature. On the contrary, NBBR leads to poor tracking performance. A possible reason is that NBBR overstretches the utilization of velocity information. ImmTrack evaluated in other sections adopts the mmClusterNet trained with BBR.

6.4 Compute and Communication Overheads

6.4.1 Server computation overhead. Fig. 19 shows the runtime latency of ImmTrack and ICTrack on the server under different N. In general, ImmTrack runnning on an Intel i7-11800H CPU can achieve 30 to 60 fps, depending on the number of users. Note that our ImmTrack implementation adopts a radar sampling rate of 8 fps. Thus, a CPU-only cloud server can support several ImmTrack tasks for different venues, or a CPU-only *in situ* edge server can support a single ImmTrack instance. ICTrack on the same i7-11800H CPU can only achieve about 15 fps processing throughput. Even with a GeForce RTX-3060 or RTX-6000 GPU, ICTrack's processing throughput is still lower than ImmTrack's, because the image processing imposes higher computation overhead than point cloud processing. By jointly considering the accuracy results obtained in §6.1, compared with ICTrack, ImmTrack achieves similar accuracy but only requires 1/4 to 1/2 processing power. Fig. 20 shows the breakdown of the time for processing 90 frames to generate trace map and perform cross-modality association, where generating trace map from radar and camera data takes most of the time.

6.4.2 Smartphone communication and energy overheads. We deploy both the IMU sampling and trace map generation modules on an Android smartphone and measure the overheads. ImmTrack uploads the velocity magnitude to the server for the mmWave-IMU pre-matching. At the end of each association time window, ImmTrack uploads the trace map to the server, which is about 30 KB. The mmUniverSense uploads the 3D velocity continuously. Our measurements show that ImmTrack's and mmUniverSense's bit rates are 7.36 kbps and 15.63 kbps, respectively. ImmTrack's bit rate is lower than the 8 kbps of G.729, an ITU's voice codec for bandwidth-constrained scenarios.

We also compare the battery energy usages of ImmTrack and three existing contact tracing mobile apps, i.e., TraceTogether, LeaveHomeSafe, Coronalert. We run these apps in the background on an Android smartphone for eight hours. We factory-reset the smartphone before each benchmark. ImmTrack keeps sampling IMU, computing trace maps, and uploading data. From publicly available information, Coronalert (which is based on Google/Apple Exposure Notification system) and TraceTogether exchange Bluetooth messages with nearby devices; LeaveHomeSafe is a passive tracing tool based on QR code scanning. During each 8-hour benchmark, we use the tested app to scan valid QR codes every hour to mimic normal daily usages. According to our measurements, battery energy usages of TraceTogether, LeaveHomeSafe, Coronalert are 55.62, 157.04, 37.37 mAh, respectively, while ImmTrack consumes 36.05 mAh. Thus, ImmTrack imposes similar/lower battery energy overhead compared with the existing contact tracing apps.

7 CONCLUSION

This paper presents ImmTrack, an interpersonal distance tracking system using one or more low-cost mmWave radar(s) and the IMUs of the users' smartphones. By associating the users' trajectories reconstructed from the mmWave radar and IMU sensing in terms of the trajectory features extracted by a Siamese neural network, ImmTrack transfers the users' pseudo identities tagged to the IMU data to the radar's global-view sensing results. Extensive experiments with up to 27 people show that ImmTrack achieves similar tracking accuracy and lower computation overhead compared with the more privacy-intrusive camera surveillance. ImmTrack achieves decimeters-seconds spatio-temporal accuracy in tracing contacts, outperforming the prevailing Bluetooth neighbor discovery approach that suffers inaccurate distance estimation and up to 80 seconds discovery delays in our experiments.

ACKNOWLEDGMENTS

This research is supported in part by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RG88/22), in part by the Innovation and Technology Commission of Hong Kong under Grant No. GHP/126/19SZ, and in part by the Research Grants Council (RGC) of Hong Kong under Grant No. 14209619.

REFERENCES

[1] 2022. Google/Apple Exposure Notifications. https://www.google.com/covid19/exposurenotifications/.
[2] Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. 2021. Vid2Doppler: Synthesizing Doppler Radar Data from Videos for Training Privacy-Preserving Activity Recognition. In *CHI*.
[3] Alexandre Alahi, Albert Haque, and Li Fei-Fei. 2015. RGB-W: When Vision Meets Wireless. In *ICCV*.
[4] Gabriel F Basso, Thulio Guilherme Silva De Amorim, Alisson V Brito, and Tiago P Nascimento. 2017. Kalman filter with dynamical setting of optimal process noise covariance. *IEEE Access* 5 (2017), 8385–8393.
[5] Ronan Boulic, Nadia Magnenat Thalmann, and Daniel Thalmann. 1990. A global human walking model with real-time kinematic personification. *Vis Comput* 6, 6 (1990), 344–358.
[6] N.-C. Chiu, H. Chi, Y.-L. Tai, C.-C. Peng, et al. 2020. Impact of wearing masks, hand hygiene, and social distancing on influenza, enterovirus, and all-cause pneumonia during the coronavirus pandemic: retrospective national epidemiological surveillance study. *J. Medical Internet Research* 22, 8 (2020), e21257.
[7] Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *CVPR*.
[8] S. Fang, T. Islam, S. Munir, and S. Nirjon. 2020. EyeFi: Fast Human Identification Through Vision and WiFi-based Trajectory Matching. In *DCOSS*.
[9] Bo Feng, Mengyin Fu, Hongbin Ma, Yuanqing Xia, and Bo Wang. 2014. Kalman filter with recursive covariance estimation—Sequentially estimating process noise covariance. *IEEE Trans. Ind. Electron.* 61, 11 (2014), 6253–6263.
[10] Hao Han, Shanhe Yi, Qun Li, Guobin Shen, Yunxin Liu, and Ed Novak. 2016. AMIL: Localizing neighboring mobile devices through a simple gesture. In *INFOCOM*.
[11] Shibo He, Dong-Hoon Shin, Junshan Zhang, Jiming Chen, and Youxian Sun. 2015. Full-view area coverage in camera sensor networks: Dimension reduction and near-optimal solutions. *IEEE Trans. Veh. Technol.* 65, 9 (2015), 7448–7461.
[12] S. He and K. Shin. 2017. Geomagnetism for smartphone-based indoor localization: Challenges, advances, and comparisons. *ACM Comput. Surv* 50, 6 (2017), 1–37.
[13] Hua Huang, Chien-Chun Ni, Xiaomeng Ban, Jie Gao, and Shan Lin. 2013. Connected wireless camera network deployment with visibility coverage. In *IPSN*.
[14] Xu Huang, Hasnain Cheena, Abin Thomas, and Joseph KP Tsoi. 2021. Indoor Detection and Tracking of People Using mmWave Sensor. *J. Sensors* (2021).
[15] J. Kempfle and K. Van Laerhoven. 2021. Quaterni-On: Calibration-free Matching of Wearable IMU Data to Joint Estimates of Ambient Cameras. In *PerCom*.
[16] Philipp H Kindt, Trinad Chakraborty, and Samarjit Chakraborty. 2021. How reliable is smartphone-based electronic contact tracing for COVID-19? *Commun. ACM* 65, 1 (2021), 56–67.
[17] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using wifi. In *SIGCOMM*.
[18] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. 2020. Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3 (2020), 1–29.
[19] CH Li and Peter Kwong-Shun Tam. 1998. An iterative algorithm for minimum cross entropy thresholding. *Pattern recognition letters* 19, 8 (1998), 771–776.
[20] Hansi Liu, Abrar Alali, Mohamed Ibrahim, Bryan Bo Cao, Nicholas Meegan, Hongyu Li, Marco Gruteser, Shubham Jain, Kristin Dana, Ashwin Ashok, Bin Cheng, and Hongsheng Lu. 2022. Vi-Fi: Associating Moving Subjects across Vision and Wireless Sensors. In *IPSN*.
[21] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. When Video meets Inertial Sensors: Zero-shot Domain Adaptation for Finger Motion Analytics with Inertial Sensors. In *IoTDI*.
[22] Michael Livshitz. 2017. Tracking radar targets with multiple reflection points. *Texas Instruments Application Note* (2017).
[23] Chris Xiaoxuan Lu, Muhamad Risqi U Saputra, Peijun Zhao, Yasin Almalioglu, Pedro PB de Gusmao, Changhao Chen, Ke Sun, Niki Trigoni, and Andrew Markham. 2020. milliEgo: single-chip mmWave radar aided egomotion estimation via deep sensor fusion. In *SenSys*.
[24] Donna Lu. 2021. Covid Delta variant is in the air you breathe. The Guardian.
[25] S. OH Madgwick, A. JL Harrison, and R. Vaidyanathan. 2011. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *ICORR*.
[26] Fangzhi Mu, Xiao Gu, Yao Guo, and Benny Lo. 2020. Unsupervised Domain Adaptation for Position-Independent IMU Based Gait Analysis. In *IEEE Sensors J.*
[27] Le T. Nguyen, Yu Seung Kim, Patrick Tague, and Joy Zhang. 2014. IdentityLink: User-Device Linking through Visual and RF-Signal Cues. In *UbiComp*.
[28] Shijia Pan, Carlos Ruiz, Jun Han, Adeola Bannis, Patrick Tague, Hae Young Noh, and Pei Zhang. 2018. Universense: Iot device pairing through heterogeneous sensing signals. In *HotMobile*.
[29] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Colored point cloud registration revisited. In *ICCV*.
[30] Olaf Landsiedel1 Patrick Rathje1. 2022. traceband:privacy preserving contact tracing on low-power wristband. In *EWSN*.
[31] Jacopo Pegoraro, Francesca Meneghello, and Michele Rossi. 2020. Multiperson continuous tracking and identification from mm-wave micro-Doppler signatures. *GRSS* 59, 4 (2020), 2994–3009.
[32] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. 2007. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *SenSys*.
[33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *CVPR*.
[35] Vitor Fortes Rey, Peter Hevesi, Onorina Kovalenko, and Paul Lukowicz. 2019. Let there be IMU data: generating training data for wearable, motion sensor based activity recognition from monocular RGB videos. In *UbiComp*.
[36] Carlos Ruiz, Shijia Pan, Adeola Bannis, Ming-Po Chang, Hae Young Noh, and Pei Zhang. 2020. IDIoT: Towards ubiquitous identification of iot devices through visual and inertial orientation matching during human activity. In *IoTDI*.
[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.
[38] Xian Shuai, Yulin Shen, Yi Tang, Shuyao Shi, Luping Ji, and Guoliang Xing. 2021. millieye: A lightweight mmwave radar and camera fusion system for robust object detection. In *IoTDI*.
[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.
[40] Fei Wang, Jinsong Han, Feng Lin, and Kui Ren. 2019. Wipin: Operation-free passive person identification using wi-fi signals. In *GLOBECOM*.
[41] Nicolai Wojke and Alex Bewley. 2018. Deep Cosine Metric Learning for Person Re-identification. In *WACV*.
[42] Chenshu Wu, Feng Zhang, Beibei Wang, and KJ Ray Liu. 2020. mmTrack: Passive multi-person localization using commodity millimeter wave radio. In *INFOCOM*.
[43] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M Ni. 2016. A survey on wireless indoor localization from the device perspective. *ACM Comput. Surv* 49, 2 (2016).
[44] Jingao Xu, Hao Cao, Danyang Li, Kehong Huang, Chen Qian, Longfei Shangguan, and Zheng Yang. 2020. Edge assisted mobile semantic visual slam. In *INFOCOM*.
[45] Hang Yan, Qi Shan, and Yasutaka Furukawa. 2018. RIDI: Robust IMU double integration. In *ECCV*.
[46] Xin Yang, Jian Liu, Yingying Chen, Xiaonan Guo, and Yucheng Xie. 2020. MU-ID: Multi-user identification through gaits using millimeter wave radios. In *INFOCOM*.
[47] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-grained acoustic-based device-free tracking. In *MobiCom*.
[48] Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. 2016. WiWho: WiFi-based person identification in smart spaces. In *IPSN*.
[49] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. 2019. mid: Tracking and identifying people with millimeter wave radar. In *DCOSS*.