Anonymous Multi-User Tracking in Indoor Environment Using Computer Vision and Bluetooth

A Thesis

Presented to the

Graduate Faculty of the

University of Louisiana at Lafayette

In Partial Fulfillment of the

Requirements for the Degree

Master of Science

Azmyin Md. Kamal

Fall 2021

Anonymous Multi-User Tracking in Indoor Environment Using
Computer Vision and Bluetooth

Azmyin Md. Kamal

APPROVED:

Raju Gottumukkala, Chair
Assistant Professor of Mechanical Engineering

Terrence L. Chambers
Professor of Mechanical Engineering

Paul Darby
Assistant Professor of Electrical and Computer Engineering

Mary Farmer-Kaiser
Dean of the Graduate School

To my beloved Wife, without whose love and support I could not have come this far.

Acknowledgments

I would like to begin by thanking Allah (SWT) The Most Gracious and the Most Merciful for blessing me with the opportunity to pursue higher education in the United States of America. This journey has certainly been a life-changing experience and it would be very remiss of me not to acknowledge the contributions of my professors, cohorts, friends, and family whose help, kindness, support, and guidance were instrumental in completing this journey.

My deepest gratitude goes to Dr. Raju Gottumukkala, my master's supervisor, thesis committee chair, and my mentor who provided me the resources, support, and guidance in helping me complete my thesis project. I am also thankful to him for making me part of the NSF project in the Informatics Research Institute (IRI) on COVID-19 through which I gained valuable hands-on experience in practical data science applications.

I also express my gratitude and thanks to Dr. Terrance L. Chambers and Dr. Paul Darby for providing me helpful suggestions, comments, and support during the project, patiently listening and providing valuable feedback during my thesis defense as committee members, and providing favorable comments in support of my research findings. Without their gracious support, I would not have been able to graduate in time to start my Ph.D. at Louisiana State University from Fall 2021.

I would also like to thank Dr. Satya Katragadda of Informatics Research Institute who taught me Python programming for data science and helped me solve various challenges and issues I faced during developing the proposed system. I would also like to acknowledge the support and guidance from Dr. Ravi Teja Bhupatiraju of Informatics Research Institute in preparation for the application to the Institutional Review Board (IRB) for using human subjects in the experiment. In this regard, I would like to extend a special thanks to Dr. Hung-Chu Lin, Professor in the Department of Psychology and chair of IRB, for providing her valuable suggestions and comments during the preparation

of the application and then her full support in its expedited processing.

I would like to express my heartfelt gratitude towards Dr. Alan Barhorst, the Head of Mechanical Engineering, who helped me in securing a tuition waiver in Fall 2019 which made it possible for me to come to the United States in the first place.

Lastly, I would like to thank all the undergraduate Research Assistants in the CPS lab and my new friends here in Lafayette for their constant help and support with my research work and, my life here in the United States. Without them, I would not have been able to muster the courage and confidence in completing this degree program and moving onto the next stage of my life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AL	Association Latency
BLE	Bluetooth Low Energy
BLE-IPS	Bluetooth Low Indoor Positioning System
CNN	Convolutional Neural Networks
CV-IPS	Computer Vision Indoor Positioning System
DNN	Deep Neural Network
IPS	Indoor Positioning System
LBS	Location-Based Services
MIPS	Multimodal Indoor Positioning System
MOT	Multi Object Tracker
OLE	Object Localization Error
OTA	Object Tracking Error
PDR	Pedestrian Dead Reckoning
PLBS	Personalized Location-Based Services
RGB	Image containing 3 color channels viz. Red, Green, Blue
RSSI	Received Signal Strength Indicator
RF-IPS	Radio-Frequency based Indoor Positioning System
SORT	Simple Online Realtime Tracking
UWB	Ultra Wide Band
UAT	Unique Association and Tracking
WLAN	Wireless Local Area Network
t_p	A single timestep (ms)
$O^i_{t_p}$	Pseudo id for the i^{th} user
$P^j_{t_p}$	Track id for the j^{th} detected object
B_{t_p}	Output matrix from BLE-IPS
V_{t_p}	Output matrix from CV-IPS
U_{t_p}	Output from MIPS at timestep t_p
(x_{t_p}, y_{t_p})	2D coordinate on continuous space at timestep t_p
$A^i_{t_p}$	Unique pseudo id, track id pair for i^{th} user

Chapter 1 Introduction

Personalized location-based service (PLBS) is one of the most important applications of Indoor Positioning Systems (IPS) that seeks to uniquely distinguish users and provide them accurate location-based contextual information using sensory data obtained from the surrounding environment. Robust and reliable PLBS is a lucrative field in the LBS market since the majority of people spend nearly 90% of their time indoors [1] which gave rise to demands for services such as indoor navigation [2], smart advertisement in retail stores [3], patient-tracking in hospitals, smart homes [4, 5], proximity-tracing during a pandemic [6], augmented reality [7], secure banking [8] and so on. Additionally, IPS which are good at PLBS can also provide important statistics such as user density, energy usage patterns which can improve and optimize space and energy management of the indoor environment itself.

Modern smartphones are the most logical choice for providing location services to users since in recent years, the sophistication of smartphones has increased ten-folds [9]. Often these handheld devices comes equipped with large screens, powerful CPUs, a plethora of communication and sensing, and are relatively inexpensive. Thus, smartphones using IoT technology can form dense interconnected sensory networks ideal for Indoor Positioning Systems. The ubiquitous nature of smartphones has established the notion of tracking a user's smartphone as analogous to tracking the user themselves [10].

Thus, it is no surprise that a large portion of IPS research is focused on smartphone implementation with majority using Radio-Frequency (RF) technologies such as WLAN, Bluetooth Low for localization. RF-IPS are easy to deploy, relatively inexpensive, and require no modification on smartphones to operate [10]. However, these IPS lacks the localization resolution necessary to track people in continuous space due to the instability of radio signals. This prompted the use of "fingerprinting" techniques that localize users in discrete locations by matching recorded signals to known patterns unique to each reference location. Hence, the major theme of RF-IPS research has been concentrated in

improving fingerprinting accuracy [11] but little consideration has been given in securing person-specific identifiers and mobility patterns from third parties [12].

Privacy is a major concern since most modern smartphones can be uniquely identified with their MAC address and location services often run passively in the background. The mobility trace of a person can be constructed by analyzing the historical trace of the smartphone's movement, both of which are unique to the user [13]. Furthermore, incorporating privacy-preserving features has often been a post hoc addition to RF-IPS that had a detrimental effect on the accuracy, latency, and complexity of the system [14].

Vision-based indoor positioning using the overhead camera is an approach to achieve high-resolution localization without using facial features [15, 16]. So these systems are unable to distinguish people at the user level. Moreover, these system have to be trained with annotated images to uniquely identify the user. This would not be a pragmatic solution and privacy for real-world application.

The primary objective of this thesis is to design a Multimodal Indoor Positioning System (MIPS) that can anonymously distinguish users and track them on the continuous space in an indoor environment for use in PLBS. The proposed system tracks the mobile device instead of the person through the fingerprints generated from BLE RSSI values and the overhead camera based localization. The anonymity of the user is maintained through the anonymous ID generated by BLE 4.2 (or BLE 5.0) and unsupervised object detection and localization from the overhead camera. The system uses inexpensive hardware, open-source software, and machine learning techniques to balance anonymization and tracking accuracy. The bi-modal indoor positioning system was both implemented and tested for various scenarios using a human subject study within the cyber-physical systems laboratory. The performance of the proposed system was compared with BLE and video based indoor localization and tracking.

This thesis report is organized as follows: Chapter 2 provides the relevant background study, Chapter 3 provides problem formulation, Chapter 4 provides the overall

system design and working methodology, Chapter 5 provides details of the experiment setup and methods of various components, Chapter 6 discusses the new dataset, metrics and experimental procedures. Chapter 7 provides the results and discussion on key findings, and finally Chapter 8 presents the conclusions and potential future direction.

Chapter 2 Background Study

In this chapter, we first discuss performance requirements of an Indoor Positioning System in the context of our proposed system, followed by the taxonomy of various indoor positioning systems, and review of existing indoor localization technologies. We discuss the concept of Multimodal Indoor Positioning System, and review existing literature in this space. Finally, we provide provide some common challenges associated with designing Indoor Positioning Systems

2.1 Requirements/Performance Evaluation of Indoor Positioning System

The first step in designing an Indoor Positioning System is to define a set of criteria (requirements) to qualitatively and quantitatively measure the performance of the proposed system. There are many competing criteria one needs to balance in the design of indoor-positioning systems that includes identifying the purpose, performance requirements, environmental factors, existing communication infrastructure, and human factors. These topics have been covered extensively in literature. Gu et al. [17] defined 8 metrics with a focus on user preference and their experience namely Security and Privacy, Cost, Performance, Robustness and Fault Tolerance, Complexity, User Preference, Commercial Availability and Limitations for IPS deployed with Personal Networks. Zafari et al. [10] defined 7 metrics namely Availability, Cost, Energy Efficiency, Reception Range, Localization/Tracking Accuracy, Latency, and Scalability as the key requirements applicable to IoT-based Indoor Positioning Systems. Mautz [18] defined 16 competing criteria, key among which are Accuracy, Coverage Area, Cost, Output Data, Privacy, Interface, Number of Users, Scalability, and Level of Hybridization as a general guideline for designing an IPS for mass-market application. Basari et al. [19] defined 6 parameters namely Data Rate, Positioning Accuracy, Coverage, Cost for users, Cost for Infrastructure, and Privacy for IPS in the context of commercial Location-Based Service (LBS). Morar et al. [20]

defined 4 parameters namely Accuracy, Computing Time, Equipment, Properties of Objects in context of vision-based Indoor Positioning Systems. Yassin et al. [11] defined 2 parameters namely Position Accuracy and Range to compare and contrast between multiple commercial positioning techniques that were available in 2016.

From the brief discussion above, it is evident that there are several metrics for evaluating IPS that are common across the studies. Some of them are Accuracy, Latency, Cost, Coverage Area, Scalability, Complexity and Privacy. Additionally, evaluation metrics used in Unimodal IPS are also applicable to MIPS provided certain conditions are met. With these constraints in mind, we present Table 1 to briefly define the requirements/performance metrics applicable to MIPS composed of RF and Computer Vision technologies.

2.2 Classification of Indoor Positioning Systems

A wide variety of classifications have been proposed to classify Indoor Positioning System. One thing to note is that the taxonomies varied quite widely **standard classification** [21]. From recent survey papers, we observed that, many researchers attempted to classifying Indoor Positioning Systems using two categories namely **Technology** and **Techniques** [10, 18, 19, 21–23]. Usually the "Technology" category formed the first tier of classification followed by "Techniques" which grouped a number of tools and techniques employed with a certain "Technology'. Note that, majority of the IPS classified with **Technology** and **Techniques** categories are implemented with RF (WLAN, BLE, UWB) and Inertial Measurement (Accelerometer, Gyroscope, Magnetometer, Barometer) technologies hence not directly compatible with MIPS composed with a Vision-based positioning technology. Additionally, a limited number of published works attempted to formally classify Vision-based IPS with the classification from Morar et al. [20] being the most comprehensive to date. Furthermore, to the best of our knowledge, there exists no formally accepted classification schema for MIPS (MIPS) since MIPS are composed of

Table 1. Evaluation Metrics of an Indoor Positioning System

Metrics	***Definition***
Accuracy	A quantitative measure of how well the system is able to track users movement on a local or global coordinate frame. Depending on the technique used accuracy may be defined differently. For discrete positioning (Fingerprinting technique) systems accuracy is defined as percentage of discrete locations correctly identified whereas for continuous positioning (Vision based tracking) accuracy is defined as the percentage of timestamps in which the IPS was able to keep track of an user correctly. Accuracy is represented as percentage
Localization Error	Also known as Localization Resolution, this metric is a quantitative term which measures the smallest displacement with which an IPS can localize and track movement of an user with high accuracy, usually greater than 90%. A smaller number indicates better IPS. Usually represented as meter or centimeter.
Coverage Area	Defined as the size of the observation area in which the system can reliably detect, localize and track multiple users. Note that coverage area does not necessarily mean the entire Indoor area. Often time, it is a subset of the total available area.
Number of Users	A quantitative measures which indicates how many concurrent users the IPS can serve at a given moment. Higher number is better.
Level of Hybridization	Number of heterogeneous technologies that constitutes an IPS. Note that, having more localization sensors does not necessarily mean more accurate IPS. On the contrary linearly increasing the number of sensing technology exponentially increases the complexity of the IPS technology.
Latency	A quantitative measure of how fast the IPS can provide localization service to multiple users. Latency is measured in milliseconds.
Complexity	A qualitative measures which ranks IPS based on the complexity of tools and techniques used.
Privacy	A qualitative measures which ranks IPS based on its capability to preserve user's data privacy.
Cost	A quantitative measure which ranks IPS based on the cost of equipment and technology used. Usually represented in dollars.

heterogeneous technologies, techniques, infrastructure, and, sensory devices that requires more more than two categories for proper classification. In the following subsections, we briefly discuss some of the proposed taxonomies for IPS and introduce a new classification schema for MIPS.

2.2.1 Classification based on technology. One of the earliest taxonomy for classifying Indoor Positioning System is based on the "Technology" category is from Gu et al. [17] who classified IPS based on the technology used in two tiers. In tier-1, 6 technology groups were defined namely Infrared, Vision-based, Magnetic, Audible Sound, Ultrasound, and Radio Frequency. The Radio Frequency group was further divided into 5 nodes namely Ultra Wide Band, RFID, WLAN, Bluetooth and Sensor Networks. Zafari et al. [10] categorized IPS into 8 technology groups viz. Bluetooth, ZigBee, RFID, Ultrawide Band, Visible Light, Acoustic Signals, and Ultrasound. The authors mainly focused on technologies for localization and did not consider vision-based positioning systems.

Mautz [18] proposed a two-tier taxonomy based on 13 technologies in tier-1 and 4 evaluation metrics in tier-2 with the assumption that Indoor Positioning Systems deployed in different indoor environments but using the same technology can be compared with one another. The technologies defined in tier-1 in this taxonomy are Cameras, Infrared, Tactile, Sound, WLAN, RFID, Ultra Wideband, GNSS, Pseudolites, Other RF, Inertial Navigation, Magnetic Systems, and Infrastructure Systems. In a meta-review, Mendoza-Silva et al. [21] proposed a general-purpose classification taxonomy based on 10 technologies namely Light, Computer-Vision, Sound, Magnetic Fields, PDR, Ultrawide Band, Wi-Fi, Bluetooth Low (BLE), RFID, Cellular, WSN, and Zigbee.

We present Tables 2 and 3 which briefly compares a list of common technologies used in Indoor Positioning system since early 2000. While an exhaustive review of all known IPS technologies is beyond the scope of this work, interested readers are referred to [10] for RF-IPS, [20] for vision-based Indoor Positioning Systems, [11,17] for a general

overview and [22] for cooperative and multi-technology Indoor Positioning Systems.

2.2.2 Classification based on techniques. The IPS performance metrics vary widely with the type of sensing technologies. Hightower et al. [24] proposed the earliest taxonomy for classifying Indoor Positioning Systems based on "Technique" where 15 systems were classified into three categories namely Triangulation, Proximity, and Scene Analysis. Similar one tier classification was also proposed by Liu et al. [25] in 2007 with the addition of a second tier that further stratified Triangulation into two sub-groups namely Laterartion and Angulation. Li et al. [23] also proposed a classification based on localization techniques. In tier-1, there are three groups namely Geometric, Database Matching and Dead-Reckoning. The authors subsumed "Proximity" based on positioning as a group of "Geometrical" indoor positioning systems whereas other authors kept "Proximity" as a separate group. In contrast to the above Mendoza-Silva et al. [21] defined the "Technique" category based on localization signals measured by the receiving devices and proposed categorizing Indoor Positioning Systems into four groups namely Angle of Arrival, Time of Arrival, Time Difference of Arrival and Received Signal Strength (RSS). These taxonomies are primarily for radio-based localization, for computer-vision based Indoor Positioning Systems, Morar et al. [20] did a 4 tier classification taxonomy to include vision-based positioning system, that was further categorized based on technique of image analysis viz. Traditional Image Analysis and Artificial Intelligence.

2.2.3 Proposed taxonomy. In this section we further extend prior taxonomies to include some new elements such as multi-modality, data processing at the edge, and mode of operation. The proposed schema is hierarchical in nature with each subsequent level having a causal relationship to the preceding level as shown in Figure Figure 1.

The 7 technologies chosen within "L1:TECHNOLOGY" are based on the technology trend discussed in recent comprehensive reviews [10, 11, 20, 22]. Wi-Fi, Bluetooth Low (BLE), RFID and Ultra Wideband were the most prevalent radio-based technologies with Inertial Measurement Systems being the most commonly integrated technology

Table 2. Comparison of Technologies in IPS - 1

Name	***Description***
WLAN(Wi-Fi)	IEEE 802.11 standard. Average range - 35 meters. Average localization resolution > 2 meters. Operates on 2.4 Ghz and 5 Ghz frequencies. Common metrics - RSSI, CSI. This technology is supported by most most modern smartphones. Common drawbacks include expensive hardware and complex algorithm makes scaling IPS difficult with this technology. Radio maps are scene specific and accuracy severely affected by phenomena such as NLos, multipath, device heterogeneity. Wi-Fi/WLAN draws more battery power in comparison to BLE, RFID.
Bluetooth Low (BLE)	Upgraded version of Bluetooth 2.0. Effective range between 3-10m. Unlike WLAN, BLE is designed specifically for battery-operated devices and machine-to-machine communication. Operates on 2.4Ghz frequencies on channels 37, 38 and 39. Common metrics - RSSI,AoA, ToF. Inexpensive modules, low power draw compared to WLAN. As RSSI data can be obtained, most algorithms applicable to WLAN are also applicable to BLE based IPS. Battery can last well over a year. Major drawback includes poor performance in micolocalization in comparison to WLAN, slower update rate, severe performance drop in NLoS, multipath, multi-user. RSSI data varies heavily from device to device.
Zigbee	Mean range 10-100m depending on power draw, Operates on unlicensed 2.4Ghz frequency. Common Metrics - RSS, RSSi. Inexpensive, allows directional communication. Can also operate on beacon mode but accuracy in this mode is low. Zigbee's major drawback is its operational frequency interferes with other communication networks in the indoor environment. Draws more power than BLE since it requires constant bidirectional handshake which depletes batteries very fast.
Ultra-Wide Band (UWB)	Ultra-short pulses with time period < 1 nanoseconds. Mean range between 15m - 50m, 460Mbps throughput, low power consumption comparable to BLE. Common metrics - ToA, TDoA. Highly accurate positioning in comparison to WLAN, BLE and Zigbee. Signals can pass through walls and is robust to multipath effect due to short temporal existence. Major drawback lies with requirement of specialized hardware called tags and very accurate timing control which makes UWB hardware very expensive.

Table 3. Comparison of Technologies in IPS - 2

Name	***Description***
RFID	Mean Range 200m. 1.67Gbps throughput, low power draw comparable to BLE and UWB. Requires specialized hardware called tags which are encoded with data that needs to be read by a device. Tags can be of three types. Passive which draws power from the reader device to function, Active which are battery operated and functions similarly to a BLE beacon and, Semi-Passive which upon receiving the reader‘s signal, broadcasts the stored data. While being very inexpensive, RFIDs have very short communication range, approximately 1-2 meters which makes it incompatible for large scale Indoor Positioning applications. This technology integrates very easily with other IPS technologies for Mulitmodal Indoor Positioning Systems. RFID reading hardware is not common in most smart devices.
Ultrasound	Operates on $>$ 20kHz soundwave. Usually has low power draw. Common metric - ToF. Ultrasound based IPS can achieve decimeter level accuracy but only for very specific setup. However, accuracy drops drastically in NLoS conditions and does not generalize well to multi-user scenario.
2D RGB Image	Range depends on camera's field of view and height (for overhead cameras). Common resolutions is 640 by 480 and working FPS is 30. RGB images are represented as 3D tensor and are readily usable with Deep Neural Networks. Using GPU on certain IoT devices, images can be processed to localize and track multiple users at centimeter level accuracy with high reliability. High quality cameras in modern smartphones greatly alleviate need for expensive camera which makes this technology suitable for large scale application. Major drawback lies in the fact that object tracking is not possible without continuous image stream but continuous image acquisition drains battery very fast in mobile devices. Deep Learning algorithm using vision requires a substantially large dataset which is labor and time intensive to prepare. Facial image based person recognition is privacy-invasive but without facial images, this technology cannot directly distinguish an person uniquely from a group.
Cellular	This technology encompasses GSM, LTE, 5G and other form of mobile communication which are not specifically designed for Indoor applications. In fact Cellular networks are never used by themselves since Cellular IPS relies heavily on strong RSS from cell towers which seldom reaches the target device in indoor environment. Positioning accuracy is well-over 50 meter which makes it further incompatible for indoor applications. Common metric is RSS.

with radio-based Multimodal Indoor Positioning. Computer Vision and Infrared Depth Measurement are also included into the strata since a substantial number of recent IPS are vision-based that determines users location by extracting location information from 2D RGB images or point-cloud generated by depth cameras using Machine Learning and Deep Learning techniques.

"L2:NUMBER OF SENSORS" is the next logical level in the taxonomy which

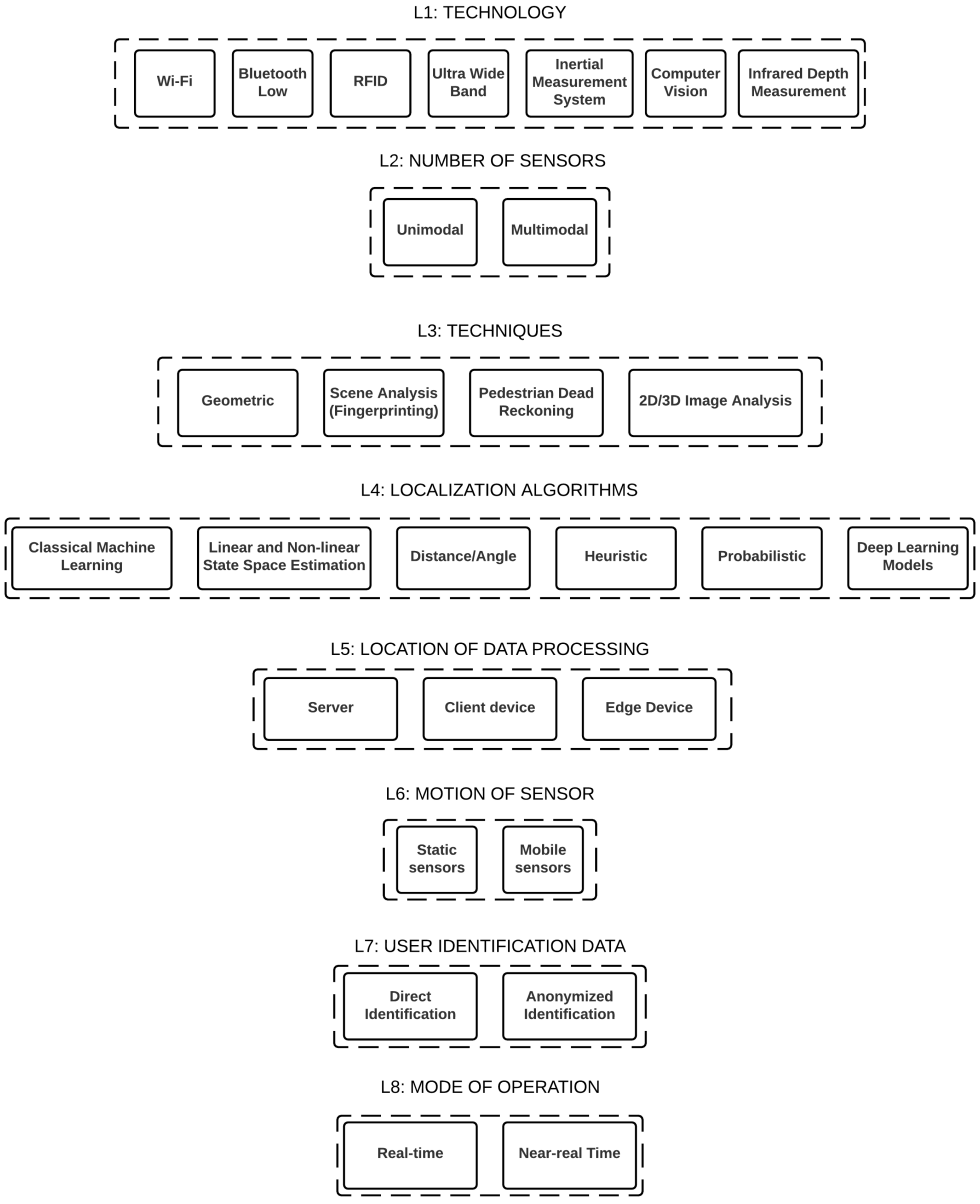


Figure 1. Proposed Classification Schema for Indoor Positioning Systems

takes into account the number of heterogeneous technologies constituting a Unimodal/Multimodal Indoor Positioning System. Our justification for considering the number of technologies first rather than following the conventional rule of placing "TECHNIQUES" in second level is the fact that, the presence of heterogeneous sensors directly dictates the techniques and algorithms that can be used in a positioning system.

For example, Faragher and Harle in [26] developed one of the first Bluetooth-Low based Indoor Positioning System using only BLE Beacons and Android smartphones. Here the authors used two algorithms viz. Proximity and k-Nearest Neighbour. Murata et al. [2] developed a Multimodal Indoor Positioning system as an assistive technology for the visually impaired with BLE and IMU for a large-scale multi-floor navigation application. Here the authors used a modified probabilistic technique that fused data from BLE and IMU to correctly predict which floor a user is currently situated on.

In [26], the authors where only restricted to using techniques from "Geometric" or "Scene Analysis" group in Level L3 (refer Figure 1) whereas the authors in the [2] were free to use techniques under "Pedestrian Dead Reckoning" along with techniques from either "Geometric" or "Scene Analysis" group to implement their solution. Having heterogeneous technologies raises the complexity and cost of the system which gives further credibility to classify a proposed positioning system with the number of sensors first before deciding upon which techniques to use for implementation.

Levels L3 and L4 depicts the "TECHNIQUES" and associated "LOCALIZATION ALGORITHM" groups which can be used to further classify an Indoor Positioning System based on its constituent sensing technology and algorithm used determine an user's position. Note that, a group in L3 may be associated with multiple groups in L4. For instance, supervised classification algorithms such as kNN, SVM, Random Forest from "Machine Learning" and RNN, Autoencoders from "Deep Learning Models" in level L4 can be attributed to level L3 "Scene Analysis" group.

Level L5 "LOCATION OF DATA PROCESSING" strata takes into account the

location where data is processed for obtaining localization information. Three groups are considered in this strata as shown in Figure 2. Blue solid lines indicates constant bi-directional communication while dotted blue lines indicates periodic communication.

The "Server" group (Figure 2a) classifies those Indoor Positioning System which requires client devices to offload raw sensory data to an offsite server for running localization algorithms [27,28]. IPS falling under this category usually achieves near real-time performance but at the cost of increased infrastructure cost, communication complexity and suffers from major privacy concerns since the data sent out from user's smart device often contains a person-specific identifier (name, facial image, smartphone's MAC address) which can be then be used to create a detailed profile of an individual that can be used to compromise people privacy and security [14]. These IPS are predominantly centralized systems.

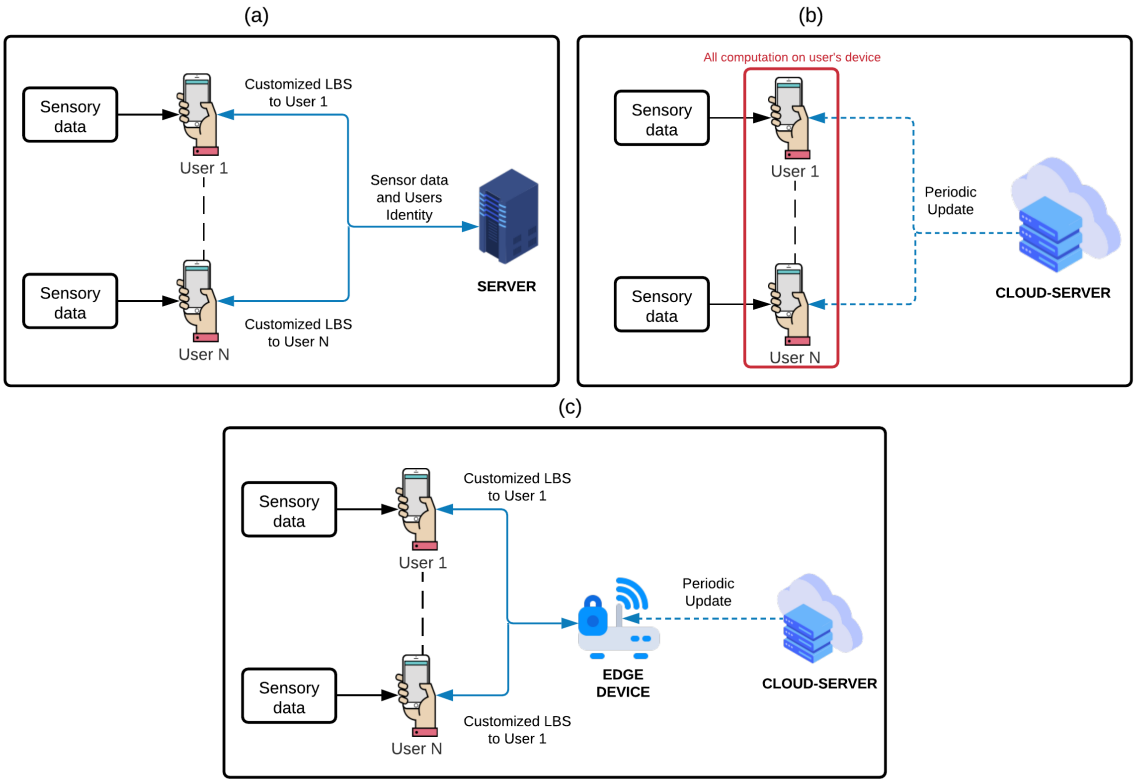


Figure 2. IoT Architectures

The "Client device" group (Figure 2b) classifies those Indoor Positioning System that performs localization and tracking computation exclusively on the user's smart device and relies on periodically communicating with a cloud server for updating environment-specific contextual information [29]. These positioning systems usually operate in real-time speeds, are inherently more privacy-friendly than their "Server" counterparts but comes at a cost of reduced localization accuracy and higher system latency due to computational resource constraints. Additionally, running all calculations on smartphones is detrimental to their battery life which reduces long-term usability and reliability. These IPS are predominantly decentralized systems.

The "Edge Device" group (Figure 2c) takes into consideration the emerging architecture where smart device offloads sensory data and receives position estimation directly from a local-machine called "Edge Device" [23]. "Edge Devices" are small single-board computers which are usually inexpensive, supports both wired and wireless communication (Wi-Fi, BLE) and sometimes contains GPUs for executing AI models in near-real time speeds. Using IoT communication protocols, Edge devices along with client devices can form a densely interconnected network that is conducive for large-scale application [30]. IPS within this group can take a hybrid approach between centralized and decentralized modes of operation and are generally faster than "Server" IPS due to the low latency of communication between the local machine and client devices.

Another novel strata introduced in the proposed classification is Level L7 "USER IDENTIFICATION DATA" which aims to classify Indoor Positioning Systems based on the type of data used to identify a user(s) from a group of people in order to provide them with personalized location-based contextual information. We incorporated this stratification due to the fact that how an Indoor Positioning System identifies a user directly dictates the communication technology, the constituent sensing technologies, the level of data security, and the amount of information that will be retained by the primary computation device to provide location estimations.

Within this strata, Indoor Positioning Systems are classified into two groups namely "Direct Identification" which as the name implies, requires the user to provide personally identifiable information (i.e. facial image for unique recognition) to receive localization information. Most commercial Indoor Positioning System falls under this category. The second group named "Anonymized Identification" classifies IPS which distinguishes users with pseudo-identifiers (i.e. Integer numbers, Alphanumeric sequences) rather than their personal information. For example, Zaho et al. [31] represented 16 participants with Integer numbers. Domingo et al. [32] represented 3 users with English alphabets. These two IPS would be classified under the "Anonymized Identification" group since none of the two systems identified users' using their personal information.

2.3 Common Signal Metrics in Radio-Frequency Based IPS

From the recent comprehensive reviews [10,21,22] on Indoor Positioning Systems, it is evident that a majority of proposed IPS are built with Radio-Frequency technology namely WLAN(Wi-Fi), Bluetooth, Ultra Wideband and RFID. In this section, we briefly introduce the common types of signal measurement techniques used in these systems since the choice of signal metric directly dictates which category of algorithms may be used for localization and tracking. It is to be noted that all of the following signal measurement techniques accumulates error from phenomena such as multipath fading, non-linear propagation loss, signal attenuation due to human body and obstacles, loss of signal due to occlusion, measurement disparity for the same location due to device-to-device hardware heterogeneity which are inherent all form of RF technologies [17]. Furthermore, signal sampling rate at the user's device significantly affects the quality of measured signals for all of the following measurement techniques.

2.3.1 Received signal strength indicator (RSSI). The most widely used signal measurement metric Radio Signal Strength Indicator (RSSI) is a relative measure of the strength of the RF signal received by a node (i.e. user's smart device) from one or more multiple transmitters (Wi-Fi Access Points, Bluetooth Beacons). This metric

is often confused with Relative Signal Strength (RSS) which is a numeric measure of actual power of the received signal expressed in decibel-miliwatts (dBM). RSSI is the most famous and most common measurement technique employed in Indoor Positioning Systems since measuring RSSI of received signal requires no additional hardware and can be used in all algorithms classified under "Geometric" and "Scene Analysis" groups [23]. It is generally accepted that RSSI values has a theoretical log normal relationship with distance from the RF transmitter. Mizmi et al. [33] represented this relationship with the following formula

$$RSSI(d) = A - L_o - 10nlog_{10}(d/d_0) + L_S H \tag{1}$$

Here $RSSI(d)$ is the relative measure of the RF signal perceived by the user's device situated at a distance of d meters from the RF transmitter (i.e. Base Station, AP, BLE Beacon). $RSSI(d)$ has no fixed scale and can take a value between 0 to 100 or -100 to 100 with a lower number indicating close proximity to the RF transmitter and vice-versa. The constant L_o is the "average propagation loss" calculated from the RF transmitter at a distance of 1 meter [34] and A is a constant value which depends upon the transmission power of the transmitter and antenna gains of the receiving device [33]. Note that, in real-world indoor scenarios, RSSI measurement does not follow the relationship depicted in Equation 1 when there is a significant distance between the user's device and the RF transmitter.

2.3.2 Angle of arrival (AoA). Zafari et al. [10] define Angle of Arrival (AoA) as a signal measurement technique that determines the angle at which a user's device received a transmitted signal using a multi-array antenna. This is achieved by calculating the difference in arrival time of a signal reported within the constituent antenna array. This signal measurement technique allows localization with as little as two angle measurement from two RF transmitters for 2D localization and three angle measurements

from three RF transmitter for 3D localization. However, AoA requires a special multi-array antenna which is not common in smartphones and requires constant calibration which is impractical for real-world indoor applications. Additionally, compared to RSSI, AoA measurement degrades more as one moves further away from the RF transmitter.

2.3.3 Time of flight (ToF)/Time of arrival(ToA). This measurement technique calculates the signal propagation time between user's device and RF transmitter which can then be linearly correlated to distance. In clean Line-of-Sight (meaning no obstacle between receiver (Rx) and transmitter (Tx)), condition, the accuracy of localization using ToF/ToA can be significantly higher than RSSi based measurements. However, ToF measurement requires very strict synchronized timing between Rx and Tx, which in a multi-user dynamic condition is not possible. Furthermore, this technique can only be used for algorithms under the "Geometric" category. Finally, significant error in measurement is accumulated in No Line-of-Sight (NLoS) condition, a condition most representative of real-world indoor scenarios.

2.3.4 Time difference of arrival (TDoA). This measurement technique calculates the difference in signal arrival perceived by the receiver from multiple RF transmitters. At minimum three RF transmitters are need and all of the transmitters must be time-synchronized is needed to measure TDoA at the receiver. Just like ToF, the quality of TDoA measurement depends upon the sampling rate, LoS and signal bandwidth. Note that, TDoA is not suitable for BLE Beacon-based IPS since BLE Beacons since, beyond a configuration stage, BLE Beacons does not support bidirectional communication which excludes TDoA measurement from most BLE based Indoor Positioning System applications.

2.4 Techniques in Indoor Positioning Systems

In the following subsections we briefly review the Geometric, Scene Analysis and 2D/3D Image Analysis techniques which are frequently used in Radio-Frequency based and Vision-based Indoor Positioning Systems.

2.4.1 Geometric techniques. This category of techniques encompasses positioning algorithms that utilizes the relationship between the measured signal property (i.e. RSSI) to distance from RF transmitters and, basic geometry to localize one or more users [11]. These methods are most suitable when the scenario can be modeled with high certainty. However, such modelling is nearly impossible due multitude of factors such as NLoS condition, multipath propagation, dynamic movement of people and obstacle which makes it difficult to establish an unique propagation model to capture the dynamics of the RF signal which by its own nature is highly nonlinear in nature. Three of the most common Geometric Techniques are briefly discussed below.

Trilateration (3 points Multilateration) algorithm is the most well-known of Geometric techniques which assumes the location of a user's device lies at the intersection point of three hypothetical circles whose centers are the location of three RF transmitters and the radii of the circles equals to the distance between the user's device and RF transmitters (Figure geometric a). Furthermore, trilateration assumes clear LoS condition and also assumes that the radii of all three circles are equal. The measured RSSI data is converted to distance using propagation model (log-normal model from Equation 1) which gives the position of the user with respect to a global reference frame. Most practical IPS will have more than 3 RF transmitters in which case multiple intersection points are obtained which then converts the problem to a nonlinear optimization problem requiring good initial estimates which for practical cases may not be easy to determine prior to localization [11]. Multilateration localization is computationally very fast but has poor accuracy in comparison to Fingerprinting techniques due to its heavy dependence on LoS condition [26, 35].

Proximity algorithm usually uses RSSI values to infer user's position by calculating a "relative" distance between the user's device and the RF transmitter. Usually this distance is then converted to a categorical attribute such as *NEAR, FAR, VERY FAR* which is then used to provide environment specific contextual information (i.e. marketing

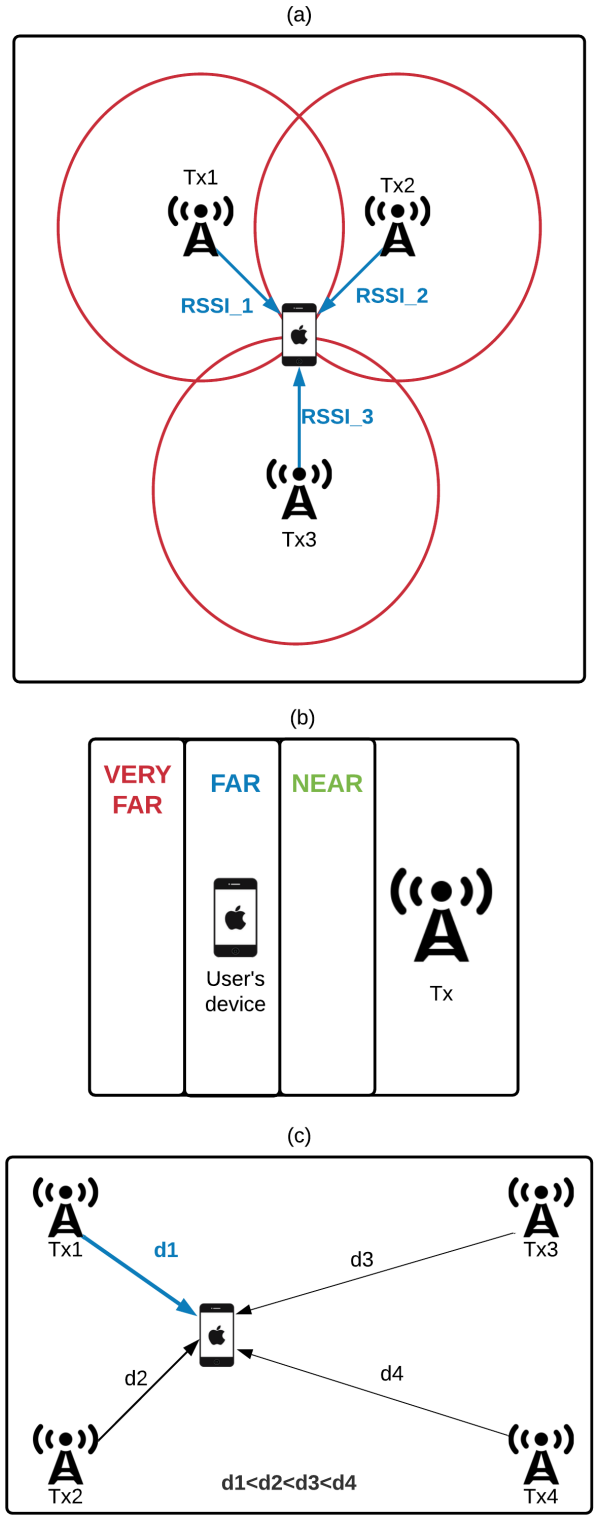


Figure 3. Geometric Techniques

alerts). The sphere of influence around the single RF transmitter is called a geofence [10]. Proximity based technique is the most simplest, cheapest solution to implement with the least number of RF transmitter requirement but with the highest uncertainty in the location estimation.

Weighted Centroid (WC) algorithm (Figure c) is a derivative of Multilateration technique where the radii of the hypothetical circles are not equal but are calculated as "weights" whose value is inversely proportional the log normal distance obtained from Equation 1. User's position would be biased towards the RF transmitter that has the lowest RSSI value (distance is inversely proportional to RSSI).Subedhi et al. [35] defines Weighted Centroid algorithm using the following set of equations

$$
\begin{aligned}
X_w &= \sum_{i=1}^{m} X_i \times \frac{w_i}{\sum_{i=1}^{m} w_i} \\
Y_w &= \sum_{i=1}^{m} Y_i \times \frac{w_i}{\sum_{i=1}^{m} w_i} \\
w_i &= \frac{1}{d_i}
\end{aligned}
\tag{2}
$$

Here (X_w, Y_w) are the estimated coordinates, d_i is the log normal distance calculated between the user's mobile device and the i^{th} and m is the number of RF transmitters accessible by the user's mobile device. WC localization technique is deterministic and has very low computation overhead. However, this technique shares the dependence on LoS similar to all other ranging based technique.

2.4.2 Scene analysis. Scene Analysis (a.k.a Fingerprinting) techniques are data-driven methods which became quite popular for radio-frequency-based Indoor Positioning Systems due to their robustness to NLoS condition and better generalization to dynamic indoor environments in comparison to their Geometric counterparts. This category of technique comprises a wide berth of algorithms that can broadly be classified into three groups namely Machine Learning, Probabilistic and Deep Learning.

While the internal mechanics of these algorithms vary, in principle all algorithms

under Scene Analysis group are essentially supervised classification techniques that takes in a vector of signal measurements for each of the accessible RF transmitters, computes its difference with known reference measurements, and finds the location point that closely matches a known reference measurement. Multiple authors described Scene Analysis techniques as three-stage process viz. *Offline Database Building*, *Generation of Localization Features*, and *Online Prediction*.

In the Offline Database Building stage, multiple measurements of RF signal metrics (RSSI, ToA, AoA) from all accessible RF transmitters (BLE Beacons, Wi-Fi routers) are collected at a given time and at a particular location to discover the spatio-temporal relationships between the radio signals and the target positions for a particular indoor scenario [21]. Note that, there is no hard consensus on the size of the database but it is generally accepted that for ML and DL methods, a considerably large database is required. Collecting such a large database is both time and resource-intensive which is one of the major drawbacks of Scene Analysis techniques [36].

In the Generation of Localization Features stage, signals are usually preprocessed to reduce noise, discard missing measurements and then feature extraction algorithms are employed that generate statistical metrics such as 4-point moving average [26], exponential moving average [37], variance [38]. It is to be noted that, what features to generate depends on the type of data, pre-processing techniques chosen and prediction algorithm employed during the inference stage.

For instance, Zafari et al. [39] developed an Indoor Positioning System using BLE beacons where raw measurements from two accessible beacons were fed through a Bayesian model to predict user's location. Feature generation from raw RSSI measurements was accomplished into two-step process. First raw measurements were fed through a moving average filter and then further refined with a Kalman Filter for additional smoothing. In another implementation, Subedhi et al. [37] performed an exponential moving average on raw RSSI measurements followed by a Weighted Centroid algorithm

to generate distance as features for each accessible BLE beacon. Contrasting to the former two examples authors in [33] moving the average filter to raw RSSI values and then converted the processed values into a binary representation which significantly improved inference time in the prediction stage. This was achieved since in Scene Analysis techniques, high dimensional data may arise when a lot of features are generated (i.e 10 features may be generated for 5 accessible RF transmitters by finding mean and standard deviation for each) that results in high computational load during inference stage [36].

How RSSI data may be represented also forms a method of feature extraction in RF-based Indoor Positioning Systems. In this regard, Torres-Sospedra et al. [40] developed a new feature generation technique called "powered" which is a modified Min-Max normalization technique coupled with a power factor that converted raw RSSI into 0-1 scale. The authors demonstrated that "powed" representation preserved the theoretical lognormal relationship between RF signal to distance better than simply representing RSSI values using traditional scales (-100 to 100). Note that, while a plethora of feature extraction techniques is available it is crucial to recognize that the extracted features must capture the spatial diversity between adjacent target location points and are temporally stable [23, 37]. This is very important since ML and DL algorithms through recursive training aims to learns the spatio-temporal relationship between measured signal to actual locations from the generated database. If two adjacent location points do not show enough variation in the generated features, then ML and DL models will fail to distinguish between them causing high level of inaccuracy during inference stage [36].

In the Online Prediction stage, real-time signal measurements are continuously taken by the user's mobile device from all accessible RF transmitters and processed to extract features. Algorithms such kNN [41], bayesian estimator [42], Autoencoders [43] can then be used to infer the user's position either as a unique (X,Y) coordinate or as a probability of occupying a certain region on the map.

2.4.3 2D/3D Image analysis. The core principle of 2D/3D Image Analysis techniques in Indoor Positioning Systems is to analyze image data using predetermined feature descriptors to find the Regions of Interests (RoIs) in the image which are occupied by the objects of a target class in order to localize their position in the image and subsequently track their trajectory over time. In computer vision literature, these two tasks are called *Object Detection* and *Multi Object Tracking* which may be performed jointly in a single model [lu2021] or separately as a sequential two-step process [44]. RGB or Grayscale images (2D Images) with/without depth information (a.k.a RGB-D or 3D Images) captured using traditional front-facing cameras, perspective cameras, overhead cameras and infrared cameras have been used in 2D/3D Image Analysis techniques and the cameras may be static or mobile which determines the type of algorithm suitable for application. Key challenges to 2D/3D Analysis techniques are change in illumination, variable pose, change of aspect ratio (apparent change in shape) of moving objects, short-long term occlusion of objects in crowded scene.

Object Detection, the first of the two tasks in 2D/3D Image Analysis techniques is defined as a sequential two step process in which the location of objects are determined in the image and then classified into pre-defined groups [31]. Using the information from Object Detector, a Multi Object Tracking (MOT) algorithm performs the tasks of assigning identifiers to the detected objects and then tracking their trajectories over time in a image sequence [45]. An *image sequence* is defined as series of images identified with incremental identifiers that preserves the chronological order in which they were acquired by the camera unit. Individual images in an image sequence may also be referred to as *frames*.

Traditional Image Analysis techniques such as stereo foreground-background segmentation [46], Histogram of Oriented (HoG) depth features [47], local maxima search [48] has been used to determine if a RoI in an image resembles a known feature for identifying an object such as head and shoulder of a human being [46]. One of the earliest examples

in this class of techniques is the work of Park and Aggrawal [49] where authors used foreground-background image segmentation technique to train a head-orientation binary classifier in detecting a person from a number of blobs (small cluster of pixels of known intensities) and finding their relative position with respect to one another. Nakatani et al. [50] employed foreground-background subtraction technique along with noise reduction to extract four features namely body size, hair color, hairstyle, and hair whorl used to identify ROI that contains a person and determine their position by computing the centroid of the ROI box determined using the histograms of the person's area.

Artificial Intelligence (AI) based ROI detection techniques have largely superseded traditional image processing techniques due to their capability of learning descriptive features of objects of multiple classes directly from the image data which greatly alleviated the need for complex, scene-specific feature extraction models [20]. AI models enable complex scene understanding for advanced vision tasks such as joint detection and tracking, image segmentation; faster detections in multi-object scenarios, better generalizability to a wider variety of indoor scenarios, and deployment into IoT edge devices for real-time applications [51]. The most representative model used with DNN based Object Detectors is Convolutional Neural Network (CNN) which has become very popular due to its capability to learn descriptive features using a network of convolution layers [52] and leverage large-scale transfer learning using very large annotated databases such as Imagenet. Some well known examples of CNN based Object Detection algorithms are AlexNet [53], You Only Look Once (YOLO) [54], SqueezeNet [55] and so on.

Mutli Object Tracking (MOT) algorithms within 2D/3D Image analysis techniques focus on solving the problem of tracking multiple detected *objects* over time. This problem is more difficult than Single Object Tracking since the number of objects varies over time along with other issues such as frequent long-term, short-term occlusions which causes track fragmentation, nonlinear rise in computation complexity with linear increase of moving objects, initialization and deletion of tracking identities and so on.

There are two major categories of MOT algorithms based on how they initialize tracklet ids and usage of future image frames. The first is the Detection-Free Tracking (DFT) which requires manual initialization of tracklet ids in the first frame and utilizes images from future timesteps to maintain object tracking. Though these features are not representative of real-world condition [45], DFT trackers are generally better at tracking objects in difficult conditions since they incorporate information from both past and future frames [56]. On the other hand, the Detection-Based Technique (DBT) techniques approaches the MOT problem as a continuous assignment problem in which tracking ids assigned to some detected objects in a previous frame is recovered in the current frame based on the similarity between the detected objects in current frame to detected objects in past frames [57, 58]. DBT techniques are suitable for real-time application since these techniques only considers frames up to the current timestep which makes them faster than DFT technique but DBT techniques suffers from two major issues. The first is the reliance on the object detector's ability to detect target objects consistently since the bounding boxes provided by Object Detectors forms the targets from which MOT determines affinity of detected objects with previously established tracks. Hence, recurrent missed detections by Object Detector causes track fragmentation which degrades performance [59]. The second major issue lies with the issue of restarting trajectories in event of track fragmentation, which is commonly known as Identity Switch. ID-Switch is particularly problematic in scenes where high number of occlusions, illumination change, erratic motion of camera is present.

2.5 Review on Multimodal Indoor Positioning Systems

To the best of our knowledge, Yassin et al. [11] and Pascacio et al. [22] attempted to formally categorize Multimodal Indoor Positioning Systems (MIPS). Based on their work and on the basis of common technologies and techniques introduced in Section

2.2, MIPS solutions may be classified into 4 categories as shown in Figure 4. **RF** represents the Radio-Frequency technologies (i.e. Wi-Fi), **PDR** represents the techniques used to measure user's gait and relative displacement using measurements from Inertial Measurement Unit (IMU) embedded into most smartphones (i.e.accelerometer, gyroscope) and **Vision** represents the tools and techniques related to 2D/3D image-based user localization. As our system proposed system is poised to track multiple users with privacy-friendly identifiers at sub 1 meter level accuracy in an edge device at near-real time speed, we emphasize on these factors for the following illustrative examples.

2.5.1 RF-PDR. These MIPS solutions are designed to enhance accuracy of position estimation from radio signals with user's motion estimate from IMUs using Linear and Nonlinear State Space estimators (refer Figure 1, L4) as Kalman Filter, Extended Kalman Filters, and Unscented Kalman Filters.

The MIPS proposed by Wang et al. [60] is a classical example of RF-PDR MIPS where the authors combined RSSI measurements from 5 Wi-Fi routers with the distance covered between two successive RSSI recordings measured with an accelerometer and topography of the observation area with a Particle Filter algorithm to improve localization accuracy by as much as 25% compared to applying a Kalman Filter estimate directly onto RSS measurements. Note that, the authors in [60] did not consider multi-user scenarios nor indicated to the speed of the combined system. Resolution of localization was greater than 5 meters at 90% accuracy level.

In another implementation, Deng et al. [61] combined RSSI from eight Wi-Fi access points to user's heading estimation calculated as a quaternion using two sequential Extended Kalman Filters. They also developed a KDE model which recursively updated the covariance matrix of the Kalman Filter to accurately capture the spatiotemporal relationship between RSSI signals and actual position. The authors demonstrated that with the inclusion of PDR data augmented with apriori knowledge of certain landmarks on the scene, mean localization error was reduced by as much 76.9% for when localization

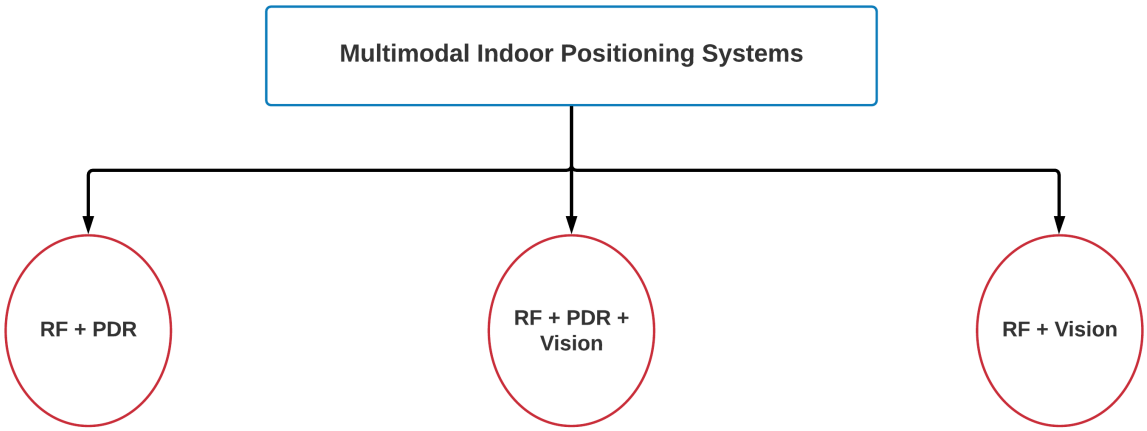


Figure 4. Classification of Multimodal Indoor Positioning Systems

resolution was 2.40 meters. However, the system only serviced one user in an experimental area of size 43.5 meters x 11.2 meters with no information as to the speed of the proposed system nor the identifier used to track the user.

Murata et al. [2] implemented a MIPS using RSSI from BLE beacons and PDR data from smartphone as an assistive navigation tool for the visually impaired in a large observation area $(21,00m^2)$ that included three multi-story buildings and an underground passageway. The core algorithm was a combination of a Particle Filter based motion model and a Bayesian fingerprinting technique optimized to run at near-realtime speeds in a mobile device. To achieve further improvements, the authors also introduced a state-machine to check for integrity of the system, combined barometric pressure sensor readings and RSSI values in a novel conditional motion model to not only reduce localization error during floor transition but actually predict the user targeted floor transition and added an adaptive RSSI signal calibration method based on a modified Kalman Filter to compensate for the offset between measured signal strength in online phase with the stored offline database. Data from 218 beacons spaced 5 to 10 meters apart was collected with two Apple iPhones during open-hours to simulate real-world adverse signal propagation conditions. 10 visually impaired people participated in the experiment with

a mean localization error of 3.2 meter at 95-percentile. The system had an effective update rate of 0.14 seconds. This experiment did not consider multi-user condition rather had the 10 participant take turns in navigating through the test path one at a time.

2.5.2 RF-PDR-Vision. Dumbgen et al. [62] implemented a MIPS in which authors recorded the return trip time (RTT) from Wi-Fi access points (AP), RSSI from BLE beacons, heading estimation from IMU and visual cues obtained by a smartphone's camera and, combined all the data together in a offline server to perform position estimation by maximizing a chain of probability functions dubbed "Conditional Random Fields". A major advantage of this MIPS was its independence from creating lengthy Radio Maps and online parameter calibration when a valid visual cue was found. While the authors mentioned presence of multiple individuals during training and testing dataset collection and the size of the observation area was greater than 10 meters by 30 meters, only one user was localized by the system in an offline manner. Furthermore, user identification during runtime was not considered. The authors showed that their proposed MIPS had an average mean localization error of 2.3 meters and concluded that Wi-Fi + IMU + BLE combination performed much better than Wi-Fi + IMU and Wi-Fi+BLE combination.

2.5.3 RF-Vision. This category of Multimodal Indoor Positioning System combines RSSI data from RF-based technology to position estimate obtained from image pixel using a variety of techniques. Our proposed method falls within this category.

Sun et al. [63] proposed a multi-modal indoor positioning system that used crowdsourced Wi-Fi positioning data to optimize a RSSI triangulation model and image data from an overhead panoramic camera to track an user in two distinct indoor environments. For the corridor scene, the optimized propagation model was used to estimate the user's position and when the user entered the room scene, the user's was first detected using a background subtraction technique followed by localizing the user using a three-layer Artificial Neural network which took the 2D pixel coordinate as input and gave user's

position on the world coordinate frame as output. For an experimental area of 51.6 meters x 20.4 meters, the mean accuracy of the Wi-Fi positioning system was 5.79 meters and the mean accuracy of the vision-based positioning system was 0.84 meters. The combined mean accuracy was 3.15 meters. However, all RSSI data were collected with a smartphone in a fixed orientation and the camera positioning was tested only for one person at a time. Multi-user interaction and user identification was not considered.

Zhao et al. [64] proposed a novel MIPS in which images from a smartphone‘s front-facing camera, heading estimation from its IMU and the Channel State Information (CSI) measurement of a 5.0 Ghz Wi-Fi AP were combined together with apriori information of the indoor environment to form a geometrical relationship between image space to the inertial space from which the user's distance and orientation with respect to detected objects can be determined with high accuracy. At first two images captured by the mobile phone's front-facing camera are passed into a pre-trained YOLOV2 DNN Object Detector which extracted coordinates of known households items in the image space. Then orientation data from IMU was measured to determine a rough estimation of the distance and orientation between the detected objects and the user using quaternion calculation. Using position estimation calculated from Wi-Fi CSI data the rough estimation is refined further by solving a linear equation. In this paper, the authors tested the system for both single users and multi-user scenarios in which 8 participant volunteered. The mean accuracy of the system was 0.2 meters at 92-percentile for objects within 5 meter of range. User's were identified using integer numbers and assigned specific actions sequence to perform. It is to be noted that, how the eight users interacted during data collection was not explained nor the speed of the system.

The MIPS proposed by Domingo et al. [27] demonstrated robust localization of multiple users moving simultaneously in a complex environment using spatial information obtained from two separate positioning system running concurrently with one another. One is a Wi-Fi based positioning (WPS) and another is a vision-based indoor positioning

system which uses depth map captured by multiple Xbox Kinetic V2 RGB-D cameras to compute the "skeletal" position of all users within its field of view. The core idea is to estimate position of a user at meter level using WPS and then refine that measurement to centimeter level using finer position estimate obtained from RGB-D cameras. During operation phase, RSSI values and their corresponding depth map are sent to the central server which then first checks if the number of trajectories generated by the two sensing modalities are equal to the number of users present at a particular time step. If equal number of trajectories exists from the two modalities, the system computes a matrix containing the euclidean distance between all possible pairs of WPS trajectories and depth map trajectories. Then it finds the one-to-one unique pairing between the radio information and skeletal position using the Balas Additive Algorithm, effectively joining an user's identity from WPS data to a skeletal coordinate from the RGB-D camera. This system while thematically similar to our proposed system has many discrepancies. Firstly the paper reported a maximum of 20 users taking 8 predefined paths but it not discuss how the users moved about the environment. Secondly, this MIPS heavily depends on server-side synchronization and 2s scanning window for WPS. This infers lower than near-real time speed in actual application.

2.6 Challenges in Indoor Positioning Systems

In this section we briefly present the common challenges associated with Indoor Positioning Systems. Note that, we will restrict the discussion only to challenges associated with RF-based Indoor Positioning, Vision based Indoor Positioning and Multimodal Indoor Positioning Systems since an exhaustive review of all challenges associated beyond the scope of this work.

2.6.1 Dependence on environment specific radio map. One of the major shortcomings of Fingerprinting based RF-IPS is its dependence on environment specific **Radio Map** for predicting location of moving objects. Radio Map is a database of RSSI signatures that uniquely characterizes certain locations on the testbed. This

database is created during the offline data collection stage which is then used to train the ML model for inference stage. Features derived from raw RSSI measurements such as mean, mode, rolling average are used to create the unique signatures as mentioned above. However, these features are dependent upon RSSI measurements whose efficacy directly depends upon various factors such as number/position of RF-beacons, number of incoming/outgoing users, change in static obstacles and system configurations such as number of reference points, number and type of RF-beacons and so on. In real-world conditions, number of users, their interactions and, position of static obstacles may change in a random manner. Hence, the Radio Map needs to be constantly updated to account for these changes. Performing this task manually is a labor and time intensive process feasible for small scale application but for large scale application automation is required. Whether manual or automatic process is used, the data collection scenario should to as close to the real-world condition the system will work in. However, replicating all real-world conditions is not a trivial task. Furthermore, within the same testbed changing the orientation, position, height or number of RF-beacons drastically changes the signal to location properties that requires creating a completely new Radio Map.

In addition to the above, updating Radio Map is also challenging due to the the fact that different smart devices (smartphones, smart watches) records RSSI value differently at a known distance due usage of different chipsets [11, 19] and their measurement varies even more during movement. Thus, updating a Radio Map not only requires collecting RSSI measurements over and over again but also requires to incorporate different models of smart devices, all of which must then participate in collecting data in a manner as they would be used during inference stage. Such knowledge of type and usage of smart devices is impractical for real-world use.

Due to the above issues, a Radio Map created in one indoor testbed is specific that environment and any MIPS utilizing RF-based fingerprinting techniques inherets this limitation. Additionally, a previously created Radio Map is not easily usable in a

new testbed since the signal properties and distribution of reference locations in latter testbed seldomly matches the former.

2.6.2 Privacy and security. IPS needs some user-specific identifier for detection, localization and tracking. One of the most common identifier used is the device identifier (such as Mac address) found in almost all smart devices. But device identifiers can be captured secretly to compromise user's privacy [13]. Zafari et al. [10] and Holcer et al. [14] attributed this limitation to the design methodology in IPS which were primarily concerned in improving accuracy and reducing cost but does not directly take into account user's privacy. Guaranteeing anonymization of device identifiers is important, especially for providing personalized services but is technically challenging to accomplish. Attempts have been made to secure users data by processing localization information directly on the user's device. This method while improving privacy is not energy efficient as the user's device is then needed to do all calculations related to positioning on top of measuring signals (RSSI, image) needed for localization. For energy constrained device such as a smartphones, this approach is sub-optimal [17]. Another approach taken to improve privacy is to use encrypted communication using traditional algorithms which in theory improves privacy guarantee compared to lack of encryption, but in practice this approach is also not energy efficient [14].

In addition to encryption, removing features from input signal is a viable option but comes with a usability trade-off. For instance, blurring out facial feature from input images completely removes the capability of the system to distinguish an user uniquely from a group. Hence, though CV-IPS can then track individuals with high accuracy, without incorporating artificial markers unique to each individual. However, this is problematic since use of artificial markers restricts the real-world applicability of CV-IPS to only a handful of scenarios [20].

2.6.3 Cost. Cost to deploy an IPS directly dictates its real-world applicability. Depending on the system architecture, costs are incurred from factors such as setting

up network infrastructure, installation, software subscription, maintenance, computer hardware and storage facilities. In some instances certain cost may be unavoidable. For instance, Domingo et al. [27] covered an office room of 80 m^2 with 20 RGB-D kinetic cameras. This many 3D cameras require a powerful back-end server for processing all data that increased overall cost. In contrast, Blanco et al. [16] covered a larger workspace consisting of more than three rooms using a small number of overhead cameras fitted with 360° wide angle lens whose data were processed in near-real time speed with standard desktop computer. Hence the latter design is much cheaper in comparison to the the former but the system architecture proposed in [27] can distinguish multiple users whereas the system in [16] cannot distinguish any users uniquely.

2.6.4 Scalability. Scalability refers to how IPS performs when more sensors, environment, network infrastructure is added to the system. The challenge lies in is creating a scalable system without incurring heavy financial and performance penalties. For RF-based IPS, performance decreases as user's mobile device moves further away from the RF-beacons [11]. Consequently, more RF-nodes is needed to be placed on the workspace which proportionally increases the complexity of the algorithm and data collection requirement. Additionally, wireless signals degrades in quality in congested area with a lot of people and radio devices, requiring update to radio map which for large application is infeasible. For CV-IPS, scalability is a major challenge since the the problem of tracking a group of users across multiple cameras is a NP-Hard problem with no known optimum solution [45]. Additionally real-time CV-IPS requires more expensive hardware and faster network infrastructure which adds to the complexity of scaling CV-IPS.

2.6.5 Large annotated database for supervised DNN. A challenge uniquely associated with CV-IPS using supervised DNN models is the lack of large annotated image database needed in training stage. Creating this database is very time consuming due to lengthy data collection process and manual annotation process that is commonly used

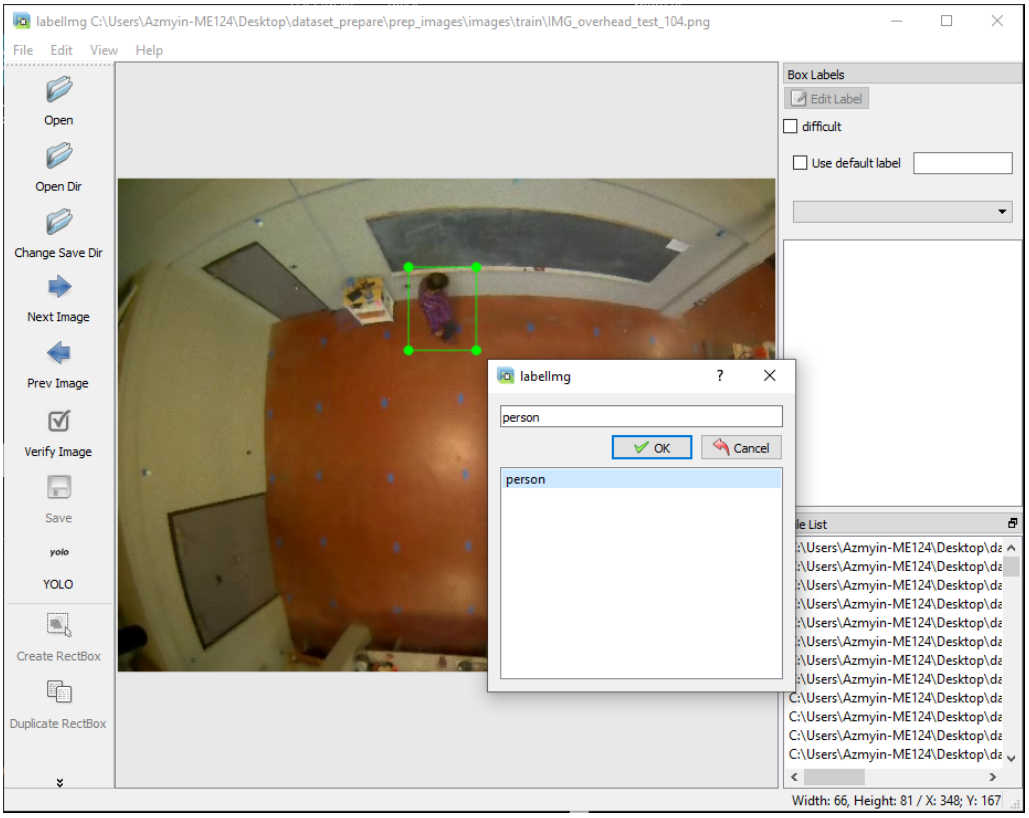


Figure 5. Manual annotation with a common "Person" class

to create annotated data. To exemplify, if a image sequence of only five minutes consists of 1000 images and if there are four users present throughout the whole 5 minutes, then the total number of annotation that manually needs to be created is 4000 which roughly takes 4 hours to complete (refer Figure 6). Automating the annotation process is a viable option but requires complex techniques which becomes a limiting factor when multiple indoor environment is considered.

To further complicate this problem, the training database needs to adequately capture the diverse interaction between users and objects in the testbed. For instance, if training data is collected only for people in walking in and out of an empty testbed for a fixed lighting condition (i.e. lab with no natural light), the model will not perform adequately for a test data where multiple chairs and tables are placed with users interacting while sitting (i.e classrooms). Thus, the training data collected needs to be as close to the actual scenario as possible which for real-world implementations is difficult

Figure 6. Annotation at individual person level

to perform due to the large diversity of interactions possible between users and objects in an indoor environment.

2.6.6 Challenges in MIPS. In section 2.5, it was established that inclusion of non RF technologies such as Inertial Measurement and Vision can greatly improve the accuracy of an Indoor Positioning System. However, the improved accuracy comes at additional infrastructure cost, complexity, latency, energy efficiency and privacy related cost. This is because the challenges/shortcomings of individual constituent technologies translates to challenges with. Multi-modal Indoor Positioning Systems. The first of such challenge is synchronizing heterogeneous datastreams that directly affects the usability, accuracy and latency of the MIPS system. To illustrate, the RF-Vision based MIPS proposed in [27], one can track multiple users in a privacy-friendly manner using position estimations obtained from Wi-Fi-IPS and Infrared depth images based CV-IPS with over 90% accuracy. However, due to the difference in measuring RSSI signal from Wi-Fi access points by the mobile phones and speed at which the depth maps from the Kinetic cameras are updated, the system requires more than 2 seconds to process 1 timestep. This coarse estimation was then fine tuned with the fine position estimation from the

depth map. Though the accuracy was high the latency of this system made incompatible for real-time operation. In contrast, the MIPS proposed in [2] fused RSSI data from BLE beacons with PDR from Inertial sensors directly on the smartphones to provide location information to the users in context of the surrounding environment.This system had an update rate of 0.14 seconds, offers better energy inefficiency since all processing takes place on the smart device that drains batteries faster, has a larger localization error in contrast to the solution proposed in [27] that makes it suitable only for large workspace applications.

The second major challenge in MIPS lies with the complexity of algorithms in fusing multimodal data. Algorithms such as Unscented Kalman Filter, Particle Filter, Conditional Random Fields are required along with Machine Learning and Deep Learning algorithms to extract accurate location information from multimodal data. However smart devices have limited computational capacities and fixed battery life which makes them unsuitable for running complex algorithms. As a result most MIPS requires a centralized server to aggregate and preprocess the multimodal data before before executing the complex positioning algorithms. Hence introduction of server computers raises infrastructure cost. But running all calculation on a centralized server comes with the additional requirement of of providing location information to each connected user individually for which a user-specific identifier is required. Ensuring error-free and low latency server-user communication adds another hurdle to MIPS whose difficulty increases since sending out location service data to multiple users who are also mobile is a complex engineering challenge. On top of streaming data to the right user, protecting their privacy and ensuring data security warrants the need for complex post-processing algorithms which adds to overall the latency of the system.

Chapter 3 Problem Statement

Let there be M users present on the testbed for the total duration of T. We define i to represent each user such that $i = 1, 2, 3.....M$. The users are in full view of the overhead camera and each user carries a mobile device that is able to record signals from K accessible BLE beacons in the testbed. All the 2D positions can be defined with respect to the image plane.

We define t_p as a single timestep within T. The testbed is divided into L_{grid} cells labeled with an integer sequence l_q such that $l_q = 0, 1, 2, ..., L_{grid-1}$. Each cell has a unique 2D coordinate assigned to it and represented by (y_{l_q}, y_{l_q}). Let each i^{th} user's smart device create a randomized alphanumeric string dubbed **pseudo id** and denoted by $O^i_{t_p}$. We assume pseudo ids do not change until end of session i.e. $O^i_{t_p} = O^i_{t_p+1} = ... = O^i_T$ and, is a token that uniquely represents the i^{th} user to the proposed system.

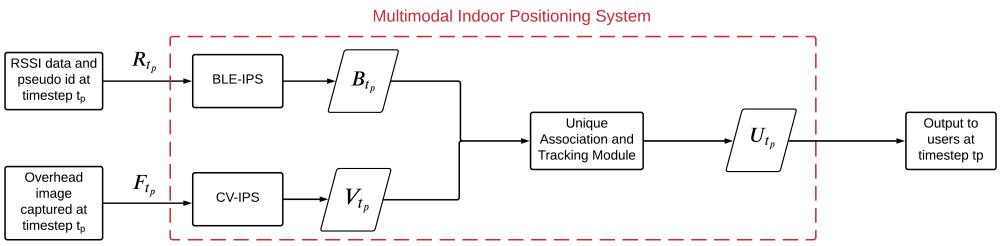


Figure 7. Block diagram of proposed MIPS

Now at timestep t_p, input to the Bluetooth Low Indoor Positioning System (BLE-IPS) is a signal matrix $R_{t_p} = \{R^i\}^{M\times 1}$ such that $\{R^i\} = [[r^1_{0:t_1}, r^2_{t_1:t_2}, r^3_{t_2:t^i_r}], O^i_{t_p}, t_p]$. Each R^i holds three RSSI vectors in the form $[r^1_{0:t_1}, r^2_{t_1:t_2}, r^3_{t_2:t^i_r}]$ collected over three epochs t_1, t_2, t_3 in a 1 second time window. Let the time at which the i^{th} user's smartphone finished collecting three RSSI vectors be the recording time t^i_r. For simplification, we assume $t^i_r \approx t_p$. Each RSSI vector is an ordered list in the form $[rssi_1, rssi_2, ..., rssi_b]$ where $b = 1, 2, ...K$ and $rssi_b$ denotes the raw RSSI measured for the b^{th} beacon. R_{t_p} is passed into the BLE-IPS where a features from the 3 RSSI signal vectors are extracted and passed through a Machine Learning model that predicts 1 cell among L_{grid} cells

for each user. The output from the model is a matrix where each row is in the form $\{M^i\} = [l_q^i, O_{t_p}^i, t_p]$. Using a linear mapping function, cell labels l_q^i are converted to their 2D coordinate mapping. Output from BLE-IPS is thus the matrix $B_{t_p} = \{B^i\}^{M\times 1}$ where $\{B^i\} = [O_{t_p}^i, (x_{l_q}^i, y_{l_q}^i), t_p]$. The output clearly shows BLE-IPS can uniquely distinguish an user but cannot track them in continuous space.

Concurrent to BLE-IPS, a 2D RGB image F_{t_p} is passed to the Computer Vision Indoor Positioning System (CV-IPS). Here the image passes through a CNN Object Detector model and a Multi Object Tracking model respectively. One pass through the pipeline directly gives 2D position estimation for J detected objects indexed by j such that $j = 1, 2, ...J$. Thus, output from CV-IPS is the matrix $V_{t_p} = \{V^j\}^{M\times 1}$ where $\{V^j\} = [P_{t_p}^j, (x_{t_p}^j, y_{t_p}^j), t_p]$. $(x_{t_p}^j, y_{t_p}^j)$ is the instantaneous 2D position coordinate for the j^{th} detected object at timestep t_p. $P_{t_p}^j$ represents a **track id** used by CV-IPS to represent a "track" for this object over time. Unlike pseudo id, track ids are system generated and may change over time for the same object. From matrix V_{t_p}, it is evident that CV-IPS can track objects on the continuous space but cannot uniquely distinguish them.

Matrices B_{t_p} and V_{t_p} are inputs to the Unique Association and Tracking (UAT) containing two sets of identity and spatial information from all M users at timestep t_p. The output from UAT is the system output $U_{t_p} = \{U^j\}^{M\times 1}$ such that $\{U^i\} = [A_{t_p}^i, (x_{t_p}^j, y_{t_p}^j), t_p]$. Here $A_{t_p}^i$ represents a pairing between $O_{t_p}^i$ and one of the track ids for the i^{th} user.

Our objective is to establish **unique** $A_{t_p}^i$ for all M users at timestep t_p and, **hold** this association until last timestep T. If $A_{t_p}^i$ is correct and unique, the proposed Multimodal Indoor Positioning System (MIPS) **uniquely distinguished** each user using the pseudo id from BLE-IPS and logical association, **track them over continuous space** using spatial information from CV-IPS.

Chapter 4 System Design

In this chapter we present the design goals of the proposed Multimodal Indoor Positioning System followed by a brief description of the System Architecture and working methodology of the proposed MIPS system. Chapter 5 will elaborate on the experimental setup along with operational details of each subsystem with discussion to why certain design choices were made.

4.1 Goals and Objectives

The proposed MIPS design aims to improve localization performance to track multiple users using Computer Vision and Bluetooth anonymously. Following the performance requirements discussed previously and privacy-preserving goals in introduced in [65], the following design goals are set for the proposed MIPS

- **Anonymity** - In order to achieve anonymization of users, the overall system design must prevent user identification. In order to achieve this requirement, we used an anonymous device identifier associated with user's smartphone that is pseudo and session specific (i.e an anonymous identifier that changes with each session) and IoT edge device running major calculations does not collect or store any personal data (i.e mac address of the phone, phone number, facial features, body pose, gesture from image data).

- **High Accuracy Tracking and Low Localization Error** - The second major requirement for the proposed MIPS is to achieve $\geq 90\%$ tracking accuracy and $\leq$ 50cm localization error.

- **Number of Concurrent Users** - Our third major requirement is to demonstrate that the proposed system can serve more than 4 users concurrently. In this research, we aim to demonstrate service to 6 concurrent users participating in 5 different mobility scenarios.

- **Latency** - The fourth requirement is to demonstrate use of inexpensive IoT Edge device, the proposed system can operate in real-time or near real-time speeds to show potential for real-world application. We expect the system to have an latency less than or equal to 100 milliseconds (i.e operate at or over 10Hz).

- **Minimal User Interaction** - Once the smartphone app is activated, the proposed system should require minimal to no input from users in order to perform unique association and tracking.

- **Edge Computing** - The proposed MIPS should demonstrate low latency to justify deployment on a modern Edge device. This allows for robust privacy-preservation since Edge Computing is decentralized by nature thereby it can provide localization information as an "on-demand" service that reduces chances of storing historical trace of the user. Furthermore, Edge devices supports M2M communication which allows the proposed system to scale up larger deployment and work with heterogeneous smart devices.

- **Generalizability** - The system architecture shown in (refer Figure 7) should support other forms of RF-based Indoor Positioning System alongside CV-IPS to function in different indoor environments.

- **Minimum Service Area and Cost** - The service area for the proposed system should be adequately large enough to distinguish,localize and track 6 concurrent users. Moreover, the system should be able to use existing WLAN infrastructure for communications and should require no additional devices to read BLE advertisements from BLE-beacons. Finally, the proposed system should be able to run completely in an Edge. Achieving these objectives will facilitate in overall low cost of deployment.

4.2 System Architecture

Figure 8 depicts the system architecture of the proposed MIPS system. Note that, the figure depicts the system in its operational stage. In a later chapter we will discuss the "data collection" stage that is needed to train the Machine Learning model in BLE-IPS and Deep Neural Network Object Detector in CV-IPS.

The proposed system architecture is divided into three distinct "layers". In the "Physical Layer", all users with their respective smartphones, Bluetooth Low Beacons and the Edge device containing the overhead camera is present. Only users visible to the camera unit (i.e present in camera's field of view) are considered for localization. All users carried Android devices, each of which is equipped with a custom Android app. The Edge device is a custom Linux Single-Board-Computer (SBC) containing a ARM CPU, an Nvidia GPU and a Raspberry Pi camera unit coupled with a wide-angle lens pointing downwards (overhead camera).

In the "Communication Layer" data from all users' smartphones is aggregated and "published" to a pre-defined topic to a MQTT broker. In this layer, a 2D RGB image captured by the overhead camera unit is passed to the Edge device via the CSI bus. Additionally, during each timestep, the Edge device using the MQTT messaging protocol "subscribes" to the pre-defined topic to acquire the Bluetooth data from users smart devices. The proposed architecture allows for the Communication Layer to be executed in a separate physical device or as a "software" layer in the Edge device itself. We have implemented the latter.

The proposed MIPS executes unique association and tracking in the "Application Layer" using three subsystems (refer Figure 7).The Bluetooth Low Indoor Positioning System (BLE-IPS) is an Indoor Positioning System which utilizes the Fingerprinting technique (a ML model) to predict the user's location out of a collection of discrete "grid points" on the observation area based on the RSSI data available at a particular time

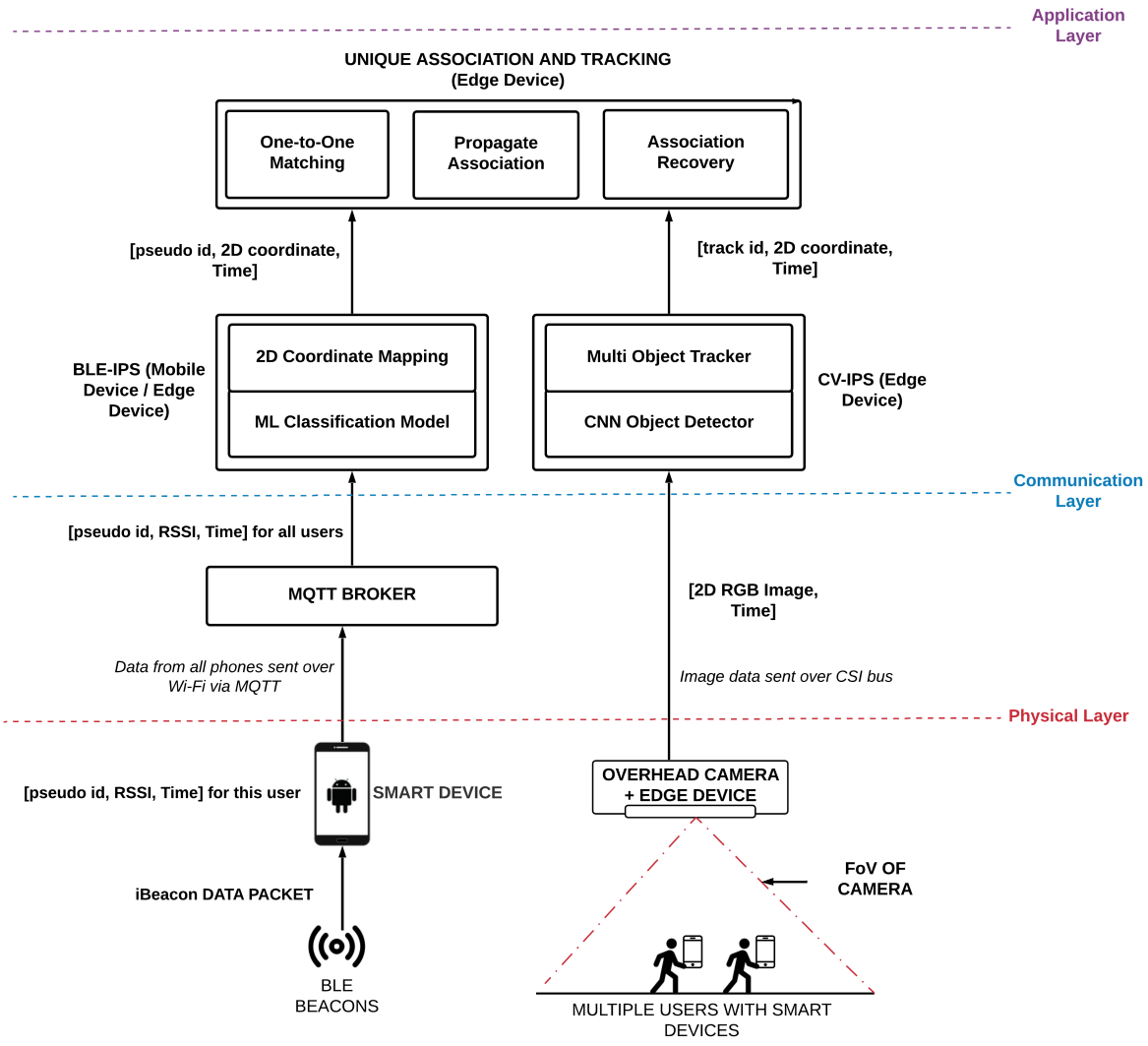


Figure 8. System Architecture (Operational Stage)

instance. BLE-IPS distinguishes an user using the "pseudo-id" obtained from user's mobile device. After determining a discrete grid point, it is converted to the 2D coordinate via the "2D Coordinate Mapping" model. This 2D coordinate is defined with respect to the image coordinate system which can then be converted to the global coordinate system using a scalar multiplier. Ideally BLE-IPS would be executed within the smartphone but due to implementation limitations, the BLE-IPS subsystem is executed on the Edge device itself. Note that, the BLE-IPS module predicts location from RSSI data for multiple users whose data is available at that time instance.

Running in parallel to the BLE-IPS is the Computer Vision Indoor Positioning System (CV-IPS) which consists of two modules a CNN Object Detector and a Multi Object Tracker which processes the raw 2D image to extract tracklet_ids and 2D coordinates for each detected "objects". Note that, unlike BLE-IPS, CV-IPS cannot distinguish users individually but can learn to recognize them as generic "objects" under a class called "Person" and assign a tracking id to track their movement across image frames. Moreover, the precision of localization in CV-IPS is in centimeter level whereas the localization precision of BLE-IPS is in meter level.

The outputs from the BLE-IPS and CV-IPS subsystems are then fed into the "Unique Association and Tracking Module" (UAT). The UAT module consists of three group of algorithms viz. "One-to-One Matching", "Propagate Assoication" and "Association Recovery". As shown in Figure 8, using the two sets of 2D position coordinate, the UAT module first matches the user's pseudo-id to the tracklet-id assigned to him/her by the CV-IPS module and then ties the high precision 2D localization from the CV-IPS module to achieve multi user unique association and tracking. This process occurs for that number of users whose pseudo-id is available to the system at that time instance and the process is repeated over and over until the last timestep in the system.

When an execution of UAT is completed, the output is a set of pseudo-ids with 2D localization from CV-IPS which are then passed back to the user over MQTT. As the analysis is post hoc, output to users were not implemented.

4.3 Working Methodology of MIPS

We begin by presenting Figure 9 that elaborates on the "Application Layer" introduced in Figure 8. The proposed system should correctly and uniquely match A **track id** from CV-IPS to a **pseudo-ids** from BLE-IPS such that the user may be tracked on the continuous space using the high resolution spatial information updated every cycle from CV-IPS and, be uniquely distinguished using the unique pseudo-id without providing

any personal information to the system.

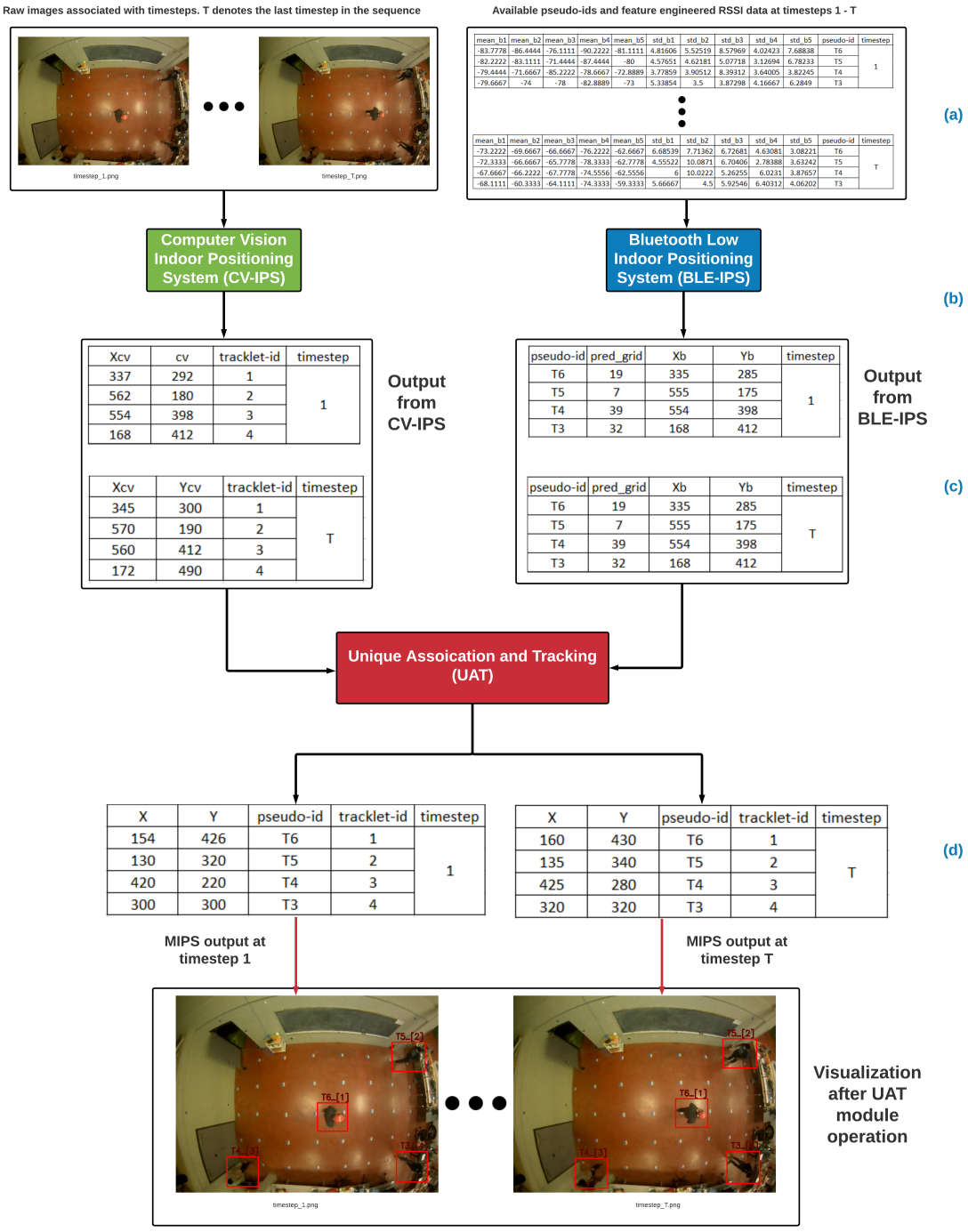


Figure 9. Working Principle of MIPS

At timestep 1, a valid 2D RGB image and RSSI data with pseudo id for the four user be present (Figure 9(a)). The image is sent to the CV-IPS subsystem that uses

the pre-trained CNN Object Detection model to predict the "bounding box ", regions which has the highest probability of containing objects belonging to the class "People". This information is then passed onto a Multi Object Detection model which compares the current box predictions to boxes available in known tracks decides either to update the 2D location information for existing tracks or assign new tracks to detected "objects". Each track is denoted by a "track id" (sometime referred as tracklet, tracklet_id in literature [66]). Concurrent to CV-IPS, the BLE-IPS takes in the RSSI data, extracts a number of "features" and sends the featured engineered RSSI data to Machine Learning model. The output from the model is processed to return a 2D location and tied to each pseudo-id as shown in Figure 9 (c).

The Unique Association and Tracking subsystem takes the output from BLE-IPS and CV-IPS as input and performs two tasks viz. match new pseudo ids to track ids or use last known pairings to update location estimation for one or more users at the current timestep. As seen in Figure 9 (c), no pseudo ids were previously paired to tracks from CV-IPS. In this condition, a MIPS matches pseudo ids to all available tracks using "One-to-One Matching" state chooses between heuristic algorithm depending on the number of pseudo id, track id pairs that needs matching. Once one-to-one matching is performed, users' pseudo ids to track ids are paired as shown in Figure 9 (d) (red bounding boxes) for timestep 1. The numeric string such "T6_[1]" depicts which pseudo id is paired to which track id at that timestep. The 2D coordinates (X, Y) for each of the pseudo ids are directly passed on from the tracks in CV-IPS system by logical association.

Now for the subsequent timestep T in which the four users were present in the observation area, if CV-IPS can "maintain" the track id it had assigned to the four users in the old timestep 1, then the pseudo id to track id association established in timestep 1 is valid in timestep T via logical association. Hence, the four users association is "propagated" into the new timestep (Figure 9 (d), timestep T) and their track information is

updated with the new 2D coordinate estimations from CV-IPS. The state is called "Propagate Association". Notice the values of (X, Y) in Figure 9 (d). For the two timesteps, the values are very close to one another. This shows that the four users have now been tracked across time with very precise localization information by the MIPS and yet they are distinguished uniquely using their phone generated pseudo id. In the subsequent timesteps, if new users are introduced or old user leaves the scenario, the MIPS system repeats the steps of association and propagation and, the cycle continues.

Chapter 5 Experimental Setup and Methods

In this chapter, we first present the details related to the experimental setup followed by brief discussion on composition and working methodology of the three major subsystems that constitutes the proposed MIPS. Table 4 lists all the important parameters/configurations used for the experiment and Figure 10 shows the physical hardware used in the experiment.

5.1 Experimental Setup

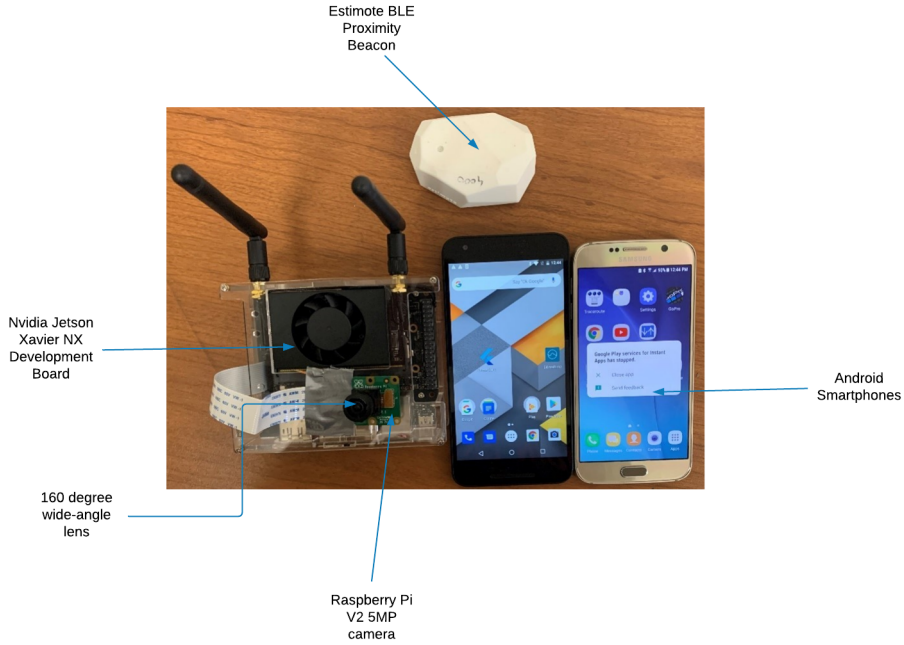


Figure 10. Hardware used in Experiment

The first major design consideration for the experiment was to determine the size of the experimental testbed. In Indoor Positioning System literature, the size of the experiment area (testbed) varied widely depending on constituent technology and application. For MIPS with a vision component, the size of testbed is limited by the camera's Field-of-View (FoV) and complexity associated with deploying multiple cameras to cover a large workspace. Having multiple cameras increases computational requirement

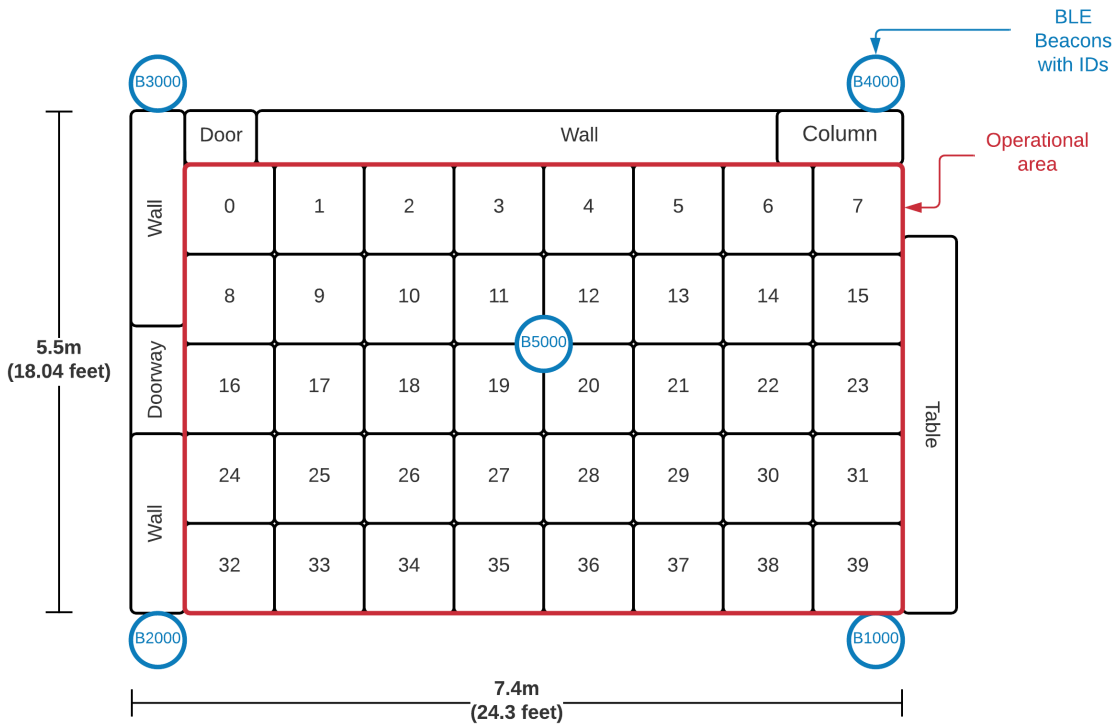


Figure 11. Top-view of the Experimental Area (40 grid configuration)

since data from all cameras needs to be synchronized either to an external server or processed asynchronously in a distributed network. This adds to latency and complexity of the overall system. Our proposed system included an an overhead camera unit coupled to the Edge device. The height at which the camera unit is placed, the camera's FoV and resolution determined the size of the testbed. While choosing these three parameters, we ensured that the size of the observation was sufficiently large to accommodate movement of 6 concurrent users and RSSI data from all Bluetooth Beacons were accessible at any point in the observation area.

Figure 11 shows a diagrammatic representation of the experimental area along with the placement of the BLE beacons (blue circles) and demarcation of the observation area (red rectangle). The area is 40.7 m^2 divided into a 40-cell grid. 5 Estimote Bluetooth Proximity BLE Beacons were placed at 5 location on the map, each of which were placed roughly 2 meters off the ground. 6 Android smartphones were used, each of which was

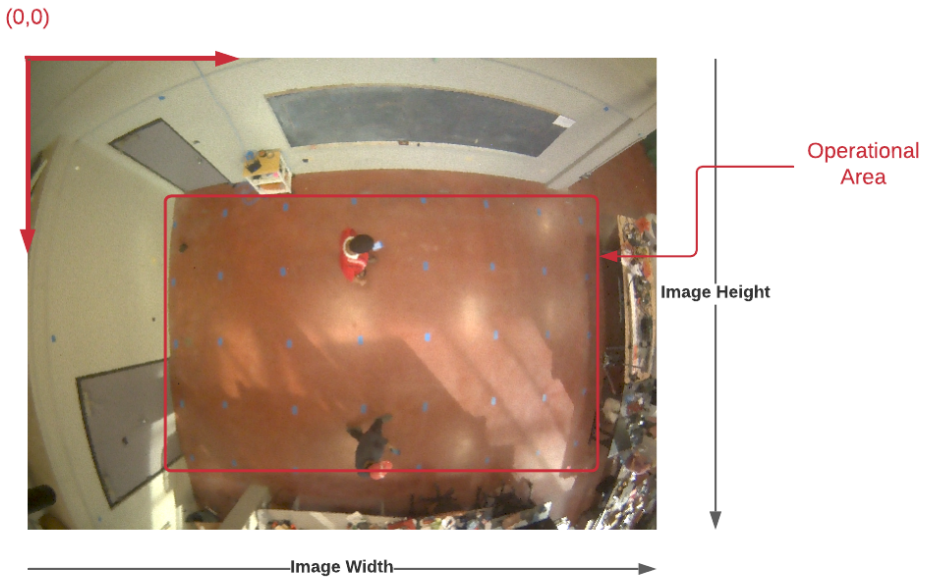


Figure 12. Overhead view of the experimental area

assigned to one user through out the experiment. The smartphones used were 4 LG Nexus 5s, 1 LG Nexus 5X and 1 Samsung Galaxy S6. Note that, due to difference in antenna design, orientation, gains, each generation of phones perceives RSSI signal value differently for the same physical location. While most IPS works with data collected by the same family of mobile devices, we opted to use multiple devices since this would more closely resembles real-world scenarios where users would have smartphones from different manufacturers.

The camera unit along with the Edge device was placed at the center of the observation area 15 feet high from the ground-level as shown in Figure 13. Figure 12 shows the observation area as seen from the CV-IPS output at this height. The 2D coordinate system is defined in terms of the image coordinate system whose origin lies at the upper-left corner of the image and each point of the image is defined in terms of pixel value. As one moves to the right, the X-axis value increases and as one moves towards the bottom the Y-axis value increases. The blue stickers seen in Figure 12 marks the center points of each grid cell which were used by the users to navigate the experimental

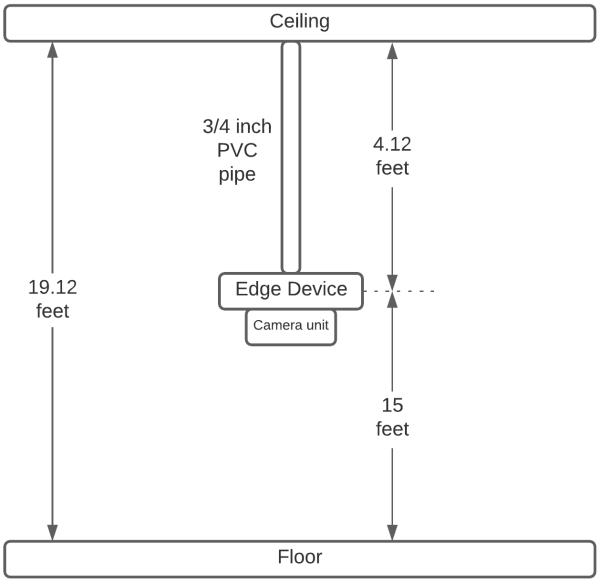


Figure 13. Schematic diagram of overhead camera setup

area. Note that, for our case, the actual FoV of the camera unit is greater than the physical limits of the testbed but this is not a norm but a by product of the combination of lens and camera height.

Another important consideration for our experimental setup is the primary computation unit. Note that our design goal requires the computation device to be cheaper but have sufficient computation power to provide localization services to multiple users in a privacy-friendly manner. We introduced in section 2.2.3 the "Edge" devices, which are local machines capable of using IoT communication protocols to communicate and interact with a wide variety of IoT devices for serving various indoor application needs. Recent works such as [15, 30, 51] have shown that modern Edge devices are suitable for Indoor Positioning Systems since they consumes less power (less than 30 Watts) are priced well below $1000 mark but can provide robust indoor localization since they come equipped with sufficiently powerful CPUS and GPUS which can run Deep Learning models.

The Edge device chosen for our experiment is the Nvidia Jetson Xavier NX AI computing kit from Nvidia which comes equipped with an integrated 384-core NVIDIA

Volta GPU with 48 Tensor Cores clocked at 1.1 Ghz, 6-core NVIDIA Carmel ARMv8.2 64-bit CPU running at 1.4 Ghz and 8GB 128-bit LPDDR4x.Furthermore, using a proprietary library, this board can poll full 5M images from inexpensive Raspberry pi via the CSI bus at 21FPS without slowing the CPU down thanks to built-in hardware acceleration and using full resolution allowed us to use the full 160 degree view. The device consumes only 15 Watts and natively supports MQTT which is a crucial design consideration since the entire communication pipeline in our proposed system is based on MQTT. The builtin GPU with the 8GB 128-bit shared RAM is essential to process the chosen CNN model at near-realtime speed without inducing large latency on the CPU.

For the BLE-IPS subsystem, the BLE beacons chosen were Estimote's Proximity Beacon which are BLE 4.2 supported System-on-Chips (SoC) powered by an ARM Cortex-M4 32-bit processor running at 64 Mhz with 512 kB Flash memory. These devices are battery operated RF=tags which continuously sends out a "beacon", a pre-formatted data in Apple's Open-Source advertizement protocol called "iBeacon". An BLE beacon configured in iBeacon periodically broadcasts a data packet within its operational area. The protocol contains a 16byte Universally Unique Identifier (UUID) coupled with optional 2 byte major and 2 byte minor values. When a advertisement packet is received by a smartphone the app first parses the UUID and then the major and minor values to identify the beacon. It then forms the RSSI value based on perceived strength of the received advertisement signal and a pre-defined signal strength value at 1 meter distance. The RSSI value is lower when the smartphone is closer to the BLE beacon and progressively increases as the smartphone moves away. Lower RSSI value indicates close proximity and vice-versa.

Figure 14 shows the User-Interface of the smartphone app that installed inside each user's smartphone. It was written on Java and utilized an open-source Bluetooth library to parse iBeacon data into Beacon ID and corresponding RSSI value.The beacon IDs are designated on the top row of Figure 14 with the bottom row showing the RSSI

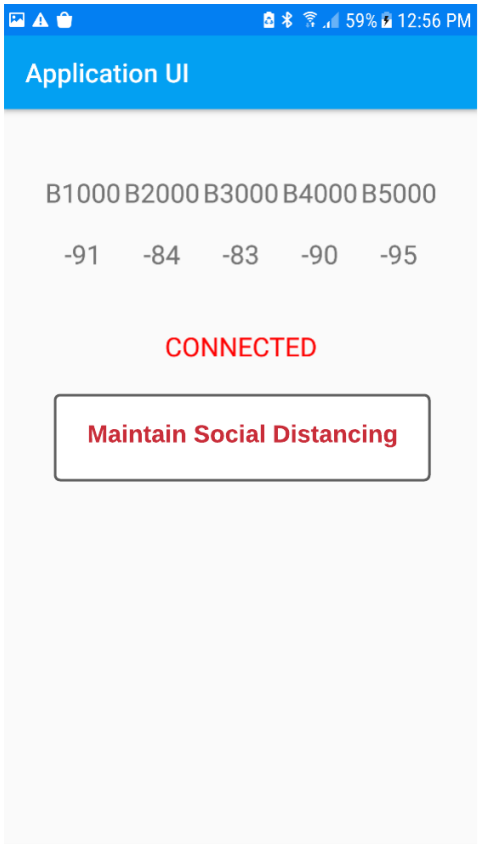


Figure 14. Smartphone Application

value in dBM at a certain interval. The row below shows the status of connection of this smartphone to overall communication network via MQTT. "CONNECTED" indicates that the smartphone is connected to the MQTT broker at that time and is transferring RSSI data at certain interval. Note that the localization output while available to the Edge device and on the MQTT network was not broadcasted to the users via the app in an effort to reduce complexity of implementation. An important point to note is that the users had to only turn the app on and move about the observation area during experimentation (data collection). No other activities were required to be performed by the user to use this app. All data collection, formatting and subsequent transfer to the Edge device was done autonomously. The app also showed a static message to inform users to maintain social distancing as per IRB regulations set for this experiment.

MQTT (Messaging Queuing Telemetry Transport Protocol) is the chosen communication protocol with which the Edge device and all smartphones maintain device-to-device communications. Introduced in 1999, it is a publisher-subscriber protocol designed to use very little bandwidth to exchange byte sized data between large number of interconnected devices [naik2014]. It is a connection-oriented communication method that uses TCP/IP as transport protocol and TLS/SSL for security. As a result, MQTT can be directly used with existing WLAN infrastructure.For our usecase, MQTT is most suitable since all the smartphones requires to send RSSI data chunks to the Edge device in a 1 second intervals. To minimize complexity and cost by having additional hardware, the MQTT broker was simulated directly inside the Edge device itself. Note that, the Edge device itself is also a MQTT client subscribing to the "ANDROID/RSSI" topic on which all Android devices are sending RSSI data along with pseudo-id associated with those smartphones. Ideally, it the proposed system would be able to accommodate data transfer from as many smart devices (i.e. users) as possible at at any given time but due to technical limitations and implementation challenges, a variable timing mechanism had to be implemented. More details on this timing mechanism is presented in Chapter 6.

5.2 BLE-IPS: Description

The BLE-IPS subsystem uses Bluetooth Low technology to localize users in one of the 40 grid cells using the principles of Scene Analysis (a.k.a Fingerprinting) techniques introduced in Section 2.4.2.

The first important parameter for the BLE-IPS is the choice of reference locations (cells) that represents the position of the user based on the RSSI signals measured from all accessible beacons. Usually deployed as a grid, the choice of number of cells and their size is not straightforward since RF signal in indoor environment is affected by a multitude of site-specific parameters that makes using the grid setup knowledge from one experiment to another very difficult, often incompatible [yasin2016]. Moreover, cell density varied

widely from experiment to experiment with each having performance unique to that testbed. For example in [67] authors developed an iterative KNN model for a 28x21 m^2 testbed were each cell was a square of size 0.6m x 0.6m and placed 1.2 meters apart. Subedhi et al. [37] divided their testbed into square cells of side length 1.35m and each was placed 2.7meters apart. Since no concrete information regarding the density and size of cells was found for a testbed similar to ours, we opted to deploy the BLE-IPS for a 40-cell grid configuration as shown in Figure 11. In this configuration, each grid cell is placed 1 meter apart and the cells are 0.5 m x 0.5 m.

The choice of BLE beacon and number of becons deployed has a direct effect on the performance of the Machine Learning model. However in literature, the number of beacon that needs to be deployed is often the outcome of the experiment itself. BLE beacons broadcasts their advertisement signal at 10Hz nominally to conserve power. This causes sharp drop in signal strength as one moves further away from the beacon and the fading occurs non-linearly [26]. As a result, cells which are farther away from the beacon have lower detection accuracy than cells which are at close proximity [67]. Another important beacon parameter to consider is the advertisement protocol. Apple's open-source iBeacon and Google's Eddystone protocols are two of the most widely used protocol with iBeacon being the most widely supported protocol by BLE-Beacon vendors. Considering all the above, we opted to place four beacons at four corners with a fifth beacon in middle. The Estimote Beacons used were configured to use iBeacon protocol and the Android app used an open-source library to read and parse beacon id and form RSSI value based on the strength of the advertisement packet received.

The Machine Learning model chosen was the Random Forest classifier. Apart from Random Forest classifier, k-Nearest Neighbor, Support Vector Classifier, Logistic Regression, Probabilistic Classifiers, Multilayer Perceptron have been used with BLE-IPS with varying results depending upon the indoor environment, beacon placement, density of users and data processing techniques used [36]. We opted to use Random Forest

Classifier since its principle of training numerous Decision Trees for ensemble voting has shown to have ¡= 2 meter localization error for ¿90% time in multiple studies [3,4]. For a detailed study on various Machine Learning models used in RF-IPS, we refer the reader to [36].

An important step in machine learning projects is the choice of "features" which are used by the ML model to characterize each classes in the response variable. In our case, the response variable is the cells that constitutes the grid and each cell needs to be uniquely characterized by the "feature" variables derived from the raw RSSI values obtained from the beacons placed on the map. Note that, choosing the right features helps in mitigating RF-specific issues such as multipath effect, signal attenuation, NLOS conditions and improves the model's capability to discriminate between different cells better. Removing outliers has also been shown to be very effective in stabilizing RSSI data in [68].

In RF-IPS, rolling mean is one of the most commonly used feature in RF-based IPS whose utility was explored in-depth in [26]. Variance and Standard Deviation was shown to be effective in extracting region specific information from k-RSSI samples for a BLE-IPS application in [xin-yu2015]. Other common features include id of the nearest beacon [41], weighted moving average and adjusted RSSI [33].

In order to create RSSI features, sufficient RSSI sample needs to be collected at a particular timestep. In this regard, the choice of scanning window is crucial since slow sampling ($\geq$ 5s) may yield stable RSSI data provided the user stands still in each timestep which does not real-world walking scenario emulated in our experiment. Setting the scanning window to a smaller value raises complexity in synchronizing RSSI data from multiple users and too frequent sampling reduces effectiveness of multipath mitigation strategies such as rolling mean. This occurs since the recording device will capture fluctuating signals without giving enough time for the RSSI signal to stabilize [26].

Considering all the factors above, we opted to create the rolling mean, standard

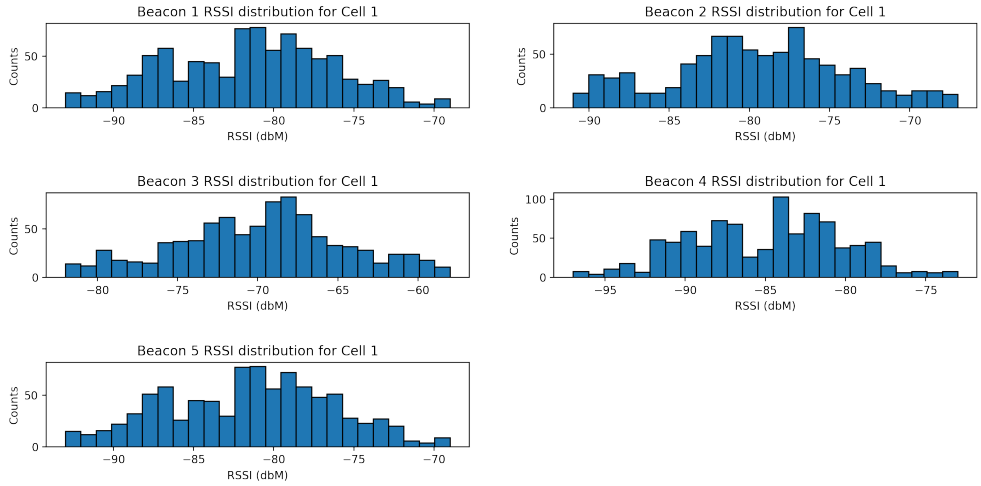


Figure 15. Cell 1 Raw RSSI data distribution from Scenario 1

deviation and skewness features for each of the 5 beacons and, had set the scanning window to 1 second. The skewness feature was selected after analyzing the Bluetooth data for individual location which often showed showed skewness either to left or right of the mean depending upon the location and beacon. A sample is shown in Figure 15.

5.3 BLE-IPS: Method

The BLE-IPS subsystem determines the location of the user in two steps. First, it generates a total of 15 features by taking 9 raw RSSI samples for each of the 5 beacon from current timestep and previous two timestep. Outlier removal is used to drop samples that did not fall within 2 standard deviations for the samples in the window. The system then forms a input matrix were each row corresponds to a pseudo-id and the feature engineered RSSI vector associated to it. This matrix is then passed to the Random Forest Classifier. As Random Forest is a deterministic classifier, it is guaranteed to choose 1 out of 40 cells as long as there is a valid featured engineered RSSI vector. Furthermore, it is also guaranteed that in a particular timestep that the BLE-IPS will predict that many cells that equals to the number of entries in the input matrix to BLE-IPS. This is possible since the quality of data transmission between the smartphones and Edge device can be

tightly controlled using the builtin QoS feature of MQTT [69].

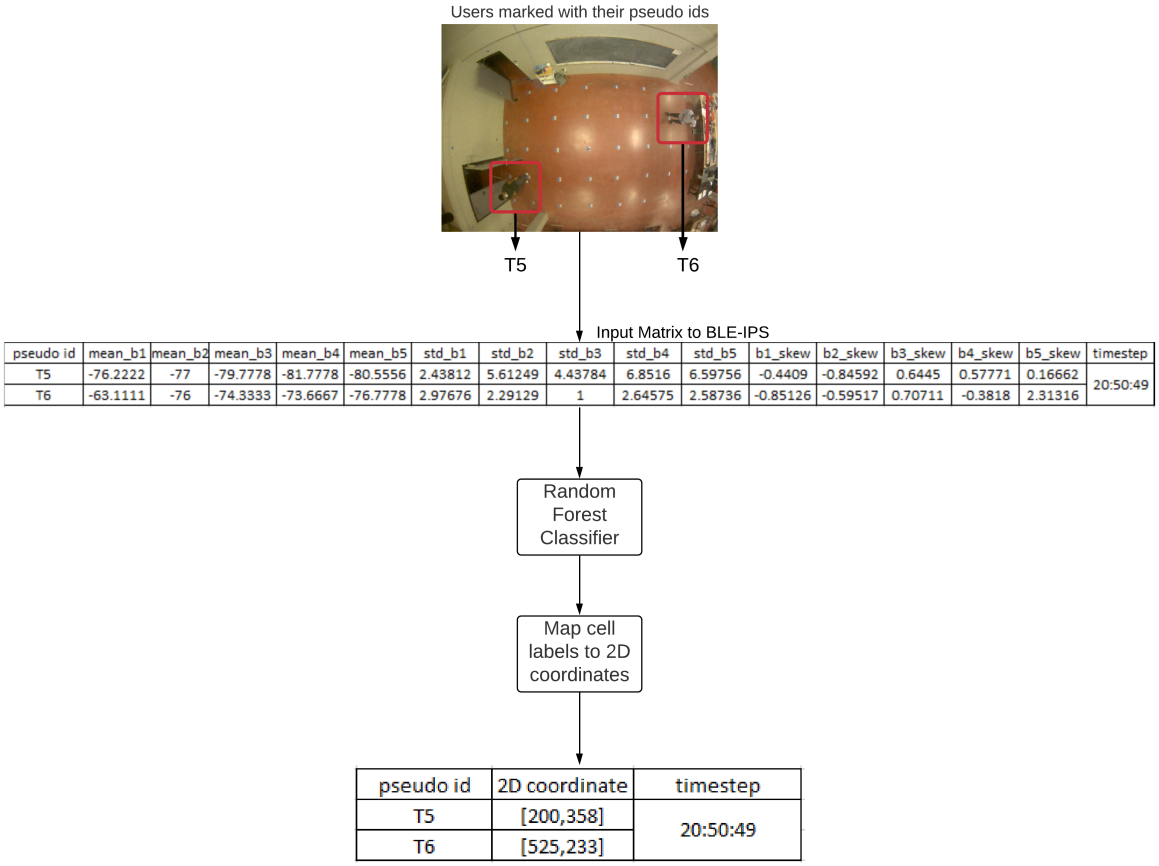


Figure 16. BLE-IPS Method

Output from the Random Forest Classifier is a matrix that contains the pseudo id and predicted cell label for each entry in the matrix. A linear mapping function converts the cell labels to a 2D coordinate based on a apriori mapping between the centroids of the grid cell and the location label. The process is shown in Figure 16. The final output from BLE-IPS is the B_{t_p} matrix in which each row contains a pseudo id and centroid of the grid cell predicted by the Random Forest classifier. It is important to note that, the pseudo ids themselves cannot be part of the feature engineered data since they session specific randomly generate by Android app during inference stage. If pseudo-ids were predetermined in the training stage, this would then become an unique identifier exclusive to that individual which would circumvent the privacy-preserving attribute of the pseudo

ids.

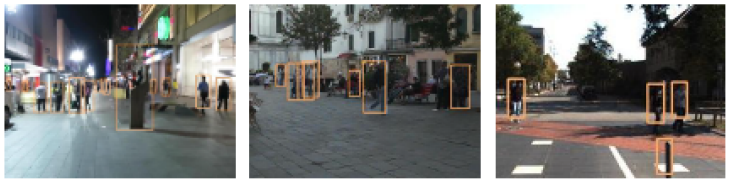
5.4 CV-IPS: Description

One of the major challenges in CV-IPS is the choice of camera view which directly impacts the performance of a CNN Object Detector. Majority of CNN Object Detectors were developed for standard front-facing camera images with the primary assumption that people appears to be in upright and their overall appearance remains relatively unchanged over the image plane [70]. However, omnidirectional images popular in overhead CV-IPS literature has two drawbacks that it makes using omnidirectional images sub optimal for use with a CNN Object Detector.

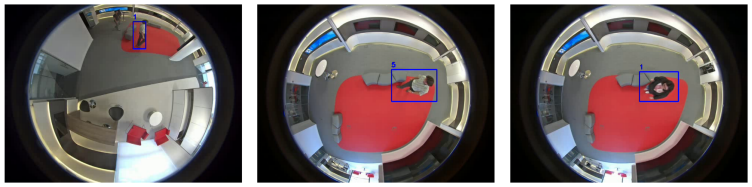
Firstly, the convolutions layers in a CNN detector assumes the visual appearance of objects do not have spatial variance, only their apparent size changes. However, as seen from Figure 17 (b), the omnidirectional overhead images of people have severe geometric distortion which dramatically changes the appearance of an individual as the traverse from the center of the image towards the periphery and vice-versa. This nullifies the spatial-invarince assumption and causes sub optimal detection [16] . Secondly, creating training samples from omnidirectional images is more time consuming since the bounding boxes needs to be radially aligned in addition to be manually drawn around people to represent ground-truth regions.

To account for these limitation, we have chosen to use a 160°wide angle lens for the overhead camera. While not covering as much area as omnidirectional camera, this image view does not suffer from server geometric distortion as shown in Figures 17(c,d). As a result, we were able to use the YOLOv3-Tiny directly on the overhead images without requiring any perspective transformation nor required to develop techniques to radially align any ground-truth bounding box thereby circumventing saving time and reducing complexity [70].

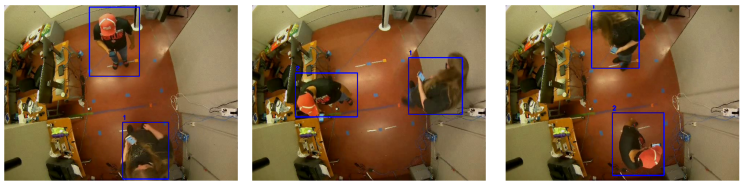
(a) Front-facing image [mot2015]



(b) Overhead omidirectional image [blanc2021]



(c) Overhead wide-angle view at 8 feet height



(d) Overhead wide-angle view at 15 feet height

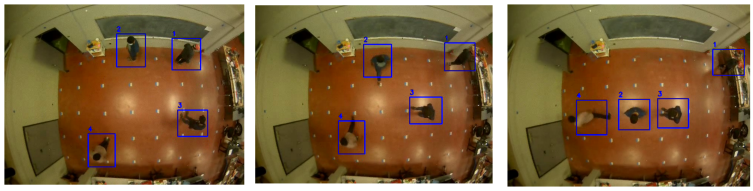


Figure 17. Object Detection and Tracking in CV-IPS

Another major consideration when choosing a CNN Object Detector for Edge application is its size and computational requirements. This choice is dictated by how many CNN layers, weight parameters, number of stages detection occurs and post-processing algorithm the model employees for object detection. To illustrate, YOLOv3, the full version of YOLOv3-tiny has 136 layers and over 20M parameters. It can reaches 30FPS on MS COCO 2017 dataset only when running on a GTX Titan, a server-class graphics card whereas Jetson NX whose graphics card has only 384 CUDA cores struggles to maintain 10FPS with the same model.

Realizing this trade-off, a number of research have emerged aimed at modifying

YOLOv3 base model to fit in to Edge devices . Mao et al. [71] greatly reduced the parameter count of YoloV3 by using depth separable convolutions and pointwise group convolutions layers instead of traditional convolution layers and, reported better accuracy than YOLOv3-tiny, the standard small size model of the base YOLOv3 model. Huang et al. [72] developed YOLO-LITE a 7 layer, 4.8 million parameters YOLOV2 model that is 3.8x faster than SSD MobileNetV1 and is capable of running at 21FPS on non-GPU computers. While very promising for our application, its detection accuracy is less than YOLOv3-tiny since YOLOv3-tiny has 36 layers which can capture much more features of the environment and people than YOLO-LITE. Another notable variation of YOLOv3-tiny is the Tinier-YOLO [73] which implemented "Fire Module", a 3x3 and 1x1 CNN combination layers that replaces max pooling layers and achieves high parameter compression (i.e. greatly reduces the number of parameters). Moreover Tinier-YOLO utilizes "Dense Connection" to improve feature detection. This CNN model has a total of 96 layers but with 3M less parameters than that of YOLOV3-tiny.

Based on the above, we have opted to use YOLOV3-tiny as the base Object Detection model for the CV-IPS subsystem. YOLOv3-tiny has 36 layers in total containing 8.66 million trainable parameters. YOLOv3-tiny predicts bounding box location and class labels twice in two densely connected layers (sometimes referred to as YOLO layer) in order to detect large and medium sized objects respectively and concat the outputs at the end. In Chapter 7, we will also used the Tinier-YOLO model to demonstrate the effect of choice of CNN model on the performance of MIPS system. Readers interested on the full working principle of YOLOv3-tiny are referred to [74] and [73] for Tinier-YOLO.

After choosing the Object Detection model, the choice of Multi Object Tracking algorithm was determined primarily from choosing a model from the "Track-by-Detection" family. Track-by-Detection models are most suitable for real-time application since they

only use information up to the current timestep to track objects (Online models). However, their performance are heavily dependent upon the performance of the Object Detection model [66].

The Simple Online and Realtime Tracking (SORT) [59] model is chosen as the Multi Object Tracking (MOT) model even though possessing relatively low tracking accuracy in comparison to recent state-of-the art online trackers such as GOTURN [75], DeepSORT [44], RetinaTrack [76], Learning to Track [57]. Although lacking in tracking accuracy, its simple architectures allows real-time operation on an Edge device. This is because most state-of-the art MOT models have separate neural networks and complex heuristic algorithms to counter short-term, long-term occlusions, missed detections, group detections, recover tracking id which requires considerable amount of processing powers to operate.

To illustrate, DeepSORT which shows 45% reduction in ID-Switching employees a separate CNN Detection Layer for appearance detection (SORT only uses the bounding box geometry) which adds an additional 2.8M parameters to be computed by the system. As a result, DeepSORT requires approximately 30ms by itself to perform one complete cycle on a GTX 1050 GPU which is a desktop class GPU having twice CUDA core count than Jetson Xavier NX. The SORT algorithm in comparison does not require any additional training since it has no separate DNN model for appearance detection nor requires a lot of computation power since the data association algorithm used is the Hungarian Data Assignment algorithm which is highly optimized for Python environment. Another key advantage is its use of Linear Kalman Filter for predicting motion of moving objects with information from current frame and the previous few frames which helps in keeping latency down but at the cost of robustness against occlusion, missed detection which on basis of scenario causes severe id-switching problem [59].

5.5 CV-IPS: Method

The CV-IPS subsystem performs the tasks of object detection and tracking in three steps as shown in Figure 18. For every timestep a 640 x 480 image is available which is passed into the YOLOv3-tiny bounding box regressor. The image is converted to a 416 x 416 letterbox size image and then passed through the network. The network predicts one bounding box for each user present in the image using a series of convolution, maxpooling, downsampling and upsampling operations followed by prediction using anchor box transformation and, deletion of spurious bounding boxes in post-processing using thresholding by object confidence and non-maximum suppression.

In Figure 18, each of the bounding boxes marked with alphabets a,b,c,d contains represents detection for the 4 individuals. Each box contains 4 parameters, the centroid of the box (b_x, b_y), box width and box height (b_w, b_h). In step 2, the information from current detection is passed to the SORT multi object tracker. For the very first detection, SORT assigns a random integer number starting from 1 for each of the detected object. In this situation, step 2 is not executed. During later timesteps, when a fresh batch of detection comes in, SORT forms a similarity matrix by calculating the Intersection-over-Union measure (also known as Jaccard's Index) by comparing the degree of overlap between the current bounding boxes to the last known bounding boxes assigned to a track using Equation 3.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

The similarity matrix is then solved for unique one-to-one matching between detections and tracks using the Hungarian Data Assignment model [77]. However, though assignment by Hungarian model is guaranteed to be optimal it not always guaranteed to be correct. Hence, a minimum IoU threshold value of 0.35 was imposed, below which assignment were rejected. The assignment matrix determines the final unique one-to-one detection to tracking id pairing, using which SORT updates the (X, Y) coordinates for

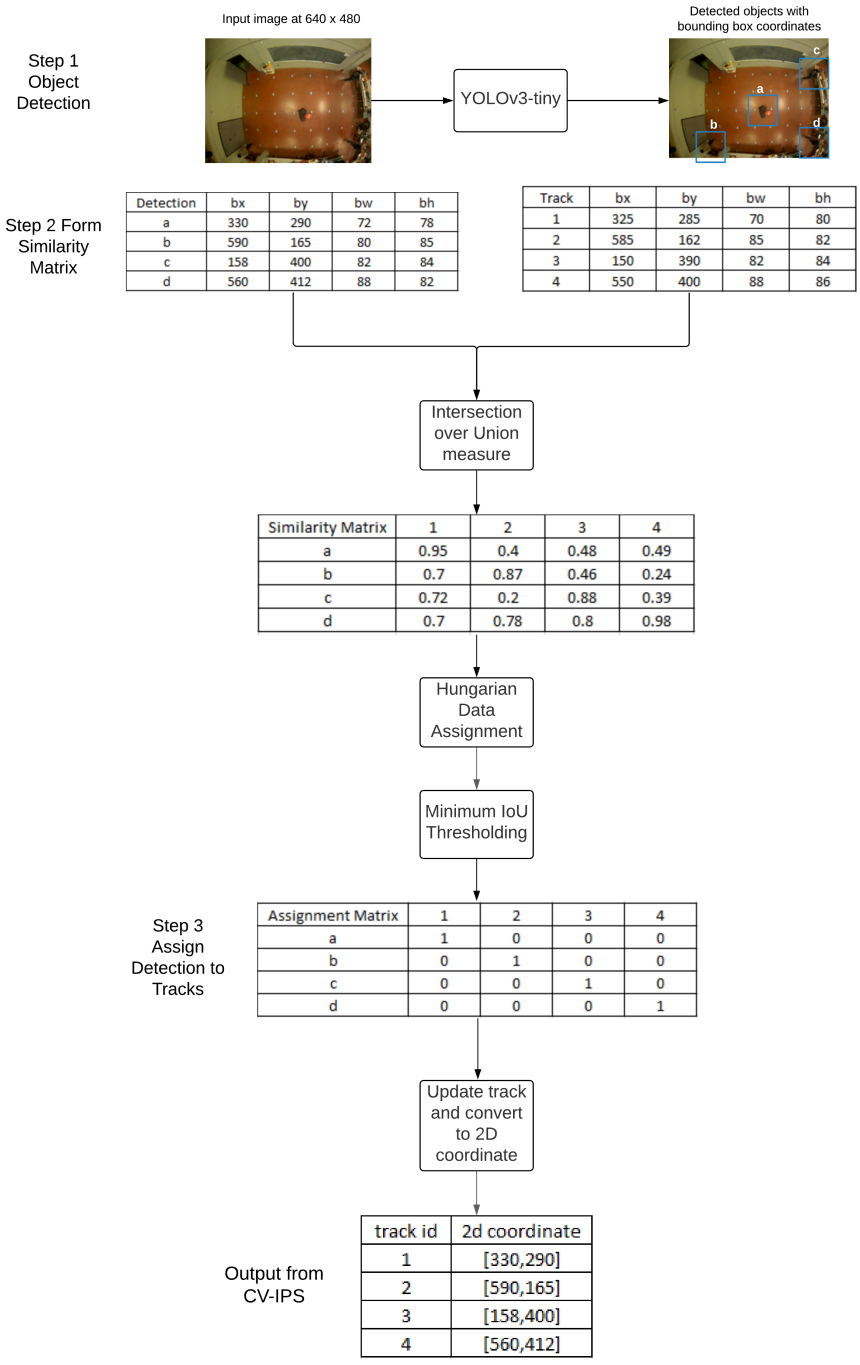


Figure 18. CV-IPS working methodology

each "track" by extracting the box centroid. Note that, the SORT is not tracking the individuals using contextual information from the image but using only bounding box parameters obtained from the Object Detector. Furthermore, to the CV-IPS, the identity

of the individual is only the "track ID" assigned by the SORT algorithm. This track id is mutable and subject to the target object's continuing detection by the Object Detector over time. If the detector misses detection of the target object, SORT deletes the last "track" assigned to that object. Thus, SORT reassigns a new track to the target object when a new bounding box estimation for the missing object is found. The output from CV-IPS is the V_{t_p} matrix that contains the assigned track id and the current (X, Y) coordinates at the current timestep.

5.6 Unique Association and Tracking: Description

In section 4.3 we introduced the Unique Association and Tracking subsystem that performed the critical tasks of unique user association and tracking using three system states. With reference to the notations and variables introduced in Chapter 3, let at timestep t_p, input to the proposed system be B_{t_p} and V_{t_p} the outputs from BLE-IPS and CV-IPS respectively at this time instance. We recall from Chapter 3, A^i_{tp} represents unique and exclusive association between $(O^i_{t_p}, P^i_{t_p})$ pairs.

If both B_{t_p} and V_{t_p} matrices have non-zero data, the system checks if there is any unmatched pseudo ids and track ids by comparing with U_{t_p-1}. If the numbers new pseudo ids to track ids are equal or if the number of track ids exceeds the number of pseudo ids, the system then executes "One-to-One Matching" state using the Nearest Neighbour Greedy Search heuristic algorithm. Nearest Neighbour Greedy Search matching requires both spatial and temporal information from B_{t_p} and V_{t_p} to perform unique association. This algorithm iteratively cycles through each pseudo-id and checks against all available tracklet-id to determine which pair is the most similar by computing a "similarity measure". The similarity measure can be computed using the well-known distance metrics such as Euclidean, Manhattan, Cosine distance since the spatial information available is a 2D vector. For the proposed system, we used a combination of Euclidean distance

and Cosine Similarity measures due to limitations of Euclidean distance for in computing same relative distance between two users occupying two adjacent grid cells. With addition of Cosine distance whose angle assumes a value between 0 - 0.05° when the 2D centroid of the bounding box is less than 70 units away from the centroid of the grid cell. The Greedy search algorithm accepts that association whose euclidean distance and angle from cosine similarity measure falls below 70 units and 0.02° respectively. However, from experimentation we observed these threshold values cannot be static throughout the whole scenario and is subject to dynamic ranging on a scenario-by-scenario basis.

The Hungarian algorithm commonly used in associating detections to tracklets in MOT [66] was a viable alternative to the Nearest Neighbour Greedy Search algorithm. However, for our experiment we did not notice any significant difference in terms of accuracy or speed when the two were compared. It is to be noted that Nearest Neighbour Greedy Search time requirement increases exponentially whereas Hungarian Algorithm increases in $O(N^4)$ [45].

If the number of unmatched $(O^i_{t_p}, P^i_{t_p})$ pairs is only 1, "Temporal Matching" heuristic algorithm is used to execute "One-to-One Matching" state. This is a naive logical algorithm built with the assumption that only 1 user requires unique matching and the pseudo id $O^i_{t_p}$ available was obtained from this individual's cellphone. Hence, association occurs directly using t_p value as basis of similarity.

Now in the next timestep, it B_{t_p} and V_{t_p} have valid data the system can choose to perform association again or propagate last known association. This is accomplished by checking if number of unpaired ids is 0 since when all users are uniquely associated or number of detection falls below the number of pseudo ids present in B_{t_p}, there is not sufficient information to do One-to-One unique association.

Now, if B_{t_p} matrix had no valid data, the UAT module utilizes current V_{t_p} and U_{t_p-1} to compare if there is a change in the collection of track ids with respect to previous timestep. If no changes occurred, the system then executes "Propagate Association"

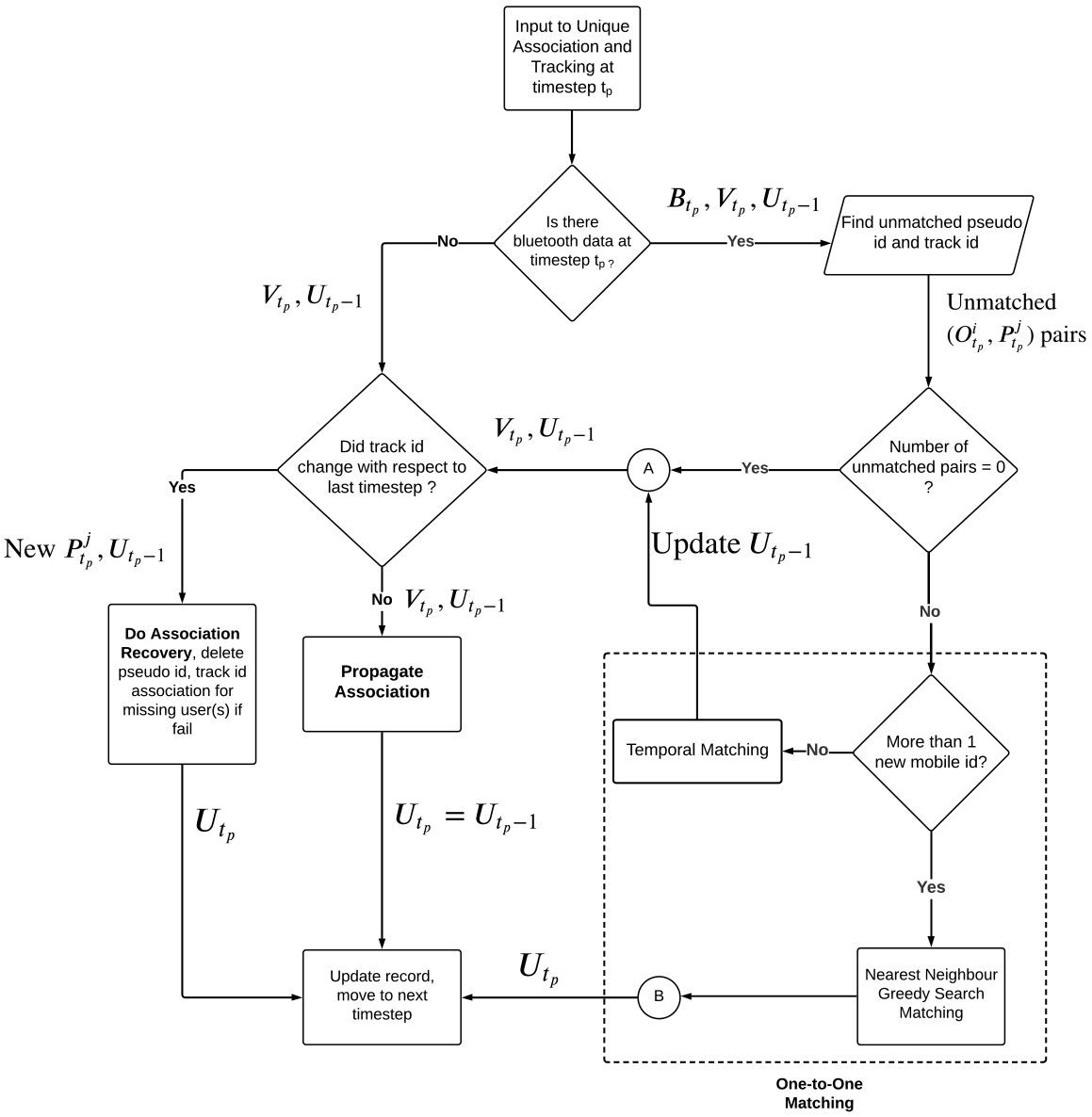


Figure 19. Flow Diagram of Unique Association and Tracking Subsystem

state, a naive heuristic which capitalizes on the fact that, for two congruent timesteps, if YOLOv3-tiny detected the target objects, SORT propagated their assigned track ids to current timestep. Hence by logical association, $P^i_{t_p} = P^i_{t_p-1}$ thus $A^i_{t_p} = A^i_{t_p-1}$ i.e association from previous timestep is valid for current timestep.

However, if the collection of track ids are different between two congruent timesteps

$t_p and t_{p-1}$, UAT module attempts to recover lost association using "Association Recovery" state. This state uses a modified Nearest Neighbour Greedy Search Algorithm that first identifies pseudo ids which lost association in current timestep by comparing with $A^i_{t_p-1}$ in U_{t_p-1} and then searches through the collection of track ids in $V^i_{t_p}$ matrix to find those $P^i_{t_p}$ ids whose spatial information is most similar to the spatial information of lost $O^i_{t_p}$ ids. If new association is valid, the pseudo ids missing lacking association is assigned to the newfound track ids and thus their track resumes. Since the data used is already available in spatial form and in memory, Association Recovery executes in real-time speed but in Chapter 7, we will demonstrate that the utility of association recovery is more scenario specific and is subject to some of the drawbacks of greedy matching strategy.

5.7 Unique Association and Tracking: Method

The three states of UAT subsystem "One-to-One Matching", "Propagate Association" and "Association Recovery" states are executed by four heuristic algorithms namely Nearest Neighbour Greedy Search, Temporal Matching, Propagate Association and Association Recovery. In the following section we briefly discuss the working principle of these four algorithms with examples from Scenario 3.

The Nearest Neighbour Greedy Search matching algorithm works in three steps as shown Figure 20. First, a similarity matrix C_{t_p} is formed in which each cell is formed by a $(O^i_{t_p}, P^j_{t_p})$ pair and for simplicity, we consider that $i = j$. Let $c_{i,j}$ represent each cell in C_{t_p} that contains two values, the Euclidean distance $d_{i,j}$ and angle from Cosine similarity measure $\theta_{i,j}$. Let q_i represent the 2D (X, Y) coordinate for the pseudo id in the i^{th} row and p_j represents the 2D (X, Y) coordinate for the track id in the j^{th} column. Then for each $c_{i,j}$, d_{ij} and θ_{ij} are computed using Equations 4 and 5 as shown below

$$d_{i,j} = \sqrt{\sum_{i,j=1}^{n} (q_i - p_i)^2} \tag{4}$$

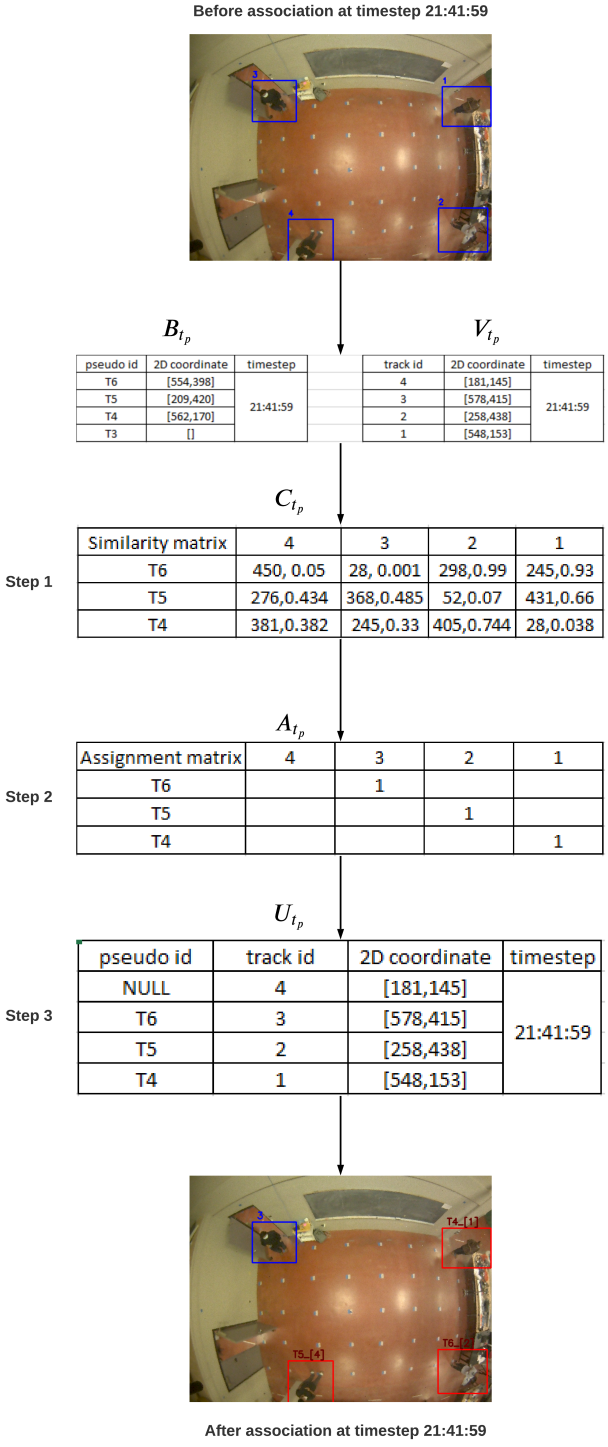


Figure 20. Nearest Neighbour Greedy Search Matching

$$\theta_{i,j} = \arccos\left(\frac{\sum_{i,j=1}^{n}(q_i p_j)}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{j=1}^{n} p_j^2}}\right) \tag{5}$$

In second step, the greedy matching algorithm cycles through all the cells in C_{t_p} and forms an assignment matrix A_{t_p} where each cell $a_{i,j}$ represents an unique one-to-one association between the pseudo id and track id pair at that cell. $a_{i,j}$ can assume a binary value and a value of 1 indicates valid assignment. To assess acceptable assignment, let the threshold Euclidean distance and angle be represented by d_{thres} and θ_{thres} respectively. Then a pseudo id, track id pair pairing is considered unique and exclusive if and only if

$$a_{i,j} = \begin{cases} 1 & \text{if } d_{i,j} \leq d_{thres} \,\&\, \theta_{i,j} \leq \theta_{thres} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Equation 6 is used to perform this operation iteratively until all pseudo ids have been paired with one track id such that all pairs are exclusive and non repeatable. Once all assignment is made, MIPS updates the matrix U_{t_p} in which each pseudo id's association with the track id is recorded and position estimate for the $O^i_{t_p}$ pseudo id is updated with the 2D position estimate from Cv-IPS. Note that, in example shown in Figure 20, user T3 was not associated with any track at this timestep since his smartphone did not set a valid RSSI data at that time.

Now in the next timestep, $t_p = 21:42:00$, user T3's smartphone sends data and since he is the only person who has no association and there is 1 unmatched track id, "Temporal Association" heuristic algorithm is invoked which directly assigns the pseudo id "T3" to the track id 4 and updates . Moreover comparing to $t_p = 21:41:59$ the track ids at timestep $t_p = 21:42:00$ are identical which entails their previous pseudo id, track id association are also valid for this timestep. As a result, the system also invokes "Propagate Association" state, executed by an heuristic algorithm of the same name that simply updates the 2D position estimation for these pseudo ids and updates U_{t_p-1}. Figure 21 demonstrates the two heuristic algorithms.

The Association Recovery heuristic algorithm works using the Nearest Neighbor Greedy Search Matching algorithm and is invoked only when there is no valid Bluetooth

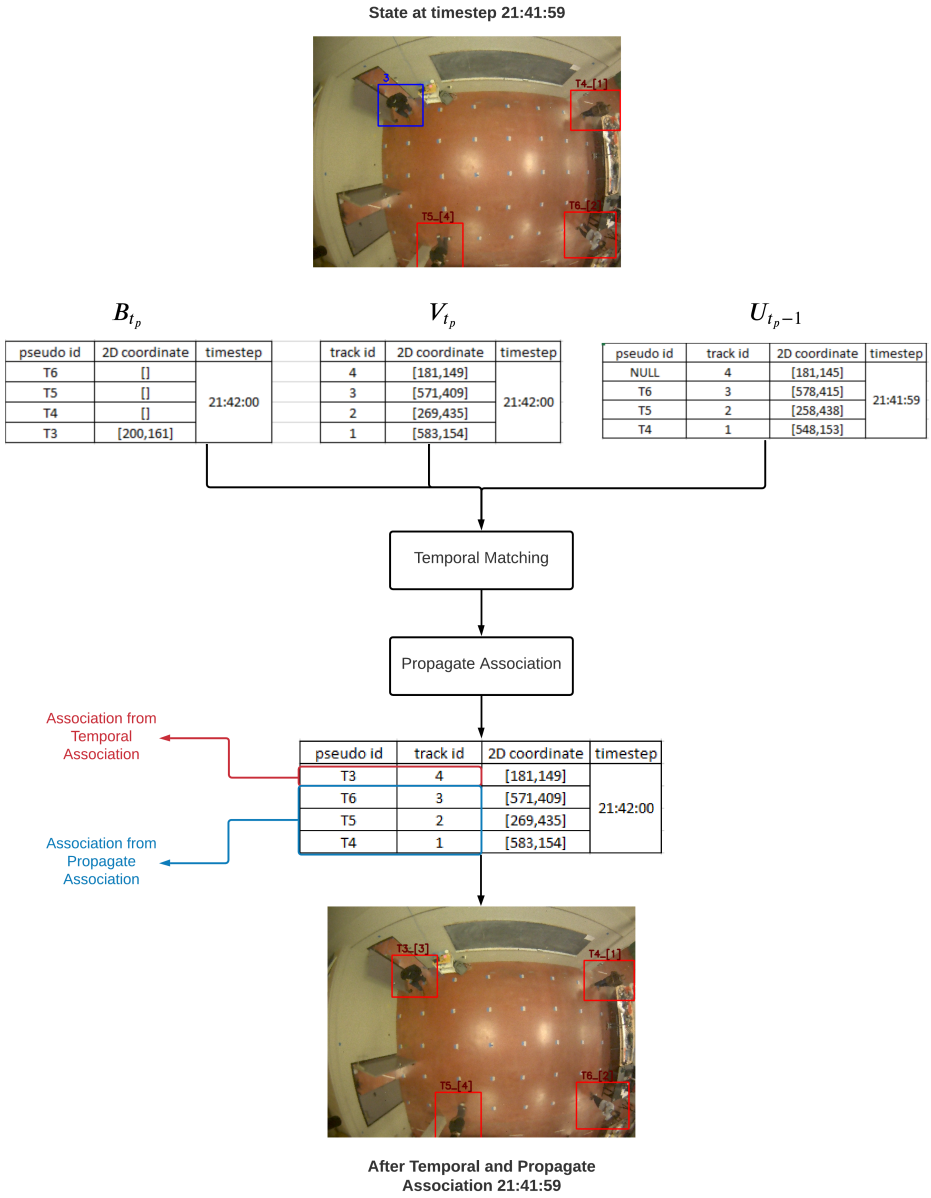
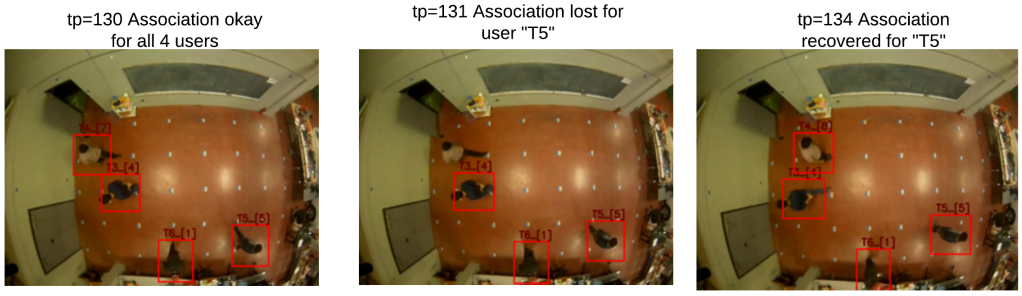


Figure 21. Temporal and Propagate Association

data at the timestep in which the system is trying to recover association. A sample is shown in Figure 22.

The algorithm forms a pseudo matrix similar to B_{t_p} matrix by taking the spatial information from the last timestep for the timestep in which one or more users lost association. In Figure 22, user "T4" user lost association at $t_p = 131$, then the pseudo matrix is formed from U_{130} matrix. At timestep $t_p = 134$, CV-IPS discovers a "new"

object and assigns track id 8 to keep track of its trajectory. But track 8 is actually tracking user "T4" as a new object from CV-IPS perspective. Now utilizing the pseudo matrix and V_{t_p}, MIPS invokes association recovery which using Nearest Neighbor Greedy Search matching matches missing pseudo id "T4" to the new track id "8".



M-IPS output				
tp	track ids from CV-IPS	Known association	lost user(s)	recovered_association
130	[1,4,5,7]	[T6,1],[T5,5],[T4,7],[T3,4]	N/A	N/A
131	[1,4,5]	[T6,1],[T5,5],[T3,4]	T4	N/A
132	[1,4,5]	[T6,1],[T5,5],[T3,4]	T4	N/A
133	[1,4,5]	[T6,1],[T5,5],[T3,4]	T4	N/A
134	[1,4,5,8]	[T6,1],[T5,5],[T4,8],[T3,4]	N/A	**[T4,8]**

Figure 22. Association Recovery

Depending on the distance traversed by the user before he/she is rediscovered by YOLOv3-tiny, this algorithm may be successful in reassigning a new track id to the pseudo id of the user. Experimentally we found that, Association Recovery works correctly when CV-IPS re detects the missing users within 1 meter of the position at which they lost unique association.

Table 4. Parameters/Configurations of the proposed system

Name	***Value***
Number of Users	Variable (2 - 6)
Raw image resolution	5 mega pixels
Downsampled image size	640 by 480 pixels
Image capture rate	21 fps
Object Detection Model	YOLOv3-tiny
MOT model	Simple Online and Realtime Tracking (SORT)
Input to CNN	416 by 416 pixels
Minimum class confidence for Object Detection	0.65
NonMax suppression threshold	0.45
Image processor	GPU
Number of missing frames before tracklet deletion	1
Minimum detection before tracklet assignment	3
Bounding box overlap threshold in SORT	0.35
Number of Beacons	5
Beacon Tx Power	-10 dB
Beacon Range	30 feet
Beacon update rate	10 Hz
ML Model	Random Forest Classifier (RF)
Number of features	14
RF estimators	500
RF minimum sample size	2
RF minimum sample split	2
Communication technology	WLAN/Wi-Fi
Messaging Protocol	MQTT
Number of Smartphones	6
Smartphone OS	Android 7.0 and above
Programming languages	Python, C++, Java
MQTT libraries	Mosquitto, Paho-MQTT
DNN Python library	Pytorch 1.7 and Torchvision 0.8

Chapter 6 Data Collection,Metrics and Experiment Procedure

In this chapter we first present a brief overview of various publicly available Bluetooth and Computer Vision datasets to discuss why none of these datasets were directly usable for training the Random Forest Classifier in BLE-IPS and CNN Object Detector in CV-IPS for the proposed system. We provide a brief description of a new multi modal dataset that was created to specifically to synchronize vision data with RSSI data for evaluating the proposed system. Finally, we conclude this chapter with a discussion on the data acquisition system, methods and processing techniques used within the three major subsystems that comprises the proposed Multimodal Indoor Positioning System.

6.1 Brief overview of existing BLE Datasets

As previously discussed, the nature of collected RSSI data is specific to an indoor positioning application that prevents model trained on RSSI data from one experiment to work correctly in another different indoor environment. Each of these databases were created from ground-up and were build for very different indoor settings. Moreover, the publicly available BLE RSSI databases did not consider multimodal data nor very close proximity positioning for multiple users, the two key requirements for our proposed system. The following is a brief overview for some of the well-known BLE RSSI datasets used in BLE-IPS research.

One of the first well-known publicly available BLE RSSI dataset is the UJIIndoorLoc data first published by Torres-Sospedra et al. [78] in 2014. Consisting of data collected from three adjacent multi floor buildings at the Jamue I University campus, this dataset has 20,000 discrete RSSI samples collected from 933 reference locations spread across three buildings. 20 participants carrying 25 mobile devices collected data from 520 Bluetooth beacons. Thus, each RSSI vector consists of 520 RSSI measures separated by beacon ids but it is to be mentioned that no mobile device at any point on the actual space got access to all 520 beacons. Undetected beacons were given a default value of

+100 dBm. On average each grid cell detected about 27 discrete beacons. The primary purpose of this dataset was building and room level localization which is not enough for our proposed system. Furthermore, no camera data is available with which our proposed MIPS model can be tested.

Another well-known public BLE RSSI dataset is presented in [41] where the primary goal was to provide a freely available BLE RSS database for testing indoor positioning methods across several devices, beacon parameter conditions in a library environment and an office enviornment with a dense deployment of beacons in both testbeds. Android phones were used to collect data with 1 second scanning window. On average 24 Bluetooth beacons configured to iBeacon protocol were placed across various locations in the two testbeds. Average coverage area by the deployed beacon were 163.54 m^2 which is much larger than the coverage area of our testbed. As a result, the RSSI data characterizing the grid cells in [41] is incompatible for our application. Moreover, the maximum number of surveyors were 3 and during data collection they stood still in each grid cells whereas the users only stand still for first 5 second in our case and then randomly walk around on each grid cell for the remainder for the experiment.

Baronti et al. [79] introduced a BLE RSSI dataset in 2018 for the purpose of testing ML models that is poised to test positioning, tracking and social interaction among subjects spread across eight rooms and a connecting corridor space. 277 grid cells spaced 0.6 meters apart where defined for a total coverage area of 237.38 m^2. The authors did not use any commercial beacons but made their own using Raspberry Pi SBCs fitted with two Bluetooth module. One module was configured to be a listener whilst the other acted as a BLE beacon broadcasting an advertisement in 10 Hz. Data were collected for 6 scenarios with three trials per scenario. Users mobile phone recorded data in a format which contained timestamp, transmitter ID, receiver ID and the RSSI value. The complete dataset has 2,820,00 samples. Similar to the issue with other two datasets discussed above, this dataset is also not usable as it does not have any image

data, grid cells were defined for a coverage area much larger than ours and only 3 users at highest participated in collecting data where as we need 6.

6.2 Brief overview of Existing Vision Dataset

The Yolov3-tiny model in CV-IPS shares the same disadvantage as any Deep Learning model, the requirement of a large annotated training dataset. Moreover, the training dataset need to adequately capture the dynamics of people walking on the observation area in order for the model to extract descriptive feature sufficient to reach greater than 90% detection accuracy required by the UAT module. However, majority of vision dataset used with YOLO family of object detectors are created for outdoor pedestrian detection with front-facing camera images. YOLO model pre-trained on a front-facing camera requires additional data from overhead camera images specific to a particular testbed in order to be usable for that particular environment. But only a handful of overhead vision dataset is publicly available majority of which is created with a 360 degree overhead images. We experimented with images from these datasets and found that the YOLO model trained on overhead images available in these datasets, performed very poorly during inference stage since the images did not capture the environment dynamics unique to our testbed.

This conformed to the findings from Blanco et al. [16] who put forward the hypothesis that, the convolution layers in CNN Object Detectors were suitable for images where people appearance were stable and did not undergo large radial distortion. In addition to the problem of 360 view images, none of the publicly available dataset considered multi modal application or dense multi user interaction that is prevalent in our experiment. As a result they do not have RSSI to grid cell mapping nor user-specific information such as a pseudo-id associated with them. While grid cells may be artificially created on the images, RSSI data that would characterize these grid cells cannot be created artificially due to spatio-temporal variance of RSSI coupled with device, beacon

configuration, number of user etc. dependencies that makes RSSI data specific to each unique indoor environments [21]. The following is a brief overview of overhead publicly available datasets used in IPS literature.

Demiröz et al. [80] published one of the earliest overhead vision dataset called BOMNI-DB in which images from Scenario 3 is relevant to overhead IPS application. Annotated images from 36 small video clips containing 5 people performing 5 actions (Sitting, Walking, Standing, Handshaking, Interested in Object) is available with bounding box drawn around each visible person along with a reference tracklet-id. Each video clip is approximately 1.5 minutes long with 750 image frames with 640 x 480 resolution. The lens used is a full 360 degree fish-eye lens. While the dataset contains sufficiently large collection of images, only three people interacted at most at any given point in time. Additionally, the dataset was designed specifically for MOT algorithms and thus did not consider multi modal application.

Another well-known overhead camera database is the PIROPO database introduced by authors in [16]. Consisting of data collected in two rooms, the first room contains three overhead cameras and the second contains one. Similar to BOMNI-DB, the camera lens used is a full 360 degree fish-eye lens. The recording speed was 10Hz and number of participants is 4. The image resolution was 800 x 600 pixels. This dataset contains images in various challenging situation such as variable illumination, multiple person walking at the same time, reflection of individuals on wall and screens and so on. The images in second room provides an additional challenge of very cluttered lab workspace environment. At the time of writing this report, PIROPO is the largest overhead image dataset with 100,000 annotated image frames. However, all annotation are point-based which cannot be directly used in bounding-box regression algorithms such as YOLO. Moreover, the maximum number of people interacting was 4 and this type of interaction was the least amount in the overall dataset.

The HDA+ dataset is another vision based data where data from 13 cameras was

collected for providing a robust dataset for solving people re-identification problem as they move between heterogeneous indoor environments covered by a network of cameras having non-overlapping view [81]. Of the 13 cameras, only one camera has overhead view. For this camera there, 9819 annotated images captured at 640 x 480 pixel resolution at 5Hz with a 140 degree wide-angle lens. A total of 9 individuals were present in the video clips for the overhead camera. Each annotated image contains the location and size of the bounding box, an unique id for a person visible, frame number, name of camera and status of occlusion. While the image size and density of people is suitable for our experiment, this dataset does not have any RSSI information associated with the 9 users since its primary objective was to test algorithm which would keep track of individuals with an initialized id as they move across different rooms.

6.3 Data Collection for Experiment

From the discussion presented in Chapter 5, Bluetooth and Vision Indoor Positioning System have two widely different update rates. The Edge device using Nvidia's "nvarguscamerasrc" proprietary library acquires 21 images per second on average but the open-source Bluetooth library, and Estimote's maximum advertisement rate of 100ms did not allow the smartphones to collect RSSI data from all accessible beacons. Furthermore, the three generation of phones used, each have different computation capabilities which meant that different mobile phones would create RSSI vectors in different time intervals even if they were synchronized to start from the same time moment. This problem is further exacerbated from the observation that during motion, the smartphones did not capture advertisement from all beacons equally in the same time interval. These limitations created the challenge of deciding from which end would the MQTT request be initiated for synchronizing the data streams. We observed that, if the MQTT request were invoked for each and every image captured, the emulated brokers greatly slowed down the Edge device due to the limitation of the Python library used and if the MQTT

request were initiated from the smartphone end than data synchronization took place at widely different time intervals which meant image frames were captured at infrequent rates which lost contextual information necessary for unique association and tracking.

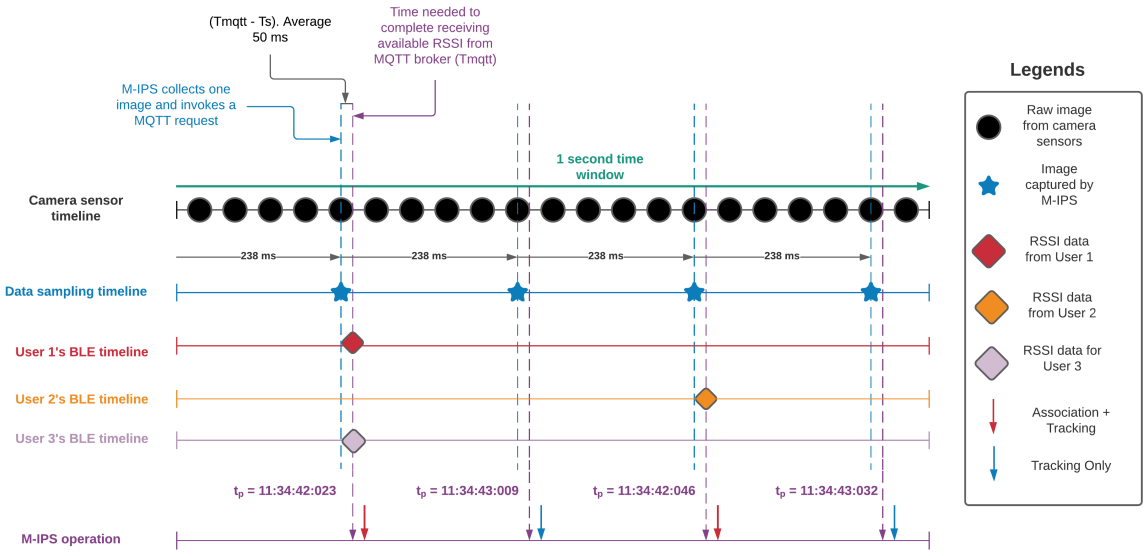


Figure 23. Data Synchronization Timeline

Taking all these practical limitation in mind and also the recommended RSSI data measurement window of 1 second, the data acquisition system was designed to collect 1 raw image every 238ms. Once a raw 2D RGB image was captured, a MQTT request to receive new RSSI data from all the connected smartphones was executed which took 50ms on average to complete.

A single **timestep** (the smallest system time interval) begins as soon as new multimodal data is collected. Figure 23 depicts the system timeline. We observed that, most of the timesteps, new RSSI data was not received from the smartphones. This occurred since updated RSSI data every 1 second whereas new image data was captured 4 times every second. This observation lead us to design the "unique association" (i.e. one-to-one association) as an "opportunistic" system whilst the "tracking" component was designed as a continuous system. Note from Figure 23 that, all system timestep is guaranteed to have an image data. The data for the experiment was collected from 5

different multi user walking scenarios divided into two groups. The first group "Fixed" contains data for Scenarios 1 and 3 in which the number of users in the testbed is fixed. The second group "Variable" contains data from Scenarios 2,4,5 in which the number of users in the testbed varies with time. The "Fixed" group scenario is designed to test how the proposed system behaves in dense user interactions and the "Variable" group is designed to test how the system behaves when the density of users varies over time in order to ascertain whether the proposed system can scale up to dynamic change in number of users and automatically handle initiation and deletion of one-to-one associations during ingress and egress of individuals from the testbed without requiring manual input from the user.

In order to minimize the probability of collecting fewer training samples than necessary for both the ML and CNN models, Scenarios 1 -3 were repeated 4 times and Scenarios 4,5 were repeated 2 times. Each repetition varied between 3.5 minutes to 5 minutes depending on fatigue of the participants and random entry and exit taken by them. All indoor conditions were kept fixed (i.e. overhead lights, arrangement of chairs, tables) and no user were allowed to swap their assigned Android phones. Within each repetitions, minor variations such as switching users entry points, variable time lengths for standing still in starting points etc. were introduced. Additionally, users were instructed to hold their phone in a manner most natural to them.

Each repetition were marked with an alphanumeric sequence consisting of 4 elements in which the first two values represents the scenario and the remaining two represents the repetition instance. For example, repetition 2 in Scenario 1 is designated as "s1a2". In total 8 entry and exist points were set close to grid cells 1, 7, 24, 32, 34, 36, 37, 39 (refer Figure 11) through which users ingressed and egressed out of the testbed.

Each repetition has three distinct data files. Firstly, a CSV file is created which records the timestep in which an image was captured where the images are aliased with a "frame_id" attribute, an integer number that represents the temporal order of a particular

frame in an image sequence. Secondly, raw RSSI samples captured by each user's mobile phone is recorded in a separate CSV file aliased by the pseudo-id assigned to that mobile phone. Finally, a CSV file containing RSSI features and aliased by the same pseudo-id is created. A sample is shown in Figure 24.

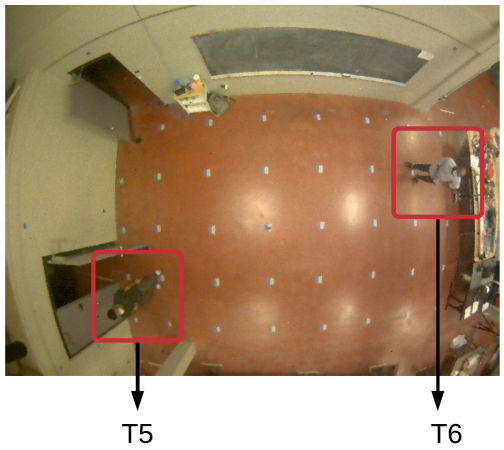
For evaluating the MIPS system, 5 repetitions representing each of the 5 scenarios were separated out while the remaining data constituted the "training" dataset. The training dataset has 9967 annotated images and approximately 3277 feature engineered RSSI vectors with ground-truth cell labels for training whilst the "testing" dataset has 5050 annotated images with 2065 feature engineered RSSI vectors with ground-truth cell labels for testing the CNN Object Detector and ML model respectively. Note that, the MIPS was not directly tested on the "testing" dataset but was tested on individual scenario data after both the ML and CNN model were trained. This is required since MIPS needs to consider the temporal ordering of image and RSSI data. The temporal ordering of image data is true for the CV-IPS but the requirement of temporal ordered data is not strictly necessary for BLE-IPS's ML model since timestep was not a input feature. A brief description of each of the 5 scenarios is presented in Table 5.

Table 5. Summary of Data Collection Scenario

Scenario ID	***Description***
s1a1	6 people enters from 6 locations. Each person halts in their starting position for 2 – 5 seconds. No one leaves
s2b2	5 people enters from one side and the 6th enters from opposite end. The 6th person enters as soon as the first 5 has started to move. After 3 minutes, 3 users leave. After some time, one user returns.
s3a1	4 people enters from 4 different locations. Each person halts in their starting position for 2 – 5 seconds. No one leaves
s4a2	4 people present in 4 corners. They stand on their spot for 3 – 5 seconds. After 3 minutes, 2 new users enters through two different points. After 1 minute, two of the original 4 users leaves the area.
s5a2	2 people present in two corners. They stand on their spot for 3 – 5 seconds. After 1 minute, 4 new user enters the observation area. After 2 minutes, 4 of the 6 users chosen at random, leaves the observation area

Note that, the assumption that user's RSSI data is only valid when he/she is

Image alised with frame_id 15



Timestep to frame_id mapping

frame_id	timestep	timestep_micro
1	21:07:19	21:07:19:238850
2	21:07:19	21:07:19:474796
3	21:07:19	21:07:19:712380
4	21:07:19	21:07:19:950722
5	21:07:20	21:07:20:188615
6	21:07:20	21:07:20:428225
7	21:07:20	21:07:20:664915
8	21:07:20	21:07:20:903233
9	21:07:21	21:07:21:141625
10	21:07:21	21:07:21:386861
11	21:07:21	21:07:21:620611
12	21:07:21	21:07:21:856375
13	21:07:22	21:07:22:092806
14	21:07:22	21:07:22:331721
15	**21:07:22**	**21:07:22:569364**

CSV file containing raw data for user T5

rssi1	rssi2	rssi3	record_time	date	timestep
[-84, -72, -90, -82, -77]	[-80, -71, -86, -86, -77]	[-78, -69, -83, -83, -76]	21:07:18	########	21:07:19
[-84, -88, -72, -82, -77]	[-80, -82, -69, -86, -77]	[-78, -69, -69, -83, -76]	21:07:19	########	21:07:20
[-81, -69, -77, -91, -81]	[-80, -68, -74, -86, -85]	[-81, -70, -80, -89, -81]	21:07:21	########	21:07:22

CSV file containing raw data for user T6

rssi1	rssi2	rssi3	record_time	date	timestep
[-68, -70, -70, -75, -66]	[-72, -78, -72, -75, -65]	[-74, -83, -74, -80, -67]	21:07:18	########	21:07:19
[-90, -73, -79, -71, -73]	[-90, -80, -86, -72, -76]	[-97, -72, -93, -74, -75]	21:07:25	########	21:07:25
[-75, -75, -82, -75, -76]	[-79, -76, -80, -77, -70]	[-77, -78, -77, -71, -73]	21:07:26	########	21:07:27

CSV file containing feature engineered RSSI data for T5

mean_b1	mean_b2	mean_b3	mean_b4	mean_b5	std_b1	std_b2	std_b3	std_b4	std_b5	b1_skew	b2_skew	b3_skew	b4_skew	b5_skew	ble_act	ble_x_act	ble_y_act	timestep
-80.6667	-73.1111	-77.7778	-85.3333	-78.5556	2.179449	7.007932	7.512952	3.162278	3.08671	-0.45251	-1.39453	-0.29237	-0.55622	-1.11991	24	156	359	21:07:22
-79.5556	-72.2222	-76.7778	-87	-82.7778	2.351123	7.429296	6.437736	3	5.717906	-0.38513	-1.44965	-0.31164	0.500867	-0.22019	24	156	359	21:07:23
-79.7778	-69.7778	-79.8889	-89.2222	-83.6667	2.438123	2.587362	4.166667	1.563472	6.103278	0.001807	-1.68247	-0.52292	0.802603	0.454097	25	200	358	21:07:25

CSV file containing feature engineered RSSI data for T6

mean_b1	mean_b2	mean_b3	mean_b4	mean_b5	std_b1	std_b2	std_b3	std_b4	std_b5	b1_skew	b2_skew	b3_skew	b4_skew	b5_skew	ble_act	ble_x_act	ble_y_act	timestep
-78.125	-76.1111	-77.5	-74.4444	-71.2222	8.025629	4.106228	5.345225	2.920236	4.352522	-0.58493	-0.13409	-0.093	-0.49112	0.28482	6	517	159	21:07:27
-81	-73.1111	-77.875	-76.3333	-74.4444	5.830952	4.594683	4.882549	6.652067	2.877113	-0.84583	0.034593	-0.23039	-1.35786	-0.39477	5	465	155	21:07:28
-78.4444	-73.8889	-74.7143	-80.8889	-69.7778	1.878238	5.230785	5.707138	7.304869	7.395569	0.448606	0.044489	0.408992	-0.07255	0.276704	2	262	158	21:07:34

Figure 24. Sample Data

present in the field was taken into consideration for the individual who left in a previous timestep and returned in the current timestep. This was done since there is a time lag for the Android app to register new RSSI data when power cycled. Furthermore, in some cases, users had turned on the app before physically entering the testbed which introduced spurious data. In post-processing these RSSI deleted. Additionally, due to the COVID 19 pandemic, limitations on user mobility, dwelling time and interactions were set by Institutional Review Board (IRB). All 6 participants were fully vaccinated and during movement they maintained a minimum of 3 feet distance fro other participants. No participants were allowed to come in physical contact with one another. Finally, no participants had to use their personal smartphones, install the data collection app in their own phones or undergo complex training session. The scenario were at maximum 5 minutes in length.

6.4 Evaluation Metrics

A number of metrics were defined to evaluate the proposed system and its sub components. Note that, all errors in the continuous space (2D plane) are calculated as Euclidean distance. The error value is computed first as a relative pixel-to-pixel distance which is then multiplied by a factor of 1.14 to convert to relative distance in centimeter. The multiplication factor is unique to the FoV of the camera and the the height at which the camera unit was placed.

For ease of explanation, let timestep be denoted as t_p and the last timestep in the image sequence be N^t. Furthermore, let the number of pseudo ids be denoted as P and the number of users present be denoted as M such that $P = M$. Definitions and mathematical formulations of the 12 chosen metrics are presented below

6.4.1 BLE-IPS object localization accuracy (OLA). This is the ratio of correct cell predictions to total predictions for all users in a scenario. This metric is calculated using Equation 7 after constructing the confusion matrix for the whole scenario. Note that, temporal ordering of RSSI data is not strictly necessary since time is not a

feature used by the ML model and the cell predictions would be same regardless of order in which RSSI data is fed into the system. OLA is denoted as percentage.

$$OLA = \frac{\sum DiagonalElements}{\sum AllElements} \tag{7}$$

6.4.2 BLE-IPS latency. Average time taken to run BLE-IPS subsystem for a scenario. Expressed in milliseconds.

6.4.3 CV-IPS object localization error (OLE). The CV-IPS is the measure of the spatial and temporal overlap between the ground-truth tracks to predicted tracks for all detected objects in a scenario. A track is defined as the collection of 2D positions arranged in temporal order that represents the path taken by an object from a initial time until timestep t_p. This metric quantitatively measures the performance of the MOT model in CV-IPS. Let $N^M_{(t_p)}$ as the number of detected objects that were uniquely paired to a track at timestep t_p. Additionally, let $dist^o_{(t_p)}$ be Euclidean distance between location estimation computed by CV-IPS for the o^{th} mapped object to its ground-truth position. OLE for CV-IPS is calculated using Equation 10 and expressed as a percentage.

$$OLE = \frac{\sum_{t_p=1}^{N^t} \sum_{o=1}^{N^M_{(t_p)}} dist^o_{(t_p)}}{\sum_{t_p=1}^{N^{(t)}} N^M_{(t_p)}} \tag{8}$$

6.4.4 CV-IPS object detection accuracy (ODA). CV-IPS is the accuracy of the CNN Object Detection model in CV-IPS to classify detected objects as the "Person" class in a scenario. This metric is a quantitative measure of the CNN Object Detector.

Let $N^G_{(t_p)}$ be the number of objects present on the testbed at timestep t_p. For a well-trained CNN model, the number of detections would always be equal to $N^G_{(t_p)}$ but in certain cases more or less bounding boxes are predicted. Let the number of extra bounding boxes predicted be defined as **false positives** denoted by $fd_{(t_p)}$ and the number of "missing" detections be defined as **missed detections** denoted by $md_{(t_p)}$. ODA for CV-IPS is then calculated using Equation 9 and is expressed as a percentage.

$$ODA = 1 - \frac{\sum_{t_p=1}^{N^t} md_{(t_p)} + \sum_{t_p=1}^{N^t} fd_{(t_p)}}{\sum_{t_p=1}^{N_t} N_{(t_p)}^G} \tag{9}$$

6.4.5 ID-SWITCH. Number of instances in which CV-IPS reassigned a track ids in a scenario. Let it be denoted as $IDS_{(t_p)}$ for timestep t_p. Expressed as an integer number, ID-SWITCH is higher for a model whose detection accuracy is lower. ID-SWITCH also occurs when SORT looses track of two more objects in short-term, long-term cluttered scenes.

6.4.6 CV-IPS object tracking accuracy (OTA). This is the measure of CV-IPS's ability to **persistently** track all objects in an image sequence. Together with ID-SWITCH, OTA of CV-IPS is a quantitative measure of how well the CV-IPS maintains active tracking of users as generic "Person" object over time. Lower ID-SWITCH and higher OTA allows MIPS to perform unique association less frequently and propagate association more. OTA for CV-IPS is calculated using Equation 10 and expressed as a percentage.

$$OTA = 1 - \frac{\sum_{t_p=1}^{N_t} md_{(t_p)} + \sum_{t_p=1}^{N_t} fd_{(t_p)} + \sum_{t_p=1}^{N_t} IDS_{(t_p)}}{\sum_{t_p=1}^{N_t} N_{(t_p)}^G} \tag{10}$$

6.4.7 CV-IPS latency. Defined as the average time taken by CV-IPS subsystem to detect and track movement of objects of a target class across chronologically ordered images in a scenario. Expressed in milliseconds (ms).

6.4.8 MIPS object localization error (OLE). Let $O_{t_p}^i$ denote the pseudo id generated by the i^{th} user with which MIPS tracks this individual. We define $dist_{t_p}^i$ as the position error between the position estimate obtained after association to the ground-truth location for the $O_{t_p}^i$ user. Let N^i be the last timestep in which the i^th user had a valid association. Note that N^i may be less than or equal to N_t.

Then OLE for MIPS is the average localization error in the continuous space for users uniquely associated with their pseudo ids in the scenario. This metric is calculated

using Equation 11 and expressed as a relative distance in centimeters. MIPS OLE conveys the information of how well the system localized an user after association on the continuous space and can also be calculated at user-level.

$$OLE = \frac{\sum_{i=1}^{P} (\frac{\sum_{t_p=1}^{N^i} dist_{t_p}^i}{N^i})}{P} \tag{11}$$

6.4.9 MIPS object tracking accuracy (OTA). Let $ea_{(t_p)}$ be defined as the **Established Association**, number of pseudo-ids that are newly associated or persistently tracked from a previous timestep to the current timestep t_p. Then let $fa_{(t_p)}$ represent **False Positive Association** and $ma_{(t_p)}$ represent **Missing Association**. Additionally we impose the condition that $ea_{(t_p)} + fa_{(t_p)} + ma_{(t_p)} = P_{(t_p)}$ where $P_{(t_p)}$ is total number of users physically present on the testbed at timestep t_p.

Then OTA for MIPS is defined as the ratio of total correct associations to summation of established, false and missing association for all users in a scenario. MIPS OTA is calculated using Equation 12 and expressed as percentage. OTA for MIPS is a quantitative measures of how well the proposed system can hold unique identification of users over time.

$$OTA = \frac{\sum_{t_p=1}^{N^i} ea_{(t_p)}}{\sum_{t_p=1}^{N^i} ea_{(t_p)} + \sum_{t_p=1}^{N^i} fa_{(t_p)} + \sum_{t_p=1}^{N^i} ma_{(t_p)}} \tag{12}$$

6.4.10 MIPS association latency (AL). Defined as the average time taken to perform one-to-one matching between pseudo ids and track ids for all users in a scenario. Denoted in milliseconds.

6.4.11 MIPS system latency (SL). Defined as the average time taken to execute one complete cycle of the proposed system. System latency is measured in milliseconds and we only consider the time it takes to process the multimodal data once it becomes available but not the latency involved in acquiring the multimodal data.

6.5 Experiment Procedure

The following are the list of assumptions and constraints imposed for this experiment

1. Bluetooth data comes only when people comes into the scenario and opens the app.
2. Waiting 2-5 seconds after initial entry is sufficient to stabilize RSSI data for correct one-to-one associations.
3. Latency to transfer RSSI data over Wi-Fi is negligible.
4. The collected multimodal data is sufficiently large enough to allow the DNN model in CV-IPS and ML model in BLE-IPS to reach ¿90

Prior to experimentation, YOLOv3-tiny model was trained with the 9967 annotated traning image set. According to [52], determining the dimensions of anchor boxes prior to training is essential for achieving high detection accuracy. This was done using K-means clustering from the training data as shown in Figure 25. Apart from achor box determination, other optimization techniques such as multi-scale training, image transformation, larger batch size were also employed.

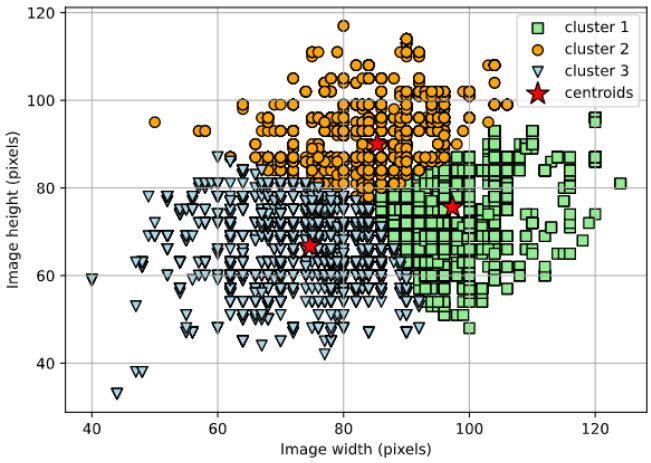


Figure 25. Determining anchor box dimensions

While training the YOLOv3-tiny was relatively straightforward, training Random Forest classifier with the 3277 proved very challenging owing to the fact that, not all cells were equally visited by the users Figure 26.

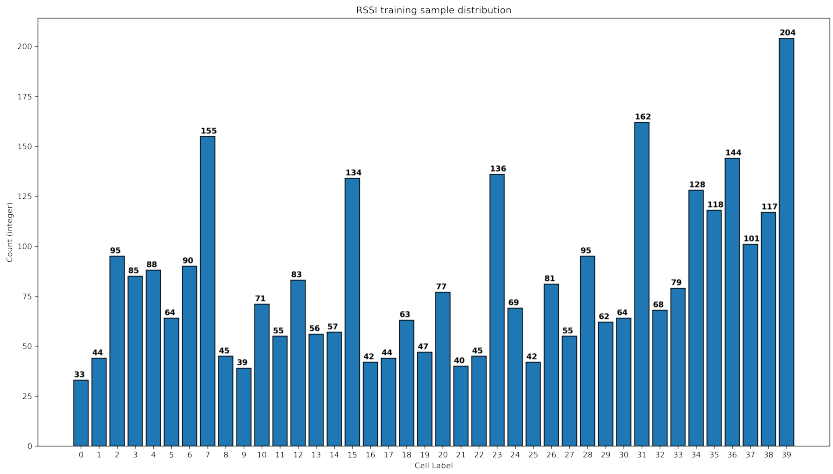


Figure 26. RSSI sample distribution based on cell location

A large portion of the samples were biased towards cell grids (i.e 7,39) which indicated a walking bias by the users not foreseen during data collection. As a result, with this training dataset, it was not possible train Random Forest classifier to use all 40 cell grids. Through trial and error, a 4-cell grid configuration was found which showed $\geq$ 90% detection accuracy for all 5 scenarios. The final grid configuration is shown in Figure 27.

With the two models trained we proceeded to test MIPS for all 5 scenarios using the procedure shown in Figure 29. For each timestep, the number of correct, missed and false association along with latencies for each of the three subsystems is recorded in one csv file as shown in Figure 28. Here, "ble_tp", "cv_tp", "al_tp" and "sl_tp" denotes the latencies for BLE-IPS, CV-IPS, UAT and System Latency respectively.

The anonymous user id is used to create one CSV file for each user that stores the 2D coordinate of their movement based on the association established to their respective

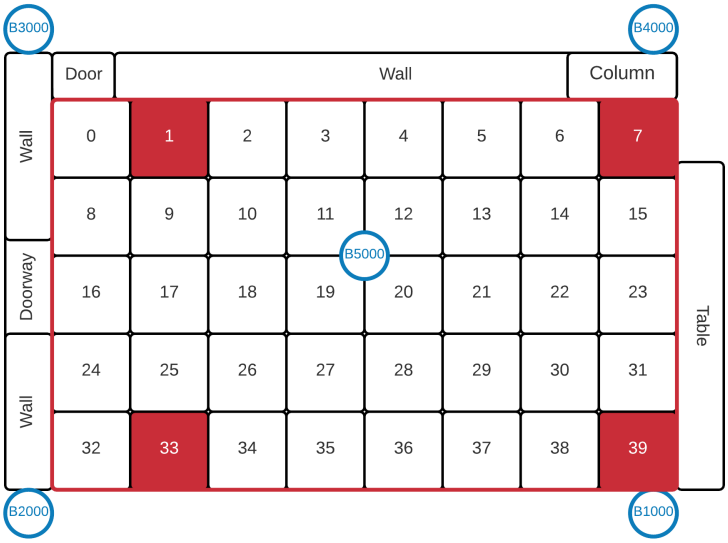


Figure 27. 4-cell grid configuration used in experiment

(a) CSV file to save associations and latencies

frame_id	timestep	ea_tp	ma_tp	fa_tp	ble_tp	gpu_t	sort_t	cv_tp	al_tp	rl_tp	pl_tp	sl_tp	id_switch
33	23:10:16	4	2	0	30.594	20.688	11.914	32.602	0	0	0.03	63.226	0
34	23:10:16	4	2	0	30.32	21.68	12.13	33.81	0	0	0.031	64.161	0
35	23:10:17	4	2	0	164.385	22.512	12.419	34.931	1.366	0	0	200.682	0
36	23:10:17	4	2	0	30.284	22.944	12.064	35.008	0	0	6.10E-05	65.29206	0

(b) Example CSV file showing 2D coordinate of an user recorded from experiment

X_mips	Y_mips	frame_id	timestep
199	367	39	23:10:18
206	367	40	23:10:18
215	368	41	23:10:18
231	369	42	23:10:18
247	371	43	23:10:19

Figure 28. Example of output CSV files

id in each timestep (Figure 29 (b)). In addition to these CSV files, the input image is processed to visualize the UAT output and saved for each timestep. Note that, MIPS does not have sufficient information to check the correctness of the established association during run time and, as such cannot fill in the "fa_tp" column. The output images are used to manually correct the associations in Figure experiment˙csv(a). Once these corrections are made, all metrics are computed and the experiment ends.

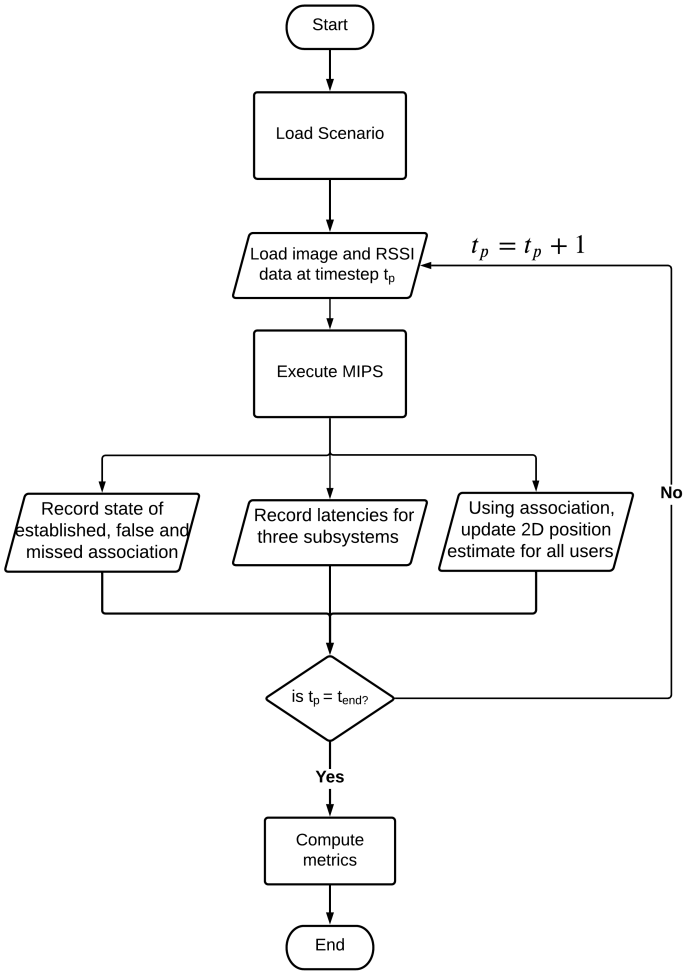


Figure 29. Experimental Procedure

Table 6. Results: OTA for CV-IPS and MIPS

Scenario ID	***CV-IPS (%)***	***MIPS (%)***
s1a1	97.9	93.9
s2b2	97.4	76.9
s3a1	97.9	95.0
s4a2	98.0	95.1
s5a2	98.1	79.6

Table 7. Results: OLE for CV-IPS and MIPS

Scenario ID	***CV-IPS (cm)***	***MIPS (cm)***
s1a1	30.1	48.3
s2b2	28.9	167.3
s3a1	26.5	33.7
s4a2	27.1	43.6
s5a2	26.6	150.8

Tables 6 and 7 shows the OTA and OLE results for CV-IPS and MIPS respectively. BLE-IPS OLA and OLE are not shown here since BLE-IPS does not track in continuous space and all 40 cells were not usable. One can observe that OLE and OTA have an inverse relationship with linear distribution for the CV-IPS it‘s OLE reaching 23.2 cm on average for the entire experiment. This corroborates the capability of a vision based tracking system to localize moving objects with high resolution provided it‘s tracking accuracy is greater than 90%. The high OTA also supports our assumption that a CNN model works efficiently with an 160 degree overhead image in localizing multiple users since the SORT algorithm is dependent upon how well YOLOv3-tiny differentiates people from other object in the image sequence. The average ODA for the entire experiment is 93.2%.

In contrast to CV-IPS, MIPS OLE vs OTA relationship shows an inverse relationship with high degree of non linearity as depicted in Figure 30. Two important observations can made namely MIPS performance is scenario-specific and OTA for MIPS

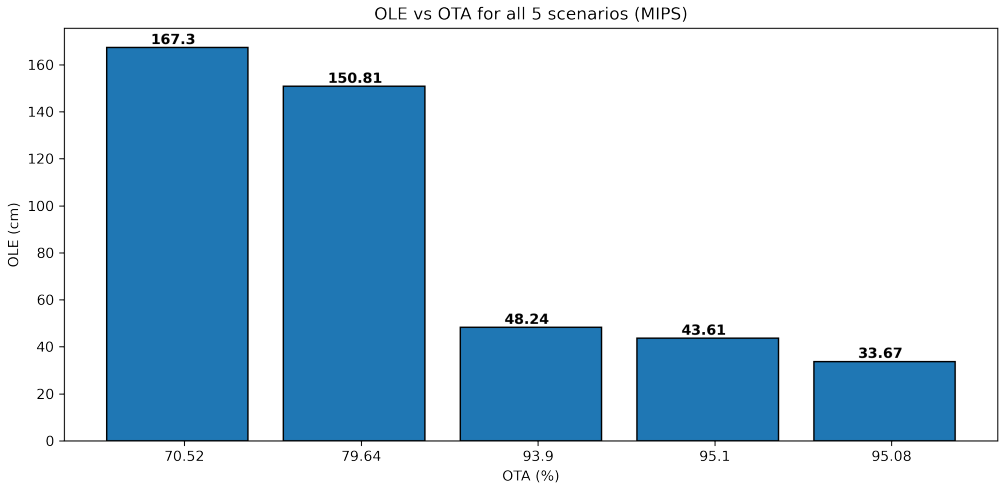


Figure 30. MIPS OLE vs OTA

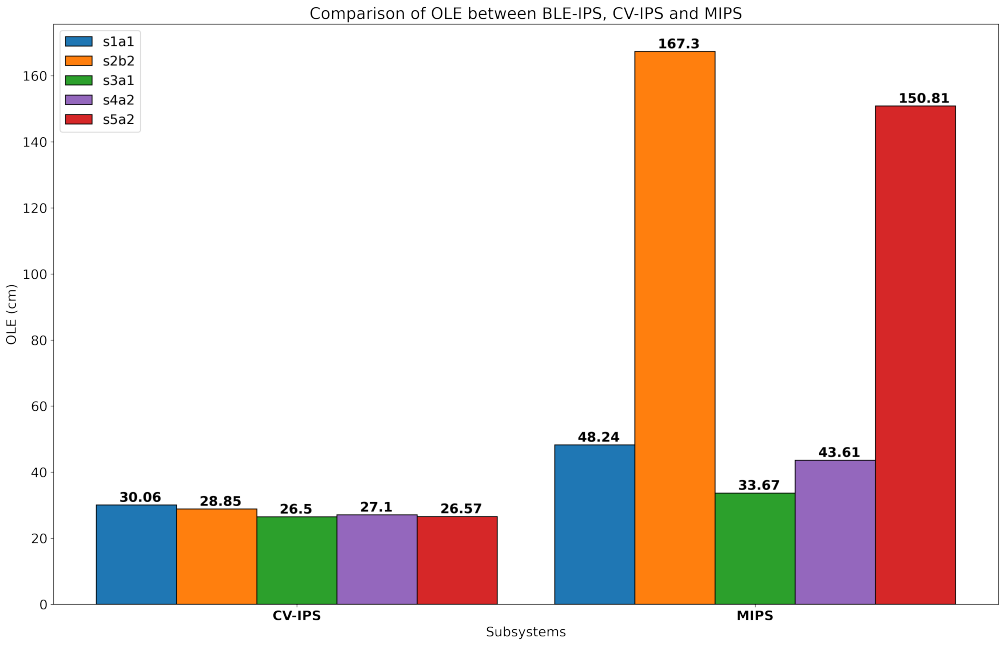


Figure 31. Comparison of OLE between CV-IPS and MIPS

needs to be greater than 90% to reach a localization resolution comparable to CV-IPS. We also observed that beyond 93% OTA, OLE did not improve significantly.

Figure 31 shows comparison of OLE between CV-IPS and MIPS for all 5 scenarios. From the figure, we observe that MIPS performed better in the "fixed" scenario group than the "variable" group. Additionally, the scenarios where MIPS OTA fell below 80%,

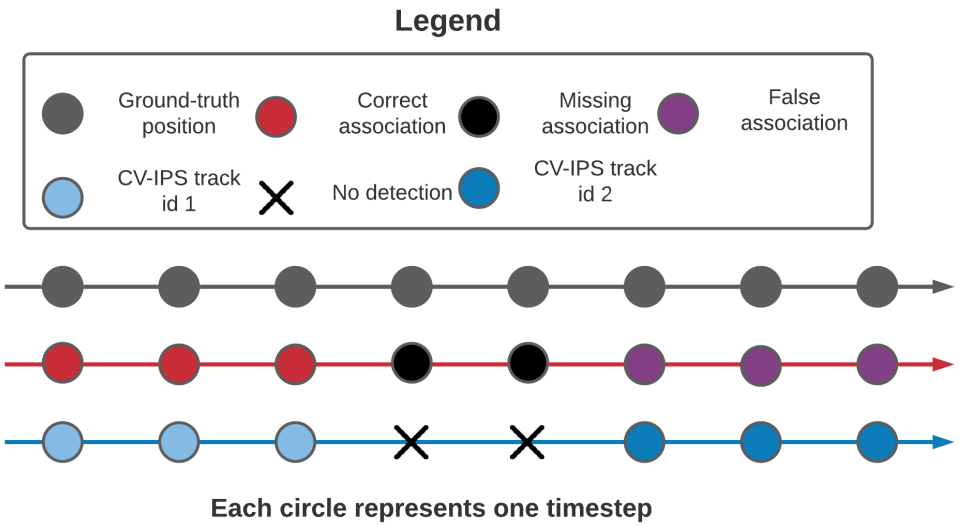


Figure 32. MIPS and CV-IPS track for one individual

OLE error rose above 150cm. The cause of wide variation in OTA and OLE for MIPS can be explained by one crucial fact, that the performance of MIPS is entirely dependent upon **longevity of correct one-to-one associations at user-level**.

As shown in Figure 32, for an individual‘s MIPS track, there are two sources of association errors namely number of frames with false associations and number of frames with missing associations. OLE error is highest when no association is established and varies in degree for false association. The figure clearly demonstrates that MIPS OLE is **additive in nature**. This is caused by the fact that MIPS cannot discontinue a track when its association is lost since the pseudo id used to represent an user is alive until end of session. In contrast, CV-IPS can discontinue and start a new track for the same user after an event of ID-SWITCH (Figure 32 light blue and deep blue circles). Once reestablished, CV-IPS counts the associations from that point in time as correct which is not possible in MIPS. After a loss of association, MIPS reestablishes tracking once a One-to-One Matching is successfully performed. Furthermore, by virtue of logical association, an association whether right or wrong is carried forward in time. Thus,

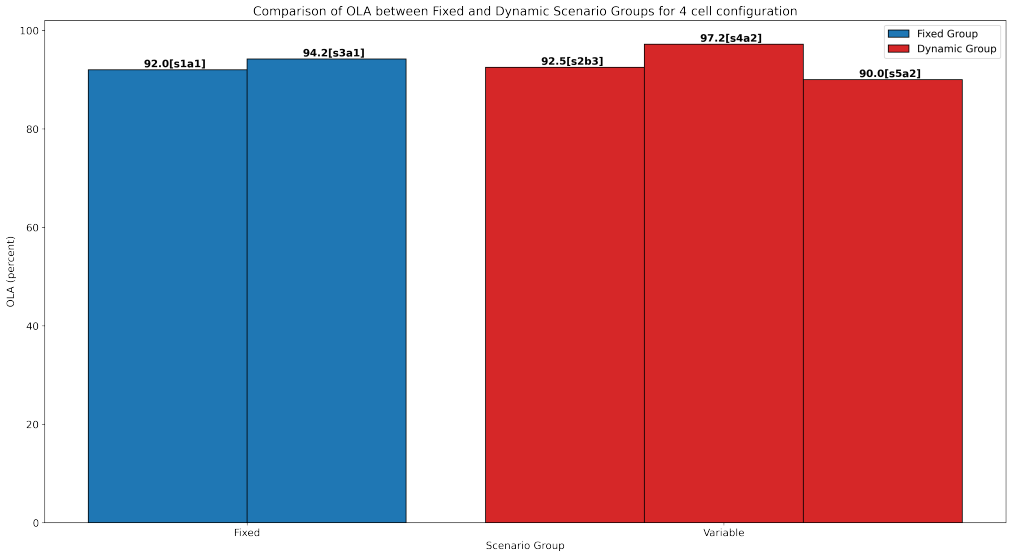


Figure 33. OLA for BLE-IPS in 5 scenarios

whether missing or false association occurs, OLE at individual level accrues over time. This observation underscores two requirements for MIPS to perform well viz. correctly associating an users as soon as they enter and performing correct One-to-One matching as soon as valid multimodal data is available for one or more user(s) missing association. Figure 32 also brings to light two relationships between OTA and number of users in MIPS. The severity of OTA error directly dependent upon how many **users had correct association** at a particular timestep. If the number users is lower, then penalty for wrong or missing association is lot higher in comparison to scenario where number of users are more. Secondly, in scenarios with even number of people, false positive errors will occurs between a pair of users.

From the discussion presented in Section 5.6, MIPS can perform unique associations but cannot verify the authenticity of association in run time. This limitation occurs due to the fact that UAT has only access to the numeric outputs from BLE-IPS and CV-IPS, not to the multimodal data itself. Furthermore, the BLE-IPS can only predict the 4 corner cells with average 93% accuracy for the experiment as shown in Figure 33. Thus, the BLE-IPS will always predict 1 out of the four chosen cells, regardless whether a

person was physically present on any of the four corner cells.

The lack of verifying one-to-one associations and limited cell detections from BLE-IPS gave rise to two major limitations of UAT module. First, the Greedy matching is correct if and only if the spatial predictions from BLE-IPS and CV-IPS is true aprior to matching which has the highest possibility if the user is positioned in one of the four corner cells. Secondly, False association, once done is carried forward in time using the "Propagate Association" as long as the the erroneous pairing survives.

These observations implies that MIPS has higher probability of performing well in those scenarios where users entered through the four corner cells only. This is clearly observed in Figure 31 where the top performing scenarios (s1a1, s3a1, s4a2) had all had users entered through the 4 corner cells only.

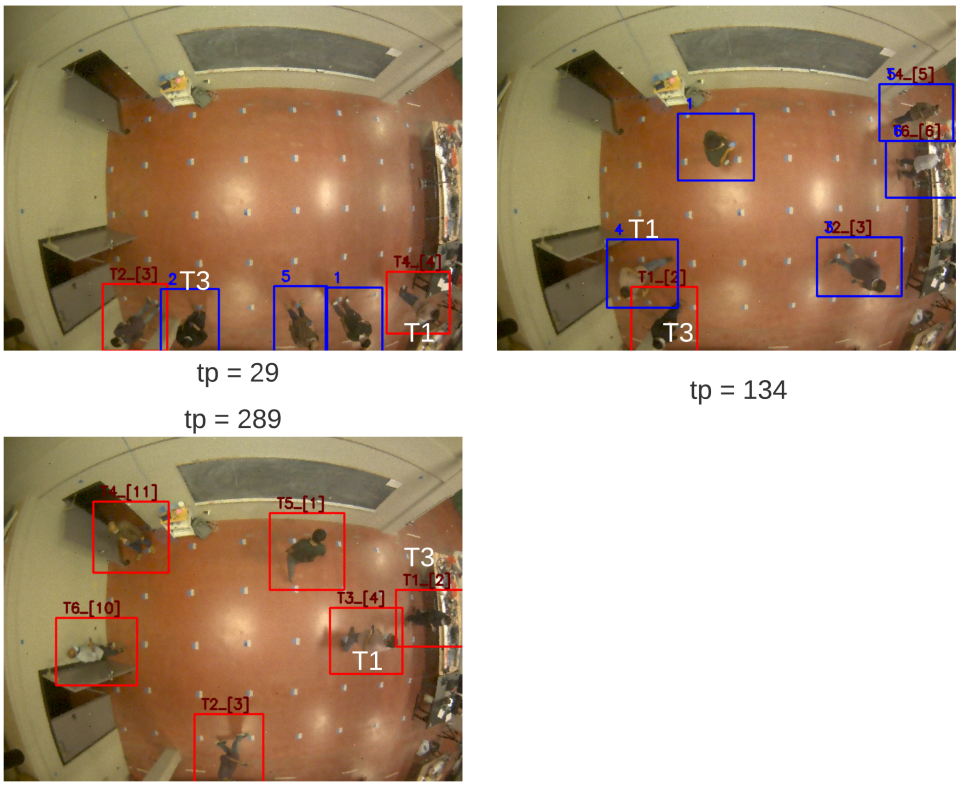


Figure 34. Example frames from Scenario 2

Scenarios where users entered through one side of the map (s2b2 and s5a2), MIPS

could not establish association and, made false association very early into the scenario that in turn would raise OLE error for affected users significantly. Consider two users "T3" and "T1" from Scenario 2 as shown in Figure 34. Pseudo id assigned to these users are marked in white. At timestep $t_p = 29$ the 2 users did not get correct association. As time progressed to $t_p = 134$ user "T3" was falsely associated with id "T1" since user "T3" was most closest to the cell 33 and both user had valid RSSI data at this time. User "T3" wrong association was not broken until very late into the scenario. Note that,even though actual "T1" user does not have any association, due to propagation, the path taken by actual "T3" would be counted towards user "T1" and, user "T3" would be incurring OLE error for missing association. At timestep $t_p = 289$ via Temporal Matching actual user "T1" is assigned id "T3" since there were one unmatched pseudo id and one unmatched track id. Thus, Temporal matching can occur anytime and can make mistake due to its simple logical association rule.

Table 8. Results: User level OLE for Scenarios 2 and 4, all values in cm

Pseudo id	***OLE (Scenario 2)***	***OLE (Scenario 4)***
T6	58.8	35.2
T5	131.5	37.9
T4	130.9	13.9
T3	408.8	60.8
T2	22.5	42.1
T1	250.9	71.7

Table 8 shows the OLE error for all 6 users in Scenario 2 and compares it with Scenario 4. As expected users "T3" and "T1" have very large errors in Scenario 2 due to them having incorrect association through most of the scenario. In comparison, scenarios where users were correctly associated from very early into the scenario, their OLE error can reach sub 20cm OLE, even lower than CV-IPS.

Apart from the longevity of correct association, MIPS performance is also impacted by the threshold parameters in Equation 6. The two threshold parameters d_{thres} and θ_{thres} can be tuned to either extend or constrict the region close to 4 cell grids that

would be considered for evaluating association.

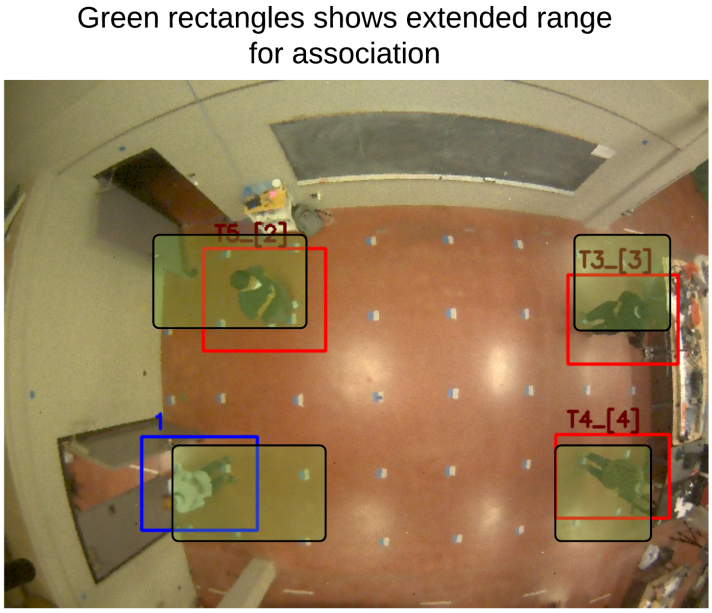


Figure 35. Effect of threshold parameters

In experiments we observed that similarity measures d_{ij} and θ_{ij} assumed values close to 60 and 0.005 respectively when an user requiring association was standing $\leq 10cm$ from one of the 4 corner cells. However, we noticed that, setting parameters d_{thres} and θ_{thres} to 60 and 0.005 respectively caused missing associations in the initial minute of Scenarios 4 and 5. In these scenarios, users after entering the testbed stood at cells adjacent to the 4 corner cells. Hence, for Scenarios 4 and 5, parameters d_{thres} and θ_{thres} were set to values 100 and 0.1 respectively. This extended the effective area of association to a 1 meter circle from the corner grid cells. However, after 1 minute, the parameters were dialed down to smaller values to minimize false associations. Figure 35 demonstrates this adjustment for the first four seconds in Scenario 4 (s4a2).

Table 9 shows the latencies of all three subsystems for the experiment and Figure 36 shows the system latency graph for the first 100 timestep of Scenario 1.

As evident from the table, running the YOLOv3-tiny model in the Jetson NX‘s

Table 9. Results: Latency, all values in milliseconds

Scenario id	*BLE-IPS*	*CV-IPS*	*MIPS (AL)*	*MIPS (SL)*
s1a1	58.0	31.8	0.02	91.6
s2a2	54.4	29.7	0.18	83.9
s3a1	48.0	28.9	0.009	76.9
s4a2	53.9	28.7	0.0085	83.6
s5a2	52.8	29.0	0.12	83.7

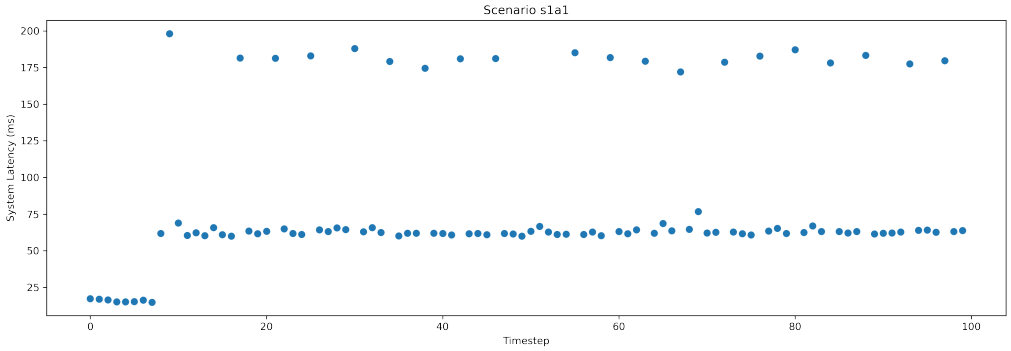


Figure 36. System Latency

power efficient GPU and choosing SORT as the MOT model enabled CV-IPS to localize multiple objects at near real-time speed (i.e. 33Hz). However, as shown in Figure 36 there are certain timesteps in which system latencies were upwards of 150ms. These are the epochs in which new One-to-One association were made. The majority of this delay came from the Random Forest Classifier whose implementation is currently CPU exclusive and computationally expensive. Thus, the proposed system opts to propagates last known associations in absence of valid Bluetooth data. This design choice has contributed to the overall low system latency but at expense of verifying correctness of association in every timestep.

Figure 37 shows the break down of average latencies of the heuristic algorithms present in UAT subsystem. It is clearly evident that, all three algorithms possess real-time performance with "Propagate Association" (and by logical reasoning Temporal Association) being the fastest.

Figure 38 shows the effect of user density to system latency for the proposed

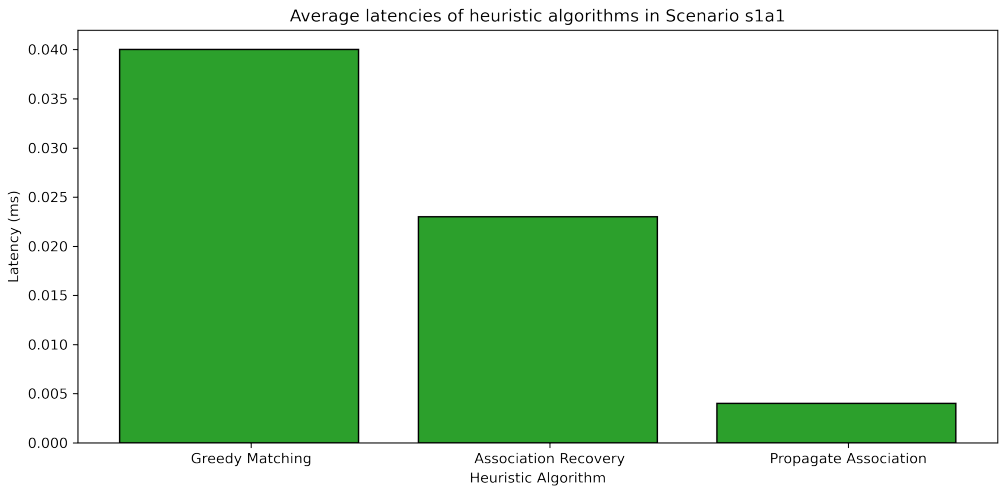


Figure 37. Latency comparison of Heuristic Algorithms in UAT

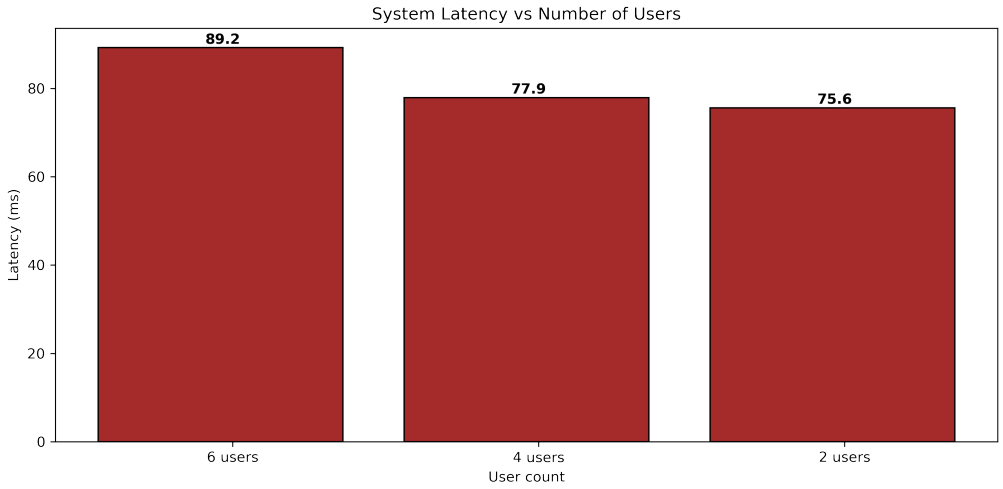


Figure 38. System Latency vs User Density

system. In the collected data, there were no standalone two user scenarios. Hence, the first 250 timesteps of Scenarios 1, 3, and 5 representing 6,4 and 2 user scenario respectively were selected to analyze this relationship. From the figure, between 6 vs 4 user scenario, system latency showed the expected inverse relationship. However, between 4 vs 2 user scenario, the result is inconclusive since the difference in time is very small. Figure 39 shows the effect of recovering lost association using the Association Recovery state. In Section 5.6, we stated the assumption that, if CV-IPS can re-detect an individual for whom it lost detection within 1 meter of their last known location, Association Recovery

can successful re-establish one-to-one association for that individual. We observed this hypothesis held true only for a number of cases. As a result, Association Recovery affected each of the five scenarios differently.

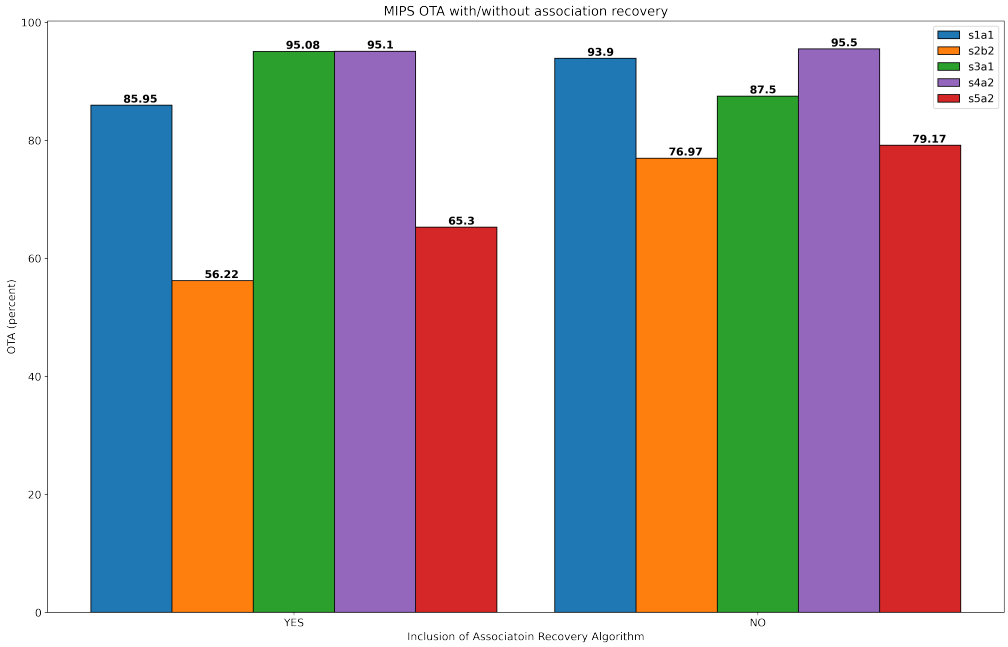


Figure 39. MIPS OTA with/without Association Recovery

To illustrate, Figure 40 shows an example where recovering association had positive impact in Scenario 3. In this case, users "T6" and "T4" lost association in timestep $t_p = 811$. At timestep $t_p = 815$ with no Association Recovery, MIPS attempts to perform One-to-One greedy matching but failed due to BLE-IPS predicting cell 33 for both the user. Due to the dynamic motion, a valid association did not occur until timestep $t_p = 962$. In contrast, with Association Recovery, both users recovered their association in timestep $t_p = 815$. As a result, MIPS held onto their Association for rest of the scenario which significantly lowered OLE for the two users. When no Association Recovery was used, OLE for users T6 and T4 were 159.9cm and 136.4cm respectively whilst with Association Recovery their OLE came down to 55.1cm and 49.8cm respectively. Thus, recovering association as quickly as possible demonstrated noticeable performance impact both at the system and the user-level.

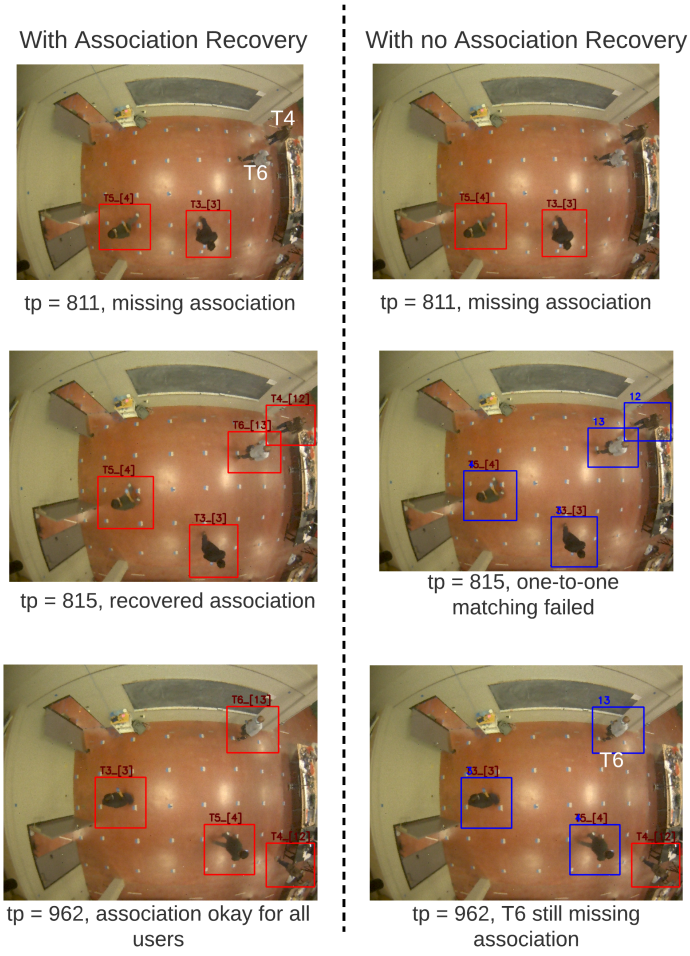


Figure 40. Example of Association Recovery

However, in Scenario 1, 3, 5 Association Recovery had detrimental effect since the algorithm reestablished erroneous association from previous timestep. This caused by the fact that, the Greedy Matching heuristic used to perform association cannot verify correctness of association just like Nearest Neighbor Greedy Search matching. As a result when Association Recovery is wrong, False Positive association continues over time which increases OLE and reduces OTA as depicted in Figure 32. In our experiment, Association Recovery had an overall negative effect on performance and marginal improved System Latency.

Table 10. Results: MIPS Performance using YOLOv3-tinier

Scenario ID	***OTA (%)***	***OLE (cm)***	***AL (ms)***	***SL (ms)***
s1a1	78.9	129.2	0.2	100.9
s2b2	75.2	137.5	0.097	94.4
s3a1	84.5	58.3	0.01	88.3
s4a2	94.8	33.1	0.015	95.3
s5a2	79.7	139.2	0.1	92.5

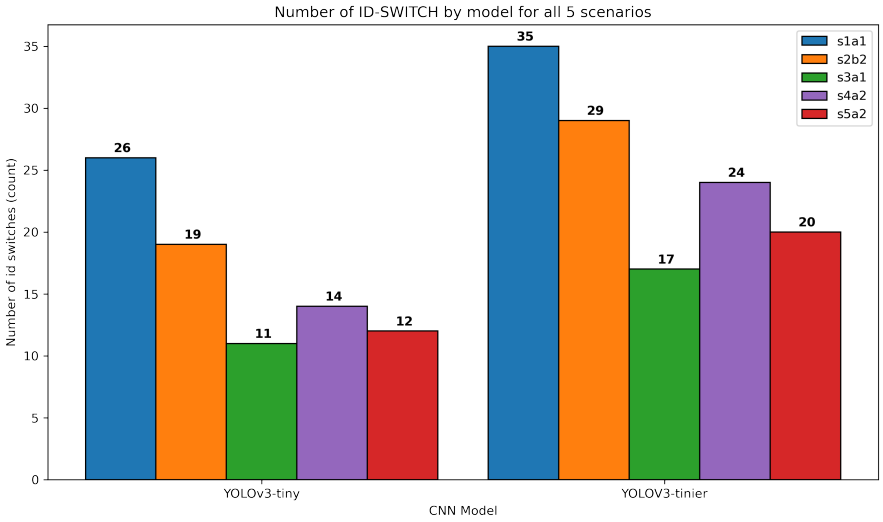


Figure 41. ID-SWITCH vs CNN Model

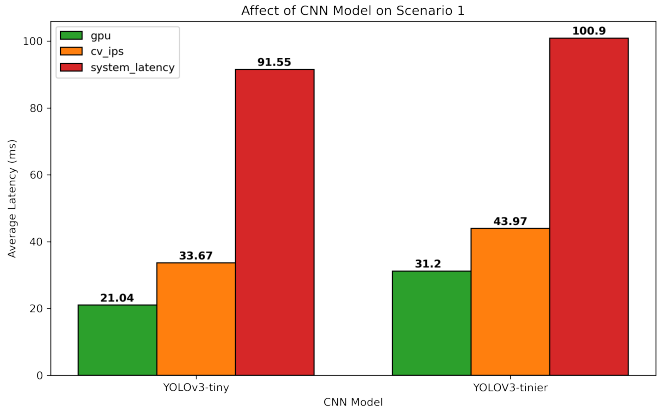


Figure 42. Latency vs CNN model for Scenario 1

The choice of CNN model in CV-IPS also showed a significant effect on the performance of MIPS. For ease of explanation, Tinier-YOLO introduced in Section 5.4 will be referred to as **YOLOv3-tinier** from hereon. Table 10 shows MIPS performance using

YOLOv3-tinier for the experiment. This model was trained with the same dataset as that of YOLOV3-tiny and all training hyperparameters were kept same. During experiment, inclusion of Association Recovery and threshold parameters association was also kept same as that of YOLOv3-tiny on scenario basis. We observed that OTA for some scenario using YOLOv3-tinier was better than YOLOV3-tiny but on average OTA it is less. This may be attributed to the higher number of ID-SWITCH shown in Figure 41 which was higher for all 5 scenarios in comparison to YOLOv3-tiny. Though YOLOv3-tinier had more layers, it's parameter count is significantly less than YOLOV3-tiny which also reduced the model's capacity to create descriptive features necessary for detecting people. In addition to OTA, choice of CNN model plays a crucial role in CV-IPS latency. From Table 10, we observe that YOLOv3-tinier is on average 10ms slower than YOLOV3-tinier. This is primarily caused by the additional time it took the GPU to process the 96 layers of YOLOV3-tinier which in comparison is 2.6 times deeper than YOLOV3-tiny. Figure 42 illustrates this observation for Scenario 1 but the same was true for the other 4 scenarios. Our findings thus far gives support to the need for judiciously choosing CNN model with strong consideration to the IoT platform in which the MIPS is deployed.

We conclude this chapter by listing the key limitations of the proposed system

1. Proposed MIPS cannot assess correctness of association during run time. As a result, False Association once done keeps on propagating forward in time.

2. Failure to establish association early on causes prolonged missing association which greatly increases OLE.

3. With only 4 cells, MIPS can only establish association at certain regions of the map using One-to-One Matching.

4. The Euclidean and Cosine similarity thresholds requires dynamic assignment and is subject to scenario-by-scenario optimization.

5. During data collection, smartphones did not update RSSI data every second. This was caused by limitation of Edge device in emulating MQTT broker along with limitation of application of Android app. As a result significant temporal gap appeared between congruent RSSI vectors in many cases which caused many false associations. Furthermore, assumption that RSSI data stabilizes 2-5 seconds after entering the field is no guaranteed to be sufficient.

6. MIPS inherits all the drawbacks of BLE-IPS and CV-IPS. For instance, MIPS needs recollect RSSI data and train the deterministic model for every new indoor environment it is deployed to.

Chapter 8 Conclusions and Future Work

8.1 Conclusions

In this thesis, we introduce a novel Multimodal Indoor Positioning System (MIPS) for anonymized multi-user identification and tracking of in indoor environment.We track the users device (i.e. smartphone) using an anonymous identifier. The system has two independent indoor positioning systems namely Bluetooth Low Indoor Positioning System (BLE-IPS) and Computer-Vision Indoor Positioning System (CV-IPS). The ingested multimodal data (RSSI measurements + 2D RGB images) provide object identifiers, spatio-temporal trajectories of users from both the IPS units, and finally these trajectories are matches against each other. The system is evaluated on four evaluation metrics namely Object Tracking Accuracy (OTA), Object Localization Error (OLE), Association Latency (AL) and System Latency (SL).

To the best of our knowledge, the proposed MIPS is the first of its kind to use multimodal data (BLE + Image) to anonymously track multiple users. Our experiments to track six subjects reveal OTA $\geq$ 90% and OLE $\leq$ 50cm for various scenarios. The average system latency was 70ms (14Hz) and the average latency to perform association was 0.05ms (0.17MHz). These system performance demonstrates potential MIPS with inexpensive Single-Board-Computers (SBC) for scaling up to large-scale indoor applications.

The UAT subsystem in the proposed MIPS only processed numeric data thereby making the system independent of the multimodal data itself. This feature allows different RF and Vision based positioning systems to be used without modification to architecture. The novel multimodal dataset provided data for 5 multi-user scenarios to evaluate performance of BLE-IPS, CV-IPS and the MIPS respectively. Additionally, during development of BLE-IPS, a novel RSSI feature **rolling skewness** was added that improve the IPS performance.

We observe that the implementation of proposed MIPS is highly biased towards

certain scenarios for two reasons. Firstly, due to uneven collection of RSSI samples per reference cells, only 4 cells were usable by BLE-IPS. Secondly due to the dependency on the longevity of correct one-to-one associations, MIPS required to detect, identify and track new users as soon as they entered the testbed. As a result MIPS OTA and OLE showed high variance across scenarios along with OLE varying in between the users in a scenario.

Our results and analysis indicated that the proposed MIPS has strong potential for real-world tracking of people with anonymization in an indoor environment provided its tracking accuracy exceeds 90% and users are **uniquely and correctly associated** as soon as they enter the workspace.

8.2 Future Work

There are a number of potential research directions one can take from this research. Some potential research ideas are listed below:

1. Current design of proposed MIPS is unable to determine the correctness of association with 100% accuracy during run time due to the nature of the multimodal data and how Nearest Neighbor Greedy Search pairs pseudo ids to track ids. Using GPS-based localization and tracking in indoor environments conducive to GPS signals or using a known tracking system with high accuracy such as Active-Bat [82], RADAR [83], a third set of identifier and spatial information may be obtained for the users walking on the testbed. This information would then act as ground-truth to assess correctness of association and tracked path during run time. Such a study would not only yield a viable solution to this limitation but also shed light to the privacy vs latency, tracking accuracy, cost, trade-offs for MIPS constituting three or more independent positioning system.

2. Another promising direction this research can take is solving the lack of uniform RSSI data collected per cell that severely impacted the performance of MIPS by

diminishing the capabilities of BLE-IPS. Using small robots retrofitted with smartphones is a viable approach to solving this problem as robots can alleviate the need for people to continuously walk over each grid cells at uniform frequency. Collecting RSSI samples with robots may lead to studies involving optimization and path-planning using the proposed MIPS.

3. One of the major drawbacks of MIPS is the dependence of creating new Radio Maps for each new indoor environment. This problem may be solved by simulating multi user walking condition for different environment. This is a viable approach since in IPS literature a large number of studies showed promising results in creating synthetic RSSI data by considering the propagation model of radio signals in different indoor environments before comparing results from real-world experiments. Such a stimulative system will greatly benefit the scalability of the proposed system by reducing the dependence of creating real-world data collection scenarios for each new environment.

4. In this study, RSSI measurements and Image data were first processed by two independent positioning systems whose outputs were then processed by a third heuristic model to localize and track users anonymously. While not part of this research scope, it is possible to fuse RSSI measurements from wireless nodes directly with image data by fusing features common features from the two heterogeneous data streams as demonstrated in [84]. This may be advantageous since a fused data could then be processed by one Machine Learning or Deep Learning model thereby reducing system complexity. Furthermore, the fused data may be able to reduce the spatio-temporal variance of RSSI data which may improve performance of the BLE-IPS significantly.

5. In this study we showcased the independence of the proposed MIPS from nature of the multimodal data itself. A followup study is needed to further verify this

observation using different data modalities such as RFID, UWB, Stereo Image, Thermal Images and so on.

6. It is essential to test the Scalability of the proposed architecture in a follow-up study for gauging its real-world applicability. In context of this MIPS system, Scalability may refer to concepts such as the number of overhead camera units needed to identify and localize users in a large workspace, number of smartphones that can simultaneously communicate with one Edge device, transferring unique associations between overlapping camera regions, scaling to multi room, multi building testbeds and so on.

7. Finding the cells in the grid that have good RSSI feature based discrimination was done manually, which is a time consuming and laborious process. This can be potentially automated with unsupervised machine learning methods such as clustering to potentially improve the performance of BLE-IPS.

8. In this study, we did not consider threat models for compromising anonymization. A future study can utilize Cybersecurity concepts based on multi-agent systems, blockchain and deep learning **??** to investigate how data security of users may be breached and what preventive measures can be taken to prevent loss of anonymization in the proposed MIPS.

Bibliography

[1] Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., Behar, J. V., Hern, S. C., and Engelmann, W. H., "The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants," *Journal of Exposure Science & Environmental Epidemiology*, vol. 11, no. 3, pp. 231–252, 2001.

[2] Murata, M., Ahmetovic, D., Sato, D., Takagi, H., Kitani, K. M., and Asakawa, C., "Smartphone-based indoor localization for blind navigation across building complexes," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, IEEE, 2018.

[3] Stavrou, V., Bardaki, C., Papakyriakopoulos, D., and Pramatari, K., "An ensemble filter for indoor positioning in a retail store using bluetooth low energy beacons," *Sensors*, vol. 19, no. 20, p. 4550, 2019.

[4] Calderoni, L., Ferrara, M., Franco, A., and Maio, D., "Indoor localization in a hospital environment using random forest classifiers," *Expert Systems with Applications*, vol. 42, no. 1, pp. 125–134, 2015.

[5] Ahmed, E., Islam, A., Sarker, F., Huda, M. N., and Abdullah-Al-Mamun, K., "A road to independent living with smart homes for people with disabilities," in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 472–477, IEEE, 2016.

[6] Castelluccia, C., Bielova, N., Boutet, A., Cunche, M., Lauradoux, C., Le Métayer, D., and Roca, V., "Robert: Robust and privacy-preserving proximity tracing," 2020.

[7] Huey, L. C., Sebastian, P., and Drieberg, M., "Augmented reality based indoor positioning navigation tool," in *2011 IEEE Conference on Open Systems*, pp. 256–260, IEEE, 2011.

[8] Islam Chowdhury, A., Munem Shahriar, M., Islam, A., Ahmed, E., Karim, A., and Rezwanul Islam, M., "An automated system in atm booth using face encoding and emotion recognition process," in *2020 2nd International Conference on Image Processing and Machine Vision*, pp. 57–62, 2020.

[9] Davidson, P. and Piché, R., "A survey of selected indoor positioning methods for smartphones," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2016.

[10] Zafari, F., Gkelias, A., and Leung, K. K., "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[11] Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., and Aboutanios, E., "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.

[12] Minch, R. P., "Location privacy in the era of the internet of things and big data analytics," in *2015 48th Hawaii International Conference on System Sciences*, pp. 1521–1530, IEEE, 2015.

[13] Lopez-de Teruel, P. E., Garcia, F. J., Canovas, O., Gonzalez, R., and Carrasco, J. A., "Human behavior monitoring using a passive indoor positioning system: a case study in a sme," *Procedia Computer Science*, vol. 110, pp. 182–189, 2017.

[14] Holcer, S., Torres-Sospedra, J., Gould, M., and Remolar, I., "Privacy in indoor positioning systems: a systematic review," in *2020 international conference on localization and GNSS (ICL-GNSS)*, pp. 1–6, IEEE, 2020.

[15] Ahmad, M., Ahmed, I., and Jeon, G., "An iot-enabled real-time overhead view person detection system based on cascade-rcnn and transfer learning," *Journal of Real-Time Image Processing*, pp. 1–11, 2021.

[16] del Blanco, C. R., Carballeira, P., Jaureguizar, F., and García, N., "Robust people indoor localization with omnidirectional cameras using a grid of spatial-aware classifiers," *Signal Processing: Image Communication*, vol. 93, p. 116135, 2021.

[17] Gu, Y., Lo, A., and Niemegeers, I., "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications surveys & tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[18] Mautz, R., "Indoor positioning technologies," 2012.

[19] Basiri, A., Lohan, E. S., Moore, T., Winstanley, A., Peltola, P., Hill, C., Amirian, P., and e Silva, P. F., "Indoor location based services challenges, requirements and usability of current solutions," *Computer Science Review*, vol. 24, pp. 1–12, 2017.

[20] Morar, A., Moldoveanu, A., Mocanu, I., Moldoveanu, F., Radoi, I. E., Asavei, V., Gradinaru, A., and Butean, A., "A comprehensive survey of indoor localization methods based on computer vision," *Sensors*, vol. 20, no. 9, p. 2641, 2020.

[21] Mendoza-Silva, G. M., Torres-Sospedra, J., and Huerta, J., "A meta-review of indoor positioning systems," *Sensors*, vol. 19, no. 20, p. 4507, 2019.

[22] Pascacio, P., Casteleyn, S., Torres-Sospedra, J., Lohan, E. S., and Nurmi, J., "Collaborative indoor positioning systems: A systematic review," *Sensors*, vol. 21, no. 3, p. 1002, 2021.

[23] Li, Y., Zhuang, Y., Hu, X., Gao, Z., Hu, J., Chen, L., He, Z., Pei, L., Chen, K., Wang, M., and others, "Toward location-enabled iot (le-iot): Iot positioning techniques, error sources, and error mitigation," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4035–4062, 2020.

[24] Hightower, J. and Borriello, G., "Location systems for ubiquitous computing," *computer*, vol. 34, no. 8, pp. 57–66, 2001.

[25] Liu, H., Darabi, H., Banerjee, P., and Liu, J., "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, 2007.

[26] Faragher, R. and Harle, R., "Location fingerprinting with bluetooth low energy beacons," *IEEE journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[27] Duque Domingo, J., Cerrada, C., Valero, E., and Cerrada, J. A., "An improved indoor positioning system using rgb-d cameras and wireless networks for use in complex environments," *Sensors*, vol. 17, no. 10, p. 2391, 2017.

[28] Haider, A., Wei, Y., Liu, S., and Hwang, S.-H., "Pre-and post-processing algorithms with deep learning classifier for wi-fi fingerprint-based indoor positioning," *Electronics*, vol. 8, no. 2, p. 195, 2019.

[29] Martin, E., Vinyals, O., Friedland, G., and Bajcsy, R., "Precise indoor localization using smart phones," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 787–790, 2010.

[30] Ye, Q., Fan, X., Fang, G., Bie, H., Xiang, C., Song, X., and He, X., "Edgeloc: An edge-iot framework for robust indoor localization using capsule networks," *arXiv preprint arXiv:2009.05780*, 2020.

[31] Zhao, Z.-Q., Zheng, P., Xu, S.-t., and Wu, X., "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[32] Duque Domingo, J., Cerrada, C., Valero, E., and Cerrada, J. A., "Indoor positioning system using depth maps and wireless networks," *Journal of Sensors*, vol. 2016, 2016.

[33] Mizmizi, M. and Reggiani, L., "Binary fingerprinting-based indoor positioning systems," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–6, IEEE, 2017.

[34] Kumar, P., Reddy, L., and Varma, S., "Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks," in *2009 Fifth international conference on wireless communication and sensor networks (WCSN)*, pp. 1–4, IEEE, 2009.

[35] Subedi, S. and Pyun, J.-Y., "Practical fingerprinting localization for indoor positioning system by using beacons," *Journal of Sensors*, vol. 2017, 2017.

[36] Nessa, A., Adhikari, B., Hussain, F., and Fernando, X. N., "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020.

[37] Subedi, S., Gang, H.-S., Ko, N. Y., Hwang, S.-S., and Pyun, J.-Y., "Improving indoor fingerprinting positioning with affinity propagation clustering and weighted centroid fingerprint," *IEEE Access*, vol. 7, pp. 31738–31750, 2019.

[38] Chen, L., Pei, L., Kuusniemi, H., Chen, Y., Kröger, T., and Chen, R., "Bayesian fusion for indoor positioning using bluetooth fingerprints," *Wireless personal communications*, vol. 70, no. 4, pp. 1735–1745, 2013.

[39] Zafari, F., Papapanagiotou, I., Devetsikiotis, M., and Hacker, T., "An ibeacon based proximity and indoor localization system," *arXiv preprint arXiv:1703.07876*, 2017.

[40] Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, Ó., and Huerta, J., "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.

[41] Mendoza-Silva, G. M., Matey-Sanz, M., Torres-Sospedra, J., and Huerta, J., "Ble rss measurements dataset for research on accurate indoor positioning," *Data*, vol. 4, no. 1, p. 12, 2019.

[42] Zafari, F. and Papapanagiotou, I., "Enhancing ibeacon based micro-location with particle filtering," in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, 2015.

[43] Xiao, C., Yang, D., Chen, Z., and Tan, G., "3-d ble indoor localization based on denoising autoencoder," *IEEE Access*, vol. 5, pp. 12751–12760, 2017.

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

[45] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T.-K., "Multiple object tracking: A literature review," *Artificial Intelligence*, p. 103448, 2020.

[46] Beymer, D., "Person counting using stereo," in *Proceedings Workshop on Human Motion*, pp. 127–133, IEEE, 2000.

[47] Spinello, L. and Arras, K. O., "People detection in rgb-d data," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838–3843, IEEE, 2011.

[48] Rauter, M., "Reliable human detection and tracking in top-view depth images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 529–534, 2013.

[49] Park, S. and Aggarwal, J. K., "Head segmentation and head orientation in 3d space for pose estimation of multiple people," in *4th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 192–196, IEEE, 2000.

[50] Nakatani, R., Kouno, D., Shimada, K., and Endo, T., "A person identification method using a top-view head image from an overhead camera.," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 16, no. 6, pp. 696–703, 2012.

[51] Cosma, A., Radoi, I. E., and Radu, V., "Camloc: Pedestrian location estimation through body pose estimation on smart cameras," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, 2019.

[52] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[53] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

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

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

[56] Ciaparrone, G., Sánchez, F. L., Tabik, S., Troiano, L., Tagliaferri, R., and Herrera, F., "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[57] Xiang, Y., Alahi, A., and Savarese, S., "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, pp. 4705–4713, 2015.

[58] Bae, S.-H. and Yoon, K.-J., "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1218–1225, 2014.

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

[60] Wang, H., Lenz, H., Szabo, A., Bamberger, J., and Hanebeck, U. D., "Wlan-based pedestrian tracking using particle filters and low-cost mems sensors," in *2007 4th workshop on positioning, navigation and communication*, pp. 1–7, IEEE, 2007.

[61] Deng, Z.-A., Wang, G., Qin, D., Na, Z., Cui, Y., and Chen, J., "Continuous indoor positioning fusing wifi, smartphone sensors and landmarks," *Sensors*, vol. 16, no. 9, p. 1427, 2016.

[62] Dümbgen, F., Oeschger, C., Kolundžija, M., Scholefield, A., Girardin, E., Leuenberger, J., and Ayer, S., "Multi-modal probabilistic indoor localization on a smartphone," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–8, IEEE, 2019.

[63] Sun, Y., Meng, W., Li, C., Zhao, N., Zhao, K., and Zhang, N., "Human localization using multi-source heterogeneous data in indoor environments," *IEEE Access*, vol. 5, pp. 812–822, 2017.

[64] Zhao, Y., Xu, J., Wu, J., Hao, J., and Qian, H., "Enhancing camera-based multimodal indoor localization with device-free movement measurement using wifi," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1024–1038, 2019.

[65] Canetti, R., Trachtenberg, A., and Varia, M., "Anonymous collocation discovery: Harnessing privacy to tame the coronavirus," *arXiv preprint arXiv:2003.13670*, 2020.

[66] Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K., "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[67] Peng, Y., Fan, W., Dong, X., and Zhang, X., "An iterative weighted knn (iw-knn) based indoor localization method in bluetooth low energy (ble) environment," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pp. 794–800, IEEE, 2016.

[68] Jianyong, Z., Haiyong, L., Zili, C., and Zhaohui, L., "Rssi based bluetooth low energy indoor positioning," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 526–533, IEEE, 2014.

[69] Naik, N., "Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http," in *2017 IEEE international systems engineering symposium (ISSE)*, pp. 1–7, IEEE, 2017.

[70] Li, S., Tezcan, M. O., Ishwar, P., and Konrad, J., "Supervised people counting using an overhead fisheye camera," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–8, IEEE, 2019.

[71] Mao, Q.-C., Sun, H.-M., Liu, Y.-B., and Jia, R.-S., "Mini-yolov3: real-time object detector for embedded applications," *IEEE Access*, vol. 7, pp. 133529–133538, 2019.

[72] Huang, R., Pedoeem, J., and Chen, C., "Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2503–2510, IEEE, 2018.

[73] Fang, W., Wang, L., and Ren, P., "Tinier-yolo: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2019.

[74] Li, T., Ma, Y., and Endoh, T., "A systematic study of tiny yolo3 inference: Toward compact brainware processor with less memory and logic gate," *IEEE Access*, vol. 8, pp. 142931–142955, 2020.

[75] Held, D., Thrun, S., and Savarese, S., "Learning to track at 100 fps with deep regression networks," in *European conference on computer vision*, pp. 749–765, Springer, 2016.

[76] Lu, Z., Rathod, V., Votel, R., and Huang, J., "Retinatrack: Online single stage joint detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14668–14678, 2020.

[77] Kuhn, H. W., "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[78] Torres-Sospedra, J., Rambla, D., Montoliu, R., Belmonte, O., and Huerta, J., "Ujiindoorloc-mag: A new database for magnetic field-based localization problems," in *2015 International conference on indoor positioning and indoor navigation (IPIN)*, pp. 1–10, IEEE, 2015.

[79] Baronti, P., Barsocchi, P., Chessa, S., Mavilia, F., and Palumbo, F., "Indoor bluetooth low energy dataset for localization, tracking, occupancy, and social interaction," *Sensors*, vol. 18, no. 12, p. 4462, 2018.

[80] Demiröz, B. E., Ari, I., Eroğlu, O., Salah, A. A., and Akarun, L., "Feature-based tracking on a multi-omnidirectional camera dataset," in *2012 5th International Symposium on Communications, Control and Signal Processing*, pp. 1–5, IEEE, 2012.

[81] Nambiar, A., Taiana, M., Figueira, D., Nascimento, J. C., and Bernardino, A., "A multi-camera video dataset for research on high-definition surveillance," *International Journal of Machine Intelligence and Sensory Signal Processing*, vol. 1, no. 3, pp. 267–286, 2014.

[82] Hazas, M. and Ward, A., "A novel broadband ultrasonic location system," in *International Conference on Ubiquitous Computing*, pp. 264–280, Springer, 2002.

[83] Bahl, P. and Padmanabhan, V. N., "Radar: An in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, vol. 2, pp. 775–784, Ieee, 2000.

[84] Jiao, J., Li, F., Tang, W., Deng, Z., and Cao, J., "A hybrid fusion of wireless signals and rgb image for indoor positioning," *International Journal of Distributed Sensor Networks*, vol. 14, no. 2, p. 1550147718757664, 2018.

Kamal, Azmyin Md. Bachelor of Science, Ahsanullah University of Science and Technology, Spring 2015; Master of Science, University of Louisiana at Lafayette, Fall 2021
Major: Engineering
Title of Thesis: Anonymous Multi-User Tracking in Indoor Environment Using Computer Vision and Bluetooth
Thesis Director: Dr. Raju Gottumukkala
Pages in Thesis: 130; Words in Abstract: 278

Abstract

Various Indoor Positioning System (IPS) has been recently developed with the goal of accurately tracking people with user-level identification for improving safety, security and, optimizing indoor spaces and services (navigation, energy management, contact-tracing, etc.). Designing IPS requires finding an optimum balance between users' privacy, tracking accuracy, system latency, cost, and ease-of-use, etc. Vision-based positioning can localize users with $\leq 100cm$ resolution but cannot distinguish them uniquely without person-specific features such as facial image, biomarkers. On the other hand, IPS using radio frequency technologies can distinguish users using smartphone's MAC address or an anonymized identifier but have cost and performance trade-offs.

The primary goal of this thesis is to design a Multimodal Indoor Positioning System (MIPS) that can anonymously distinguish users and track them in indoor spaces. The proposed system takes advantage of the best of both sensing technologies i.e. a BLE 4.2 IPS generates an anonymous user identifier which is then paired with the object tracking capability of Overhead Vision IPS for privacy-preserving multi-user tracking. The system architecture uses inexpensive hardware, open-source software, and machine learning techniques to reach an optimum compromise between anonymity and tracking accuracy. We evaluate the performance of the MIPS system with three major metrics viz. Object Localization Error (OLE), Object Tracking Error (OTA), and System Latency. We found the overall MIPS performance for OTA $\geq 90\%$ and OLE $\leq 50cm$ for 3 out 5

scenarios. In these 3 scenarios, MIPS demonstrated near-perfect anonymous identification for as many as 6 users walking concurrently on the testbed. The average association latency was $\leq 0.06ms$ and the average system latency was $70ms$. These findings show the strong potential of MIPS for real-world multi-user anonymous tracking using Bluetooth and vision multimodal data provided certain conditions are met.

Biographical Sketch

Mr. Azmyin Md. Kamal is an international graduate student born on the 14th of July, 1993 in the port city of Chattogram, Bangladesh. He obtained a bachelors's degree with distinction in mechanical engineering from Ahsanullah University of Science and Technology (AUST) located in the capital city Dhaka, Bangladesh on 30th December 2015. After graduating, he served as a Junior Lecturer in the same department for 3 years from April 2016. During his time as an instructor, he taught courses on MATLAB, Engineering Design, Mechatronics, and Measurement and Instrumentation. Mr. Kamal's research interests lies in Robotics, IoT, and Deep Reinforcement Learning.

In August 2019, Mr. Kamal began his master's program in mechanical engineering at The University of Louisiana at Lafayette under the supervision of Dr. Raju Gottumukkala, Assistant Professor of Mechanical Engineering. There he worked as a Graduate Research Assistant in the Cyber-Physical Systems Lab where he received advanced training on data science and IoT technologies. His research work culminated in the creation of a novel positioning system which utilized Bluetooth Low and Computer Vision technologies to track human users for providing customized localization services without compromising their privacy. Mr. Kamal is scheduled to join the Louisiana State University as a Ph.D. student in the Department of Mechanical Engineering from Fall 2021.

Apart from academics, Mr. Kamal enjoys driving, playing video games, and helping his friends and community members. He hopes to return to his country after completing his Ph.D. and necessary research training here in the US to advance research in robotics in Bangladesh.