

Improving Indoor Pedestrian Detection and Tracking in Crowded Environments: Deep Learning Based Approaches

YU ZHANG

Doctor of Philosophy

Supervisor: Dong Yuan
Associate Supervisor: Huaming Chen, Wei Bao

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Electrical and Information Engineering
Faculty of Engineering
The University of Sydney
Australia

3 March 2024

Abstract

With the increasing demand for location-based indoor services and the rapid advancements in related technologies, indoor positioning has gained significant traction. This research dives into several leading indoor localisation technologies, introducing enhancements through different technologies.

GauPro, a beacon probabilistic fingerprint algorithm, leverages the Gaussian kernel model and an LSTM model to refine the precision of conventional algorithms. Our system offers positioning services for standard smartphone users without requiring hardware modifications on the receiving devices. Evaluations in various environments, including rooms, corridors and an open room, reveal that GauPro surpasses the conventional Bayesian probability-based approach by 17.3% in error distance.

Computer vision-based indoor positioning grapples with challenges in object detection and tracking data association. Addressing this, we introduce **GATracker**, a two-stage pedestrian tracker. This model employs bipartite matching for tracklet proposals and a trainable graph attention network to pinpoint split points, enhancing tracking accuracy. To increase detection performance in densely populated scenes, we propose an anchor-free joint head and body detector **JointTrack**. This model dynamically learns the head-body ratio, eliminating the need for statistical data during training. Evaluations on datasets such as MOT20, Crowdhuman, and HT21 confirm its superior performance, especially for small and medium-sized pedestrians.

The association of visual detection with users in crowded settings remains a challenge. Our solution **PediFuse** integrates real-time video analytics with wireless signal-enabled devices using a deep reinforcement learning scheme. The agent extracts motion patterns from both visual tracking and wireless signals, merging these features to identify the best match. Experiments in a simulated warehouse setting underscore the efficacy of our approach, which markedly outperforms existing analytical methods.

In summary, we have taken a multifaceted approach to advance the state-of-the-art in indoor positioning systems. Each method we propose represents a significant step forward that addresses the distinct challenges faced in the field of indoor localisation. Collectively, these advancements not only highlight the gaps in current systems, but also pave the way for more precise, efficient, and adaptable indoor positioning services in diverse environments.

Statement of Originality

I hereby affirm that, to the utmost of my understanding, the material presented in this thesis is a result of my individual effort. This thesis has not previously been submitted for any academic degree or other evaluations.

I confirm that the intellectual contributions in this thesis are solely my own, and any guidance or resources utilised in its creation have been duly acknowledged.

Yu Zhang

March 3, 2024

Authorship Attribution Statement

Publications or accepted papers included in this thesis:

- **Y. Zhang**, Z. Lai, D. Yuan, W. Bao, B. Zhou, J. Qiu, S. Wang and S. Adam. "GauPro: An Accuracy-Improved Indoor Positioning System Based on Beacon Probabilistic Fingerprint." In: *2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*. 2021, pp. 276-283.

I designed the study, analysed the data and wrote the drafts of the MS.

- **Y. Zhang**, H. Chen, Z. Lai, Z. Zao and D. Yuan. "Handling Heavy Occlusion in Dense Crowd Tracking by Focusing on the Heads." In: *Advances in Artificial Intelligence*. 2023, pp. 79-90.

I designed the study, analysed the data and wrote the drafts of the MS.

- Z. Lai, D. Yuan, W. Bao, **Y. Zhang** and B. Zhou. "DeepWiSim: a wireless signal simulator for automatic deep learning." In: *Proceedings of the SIGCOMM'22 Poster and Demo Sessions*. 2022, pp. 13-15.

I co-designed the study with the co-authors and interpreted the analysis.

- Z. Lai, D. Yuan, H. Chen, **Y. Zhang** and W. Bao. "WirelessDT: A Digital Twin Platform for Real-Time Evaluation of Wireless Software Applications." In: *IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. 2023, pp. 146-150.

I co-designed the study with the co-authors and interpreted the analysis.

The papers included in this thesis that are under review are listed as follows. Portions of this thesis are derived from papers I have co-authored and submitted to a double-blind

reviewed journal, and due to the anonymity constraints of the review process, specifics about the journal and paper cannot be disclosed at this stage.

- **Y. Zhang**, D. Yuan, Z. Lai and W. Bao. "GATracker: Graph Attention Network Based Tracklet Separation for Multiple Object Tracking." Submitted to IEEE Transactions on Intelligent Transportation Systems.
- **Y. Zhang**, J. Jin, H. Chen, Z. Lai and D. Yuan. "End-to-End Multimodal Indoor Pedestrian Identification With Deep Reinforcement Learning." Submitted to a double-blind top journal.
- Z. Lai, H. Chen, J. Grundy, **Y. Zhang** and D. Yuan. "WirelessDT: A Digital Twin System for Wireless and Mobile Applications." Submitted to IEEE Transactions on Mobile Computing.

Publications or accepted papers that are not included in this thesis:

- Z. Zhang, D. Yuan, **Y. Zhang** and W. Bao. "Trajectory Alignment based Multi-Scaled Temporal Attention for Efficient Video Transformer." *In: IEEE International Conference on Multimedia and Expo (ICME)*. 2023, pp. 1409-1414.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Yu Zhang

March 3, 2024

As a supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Dong Yuan

March 3, 2024

Acknowledgements

I would like to express my deepest gratitude to my lead supervisor, Dr. Dong Yuan, for his invaluable guidance, unwavering support, and mentorship throughout this research journey. My sincere thanks also go to my associate supervisors, Huaming Chen and Wei Bao, for their insightful feedback, expertise, and continuous encouragement.

I am also indebted to the entire research team, whose collective efforts and collaboration have been instrumental in the progress and success of this work. Their camaraderie, shared wisdom, and constructive critiques have enriched this research experience.

Furthermore, I would like to extend my appreciation to the Engineering School of USYD for fostering an environment of academic excellence and innovation. A special mention to CHUR Networks for their generosity in providing the scholarship opportunity, which has been pivotal in facilitating this research.

This journey would not have been possible without the unwavering support and belief of all those mentioned above. To each one of you, thank you.

Contents

Abstract	ii
Statement of Originality	iv
Authorship Attribution Statement	v
Acknowledgements	vii
Contents	viii
List of Figures	xii
List of Tables	xviii
Chapter 1 Introduction	1
1.1 Overview of Indoor Positioning Service	1
1.2 Research Gap and Motivation	5
1.2.1 Wireless Approach Research Gap	8
1.2.2 Computer Vision Approach Research Gap	8
1.2.3 Multimodal Approach Research Gap	10
1.3 Contribution.....	11
Chapter 2 Literature Review	14
2.1 Indoor Positioning Overview.....	15
2.2 Indoor Positioning Approaches.....	15
2.2.1 Traditional Approaches	18
2.2.2 Deep Learning Approaches	23
2.2.3 Concepts in Deep Learning	24
2.2.4 Example Research Works	25
2.3 Indoor Positioning Technologies	29

2.3.1	Wireless Indoor Positioning	30
2.3.2	Computer Vision Indoor Positioning	32
2.4	Multimodal indoor positioning	42
2.4.1	IMU with Bluetooth Approach	43
2.4.2	IMU with Computer Vision Approach	44
2.4.3	IMU with Computer Vision & Bluetooth Approach	46
2.4.4	Wi-Fi with Computer Vision Pedestrian Identification	47
Part 1.	Wireless Signal-Based Approach	49
Chapter 3	Wireless Signal Based Deep Learning Fingerprint IPS	52
3.1	Introduction	52
3.2	Related Work	54
3.2.1	Overview of Wireless Indoor Positioning	54
3.2.2	Wireless Positioning Techniques	55
3.3	System Design and the GauPro Algorithm	56
3.3.1	System Overview	56
3.3.2	Offline Phase	57
3.3.3	Online Phase	58
3.3.4	Temporal Feature Extraction for IMU Orientation	62
3.3.5	Training Detail of the Temporal Feature Extraction Model	63
3.4	System Implementation	64
3.4.1	Mobile App Implementation	64
3.4.2	Back-End Server Implementation	66
3.5	Evaluation	68
3.5.1	Experimental Setup	68
3.5.2	Benchmark Approach and Testing Criteria	69
3.5.3	Positioning Result	70
3.6	Conclusion	72
Part 2.	Computer Vision-Based Approach	75

Chapter 4	Graph Attention Network based Multiple Object Tracking	78
4.1	Introduction	78
4.2	Related Works	81
4.3	Methodology	83
4.3.1	Overview	83
4.3.2	Feature Extraction and Bipartite Matching	84
4.3.3	Classifying Nodes in Tracklet	86
4.4	Experiments	89
4.4.1	Datasets and Evaluation Method	89
4.4.2	Implementation Details	90
4.4.3	Ablation Study	91
4.4.4	Benchmark Evaluation	95
4.5	Conclusions	96
Chapter 5	Joint Head Body Detection for Crowded Environment	97
5.1	Introduction	97
5.2	Related Work	99
5.2.1	Object Detection and Occlusion Handling in Pedestrian Tracking	99
5.2.2	Head and Body Detection in Dense Crowd Tracking	101
5.3	Methodology	102
5.3.1	Framework Overview	102
5.3.2	Anchor-Free Head-Body Detection	103
5.3.3	Joint SimOTA	104
5.3.4	Tracking Framework	105
5.3.5	Loss Function	107
5.3.6	Training Details	107
5.4	Experiments	109
5.4.1	MOT Challenge	109
5.4.2	Qualitative Result on MOT20	110
5.4.3	Ablation Study on Joint SimOTA	111
5.4.4	Crowdhuman	112

5.4.5	Head Detection and Tracking	113
5.5	Conclusion	113
Part 3.	Multimodal Approach	115
Chapter 6	A Deep Reinforcement Learning Based Multimodal IPS Approach	118
6.1	Introduction	118
6.2	Background Technology and Preliminaries	122
6.2.1	Real-World Experiment vs. Emulator	122
6.2.2	Shopping Mall Environment	126
6.2.3	Involved Sensors	127
6.2.4	Background of Deep Reinforcement Learning	132
6.3	DRL Environment of Pedestrian Identification	146
6.3.1	Problem Formulation	146
6.3.2	State Space	147
6.3.3	Action Space	149
6.3.4	Reward	150
6.4	Training RL Agent for Identification Matching	151
6.5	Evaluation	155
6.5.1	Experiment Setup	155
6.5.2	Evaluation of Different Modalities	157
6.5.3	Evaluation of User Identification Matching	158
6.5.4	Influence of Accumulating Reward	159
6.5.5	Analysis with Case Study	160
6.6	Conclusion	163
Chapter 7	Conclusion	164
7.1	Future Outlook	167
Bibliography		168

List of Figures

1.1	Examples of the Indoor positioning service use cases. Top left: An advanced camera system integrated into intelligent furniture, enabling precise indoor positioning for enhanced user experiences; Top right: A sprawling shopping mall that uses indoor positioning services to guide visitors, improve shop recommendations, and optimise store layouts. Bottom left: A modern museum that utilises indoor positioning to provide interactive tours, direct visitors to exhibits, and offer augmented reality experiences. Bottom right: A bustling train station that employs indoor positioning for real-time train tracking, platform guidance, and helping commuters navigate the complex terminal.	2
1.2	Indoor technologies in dependence on accuracy and coverage.	6
1.3	Indoor technologies in dependence on accuracy and cost.	7
1.4	The structure of this paper.	11
2.1	Example of using RSSIs to locate the user given multiple Access points. . .	16
2.2	Example of using AoAs to locate the user given multiple Access points. . .	19
2.3	Example of indoor radio map for the fingerprinting methods using a single Access Point. Red dots are the sample location used for map construction (offline training phase), green dots are used for testing.	22
2.4	The structure of a typical Convolutional Neural Networks (CNN) by Sumit Saha [39], which starts with the feature extraction layers and is followed by fully connected classification layers.	33
3.1	Overview of the indoor positioning system.	56
3.2	Workflow of fingerprint construction.	65
3.3	Workflow of back-end server	67
3.4	Testing areas.	69

3.5	Comparison of error distances for the empty hall area.	71
3.6	Precision comparison for empty hall area.	72
3.7	Comparison of error distances for the workplace area.	73
3.8	Precision comparison for the workplace area.	73
4.1	Concept of GATracker. 1) Given all the detection in a MOT video sequence, 2) we produce raw tracklets by clustering detection with similar features in time order with bipartite matching. 3) Then, we train a graph attention network model to separate mismatched tracklet.	79
4.2	Overview of our proposed method. (a) We receive detection results from object detection model and extract the visual features with a pretrained ReID model. (b) We initialize the tracking by taking the detection result from the first frame as starting objects, and match new detection with the objects. The matching process consider both the spatial-temporal features and the appearance features. (c) We take the generated tracklet proposals, construct graphs according to their features. Each node represent one frame of detection in the whole tracklet. (d) We feed the graphs into our trained GAT model to find out the split point on each tracklet (i.e. to label every node in the graph). (e) We separate the tracklet according to the output of our GAT model to obtain the final tracking result.	81
4.3	Example of tracklet visual extraction and processed with GAT to find the split point.	84
4.4	Example of two tracklets with large feature changing.	89
4.5	MOTA and IDF1 vs K values.	92
5.1	MOTA-IDF1 comparisons of different trackers on the test set of MOT20. With the help of our supplementary body detector, the proposed tracking framework achieves 78.2 MOTA and 75.5 IDF1 scores, outperforming all previous trackers.	98

5.2	Overview of the proposed model. The image is first processed by a FPN structure to extract features of different scales. Then, a 1×1 stem conv layer followed by two 3×3 branches extract the classification and regression features of the image. For the regression branch, we add two additional prediction heads to predict the head and body simultaneously. The predictions are evaluated by a Joint SimOTA module to filter out the grids that have either bad head prediction or bad body prediction. We calculate losses for the rest of the grids.	100
5.3	Heatmap comparison between head detection and body detection model under the same model structure. (a) is the input image from the MOT20 data set. (b) is the heatmap from the baseline. (c) Heatmap of our model. Compared to the baseline, our heat map is more distinguishable and can detect more pedestrians in the scene.	103
5.4	Illustration of our anchor-free approach vs. traditional static head-body ratio anchor-based approach. a). Anchor-based approach, where head and body boxes are regressed from separate RoIs. b). Our anchor-free approach, where head and body boxes are predicted simultaneously from the same grid box, achieving a faster inference speed and fewer parameters for training.	105
5.5	Qualitative results on MOT20 dataset. Pedestrians detected by the baseline are shown in grey boxes. Pedestrians detected by our method which are not detected by the baseline are shown in red boxes. Best viewed in colour. . . .	107
5.6	Demo of creation of our mixed training dataset. We use bipartite matching to assign head labels to body labels. a): Red boxes represents the body labels from MOT20 dataset, green boxes represents the head labels from HT21 dataset; b): after bipartite matching, each head label is assigned a body label similar to Crowdhuman dataset. Same colour represents the label box belong to the same person.	108

5.7	Comparison of head detection between our model and the baseline model. The person dressed in pink covered their head with an umbrella, making the features of the head invisible to CNN. a). One of the baseline models only detects two heads in the scene; b). Another baseline model predicts the wrong location of the head, represented with a red box; c). Our model can correctly predict the three locations of the head of the person.	112
6.1	Illustration of the testing ground for our proposed multimodal IPS algorithm. (A) Overview of the shopping mall: A comprehensive structure mainly constructed of concrete and glass, surrounded by trees and roads. The total area is approximately 20 km^2 . (B) Interior of the mall: A diverse assortment of establishments, such as food courts, entertainment areas, retail stores, and bookshops, are present. To mimic real-world scenarios, various furnishings, such as sofas and indoor plants, are strategically placed. (C) Ground-level layout: This provides a clear representation of the interior architecture and arrangement. (D) Detailed grid layout: Offers a more precise visualisation, highlighting structural elements such as walls and pillars.	123
6.2	Wireless transmitter sensor example in the shopping mall environment. . . .	128
6.3	Wireless receiver examples for the shopping mall environment. We combine the default UE5 AI and our programmed wireless receiver object to simulate pedestrians in the real world.	129
6.4	A detailed blueprint implementation of the camera sensor for our simulated environment.	130
6.5	A detailed blueprint implementation of the IMU sensor for our simulated environment.	132
6.6	A example of the Markov Decision Process (MDP).	134
6.7	An example of the Markov Decision Process for agent training. Given every state s_t , an action a_t is applied to the environment, causing the state to transit to another state s_{t+1} . The reward for this action r_t is also gained in this process. We collect the reward and proceed with the training procedure. . .	134
6.8	A Taxonomy of RL Algorithms according to OpenAI.	137

6.9	Value-Based RL and Policy-Based RL.	140
6.10	An overview of the standard Actor-Critic framework.	141
6.11	Overview of our proposed DRL training process for pedestrian identity matching. Three types of input are extracted from the emulated environment: visual tracking results from IP Cameras, acceleration and orientation of the device, and received wireless signal strength. These inputs are processed as observation and fed to the agent to perform identity matching. At each timestamp, the agent maps the device to a visual detection. The predictions are evaluated by the reward generated by the environment.	147
6.12	The provided illustration presents a comprehensive setup of a warehouse - one of the most classic indoor positioning application environments. The top left section provides a camera view of the indoor environment, the detailed floor plan which delineates the warehouse's architectural blueprint is shown in the top right, and the RSSI distribution of the environment is shown on the bottom. Wireless transmitters are positioned near the shelves. The strength of the wireless signal is intuitively represented through a gradient of colours; brighter shades correspond to a more robust signal. We disable some of the transmitters for better visualization.	154
6.13	Ablation study on different modalities with three (top left), five (top right), ten (bottom left), and twenty (bottom right) pedestrians simultaneously present in the scene.	159
6.14	Benchmark evaluation against RFCam and PHADE for three (top left), five (top right), ten (bottom left), and twenty (bottom right) people scene.	160
6.15	Ablation study for training w/o cumulative reward.	161
6.16	Visual trajectory vs. motion feature for one tracklet, time length is 150. The X and Y axes represent the coordinates of normalised objects. The tracklet is depicted through a colour gradient that transitions from blue to red. This colour transition signifies the progression of movement from the starting point (blue) to the ending point (red). Best viewed in colour.	162

6.17	Visual trajectory vs Visual feature extracted by the agent. Different trajectories shows very different features.	162
6.18	Matching reward example for a trajectory of 150 time steps, 1 indicates correct matching and 0 otherwise.	163
7.1	Intrinsic connections of the algorithms proposed in this thesis.	165

List of Tables

2.1	Traditional and deep learning Indoor Localization Approaches	18
2.2	Indoor Positioning Technologies	29
4.1	Performance comparison with bipartite matching is not performed.	92
4.2	Performance comparison when GAT model is not included (top: detection from [120]; bottom: detection from [144]).	93
4.3	Performance comparison with state-of-the-art on MOT17 and MOT20. Detection preprocess by Tractor, PermaTrack and TransCtr.	93
4.4	Performance comparison with different feature encoding.	94
4.5	Performance comparison between GAT and GCN.	94
4.6	Performance comparison when tracklet reconnect is performed.	94
5.1	AP and MR^{-2} performance of different detection methods on MOT20. . . .	110
5.2	Performance comparison on the test set with state-of-the-art on MOT20. . .	110
5.3	Performance comparison on the test set with state-of-the-art on HT21. . . .	111
5.4	Head and body AP performance of different detection methods on Crowdhuman. .	111

CHAPTER 1

Introduction

1.1 Overview of Indoor Positioning Service

Indoor Positioning Service (IPS) refers to a system that locates objects or people inside a building using sensors, radio waves, magnetic fields, or other informational cues. Unlike the Global Positioning System (GPS), which provides outdoor navigation, IPS operates within enclosed spaces, offering high precision in complex indoor environments. There are many use cases for IPS. For example, in retail and shopping malls, retailers can use IPS to guide customers to specific products, offer promotions when a buyer is near a particular aisle, or analyse foot traffic patterns to optimise store layouts. At airports and train stations, travellers can be guided to their gates, baggage claim areas, or other points of interest, improving the overall travel experience. For museums and tourist attractions, visitors can receive detailed information about exhibits as they approach them or be guided on a curated tour route. We show some use cases in Figure 1.1.

In the above-described use cases, GPS does not work in indoor environments due to many factors:

- **Signal Attenuation:** The materials used in modern buildings, such as metal, concrete, and brick, can significantly weaken or block GPS signals. This attenuation makes it difficult for a GPS receiver inside a building to pick up weak signals.
- **Initial Position Acquisition:** Even if some GPS signals penetrate indoors, acquiring an initial position (cold start) can take a long time due to the weakened signal strength.



FIGURE 1.1. Examples of the Indoor positioning service use cases. Top left: An advanced camera system integrated into intelligent furniture, enabling precise indoor positioning for enhanced user experiences; Top right: A sprawling shopping mall that uses indoor positioning services to guide visitors, improve shop recommendations, and optimise store layouts. Bottom left: A modern museum that utilises indoor positioning to provide interactive tours, direct visitors to exhibits, and offer augmented reality experiences. Bottom right: A bustling train station that employs indoor positioning for real-time train tracking, platform guidance, and helping commuters navigate the complex terminal.

- **Signal Noise and Interference:** For indoor environments, people carry many electronic devices and other equipment inside buildings, which can produce a lot of electromagnetic interference, which can introduce noise into the GPS signal, further reducing its accuracy.
- **Dense Urban Environments:** In urban canyons, where tall buildings surround narrow streets, GPS faces challenges similar to those in indoors. Buildings can block, reflect, or refract GPS signals, leading to inaccuracies.

Due to these challenges, alternative methods and technologies, such as Wi-Fi-based positioning, Bluetooth Low Energy (BLE) beacons, surveillance cameras, and sensor fusion, are often used for indoor positioning and navigation. With the proliferation of smartphones and IoT

devices, coupled with advances in sensor technology and data processing, IPS has become more accessible and accurate.

In addressing the challenges of IPS, this thesis concentrates on Wi-Fi-based Positioning and Surveillance Cameras combined with Computer Vision, primarily due to their widespread availability, scalability, and the depth of data they can provide. Wi-Fi-based Positioning is selected because of the ubiquity of Wi-Fi infrastructure in urban environments, offering a cost-effective and readily available means to deploy IPS without the need for additional specialised hardware. This approach leverages existing network equipment and can be easily scaled in large areas, making it an attractive option for indoor navigation solutions in commercial, educational and public buildings. Surveillance cameras and computer vision are chosen for their ability to offer high-accuracy positioning and the potential for rich analytics, such as foot traffic analysis and behaviour monitoring. With advances in artificial intelligence and machine learning, computer vision algorithms have become highly effective in processing and interpreting video data, enabling precise tracking of individuals within complex indoor spaces.

- **Wireless Signal based** approaches use Received Signal Strength Indicator (RSSI) from Wi-Fi or Bluetooth access points (APs), which are often used to triangulate the user's position. It is one of the most common methods due to the widespread availability of Wi-Fi infrastructure. One of the advantages of these approaches is their ubiquity of infrastructure. Many indoor environments, such as shopping malls, airports or office buildings, already have Wi-Fi or Bluetooth infrastructure in place. This makes it cost-effective to implement an IPS without the need for additional hardware. Therefore, implementing an RSSI-based IPS can be as simple as implementing a software solution that takes advantage of existing Wi-Fi or Bluetooth APs. However, the accuracy of RSSI-based IPS can be affected by various factors, including interference, multipath propagation, and signal attenuation. Physical obstacles such as walls, floors, and even people can affect signal strength, leading to potential inaccuracies. In addition, dynamic changes in the environment can also affect RSSI readings.

- **Vision-based** methods employ cameras and advanced image processing technologies to detect landmarks or track pedestrians within an environment. A plethora of techniques are available, with deep learning-based approaches being the most prevalent and effective. These vision-based strategies can be broadly divided into two categories: navigation and surveillance. The navigation method is primarily used in robot navigation, leveraging first-person view cameras to identify landmarks and cross-reference them with a preexisting database. However, surveillance techniques use environmental cameras to monitor and pinpoint the locations of individuals indoors.

Regardless of their specific application, vision-based methods share certain advantages and disadvantages. When equipped with the appropriate algorithms and training datasets, these systems can achieve remarkable precision in landmark recognition and position estimation. The depth of visual data captured by cameras provides a comprehensive understanding of the environment, which proves beneficial for tasks beyond mere positioning, such as identifying objects or analysing activities. The presence of existing surveillance or camera infrastructure further enhances the utility of these systems. Moreover, deep learning models can be adeptly trained to discern a diverse range of landmarks, from static entities like artworks or statues to dynamic subjects like humans or robots.

However, there are challenges associated with vision-based methods. Deep learning models, particularly convolutional neural networks (CNNs), require significant computational resources, often necessitating robust processors or GPUs for real-time operations. Additionally, these systems exhibit a pronounced sensitivity to lighting conditions. Variations in illumination or suboptimal lighting can significantly impair their performance. Another limitation arises when objects or individuals obstruct the camera's view, complicating the task of recognising landmarks or persons. Processes like pedestrian or landmark re-identification can be both resource-intensive and lacking in efficiency.

- **Sensor Fusion** approaches combine data from multiple sensors (such as accelerometers, gyroscopes, and cameras) present in smartphones to estimate position and

orientation. This approach combines the above-mentioned techniques to cover each other's disadvantages. By integrating data from multiple sensors, sensor fusion can provide more reliable and consistent positioning information, especially when one or more sensors fail or provide inaccurate data. Also, combining data from different sources can lead to more accurate estimates than relying on a single sensor type. However, implementing sensor fusion requires sophisticated algorithms that may consume a large amount of computing resources from the central server. Calibration is also a great challenge, as ensuring that all sensors are properly calibrated and synchronised can be challenging. Any misalignment or timing difference can introduce errors. With more sensors involved, there is a higher chance of one or more sensors failing or needing a recalibration, increasing maintenance demands.

This thesis aims to explore a synergistic approach that maximizes the strengths of both Wi-Fi signals and visual data, potentially overcoming the limitations faced by other IPS methods and pushing the boundaries of what can be achieved in indoor positioning accuracy and application.

1.2 Research Gap and Motivation

Figure 1.2 provides a graphical overview that compares the precision and coverage of various common indoor positioning techniques. The circle size roughly represents how many related methods are proposed. Vision-based methods stand out for their superior accuracy, but are constrained by a limited coverage area. In contrast, wireless signal-based methods may offer less precision, but excel in extensive coverage due to the inherent nature of the signals. Within this category, UWB and RFID typically exhibit greater accuracy compared to Wi-Fi and Bluetooth. Sound-based techniques, though less prevalent, align closely in accuracy and coverage with Bluetooth signals. Earth magnetic-based and Pedestrian Dead Reckoning (PDR) based methods are not constrained by coverage limitations. However, they may either lag in accuracy or be susceptible to the drift effect.

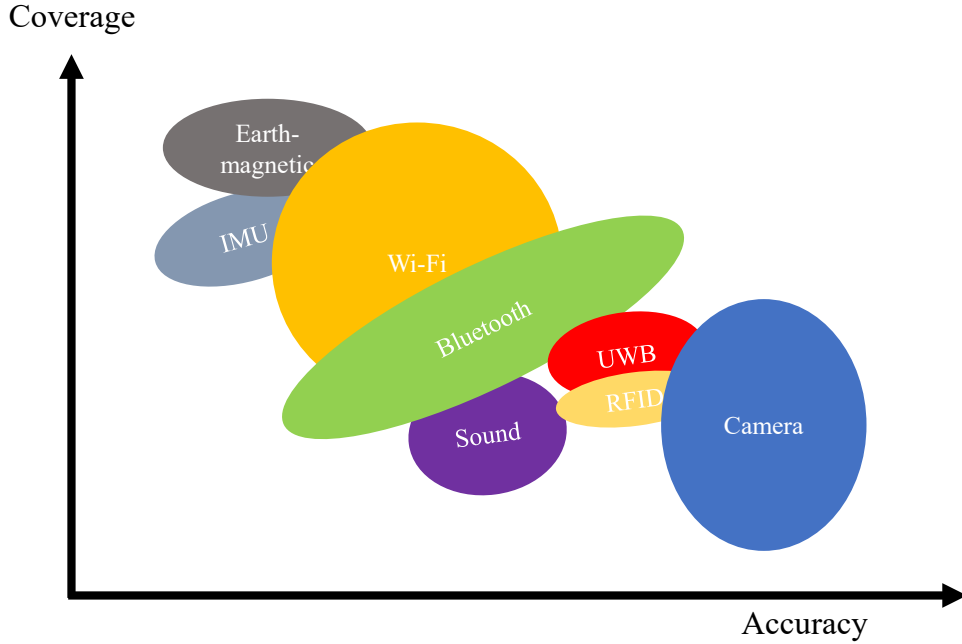


FIGURE 1.2. Indoor technologies in dependence on accuracy and coverage.

Figure 1.3 illustrates the relationship between accuracy and computational cost for various indoor positioning methods. Here, the “cost” refers to the computational resources required by each technique. PDR-based approaches are the most resource-efficient, relying solely on the smartphone’s Inertial Measurement Unit (IMU) sensors, resulting in minimal computational overhead. Multilateration approaches demand a dense network of transceivers, combined with sophisticated data processing software. Consequently, it is resource-intensive, despite offering accuracy similar to fingerprint-based methods. Fingerprint-based approaches have similar accuracy to multilateration methods, which is more cost-effective. Its primary expenditure is the extensive data collection required during the offline map construction phase. Deep Neural Networks (DNNs) can enhance the aforementioned techniques by supplementing their original algorithms with trained neural networks. Although they require substantial datasets for training and robust computing capabilities, the inferencing phase is relatively less resource-intensive than training. Given the rapid advances in DNNs, these approaches often deliver superior performance, albeit at a medium to high computational cost.

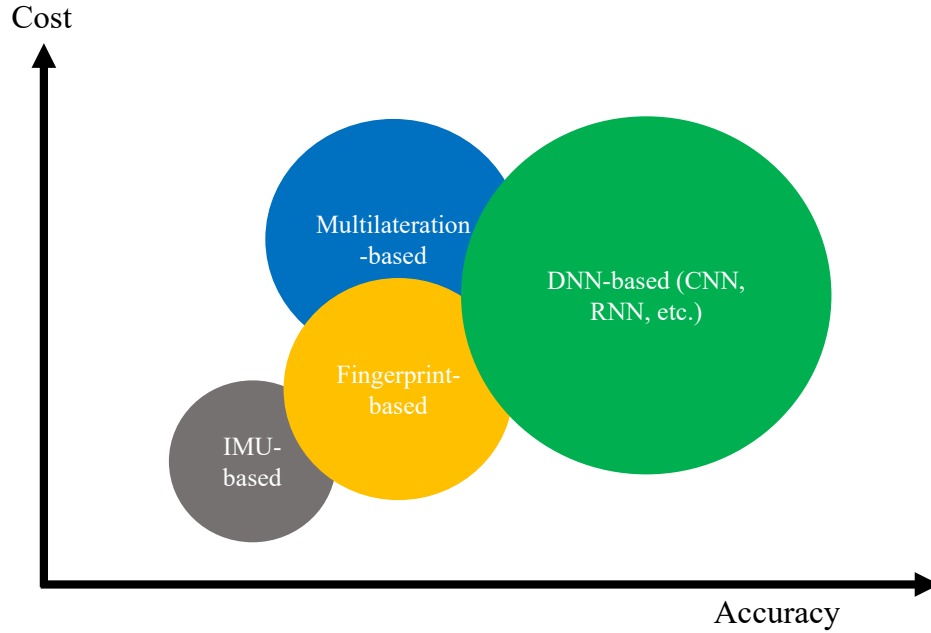


FIGURE 1.3. Indoor technologies in dependence on accuracy and cost.

The research gap exists in many aspects of current IPS solutions. For wireless signal-based approaches, the main challenges are the environmental dynamics. Human movements, the opening/closing of doors and changes in the furniture layout can affect the wireless signal propagation significantly, leading to inaccuracies in positioning. One of the most recent large-scale indoor localisation research articles “Experience: Pushing Indoor Localisation from Laboratory to the Wild” [83] by Tencent Inc. mentioned that the main research gap for indoor localisation is the transition from laboratory settings to real-world applications. When we do the transitions, many unforeseen challenges arise, such as the environmental variability and the hardware inconsistencies. The physical layout, building materials, and presence of obstacles can vary greatly in real world settings. This can affect signal propagation and lead to unpredictable interference. Different devices with varying capabilities might be used in real-world deployments. These devices can have different signal transmission/reception characteristics, which can introduce variability in system performance.

1.2.1 Wireless Approach Research Gap

In real-world IPS environments, conducting experiments presents significant challenges compared to laboratory settings. In laboratories, experiments are typically conducted on a smaller scale, often involving fewer than 20 participants. On the contrary, real-world scenarios can encompass thousands of individuals in a single indoor space, amplifying the complexities mentioned earlier. This vast difference in scale not only exacerbates the challenges, but also intensifies the demands for computational resources. Controlling variables becomes increasingly difficult in such settings, especially factors such as crowd density, movement patterns, and signal strength variations. The paper by Tencent underscores this challenge by conducting experiments in the world's third largest shopping mall, a venue that is not easily accessible for most researchers and far from a typical experimental environment.

This problem drive our pursuit for a more refined solution for IPS. In this paper, we aim to tackle challenges from multiple dimensions. Addressing wireless challenges, we note that in fingerprint-based IPS methods, after the map construction phase, the mobile device, upon receiving real-time signal strength, seeks the optimal match in the database. Typically, probabilistic matching techniques such as the Bayesian algorithm [1] are employed. In these methods, the prior probability assigned to the checkpoints remains constant. Yet, in a dynamic environment, one's direction of movement should influence the weightage given to checkpoint selection.

1.2.2 Computer Vision Approach Research Gap

The research gap also exists in the technologies themselves. Typically, for computer vision approaches, the camera detection performance often limits the tracking performance. Detecting every pedestrian in a crowded scene has been a great challenge over the past few decades. In the view of the surveillance camera, pedestrians often overlap or block each other. This makes it difficult for algorithms to detect and differentiate between individuals, especially when only a small portion of a person is visible. In addition, pedestrians come in various sizes, shapes, and clothing. The wide range of appearances can make it challenging for traditional

algorithms to generalise and detect all individuals accurately. Scale variations are also a big problem. In a single image, pedestrians closer to the camera appear larger than those farther away. Algorithms need to account for these scale variations to detect all individuals. Moreover, depending on the camera angle, position, and lens type, there can be perspective distortions that affect the perceived size and shape of pedestrians, making detection more challenging.

We address the intricacies associated with vision-based indoor positioning techniques. A significant observation is that in crowded scenarios, heavy occlusions often compromise detection performance. Traditional detection methods typically aim to predict an individual's entire body, akin to how they predict other objects. However, specific body parts, especially the head, can be instrumental in human identification. Drawing inspiration from body part detection studies [22, 95, 35], we explore the efficacy of anchor-free body part detection methods to enhance the accuracy of detecting individuals in crowded settings.

People tracking performance relies on the initial detection accuracy. If the detection phase misses a pedestrian or falsely detects a non-pedestrian object, the tracking phase will inherit these errors. Even given perfect detection results, tracking every pedestrian is still a formidable challenge for several reasons. First of all, there is the identity switch problem. In dense crowds, tracking algorithms can confuse one individual with another, leading to identity switch. This is especially common when individuals cross paths or move in close proximity. Due to the challenges mentioned above, tracking algorithms might produce fragmented tracks for a single individual, where the person is lost and then re-acquired multiple times. Human reidentification [108] is applied to solve this problem, but it often requires a lot of computational costs, which makes it very hard for real-time processing.

To capitalise on the enhanced detection capabilities, we also probe potential refinements for two-stage tracking algorithms. The term "two-stage tracking" refers to algorithms that take advantage of the provided detection results to facilitate the tracking. Although improved detection can inherently boost tracking accuracy, the standalone tracking performance can also be increased by taking low-score detection into consideration. We have taken note of the recent advancements in graph neural networks (GNNs) and recognise that the Multi-Object

Tracking (MOT) challenge can be conceptualised as a graph problem [120]. Within this graph, nodes can be assigned varying weights using attention mechanisms, enabling the modelling of nonlinearities and intricate interconnections.

1.2.3 Multimodal Approach Research Gap

Multimodal approaches, which combine data from multiple sources to enhance the accuracy and robustness of a system, have gained popularity in indoor positioning systems. However, the integration of different modalities, such as Wi-Fi received signal strength (RSSI) and visual tracking results, presents its own set of challenges. Beyond the evident challenges of data synchronisation and computational demands, many existing data fusion techniques focus on extracting user motion patterns to craft user profiles, subsequently aiming to align these profiles. Although basic motion features, such as standing, moving, and turning, can offer a degree of identity discernment, their effectiveness diminishes in crowded settings where individual movements often overlap and are less distinct. Although more intricate motion patterns can be delineated using hand-crafted feature extractors, these often fall short in performance and are computationally taxing.

We have discerned opportunities to enhance the fusion of different modalities for IPS. A notable observation is that during the extraction of pedestrian motion patterns, many subtle and nuanced motions are often overlooked. This is mainly because the system is typically calibrated to identify only the most prominent or representative patterns. We see potential in deploying a neural network-based approach that can autonomously extract a plethora of motion features from crowded scenes without the need for manual oversight and seamlessly integrate them. As highlighted previously, executing experiments that involve large-scale data fusion can be resource intensive and inefficient. However, we have found a way to circumvent this challenge using a wireless signal emulator, WirelessDT [179], built on the Unreal Engine 5 (UE5) platform. This robust digital twin platform allows us to design, refine, and test our algorithms with greater ease and efficiency.

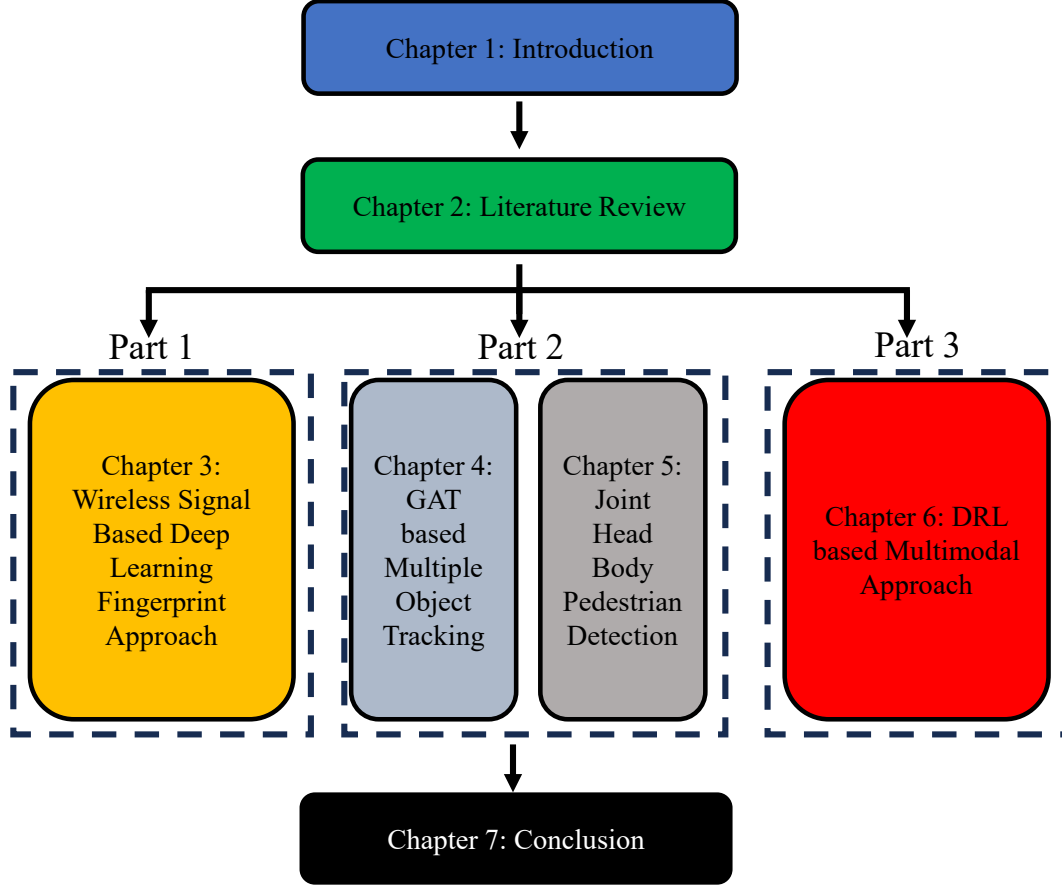


FIGURE 1.4. The structure of this paper.

1.3 Contribution

Based on the research gap and the motivation that inspired us, we made several improvements to indoor positioning approaches. The contribution of this paper is threefold:

- We introduce GauPro, an innovative Gaussian Kernel-based Bayesian algorithm tailored to refine the accuracy of fingerprinting IPS. What sets GauPro apart is its dynamic adaptation; it modifies the prior probabilities of each checkpoint based on the user's current position and movement direction. Our comprehensive solution encompasses both a client application and a back-end server to support the

implementation. GauPro outperforms conventional wireless IPS methods, marking a significant advancement in this domain.

- We have pioneered a tracker rooted in Graph Attention Networks (GATs). This tracker capitalises on low-score detections to aggressively create tracklets. Subsequently, GATs are employed to refine and split these tracklets. Our detection methodology markedly diminishes false negatives in contrast to standard detection techniques. Additionally, our GAT-based tracker not only enhances tracking precision but also excels in preserving individual identities throughout the tracking process.

Furthermore, we introduce a cutting-edge joint head-body detector designed to simultaneously detect a pedestrian’s head and body. This stands in contrast to traditional detectors that focus primarily on identifying the entire body. Our innovation shines particularly in crowded scenarios. When observed through surveillance cameras, individuals often face significant occlusion, rendering their bodies obscured. However, because of the top-down perspective of these cameras, their heads remain distinctly visible. Leveraging this insight and building upon prior research, we developed an anchor-free detector that identifies both an individual’s head and the associated body.

- Merging the strengths of both wireless and vision-based IPS approaches, we have crafted a Reinforcement Learning (RL)-driven multimodal method to seamlessly integrate diverse data sources for precise indoor pedestrian identity matching. While cameras adeptly pinpoint individuals within a scene, they fall short in discerning personal identities (i.e., they cannot directly link an individual to their specific mobile device). This shifts the indoor positioning challenge from mere location estimation to a more intricate identity matching problem.

Our primary objective is to devise an optimal strategy for aligning detection/-tracking outcomes with the corresponding user’s receiving device. Our innovative RL-based technique is adept at discerning and correlating users’ motion features across different device endpoints, facilitating automatic matching. To validate our

approach, we conducted rigorous testing within a digital twin platform. Empirical results underscore the superiority of our RL agent, which consistently eclipses conventional multimodal strategies, especially in densely populated settings.

We show the structure of this article in Figure 1.4. The rest of the paper is structured as follows. In Chapter 2, we give an extensive background review of related technologies and existing researches, including the wireless positioning technologies, vision-based detection and tracking approaches, and multimodal indoor positioning approaches. In Chapter 3, we introduce the proposed wireless fingerprinting IPS method. In Chapter 4 and Chapter 5, we introduce our proposed GAT-based tracker and the joint head & body detector. In Chapter 6, we introduce our DRL-based multimodal approach. We summarise and conclude our work in Chapter 7.

CHAPTER 2

Literature Review

In this section, we present an exhaustive literature review related to the research themes of this paper. We begin with an overview of indoor positioning, highlighting various approaches and methodologies. Broadly, the technologies employed can be categorised into two types: wireless technologies and vision-based technologies.

For wireless technologies, we discuss the primary wireless signals prevalent in indoor positioning scenarios, such as Wi-Fi and Bluetooth. While these are the dominant technologies, we also discuss the less conventional ones like UWB, geomagnetic, and RFID.

Turning our attention to vision-based technologies, we provide a concise summary of the evolution of computer vision technology. The focus in this segment is on methods underpinned by deep learning. We discuss image detection and video tracking as distinct entities, leading to an exploration of studies centred on vision-based positioning.

In conclusion of our review of the literature, we explore research on multimodal indoor positioning. The strength of these approaches lies in their ability to amalgamate the benefits of various technologies, mitigating the limitations inherent to each. Vision-based methods, for instance, offer remarkable localisation accuracy when calibrated correctly. The precise location of a detected pedestrian in the world can be easily computed using the camera matrix. However, these methods face challenges such as maintaining user identity and detecting individuals obscured by indoor obstacles, such as pillars or walls. These challenges can be overcome when coupled with wireless signal technologies, which excel at consistently tracking a device address, regardless of line of sight. A quintessential example of multimodal

indoor positioning involves leveraging cameras for precise user localisation while employing Wi-Fi signals to maintain user identity across various rooms.

2.1 Indoor Positioning Overview

IPS use sensors and communication technologies to locate objects in indoor environments, which have been studied for decades. It can be very useful in many different ways, such as indoor navigation for large shopping centres or train stations, public surveillance, and crowd control. The ability to position a user in an indoor environment is the most important and fundamental element of contextual information. We introduce different approaches and technologies for IPS. The word 'approaches' represents abstract approaches people use for locating users that do not strictly depend on any physical media like wireless signals or vision signals.

2.2 Indoor Positioning Approaches

The approaches involving indoor positioning vary greatly. Traditionally, there are mainly three ways of estimating the position of a user given a number of different reference devices. Multilateration & Triangulation and Fingerprinting. The first two approaches are more commonly used with wireless signals that can be detected with some specific receiving ends, such as users' smartphones or some configured chips. By analysing the signal one can estimate the distance between the user and the referencing device. For example, received signal strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Angle of Arrival (AoA) are some different measurements for wireless signals that can be used for indoor positioning. We now introduce some of them in detail to provide a better background introduction for our study.

RSS or Received Signal Strength Indicator (RSSI) is a measurement used in wireless communication to determine the signal power received by the user, which is usually expressed in decibels (dB). In the context of IPS, RSSI can be used to estimate the distance between

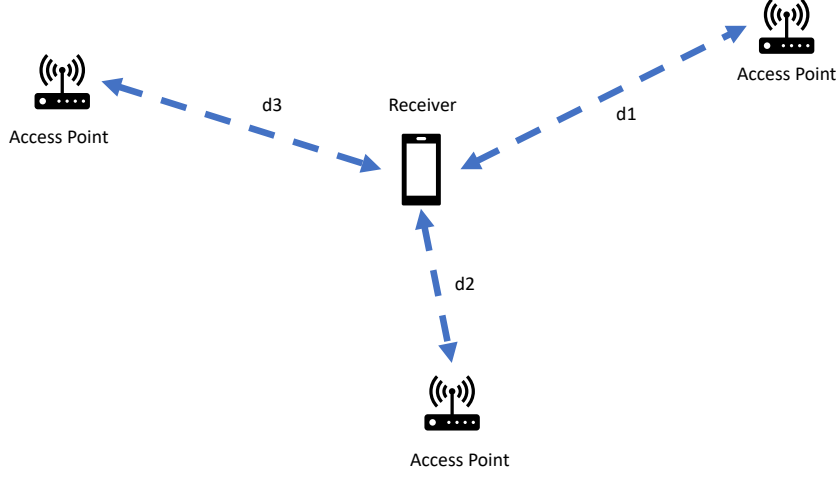


FIGURE 2.1. Example of using RSSIs to locate the user given multiple Access points.

the receiving devices (e.g. smartphone or tablet) and the transmitters (WiFi access point or Bluetooth beacon), since the signal strength gets weaker as the distance for the transmission increases.

The distance between the transmitter and the receiving device can be determined using a signal propagation model [72]. An example is to use the following equation:

$$d = d_0 \times 10^{\frac{PL(d_0) - RSSI}{10n}} \quad (2.1)$$

where d_0 is the reference distance from the transmitter, which is usually 1 metre. $PL(d_0)$ is the RSS at the reference distance. RSSI represents the received signal strength indicator at the unknown distance and n is the path loss exponent. This equation allows us to calculate the distance d between the unknown location and the transmitter based on RSSI.

Figure 2.1 shows a simple example of using RSSI to locate the receiving device, which requires the use of multilateration. We will introduce this approach in detail in Section 2.2.1.1.

The Time of Arrival (ToA) is another common measurement that records the signal travel time from the transmitter to the receiver. Assuming that there are no obstacles between the transmitter and the receiver, we can easily compute the distance based on the ToA and the speed of light ($c = 3 \times 10^8 m/s$). Similarly to the RSSI approach, we can draw circles from the transmitter and use the distance as a radius. Usually, two access points are enough to solve the location of the receiving device. However, since distance alone may not be sufficient to determine the location, we use ToA to assist in the process.

Angle of Arrival (AoA) is a measurement that describes the angle of the received signal corresponding to the transmitter. Given multiple transmitters, we first define a reference line to measure the angle of the received signal. As shown in Figure 2.2, θ_1 and θ_2 are the angles of the receiving device that correspond to different access points. With these two angles, we can compute the intersection point of the direction line, which results in positioning the receiver device (i.e., intersection point P).

We have summarised some typical IPS approaches that can make use of this information, including traditional and modern ones, in Table 2.1. More details are discussed in the following section.

TABLE 2.1. Traditional and deep learning Indoor Localization Approaches

Approach	Description	Advantages	Limitations
Triangulation	Uses the geometric properties of triangles to estimate positions based on signal measurements.	Simple to implement; moderate accuracy.	Requires line-of-sight; affected by multipath.
Fingerprinting	Matches observed signal characteristics with a pre-mapped database.	High accuracy in stable environments.	Requires extensive initial mapping; maintenance.
Convolutional Neural Networks (CNNs)	Utilizes deep learning for feature extraction and localization from signal patterns.	High accuracy; good at handling variability.	Requires large labeled datasets; computational resources.
Recurrent Neural Networks (RNNs)	Optimized for sequential data, making it suitable for time-series signal processing.	Effective in dynamic environments.	Complex model architecture; training challenges.

2.2.1 Traditional Approaches

In this section, we introduce some traditional approaches for indoor positioning tasks.

2.2.1.1 Multilateration and Triangulation

According to [3], the multilateration problem can be solved using linear algebraic methods, which have low computational complexity and can be applied to real-time application in wireless sensor nodes. In this paper, the author discussed the case where the multilateration problem cannot be solved due to too short distances, leading to no real solutions mathematically. In such cases, the signal strength is strong enough so that we can use the approximation for the solution and still get very accurate results. The author conducted experiments with Ultra Wide Bandwidth (UWB) transceivers and has proved the method to be highly effective and low error.

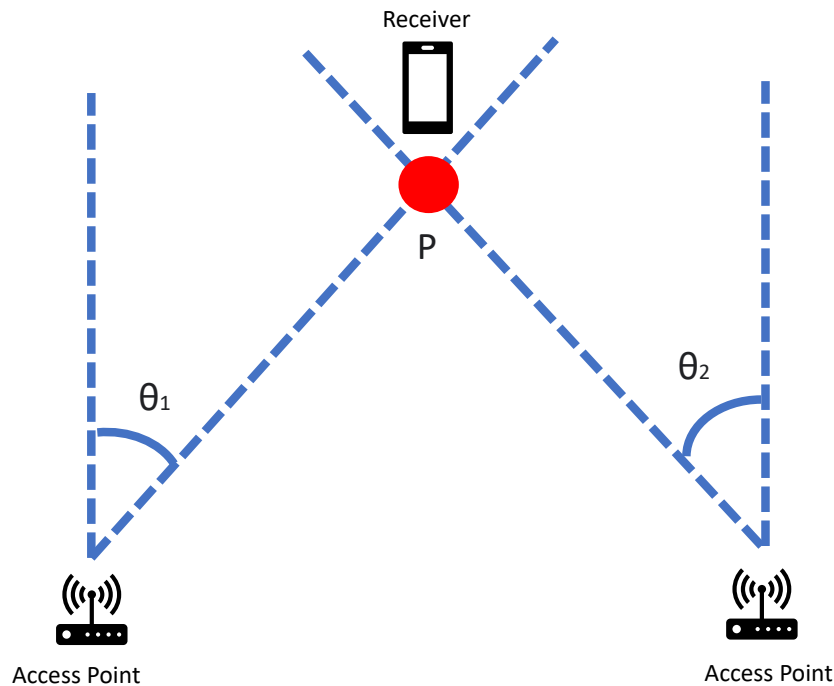


FIGURE 2.2. Example of using AoAs to locate the user given multiple Access points.

Jeffrey and Gaetano [77] introduced a novel approach and survey on estimating the physical location of the mobile device given a wireless signal environment. It makes use of three different approaches to measure the distance: Direction, which equals the Angle of Arrival previously described, Time-of-Flight (ToF), and Attention, which describes the intensity of an emitted signal, and can be considered equal to RSSI. This paper categorises location-sensing approaches into three main types: triangulation, scene analysis, and proximity. Triangulation uses the geometric properties of triangles to compute the location of objects. Scene analysis uses features of a scene observed from a particular vantage point to draw conclusions about the location of the observer or of objects in the scene, while proximity involves determining when an object is “near” a known location. This paper suggests important future research directions for IPS by providing a thorough analysis and evaluation of the many options in the domain of location systems for ubiquitous computing.

Later in the years, Yapeng et al. [153] discuss the use of triangulation methods that use time, time difference, AoA, and signal strength for Bluetooth indoor positioning problems. This paper presents a novel approach, which utilises a new Bluetooth standard to retrieve the RSSI readings. A new signal propagation model is then proposed, and the triangulation-based positioning scheme is defined in order to achieve the user localisation. The experiments were carried out in a simple indoor environment and the results were satisfactory at that time.

Another paper by Bumgon et al. [26] presents a novel indoor localisation method for mobile devices that also uses RSS to estimate the distance. A least squared error (LSE) triangulation is used to determine the position of the user. In this paper, the user discussed the challenges of the indoor channel characteristics, which can fluctuate significantly due to various types of perturbations like electronic devices or people walking inside the room. The use of a reference point (RP) was proposed to compensate for the challenge. The author presents two main contributions: cross-monitoring AP eliminates the need for deploying RPs, as other APs can play the dual role of regular AP and RP. The second contribution of this paper is the weighted triangulation that can improve the accuracy of the LSE estimation of the mobile device's position.

“Rfid Indoor Location Identification for Construction Projects” [11] is another example of using RSSI to compute the location of a mobile receiver. Instead of Wi-Fi or Bluetooth, the author of this paper uses Radio Frequency Identification (RFID) technology for construction projects. The authors discuss the challenges of indoor location identification, such as the fluctuation of signal strength due to various factors. To overcome this, they proposed the use of reference tags to compensate for signal instability. The authors present a system that includes several entities, such as RFID signal, mobile RFID reader, material tag, and zone. The authors conducted experiments in two case studies, one at the construction site of the Centre for Structural and Functional Genomics at Concordia University and another in the Construction Automation Lab at the same university. The results showed that the proposed method can provide a reliable indoor location for workers and materials at construction sites.

A 3-dimensional triangulation scheme [61] was proposed to improve the traditional triangulation by introducing the third dimension in space to boost the distance accuracy and mitigating

the influence of the “malicious beacon (transmitter)”, which can be a faulted or bad beacon. The authors conducted experiments in an indoor environment without obstacles, using a physical web transmitter device attached to a wall and an Android device as a Bluetooth receiver. The results showed that their proposed method can provide a reliable indoor location with improved accuracy.

2.2.1.2 Fingerprinting

Another famous approach for IPS is the fingerprint method, also known as the fingerprinting or radio map-based method. This method involves two main stages: the offline training phase (or map construction) and the online positioning phase.

For the offline training phase, a radio map of the indoor environment is created with fingerprint collection. More specifically, the researcher walk through the indoor space and collect RSSI values from various known APs. The collected set of RSSI values are referred to as reference points or fingerprints, consisting of the unique combination of signal strengths from these different transmitters. These data are then stored in a database to form the radio map and the offline training phase is completed.

As for the online positioning phase, the mobile device keeps receiving RSSI values in real-time, and these values are compared to the radio map stored in the database to find the best match. The locations corresponding to the best match are considered to be the estimated position of the device. A lot of matching algorithms can be used here, such as the K-nearest neighbour method, Bayesian methods, or machine learning methods.

The advantage of the fingerprint approach is that it can provide high localisation accuracy without requiring knowledge of the signal propagation model, which means that we do not need to compute the distance according to RSSI or AoA information. The only thing we need is the unique combination of RSSI values. However, the drawbacks are also obvious, as they can easily be affected by changing the environment, such as increasing, decreasing, or moving the APs. And the offline fingerprint collection process is labour intensive. Figure 2.3 shows a

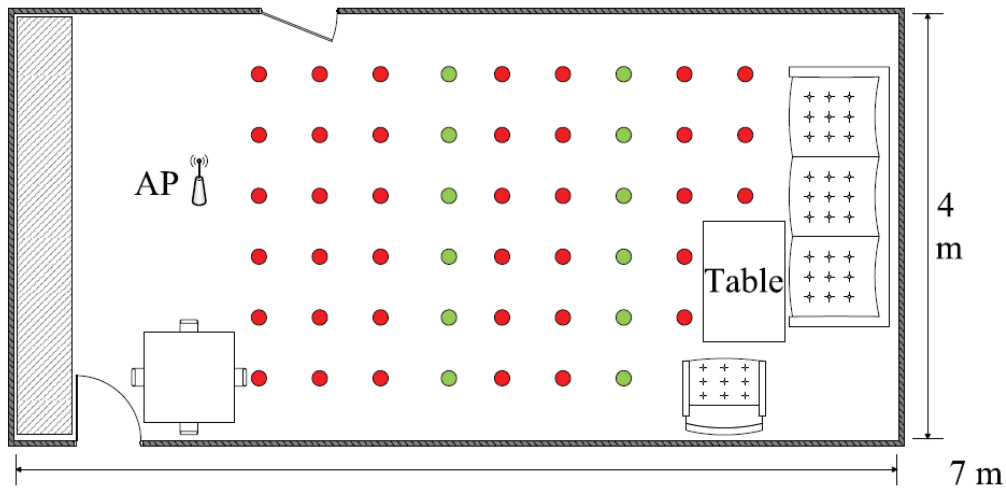


FIGURE 2.3. Example of indoor radio map for the fingerprinting methods using a single Access Point. Red dots are the sample location used for map construction (offline training phase), green dots are used for testing.

radio map example of the fingerprinting method constructed in a room that is four by seven metres.

Youssef et al. proposed an early fingerprint positioning method [1] in the year 2002. They present the design, implementation, and evaluation of two novel probabilistic indoor location determination approaches: the Joint Clustering approach and the Incremental Triangulation approach. Both approaches depend on the RSSI distribution of the signal to locate the receiver. The Joint Clustering approach has better accuracy but requires more operation, while the Incremental Triangulation method is much faster.

Zheng et al. [177] present a novel method for device-free localisation using fine-grained subcarrier information with low human effort. To overcome the challenge of fluctuation of signal strength due to various factors, they propose the use of a multidimensional scaling (MDS) approach to create a high-dimensional space where fingerprints are represented by points and their mutual distances are preserved. The authors conducted experiments in an indoor environment, using Wi-Fi signals and accelerometer readings to detect the number of steps. The results showed that their proposed method can provide a reliable indoor location with improved accuracy.

DeepFi [163] proposed by Xuyu et al. discussed the a fingerprint method using Channel State Information (CSI) with the help of Deep Learning (DL). The authors of DeepFi proposed to use weights in the deep network to represent fingerprints and incorporated a greedy learning algorithm for weight training to reduce complexity. A mean error of 0.95 metres with a single access point can be achieved within a living-room environment.

A paper by Ali et al. [10] summarised the modern wireless local area network (WLAN) IPS methods and deployment challenges. They categorise the techniques into three parts: conventional approaches, state-of-the-art solutions, and deployment challenges. Many conventional approaches include deterministic methods, such as KNN and Median filtering, use a single RSS fingerprint for localisation; probabilistic approaches, like Maximum A Posteriori (MAP) and Maximum Likelihood (ML), use the ensemble of RSS fingerprints to provide statistical characteristics of the area; Pattern recognition methods are based on classifiers, such as Canonical Correlation Analysis (CCA), Support Vector Machine (SVM), and Neural Networks (NN), use machine learning algorithms for localisation, are included.

More advanced solutions are refinements and enhancements to conventional approaches. Kernel Density Estimation (KDE) Method use a kernel function to estimate empirical distributions of the fingerprints. It can also be used for weight computations. Principal Component Analysis (PCA) method is an alternative approach for computing the conditional probability of the location given the received online measurement. Other state-of-the-art methods such as KL-Divergence Method and Geometry-based Localisation are also discussed in the paper.

2.2.2 Deep Learning Approaches

Deep Learning (DL) has been a hot topic recently in almost all research fields, including indoor positioning. It can be used with both multilateration methods and fingerprinting approaches. Deep Learning are suitable for IPS for the following reasons:

- **Automatic Feature Extraction:** DL techniques are capable of automatically extracting necessary features from the data, eliminating the need for manual feature

engineering. This is especially beneficial for dealing with high-dimensional features required for radio frequency and image fingerprinting, which are not easily extractable features.

- **Efficient Processing of Massive Data:** DL can process large volumes of raw data effectively and efficiently as its performance improves with more training samples, allowing it to use unlimited fingerprint data for training.
- **Speed and Scalability:** Due to their parametric nature, the computational complexity of DL algorithms does not depend on the size of the data set, making them fast and scalable. DL methods, in combination with GPU technology, offer near-instantaneous inference latency, suitable for real-time positioning applications.
- **Ability to Handle Complex Non-linear Boundaries:** DL is effective for classification and regression problems with complex, nonlinear boundaries, such as distinguishing between close spatial locations in fingerprinting that are separated by a few centimetres or less
- **Applicability for Transfer Learning:** DL can effectively utilize transfer learning, where knowledge from pre-trained networks is used to reduce data collection and the training efforts. This can reduce the overall cost of fingerprinting approaches. Furthermore, unsupervised and semi-supervised DL methods can be successful when fingerprint data are scarce or unlabelled, further reducing the cost.

2.2.3 Concepts in Deep Learning

We now introduce a few popular concepts in deep learning.

FFN [97] is one of the first ideas of DL and was proposed by Kurt Hornik, Maxwell Stinchcombe, and Halbert White; they discuss the concept of multilayer feedforward networks and their ability to serve as universal approximators. The authors present several theorems and proofs to support the arguments, and argue that there are no theoretical constraints for the success of FFNs, the lack of success is due to inadequate learning and the lack of hidden units. Their arguments have proven to be correct, thanks to the explosion development of hardware and computational resources these days.

CNN [166] is also an early concept of DL for image & document recognition. It was discussed by Yann Lecun et al. back in the 1990s, inspired by [96]. This paper demonstrated that a CNN model can aggregate simpler features into more complicated features and can be used for handwritten character recognition. They test a CNN with the MNIST database [99], consisting of images of handwritten digits ranging from 0 to 9. The model can predict the identity of the digits based on the appearance of the image. CNN has also grown greatly and we will discuss it in Section 2.3.2.

Recurrent Neural Network (RNN) [48] is a type of artificial neural network designed to recognise patterns in sequences of data, such as text, genomes, handwriting, or spoken words. It focusses on exploring the temporal information of the input sequences. It can process sequential data by maintaining a form of memory about previous inputs in the sequence. This is particularly useful for tasks such as language modelling, time-series prediction, and machine translation. The word “Recurrent” indicates loops of the model structure. Modern RNNs such as the Long-Short-Term Memory (LSTM) [134] are very handy when processing the IPS request since it can analyse the movement of users for a period of time rather than just a frame.

2.2.4 Example Research Works

PhaseFi [162] was presented by Xuyu et al. in late 2015 to propose a system that focusses on leveraging phase fingerprinting combined with deep learning techniques. The core idea behind PhaseFi is to extract and calibrate phase information from the antennas of the Intel WiFi Link 5300 NIC. The data of this phase is then processed using a deep neural network with three hidden layers during the offline stage. The network is trained using a greedy learning algorithm, where each subnetwork between two continuous layers forms a Restricted Boltzmann Machine (RBM) that is approximately solved using a Contrastive Divergence-1 (CD-1) algorithm.

There are two stages of the proposed DNN methodology: Offline Training Stage and Online Testing Stage. For the offline stage, PhaseFi incorporates deep learning to train weights and

then stores them to build a feature-based fingerprint database. Firstly, a deep network with one input layer and three hidden layers is used for pre-training. The joint distribution for the deep network is maximised by using a greedy learning algorithm. This algorithm trains the weights layer by layer using a stack of RBMs to reduce complexity. After the pretraining stage, the calibrated reconstructed phase data is obtained by unrolling the deep network with forward propagation. The back-propagation algorithm is used to train all weights in the deep network by computing the error between the input calibrated phase data and the reconstructed calibrated phase data.

During the online stage, a Bayesian method based on Radial Basis Function (RBF) is employed for location estimation. A probabilistic method is developed to estimate the location of the mobile device based on the fingerprint database and new calibrated phase data. The posteriori probability is calculated on the basis of Bayes law. The position of the mobile device is then computed as a weighted average of all reference locations.

The performance of PhaseFi was validated in two distinct indoor environments. The results demonstrated that PhaseFi outperforms three benchmark schemes that are based on CSI or RSS. The paper concludes by emphasising the advantages of using phase information, which is more robust than amplitude, especially in cluttered indoor environments.

DeepFi [33, 163] introduced a deep learning application for indoor localisation using transmission channel quality metrics, specifically RSSI and channel state information (CSI). The paper presents a method in which a room is partitioned into two-dimensional blocks, with each block treated as a class. The localisation task is then formulated as a classification problem. The authors explored two types of DNNs: Multi-Layer Perceptrons (MLP) and 1-Dimensional Convolutional Neural Networks (1-D CNN). The localisation problem was treated as a classification task with $M \times N$ classes, where each class corresponds to a block in the room. The authors designed multiple architectures for both MLP and 1-D CNN and evaluated their performance based on prediction accuracy and network complexity. For MLP, The authors designed eight different MLP architectures. The best performing architecture had five layers with varying numbers of neurones in each layer. This architecture achieved good prediction accuracy with relatively lower network complexity. For 1-D CNN, the architectures

varied in the number of convolution blocks and kernel sizes. The best-performing architecture used three hidden convolution blocks with a kernel size 16.

The authors conducted experiments in a room filled with furniture and obstacles to increase the environmental complexity to evaluate the proposed system. The room was divided into 2×8 blocks, with each block representing a quantised location. A personal computer equipped with an Intel 5300 network interface card was used as a receiver and a TP-Link WA901-D wireless network router served as the transmitter. Three individuals with varying body shapes were chosen for the evaluation, and they stood at training points for one minute while the network receiver captured packets. The extracted RSS and CSI data was then used for training and testing.

The advantages of the proposed system are that the use of DNNs allows for the automatic extraction of relevant features from the data, potentially improving accuracy. The system is also robust to complex indoor environments. However, the accuracy of the system might be affected by rapid changes in the environment or the presence of multiple individuals. In summary, the paper presents a novel approach to indoor localisation by combining RSS and CSI data and processing them using deep neural networks. The method shows promise, but its performance can be influenced by various factors in real-world scenarios.

ConFi introduced by Chen et al. present a novel approach to indoor Wi-Fi localisation using CSI and Convolutional Neural Networks (CNNs) [67]. The objective of this paper is to leverage the power of CNNs to process CSI from multiple antennas for indoor localisation. The input to the CNN is the CSI from multiple antennas, which is organised into matrices that represent the CSI over time and frequency domains. The CNN architecture comprises three convolutional layers and two fully connected layers. The convolutional layers are designed to capture the spatial and temporal features of the CSI. The output layer uses the softmax function, allowing the outputs of all neurones to be summed to one. This means that each output can be interpreted as the probability that a device is at a particular Reference Point (RP). The network is trained using the cross-entropy loss function with a regularisation term to avoid overfitting.

During the localisation phase, the CSI feature image of the target device is input into the model. The model then outputs a probability for each RP, indicating the probability that the device is at that location. The final location of the device is estimated using a probability-weighted centroid method, typically considering the top 3 RPs with the highest output values.

The experiments were carried out in an indoor environment measuring approximately 16.3 m by 17.3 m with various walls and reflectors. A laptop with a wireless network card was used as the target device, while a Wi-Fi router with three antennas served as the Access Point (AP). The paper compares ConFi's performance with other neural network-based methods and concludes that CNNs are more effective for localisation than fully connected neural networks, which outperform most existing methods, demonstrating the power of CNNs in pattern recognition.

SDR-Fi [52] is another example of a Deep Learning-based IPS approach. The authors describe a system where the received signals are digitised and the digitised data are processed through software. This is useful because it can provide full control of the receiver module, which is advantageous for fast prototyping. The author claims to be the first to propose a software-defined radio (SDR) receiver for CSI measurement extraction. They use two types of neural networks, Feed Forward Network (FFN) and Convolutional Neural Network (CNN) to estimate the location via the CSI measurements.

The architecture of the proposed FFN is a specific type of pattern recognition network called PatternNet [155] from MATLAB. The FFN is trained on preprocessed CSI data to classify input into N target classes, where each class represents a specific location on the radio map. It learns the patterns and relationships in the CSI data to predict the location. CNN is used to process 1D flattened CSI-based input. The convolutional layers in the CNN are adept at extracting local features from the CSI data, which are crucial for accurate positioning. CNN's hierarchical structure allows it to capture both low- and high-level features from the CSI data, which are then used to predict the indoor position.

The paper evaluates the performance of both FFN and CNN in various indoor scenarios. The results indicate that the CNN outperforms the FFN in most scenarios, especially when the

number of locations is large. CNN's ability to extract hierarchical features from the CSI data gives it an edge over FFN in accurately predicting indoor positions. In summary, the paper uses the pattern recognition capabilities of FFN and the feature extraction prowess of CNN to interpret CSI measurements for indoor positioning. By training these networks on CSI data, the system can predict indoor positions with high accuracy, demonstrating the potential of deep learning to improve IPS.

2.3 Indoor Positioning Technologies

Table 2.2 is a list of technologies that are discussed in this section. People choose different technologies in different scenarios. The details are introduced in the following sections.

TABLE 2.2. Indoor Positioning Technologies

Technology	Key Characteristics	Advantages	Limitations
Wi-Fi	Utilizes signal strength from Wi-Fi access points.	Widely available; moderate accuracy.	Affected by environmental factors; privacy concerns.
Bluetooth (BLE)	Uses signal strength from BLE beacons.	Low energy; easy deployment.	Requires dense beacon deployment; medium accuracy.
RFID	Employs electromagnetic fields to automatically identify and track tags attached to objects.	High accuracy; quick scanning.	Requires tag and reader; limited range.
UWB	Uses time-of-flight of ultra-wideband signals for precise localisation.	Very high accuracy; works well in multi-path environments.	Requires UWB infrastructure; higher cost.
IMU	Incorporates accelerometers, gyroscopes, and magnetometers.	Self-contained; good for step counting.	Prone to drift; requires initial position.
Computer Vision (CV)	Utilizes cameras and image processing.	High accuracy; leverages existing infrastructure.	Privacy concerns; lighting conditions.

2.3.1 Wireless Indoor Positioning

Wireless indoor positioning is a technology used to locate objects or people inside a building using radio waves, magnetic fields, acoustic signals, or other sensory information collected by mobile devices. Due to the lack of precision of the Global Position System (GPS) for indoor scenarios, they can help locate users.

Wi-Fi signals are the most commonly used in IPS due to their widespread availability in indoor environments. As introduced above, many successful IPS systems are Wi-Fi based. Many useful measurements can be used, such as RSSI, AoA, ToA, etc.

Bluetooth Low Energy (BLE) is similar to Wi-Fi but with a lower power consumption, which makes it ideal for battery-powered devices that need to operate for long periods on a single charge. Also, Wi-Fi typically has a longer range than BLE. A Wi-Fi network can cover a large house or office building, while BLE is typically used for short-range connections of up to 100 metres. Another important reason that BLE is appropriate for IPS is that BLE devices are generally less expensive to manufacture than Wi-Fi devices, which can make them a more cost-effective choice for certain applications. Chao et al. [31] discuss a novel approach to indoor localisation using BLE signals. The system uses a denoising autoencoder, a type of artificial neural network, to process the Received Signal Strength Indicator (RSSI) from BLE beacons. The system does not require any infrastructure other than a smartphone and a magnetic fingerprint map of the building. Future work will involve combining a WLAN RSSI method with the system to effectively localise the initial position within 10 metres.

Radio Frequency Identification (RFID) is a wireless communication technology that uses radio waves to identify and track objects. This technology is used in many industries for a variety of applications, including inventory management, asset tracking, personnel tracking, controlling access to restricted areas, and prevention of retail theft. Each object to be tracked is equipped with an RFID tag, which is a tiny device that contains a microchip and an antenna. The microchip stores information about the object, such as its identification number, and the antenna allows the tag to communicate with RFID readers. An RFID reader is a device that emits radio waves and receives signals from the RFID tags. When a tag comes within range of

a reader, the reader can read the information stored on the tag microchip. This technology can also be used for IPS, but the use cases are generally different than the technologies described above, as it is used more in factory for cargo localisation.

Guangyao et al. [64] discusses an improved mechanism for indoor localisation using RFID technology. The proposed mechanism in the paper can reduce the computing load by reducing the number of candidates for neighbouring tags, thus reducing the latency in locating. The mechanism obtains the target coordinate by adding the computed coordinate result with the average error range of all neighbouring tags, resulting in higher locating accuracy. The proposed mechanism was found to significantly reduce the estimated error compared to the existing system in the same indoor environment.

Ultra-wideband (UWB) is a wireless communication protocol that uses a very low energy level for short-range, high-bandwidth communications over a large portion of the radio spectrum. It has recently gained popularity in the field of indoor positioning systems due to its high accuracy, often within a few centimetres. This is because UWB signals have a very short pulse duration, which allows for highly accurate TOA measurements. UWB signals can penetrate through walls and other obstacles, making them suitable for indoor environments where signals might be blocked. The signal is also less susceptible to multipath interference and has very low power consumption. However, despite its advantages, UWB also has some challenges. The infrastructure setup for UWB can be complex and costly, especially for large areas. Furthermore, the signal strength of UWB decreases rapidly with distance, which can limit its range.

Segura et al. [107] proposed a novel method of indoor navigation using UWB, which allows accurate mobile robot (MR) navigation in indoor environments. The navigation system is composed of two subsystems: the localisation system and the MR control system. The statistical analysis shows that the UWB approach achieves a prime positional accuracy since the 95% of the measurements result in errors below 20 cm. The UWB navigation system outperforms odometry navigation and avoids the problem of accumulated errors.

2.3.2 Computer Vision Indoor Positioning

Computer Vision (CV) is a significant technology that can be employed in IPS due to the extensive use of surveillance cameras in public spaces. With CV, we can pinpoint pedestrian locations with the camera, thereby directly obtaining the user's position with minimal error. Unlike wireless signal methods, if the location is successfully determined, the error distance can be easily maintained within 50 cm, provided that the camera is calibrated correctly.

However, CV approaches also have their drawbacks. They can be easily influenced by false positives and are prone to occlusion issues. Additionally, unlike wireless approaches where personal IDs can be directly obtained (considered to be the devices' MAC addresses), CV methods do not readily provide this information.

For CV-based IPS, the problem can be considered as the MOT problem. We use the cameras to track pedestrians and keep a record of their location in real-time. The general solution is to first detect and then track the information. In the following sections, we are going to introduce these two aspects of CV respectively.

2.3.2.1 History of Object Detection

In the early days of computer vision, the **Template Matching** [147] was used for image detection. It involves sliding a template over a large image to find matches. The template can be a small image that contains different kinds of objects. This method was not robust to changes in scale, rotation, or lighting. However, the idea was inspiring and was adopted by some modern deep-learning techniques which we will introduce later.

The next advancement came with the development of **feature detection algorithms**, such as Scale-Invariant Feature Transform (SIFT) [60] and Speeded Up Robust Features (SURF) [71]. SIFT extracts object keypoints from a set of reference images and stores them in a database. To recognise objects in a new image, the features from the new image are compared to those stored in the database, and the best matches are found by calculating the Euclidean distance of the feature vectors. From the full set of matches, subsets of keypoints that agree

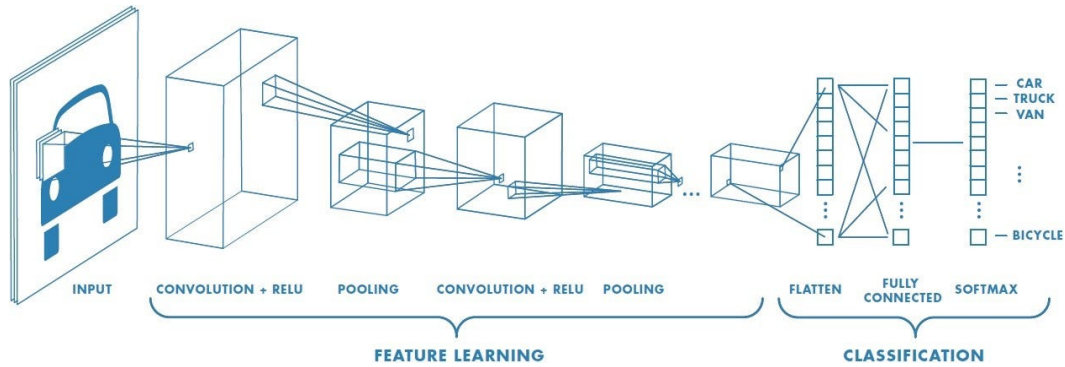


FIGURE 2.4. The structure of a typical Convolutional Neural Networks (CNN) by Sumit Saha [39], which starts with the feature extraction layers and is followed by fully connected classification layers.

on the object and its location, scale, and orientation in the new image are identified to filter out good matches. SURF is a robust local feature detector, first presented by Herbert Bay et al. in 2006, which is faster to compute and still provides robust features that are invariant to scale, orientation, and affine distortions. It was implemented by Anqi et al. and outperformed SIFT due to its speed and robustness. The authors developed a full-fledged SURF framework using MATLAB, based solely on the original SURF authors' publications. They made several assumptions in their implementation, which resulted in poorer recognition rates than the original authors' version. However, they gained a tremendous amount of useful information on SURF and image features in general by writing all the algorithms themselves.

Histogram of Oriented Gradients (HOG) [115] descriptor was introduced for the detection of objects, particularly for the detection of pedestrians. It involved counting the occurrences of gradient orientation in the localised portions of an image. The gradient of the image is computed by filtering the image with a simple Sobel filter to get the first derivative in the horizontal and vertical directions. The gradient directions are then binned into a histogram. Each bin covers a range of directions, and the magnitude of the gradient is added to the corresponding bin. This creates a Histogram of Oriented Gradients for each cell of pixels. When the HOG descriptors are created by normalising the histograms in overlapping blocks, it is passed to a machine learning algorithm, such as a Support Vector Machine (SVM), to perform the object detection.

Then we enter the **Deep Learning** era. The Convolutional Neural Networks (CNNs) are the most commonly used neural network. CNNs could learn hierarchical representations of images, which makes them very effective for this task. As shown in Figure 2.4, the fundamental layer of a CNN is the convolutional layer. In this layer, several filters (also known as kernels) are applied to the input data to produce feature maps. These parameters of these filters are learnt during the training process. The convolution operation allows CNNs to automatically and adaptively learn spatial hierarchies of features.

Then there is the pooling layer, which reduces the dimensionality of the feature maps produced by the convolution layer. The most common type of pooling is max-pooling, where the maximum value is taken in a patch of the feature map. The final step of CNN is the Fully-Connected (FC) layer. After going through several convolutional and pooling layers, the resulting feature vectors are flattened and fed to the FC layers to obtain the final predicting results of the input image. Neurones in a fully connected layer have connections to all activations in the previous layer, as seen in regular Neural Networks. They use these features to classify the input image into various classes based on the training data set. Backpropagation is used to train CNNs.

Until recent days, CNN-based approaches can still achieve state-of-the-art performance for object detection. In this paper, we also propose a novel pedestrian detector, the details can be found in Section 5. Now, we first introduce some popular modern DL-based image detectors.

2.3.2.2 Deep Learning Based CV Approaches

For example, **Faster-RCNN** [135] is a popular model for object detection because it can provide excellent accuracy and speed and can be trained end-to-end. Before Faster-RCNN was proposed, the bottleneck for image detection was the first step - to extract region of interest from the image for CNNs to classify. This paper proposed a novel Region Proposal Network (RPN) to simultaneously predict object bounds and objectness scores at each position. After the region proposals are generated, a Region of Interest (RoI) pooling layer is used to extract a fixed-length feature vector from each proposal. The processed features are then sent to CNNs to predict the object type as well as the four coordinates of the bounding box. The network is

trained with a multi-task loss. Parameters are learnt using stochastic gradient descent. The parameters of the region proposal network and the convolutional layers are shared for faster and end-to-end training.

Mask-RCNN [92] extends Faster R-CNN by adding a branch to predict segmentation masks on each RoI, in parallel with the existing branch for classification and bounding box regression. It also introduces a new layer, RoIAlign, that fixes the location misalignment caused by quantisation in RoI pooling in Faster-RCNN.

YOLO [90], which stands for “You Only Look Once” is a real-time object detection system that represents a different approach to object detection. Previous methods see object detection problems as a classification problem and apply the CNNs to different RoIs; it sees object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Divide an input image into $n \times n$ grids containing potential objects. And for each grid cell, it predicts a fixed number of bounding boxes. This approach is extremely fast compared to traditional two-step object detection algorithms. There have been several versions of YOLO, each improving on the last. From YOLOv1, YOLOv2 to YOLOv7, each version introduces improvements in terms of speed, accuracy, or both.

DETR [117] was proposed by Nicolas et al. that uses a transformer [17] encoder-decoder architecture to perform object detection. This approach treats object detection as a set-prediction problem. It outputs a fixed-size set of bounding boxes and class labels directly, eliminating the need for anchor boxes and nonmaximum suppression steps. The transformer architecture used by DETR was primarily used for natural language processing tasks, but later on it appears that it can also help to model the relationships between objects and their context in the image. The system is also end-to-end trainable, and it achieves competitive results on object detection benchmarks (like COCO) while operating in real-time. It performs particularly well on large objects and objects that require context for recognition.

The **Swin Transformer** [175] is another type of vision transformer proposed to handle image recognition tasks introduced by Liu et al. A novel mechanism called “window-based self-attention” was applied to the object detection task. This mechanism partitions the image

into non-overlapping windows and computes self-attention within each window for the local context by sliding the window line by line down the image. This is a departure from the standard Transformer, which computes self-attention globally and can be computationally expensive for large images. Similar to CNN, The Swin Transformer has a hierarchical structure, which progressively reduces the spatial resolution of the feature maps, allowing it to model the semantic information more effectively. The Swin Transformer represents a significant advancement in the application of Transformer models to computer vision tasks. It has achieved state-of-the-art performance on several benchmarks for tasks such as image classification, object detection, and semantic segmentation.

The **Segment Anything Model (SAM)** [7] is a novel, promptable segmentation system that revolutionises the approach to object detection. Drawing inspiration from ChatGPT [103], SAM reimagines the object detection task as a process of issuing prompts to the model, specifying which elements to segment within an image. This innovative model, equipped with a comprehensive understanding of the characteristics of objects, demonstrates remarkable zero-shot generalization capabilities. It can accurately identify and segment unfamiliar objects and images without the need for additional training. This pioneering approach signifies a potential future direction for object detection in the era of big data.

2.3.2.3 Introduction to Multiple Object Tracking

Once we get the detection results, we can combine them to be the tracking results of the users. One of the most difficult challenges is dealing with occlusions. Occlusions occur when an object or person of interest is partially or fully blocked from the camera's view by another object or person. This can make it difficult for the system to accurately track the occluded object or person. For example, if a person walks behind a pillar, the camera may lose sight of them, causing the system to lose track of their location. Similarly, in crowded environments where people may overlap in the camera's view, identifying and tracking each individual can be challenging. Another challenge is human reidentification, which involves recognising the same person across different camera views or at different times. A person's appearance can look different from different camera angles, making it difficult for the system to accurately

re-identify individuals, which can impact the system's ability to accurately track a person's location over time.

To solve the issue we described above, one solution is to extract the appearance characteristics of the pedestrians. This involves extracting distinctive features from an image that can be used to identify an individual. These features can include colour, texture, and shape. For example, colour histograms can capture the colour distribution of a person's clothing, while texture features can capture patterns on the clothing. Shape features can capture the person's silhouette or body shape. When extracting the features, deep learning models are often used, particularly CNNs. CNNs can automatically learn hierarchical feature representations from the data, which can capture complex appearance patterns. Some approaches use part-based models that focus on specific body parts, as these are less affected by changes in pose [164].

2.3.2.4 Examples for Multiple Object Tracking

Simple Online and Realtime Tracking (SORT) [2] is a pragmatic approach to multiple object tracking with a focus on simple and effective methods. The algorithm is designed to be lightweight and reliable, providing good tracking for applications that do not require the complexity of larger, more advanced systems. The idea of SORT is to use a Kalman filter to predict the state of an object; the size of the bounding box is also adjusted based on the observed size in each frame, which helps to handle changes in object size and aspect ratio. This algorithm uses Hungarian Algorithm to associate the detected objects in the current frame with the existing objects being tracked. This algorithm computes the optimal assignment (lowest total cost) of detected objects to predicted objects to minimise the overall tracking error. In order to handle occlusions and missed detections, if an object is not detected in a frame, SORT will keep the object in the state of being tracked for a few more frames based on its velocity before removing it. SORT is capable of real-time tracking on standard hardware, but could not handle long-term occlusions or re-identifications due to its simple tracking mechanism.

DeepSort [116] is an extension of the original SORT algorithm. It incorporates a deep learning association metric to improve the performance of the tracker, particularly in handling

appearance changes, occlusions, and missed detections. It uses an extension of the original SORT algorithm. It incorporates a deep learning association metric to improve the performance of the tracker, particularly in handling appearance changes, occlusions, and missed detections, which involves the training of a CNN on a large-scale person re-identification dataset to extract features from detection bounding boxes. These features are then used to calculate the appearance cost in the Hungarian algorithm, which improves the association of detections and tracklets.

While the above approaches use two-step feature extraction (i.e. extract features from existing detections), there are also ways to simultaneously predict the bounding box location and extract the visual features. For example, FairMOT [168] by Yifu et al. addresses the imbalance between detection and reidentification (Re-ID) in MOT systems. It proposes a single-shot framework that jointly learns the detection and Re-ID tasks. This is achieved by designing a shared backbone network that extracts features for both tasks, which are then used by separate heads for detection and Re-ID. Unlike previously described anchor-based detection framework (RCNNs), FairMOT adopted an anchor-free method that simplifies the detection process and reduces the number of hyperparameters. The identification branch is learnt in a self-learning fashion so that each pedestrian has their own class. It also introduces a new loss function, called fairness loss, to balance the learning of detection and Re-ID. This ensures that both tasks contribute equally to the final performance of the system. FairMOT achieved state-of-the-art performance in the MOT task.

ByteTrack [169] is a state-of-the-art MOT system designed to estimate the bounding boxes and identities of objects within video sequences. It employs a YOLOX model to predict both high- and low-score bounding boxes, and then utilises enhanced tracking algorithms to associate each detected bounding box. The significant performance improvement of ByteTrack is primarily attributed to the enhanced capabilities of the detector. Interestingly, the method incorporates the innovative idea of associating lower score bounding boxes, which contributes substantially to its effectiveness. In particular, ByteTrack does not rely on feature extraction at all. Nevertheless, if deep feature extraction techniques were to be applied, it is anticipated that the system's performance could be marginally improved further.

2.3.2.5 Vision-Based Indoor Positioning

In addition to the top-down camera perspective for indoor location solutions, a prevalent approach is the utilisation of a first-person view camera for IPS. This method closely resembles navigation systems, typically leveraging the user's smartphone camera to capture the surrounding environment. Identifies landmarks from these captures and cross-references them with a database of known landmarks for accurate positioning. Many of the related image- or vision-based indoor positioning approaches take advantage of the maps generated with a technology called visual Simultaneous Localisation and Mapping (SLAM) [74]. We first introduce some basic concepts of this technique.

SLAM is a computational problem in which a device, often a robot or drone, aims to construct or update a map of an unknown environment while simultaneously keeping track of its location within that environment. Essentially, it is about understanding the surroundings and knowing one's position within it at the same time. In our case, the device is the user's smartphone. Visual SLAM is a specific variant of SLAM that uses visual data, typically from cameras, to construct a map of the environment and determine the device's position within that environment. It captures images of the surroundings, identifies key features in those images, and uses these features to build a map and estimate the device's position and orientation.

A comprehensive survey [13] provides an in-depth review of various approaches and methodologies to implement visual SLAM. The authors present the three primary visual-based SLAM approaches: visual-only, visual-inertial, and RGB-D SLAM. They provide a comprehensive review of the main algorithms for each approach using diagrams and flowcharts, emphasising the pros and cons of each method.

- For visual-only SLAM, it relies solely on visual data, typically from a monocular camera or stereoscopic camera pairs. The advantage is that it is lightweight and cost-effective, as it requires minimal hardware. It is suitable for environments where other sensors might not be reliable. This is also the most common solution for

image-based indoor localisation & navigation. This approach is susceptible to rapid camera motions that might cause motion blur or feature loss.

- Visual-Inertial SLAM fuses visual data with inertial measurements from an IMU. The IMU typically provides acceleration and angular velocity readings, which can be integrated over time to estimate motion. This approach combines the strengths of visual data for tracking and mapping with the high-frequency motion data from the IMU, and can handle brief occlusions or feature-poor environments better than visual-only SLAM. However, this method requires careful calibration and synchronisation between the camera and the IMU.
- For RGB-D SLAM, this approach uses RGB-D cameras, which capture regular RGB images along with depth (D) information for each pixel. Depth is typically obtained using infrared projectors and sensors or time-of-flight technology. This method can provide rich 3D information directly, making the mapping process more straightforward and suitable for indoor environments and applications where accurate depth information is crucial. The disadvantage of this method is that it typically has a limited depth sensing range.

2.3.2.6 Examples for Vision-Based IPS

Each of these approaches has its own set of advantages and challenges, making them suitable for different applications and environments. The choice of approach often depends on the specific requirements of the application, the available hardware, and the environmental conditions.

Han et al. [66] proposed a system that delve into the challenges and solutions of WiFi-based indoor localisation. The authors recognise that while WiFi localisation is a promising technique, it often faces issues due to fingerprint ambiguity. To address this challenge, the authors introduce Argus, a system that leverages geometric constraints from ubiquitous photos to enhance WiFi-based localisation. When a user wishes to be localised, they activate the Argus app on their smartphone. The phone's camera and WiFi scanning are automatically turned on. The user is then instructed to capture a photo of a Point of Interest (POI) near the

centre of their viewfinder. The system then uses these visual data in conjunction with WiFi fingerprints to improve localisation accuracy.

Another paper aims to improve indoor tracking using crowd-sourced maps [81]. The proposed method focusses on improving indoor tracking accuracy using crowd-sourced 3D point clouds from indoor environments. The authors propose an enhanced particle filtering algorithm that takes into account potential obstacle information that might be missing from crowd-sourced maps. This approach is particularly beneficial because traditional indoor mapping techniques, such as SLAM, require expensive and specialised equipment, whereas crowd-sourced visual data from devices like smartphones offer a more accessible and cost-effective solution.

The paper emphasises the use of 3D point clouds generated from crowd-sourced photos using VisualSfM, a state-of-the-art Structure from Motion (SfM) toolchain. The system detects 3D points representing floors and ceilings and projects the remaining points, which represent obstacles, onto a ground plane. These data are then discretised into an occupancy grid map. The enhanced particle filtering algorithm (EPF) developed by the authors takes into account the density of these 3D points in each grid cell to detect potential obstacles and collisions. The algorithm also uses history-based particle regeneration to improve tracking accuracy.

iMoon [80] is another example of using a smartphone by a user for indoor image-based navigation. It uses crowd-sourced photos and sensor data from smartphones to build 3D models of indoor environments. These models serve as the foundation for providing navigation services to users. The operation of iMoon is divided into two modes: initialisation and navigation. In the initialisation mode, users contribute photos and sensor readings, which are then processed to generate 3D models of the environment. The system employs SfM techniques to reconstruct 3D models from 2D images. In navigation mode, users can request directions to a specific location. The system identifies the user's current location by matching their live photos with the preconstructed 3D model. Once the location is determined, iMoon provides real-time navigation instructions.

However, the limitation of iMoon also potentially leads to inaccuracy or incompleteness of the crowd-sourced maps. The authors propose deploying iMoon in a hybrid cloud infrastructure to

balance the computational demands of 3D modelling with the need for low-latency navigation services.

Mingkuan et al. propose an automatic key frame detection system called **SweepLoc** [113]. SweepLoc addresses the challenge of the increasing demand for accurate indoor localisation by proposing a novel approach that takes advantage of the synergy between video data and Channel State Information (CSI). Recognising that both video analytics and CSI offer unique insights into user location and behaviour, the authors aim to combine these modalities to achieve a more precise and reliable indoor localisation system. The proposed system uses a fine-tuned Faster R-CNN to select Regions of Interest (ROIs) with potential landmarks in video frames.

The authors constructed a landmark database with manually annotated bounding boxes for each landmark in the image. They augmented this database using various image transformations to prevent overfitting and facilitate accurate ROI detection. The proposed system integrates video and CSI data to represent various parameters such as location, movement patterns, and activities. The authors employ various DNN architectures to process and extract relevant features from both the CSI and video data. The system is also designed to recognise three primary activities: walking, standing, and turning around, which are predicted using CSI data and video analysis. By amalgamating the angle, distance, and motion pattern data, SweepLoc attempts to match each detected pedestrian in the camera's view with the corresponding person.

2.4 Multimodal indoor positioning

Multimodal indoor positioning is a technique that combines multiple types of data or technologies to improve the accuracy and reliability of indoor positioning systems. This approach takes advantage of the strengths of different positioning technologies and mitigates their individual weaknesses, leading to more robust and accurate positioning. It can combine different technologies such as Wi-Fi, Bluetooth, IMU sensors and surveillance cameras; each

technology has its own strengths and weaknesses in terms of accuracy, range, and infrastructure requirements. When multi-technology combinations are used, multimodal systems can achieve higher accuracy and robustness than systems based on a single technology. For example, a system might use Wi-Fi for coarse positioning and then uses a smartphone's IMU sensor for relative moving direction prediction.

Multimodal indoor positioning systems have a broad spectrum of applications, including providing navigation assistance for visually impaired individuals, facilitating asset tracking in warehouses, enabling indoor robotics, and offering location-based services in shopping malls, hospitals, and museums. These systems offer substantial benefits to users, particularly in terms of improved accuracy in indoor positioning. Complex indoor environments often present challenges for camera-only positioning systems, making it difficult to accurately identify and locate users. However, when these systems are combined with Bluetooth Low Energy (BLE) signals or Wi-Fi signals, the identification problem can be effectively addressed. This is because a user's Media Access Control (MAC) address, which represents the physical address of a device, remains constant. By analysing the movement of the device and matching it with visual tracking results, users can be located with a high degree of precision.

2.4.1 IMU with Bluetooth Approach

The system "TAR: Enabling fine-grained targeted advertising in retail stores" [157] is designed to enhance the retail shopping experience by delivering targeted advertising. This is achieved through the integration of Bluetooth Low Energy (BLE) and video-based technologies to identify and track customers within a retail setting, subsequently providing personalised advertising based on their shopping behaviours and preferences.

The system employs a two-step matching algorithm for indoor pedestrian tracking. The initial step involves the amalgamation of BLE signals with visual tracking results from a single camera. In this design, customers are required to have an application installed on their smartphones that broadcasts BLE signals. The receivers of these signals are co-located with surveillance cameras. By recording the Received Signal Strength Indicator (RSSI) from

various users, the central server can construct user profiles based on motion patterns. These profiles are then compared with local visual tracking tracklets, obtained through Deep Neural Network (DNN) based tracking algorithms, to identify the best match.

The subsequent step involves cross-camera tracking and identification. This is achieved through a combination of ID filtering, a probabilistic matching algorithm based on Dynamic Time Warping (DTW), and an ID assignment algorithm. These methods collectively determine the correct identity from a global perspective. This comprehensive approach ensures accurate customer tracking and identification, facilitating the delivery of targeted personalised advertising.

The authors conducted several experiments to evaluate the proposed system. They conducted experiments that involved up to 27 participants captured simultaneously by cameras. Each participant had the TAR application installed on their devices and moved around the store according to their interests. Approximately 1 hour of video data is collected; some of the challenges faced included dark and blurry snapshots, making it difficult to identify individuals using only visual approaches. The system achieves an accuracy of 90%, which is 20% better than the state-of-the-art multicamera person tracking algorithm at that time. TAR can still maintain more than 80% accuracy even when half of the nodes (cameras or BLE receivers) are down. This is attributed to the fact that each working node can independently identify and track customers.

2.4.2 IMU with Computer Vision Approach

Siyuan et al. proposed a system called PHADE [141]. This paper discusses a system that allows public cameras, such as those in museums, to interact with individuals based on their behaviour and movements captured in videos. The system customises messages for individuals based on certain external characteristics and uses people's behavioural patterns as identifiers. These identifiers are extracted from their movements in the video. For example, a camera can recognise that a user is pointing or looking at a specific exhibit and send

relevant information to them. The behavioural information on the user side is collected via the smartphone's IMU sensors.

The paper highlights the challenges of extracting step-related motion features from videos due to the intensive computation cost, such as step phase and gait. This makes it less scalable when dealing with a large number of people in a crowded scene. On the other hand, walking speed can be easily calculated from videos, while obtaining walking speed from sensors presents its own set of challenges. The performance of methods that integrate accelerometer readings, combine step count with stride length, or use feature-based estimation is influenced by the orientation and position of the phone. Furthermore, determining the walking direction using the compass is not always reliable, especially in indoor scenarios, as it can be easily affected by signal and ferromagnetic interference.

The proposed method extracts two types of motion characteristics: "isWalking" and "Relative Walking Direction". For the "isWalking" feature, it determines whether a person in the video is walking. On the video side, the velocity of bounding boxes generated by a Kalman filter during the tracking process is checked. On the user side (i.e. IMU sensor), the system projects sensor acceleration onto gravity and calculates the variance of the projected values within a two-second window. If the variance is above a pre-set threshold, "IsWalking" is marked as "Yes". The "Relative Walking Direction" feature defines the direction in which a person moves with reference to their initial direction at the beginning of a movement period. The relative direction can be derived from the velocity of the binding boxes in the video without extra computation. On the user side, the system projects rotation rate onto gravity and then integrates it to obtain relative rotation.

This motion pattern matching addresses the issue where the smartphone only chooses to receive the messages corresponding to itself since all the messages are broadcast from the central server. To prevent broadcasting packets with raw video motion features, the paper proposes using principal component analysis (PCA) to transform raw motion features into low-dimensional address codes to prevent information leaks.

The overall performance of the proposed system is satisfactory. It was tested with up to 10 users in the scene, PHADE achieved 87% matching correctness using 18 seconds of motion features. Performance degrades as the number of users increases, mainly due to occlusions between people in a limited area. This can be mitigated by placing the cameras on top of or on the ceiling. The discussed approach is one example of fusing the IMU and video sensor for applications like indoor localisation and navigation, gesture-based messaging, etc.

2.4.3 IMU with Computer Vision & Bluetooth Approach

There is another paper that introduced the system “iVR” [84], which fuses three different modalities: IMU sensor, BLE and video. For the visual modality on the server side, the system uses IP cameras to record the scene with continuous-stream videos to capture the movement of individuals within the scene. The image detection is done with Mask-RCNN pre-trained on a modified COCO dataset. On the client side, implemented on Android devices, the accelerometer and gyroscope readings are recorded. These readings help in dead-reckoning, which is a method to estimate one’s current position based on a previously determined position, similar to PHADE. The client also samples BLE signals for RSSI of different beacons deployed inside the indoor environment, indicating the estimated distance between the mobile device and each beacon. This provides another set of localisation data points. The proposed system combines the three sources of input to achieve a better indoor positioning accuracy.

Before merging the results from the different modalities, the system associates the visual detection results of each pedestrian with the pedestrian’s dead-reckoning trace. This ensures that the visual data and IMU data are correctly matched for each individual. The matching is achieved by finding the best matches between the video detection and the PDR result with the Hungarian algorithm. The next step is to combine the intermediate results from vision detection, dead-reckoning, and wireless localisation into an augmented particle filter. The state of each particle in the filter includes its current location, heading direction, and weight. The particle filter helps to estimate the most probable location of an individual based on the combined data from all modalities.

The iVR system boasts several notable advantages. Foremost among these is its ability to automatically construct a semantic map, eliminating the need for labour-intensive site surveys. Additionally, its multimodal fusion approach takes advantage of the strengths of various data sources, ensuring that the limitations of one modality are offset by the strengths of another. This integrated approach significantly enhances the overall accuracy and robustness of the system. However, the system is not without challenges. A key limitation is the requirement for a direct line-of-sight for the camera, which can pose challenges in certain environments. Additionally, the system's frame rate is constrained by bandwidth and computational resources, potentially limiting its real-time tracking capabilities. Another point of consideration is the system's reliance on an empirical threshold to distinguish between stationary and walking states. This means that the manner in which a user carries their smartphone can influence accuracy. For example, holding the phone horizontally in front of the body offers an average accuracy of 0.76m. On the contrary, holding it naturally in hand results in 1.55m, stowing it in a trouser pocket gives 1.25m, and placing it in a backpack yields 1.09m.

2.4.4 Wi-Fi with Computer Vision Pedestrian Identification

RFCam [73] is a system designed for human identification by combining data from the Wi-Fi and the camera. RFCam uses a multiantenna Wi-Fi radio to collect Channel State Information (CSI) of Wi-Fi packets sent by mobile devices and a camera to monitor users in the area. It derives features on location, motion, and user activity from CSI data and combines them with visual user features from video analytics to identify users. The system addresses challenges posed by multiuser environments and uncertainties in Wi-Fi data by developing video-assisted learning models.

The central concept of RFCam hinges on the combined use of video and Channel State Information (CSI) data to depict location, movement patterns, activities, and more. The authors used various Deep Neural Network (DNN) models to process and extract valuable information from both CSI and video data. This extraction was designed to determine the angle and distance between the wireless Access Point (AP) and the mobile device, as well as to identify specific user activities. For activity recognition, the system was trained to

discern three distinct actions: walking, standing, and turning around. The predictions for these activities were made by analysing both the CSI data and the video footage. By integrating information about angle, distance, and motion patterns, RFCam, for every pedestrian detected by the camera, aims to pinpoint the corresponding individual.

In the study, experiments were conducted in two different testing environments. The camera was strategically positioned directly above the Wi-Fi transmitter, ensuring that the receiving angle and distance were aligned. In a test scenario involving 10 individuals, RFCam demonstrated an identification accuracy of 82.2% without any retraining, using only 200 Wi-Fi packets. When tested with 25 volunteers indoors, the system achieved an impressive average identification accuracy of 97.01%. However, the research also highlighted that environmental modifications, such as changes in area layouts or the introduction of large furniture, could generate Wi-Fi packet noise, which might compromise RFCam's efficacy. The system showcased robust performance in moderately populated areas, such as schools or store entrances. Yet, in densely populated settings, like train stations, RFCam encountered challenges related to misidentification. Despite these challenges, the system adeptly captured overarching user behaviours. To address the challenges presented in extremely crowded environments, this paper introduces a Reinforcement Learning (RL)-based matching system.

Part 1

Wireless Signal-Based Approach

Indoor positioning using wireless signals has emerged as a prominent method due to the ubiquity of wireless devices and infrastructure. The advantage of wireless signal-based approaches is that they preserve the user's privacy compared to other vision-based approaches that use surveillance cameras. These approaches primarily leverage RSSI from wireless technologies such as Wi-Fi, Bluetooth, and UWB. Deep learning improves the accuracy and robustness of this method by capturing intricate patterns in signal propagation affected by multipath fading, wall reflections, and other environmental factors.

Traditional methods often relied on deterministic models or fingerprinting techniques, which required extensive manual calibration. Deep learning models can automatically learn the spatial characteristics of the indoor environment. Techniques such as CNNs and RNNs have shown promise in improving positioning accuracy by processing sequences of RSSI values, thereby capturing temporal and spatial dependencies. In this part, we introduce our work GauPro, a Gaussian Kernel-based beacon probabilistic fingerprint algorithm. An LSTM model is used to mitigate the noise. The experimental results show that our approach improves the error distance of the conventional method by a large margin.

Symbol Used in Part 1

Symbol	Description
i	checkpoint index
j	accesspoint index
F_{ij}	Fingerprint value
μ_{ij}	Mean RSSI value
σ_{ij}	Standard deviation
R_{ij}	RSSI value
r_i	set of reading
g_x, g_y, g_z	gyroscope readings for the x, y, z directions
q	orientation vector
Z_i	Kernel value for checkpoint
$\mathcal{M}$	LSTM model
o_t	orientation data

Wireless Signal Based Deep Learning Fingerprint IPS

3.1 Introduction

Location-based service has become more and more popular in recent years. The aim of such a service is to provide detailed location information to customers and help them in different scenarios. The Global Positioning System (GPS) technology has made outdoor positioning highly successful and reliable. However, due to the high interference of the GPS signal inside a building, the accuracy of the indoor GPS is not as satisfactory as that of the outdoor GPS.

Many systems have been built to solve the indoor positioning problem with various technologies, including and not limited to infrastructure-based [127], RF-based [19, 146], magnetic-based [70] and camera-based [46]. However, many solutions involve modifications of the user's receiving-end device or require the user to carry a specific tag to assist the positioning process, which brings difficulties for their promotion in industry and brings troubles for the user.

Meanwhile, with the fast-growing of smart mobile phones and ubiquitous computing technology, it has become easier for us to build a positioning system to track the mobile phone inside a building without too many other devices. Among all the ubiquitous technologies for indoor positioning, the Bluetooth-based approach is one of the most widely used technologies. It works similarly to Wi-Fi, but provides better accuracy since it covers fewer ranges and consumes less power. Two main directions for Bluetooth-based approaches can be found in previous research. The first is the triangulation method that studies the relationship between signal strength and distance of Bluetooth signal [167]. Another direction for Bluetooth

positioning is the fingerprint method. A fingerprint, which usually consists of information from several sources of signals, represents a position on the map. It can be a piece of single-dimensional or multi-dimensional information. People use the signal data that they receive in real-time to match the fingerprint stored in the database and locate the receiving device on the map. In this chapter, we mainly discuss the fingerprint approach.

In the fingerprinting approach, checkpoints are strategically chosen locations within an indoor space where fingerprints are recorded. While traditional methods, such as the Bayesian approach [1], assume equal prior probabilities for each checkpoint, this assumption may not always hold in practical scenarios. Specifically, the positioning result from the previous timestamp should significantly influence the current estimation.

In this chapter, we introduce the Gaussian kernel as a mechanism to modulate the prior probabilities of checkpoints. The Gaussian kernel, commonly employed in machine learning, especially in support vector machines (SVMs)—serves as a powerful tool for pattern analysis, enabling the identification and study of various relational patterns. Our approach begins with the utilisation of the Gaussian kernel to compute the prior probabilities for all checkpoints. Subsequently, an LSTM model processes the IMU readings in a time sequence, effectively filtering out noise and enhancing the prediction accuracy of the user's subsequent position. By integrating these two steps, we ensure that the prior probabilities of the checkpoints are dynamically updated in alignment with the user's movements.

The primary contributions of our work can be summarised as follows.

- We design GauPro, a Gaussian kernel-based Bayesian algorithm. In this algorithm, the prior probabilities of each checkpoint are not equal; instead, they will change according to the user's current position and moving direction.
- We implement a real indoor positioning system based on the GauPro algorithm that includes the client application and the back-end server. The system can be easily deployed in any indoor space without any hardware modifications of the receiving end devices; hence, it can provide services for general smartphone users.

- We deploy the implemented system in a work space and conduct field experiments in difference scenarios. Using the real experimental data obtained in field experiments, we perform evaluations of our GauPro algorithm-based indoor positioning system.

The remainder of this chapter is organised as follows. Section 3.3.1 introduces the overall architecture of the system and the design of our GauPro algorithm. Section 3.2 discusses the related work to this chapter. Section 3.4 presents the implementation of the indoor localisation system, section 3.5 evaluates the proposed system through field experiments, and section 3.6 draws the conclusion.

3.2 Related Work

3.2.1 Overview of Wireless Indoor Positioning

A survey [56] has discussed different technologies for indoor localisation from hundreds of articles in the literature. In project RADAR [19], the authors use a Viterbi-like Algorithm for continuous user tracking and use it to guess the user's location. In "Location fingerprinting with low energy bluetooth beacons" [54], the authors divide the entire area into 1 m grid cells and estimate the probability of each cell by calculating the Euclidean distance between the set of signal strength values in a map cell and the current fingerprint. A system called MoteTrack [98] proposed a decentralised location estimation protocol that relies only on local data, local communication, and operational nodes. It allows the base station nodes themselves to perform location estimation rather than relying on a central server.

For the Wi-Fi approach, Chronos [146] measures subnanosecond Time of Flight (ToF) on commercial Wi-Fi radios. Another example is SpotFi [94], which provides a better angle of arrival (AoA) estimation and direct path filtering. For Bluetooth technology, a research [55] was carried out using the triangulation method combined with least squares estimation. Sudarshan in his work [130] has described a study about choosing the location of APs inside a room. Their solution is to find a maximum resolution subhypergraph of the indoor Bluetooth signal.

In addition to Wi-Fi or Bluetooth, the idea of a fingerprint database or landmark-based localisation based on the Earth's magnetic field was also conceived. A system with an enhanced particle filter [28] was proposed, which builds a database of magnetic field landmarks. Another solution proposed by previous researchers is to try to make use of the backscatter signals introduced in systems like HitchHike [118] and WiTag [91].

3.2.2 Wireless Positioning Techniques

3.2.2.1 Triangulation Positioning Method

Among all the techniques for indoor Bluetooth positioning, one of the easiest ways is the triangle localisation method based on the strength of the received signal. [63, 167] has analysed several different ways of implementing the indoor triangulation algorithm.

Given the location APs, the triangulation method can be used to determine the position of a mobile device inside a building. The results provided by [167] are less than 3.5 metres for 95% of the time. This approach runs very fast and costs less computing resources. However, it relies heavily on the location of the APs in the algorithm.

3.2.2.2 Fingerprint Method

Another common solution for Bluetooth indoor positioning is the fingerprint method. There are two different kinds of approach: deterministic and probabilistic. For the deterministic method, a Microsoft paper [20] described the relationship between the received signal and the signal fingerprint and how to compare them. It is called the k-NN method. Based on that, an algorithm called the weighted k-NN algorithm [21] was also studied to improve the drawback of the original k-NN algorithm where all k neighbours have the same vote.

The first probabilistic wireless positioning method was proposed by Youssef et al. [1]. They use the joint probability distribution as the fingerprint and use the Bayesian formula to estimate the position. In Swangmuang and Krishnamurthy's work [131], they introduced a new analytical model for estimating the probability distribution of the indoor Wi-Fi fingerprint.

Based on their work, our designed system further improved positioning accuracy by leveraging the users' movement information and by redistributing the probabilities of each checkpoint.

3.3 System Design and the GauPro Algorithm

3.3.1 System Overview

The purpose of our system is to locate users while inside a building using fingerprint technology. We designed and implemented our system, which provides indoor positioning service based on the Beacon device, user's smartphone, and several back-end servers. The phone application was developed on iOS system, but the same Beacon protocol can also be used on Android.

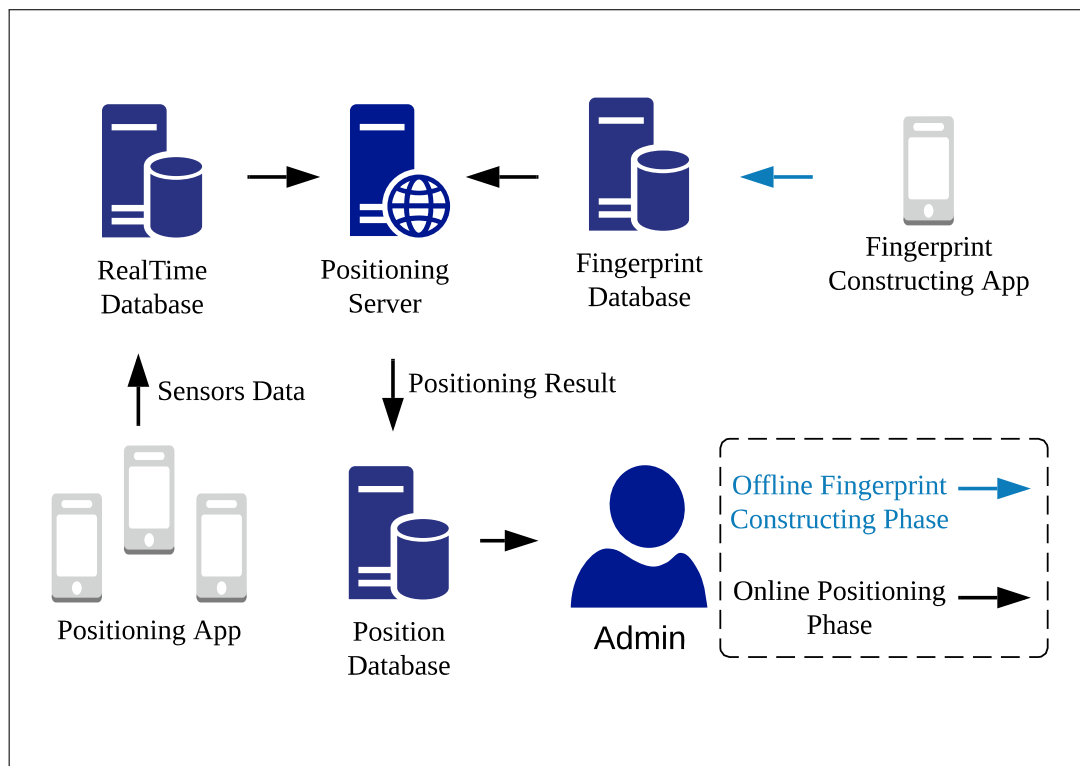


FIGURE 3.1. Overview of the indoor positioning system.

Our system consists of two main phases: **offline fingerprint database constructing** and **online positioning**. During the offline (fingerprint constructing) phase, we set checkpoint on

the map and record the observed Bluetooth Received Signal Strength (RSS) value and store them in a static database. For the online (positioning) phase, we use a matching algorithm to match the observed run-time value with the values we stored in database.

Figure 3.1 shows an overview of our beacon positioning system. We have the following components:

- **Positioning App**, smartphone App for user. Receive Bluetooth signals and send them to the real-time database
- **Fingerprint constructing App**, smartphone App for admin. Collect RSS fingerprints and upload them to the fingerprint database
- **Real-time database**, temporarily stores RSS data uploaded by users to avoid data traffic problem.
- **Fingerprint database**, stores fingerprint information for all checkpoints.
- **Positioning server**, estimate user's position with our designed algorithm.
- **Position database**, stores the calculated position information of each user.
- **Admin** can read all user position information from the position database.

For the offline fingerprint-building phase, we use a fingerprint-collecting application to upload the gathered Received Signal Strength Indicator (RSSI) as fingerprint for each checkpoint. For the online positioning phase, the user's smartphone can receive wireless signals from several Beacon devices inside the building. The smartphone sends all the data to a real-time database regularly to avoid data traffic issues. Then, the core back-end server (Positioning Server) will cache data from both the real-time database and the fingerprint database, estimate the user's location, and then send it to system admin or back to the user.

3.3.2 Offline Phase

The RSSI values (in dBm) observed from a checkpoint are subject to the Gaussian distribution.

$$X \sim \mathcal{N}(\mu, \sigma^2). \quad (3.1)$$

The fingerprint (from one single AP) of a checkpoint i can be described as

$$F_{ij} \triangleq (CP_i, AP_j, \mu_{ij}, \sigma_{ij}), \forall i \in 1, \dots, N, \forall j \in 1, \dots, n, \quad (3.2)$$

where N is the total number of checkpoints, n represents the number of beacons a checkpoint can hear. CP_i represents the checkpoint ID, AP_j is the AP ID, μ_{ij} represents the mean RSSI value, and σ_{ij} represents the standard deviation of that AP. For each checkpoint i , we define its coordinates as (x_i, y_i) . All the values in the fingerprint are measured and stored in the database in the offline phase.

We use the joint probability distribution function of a checkpoint's RSSI to represent the fingerprint. In our design, we consider the RSSIs collected from different access point checkpoints independent from each other. We can hear n Beacons at checkpoint i . The fingerprint at checkpoint i can be considered as:

$$f_i \triangleq [F_{i1}, F_{i2}, F_{i3}, \dots, F_{in}].$$

We build the fingerprint database with the fingerprint building app and an upload script. After we have built a radio map and deployed the beacon devices inside the area, we manually set the checkpoints on the map. Then, we hold our phone (installed the fingerprint building App) and record the RSSI values of each checkpoint. Finally, we upload all the recorded fingerprints to a fingerprint database via a Python-written upload script.

Then, we are going to use this database for our position estimation in the online phase.

3.3.3 Online Phase

3.3.3.1 Traditional Bayesian Algorithm

When we collect a signal with a mobile phone at the checkpoint i , we will receive multiple RSSIs from different APs. For a single access point j , the received RSSI R_{ij} is subject to the

Gaussian distribution. Let the probability density function $g(R_{ij})$ as

$$g(R_{ij}) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp \left\{ \frac{-(R_{ij} - \mu_{ij})^2}{2\sigma_{ij}^2} \right\}. \quad (3.3)$$

The reading of signal strength is discrete (integer). We normalise the probability distribution. During our test, we receive RSSI between -50dBm and -90dBm. Thus, the signal strength is in $[-50, -90]$. Any reading we receive is an integer in this range. Thus, the normalized probability of having a specific reading $\hat{R}_{ij}$ at checkpoint CP_i is:

$$P(\hat{R}_{ij}|CP_i) = \frac{g(\hat{R}_{ij})}{\sum_{k=-50}^{-90} g(k)}. \quad (3.4)$$

The number of beacons we can hear is n . Thus, the set of readings can be expressed as $\hat{r}_i = [\hat{R}_{i1}, \hat{R}_{i2}, \hat{R}_{i3}, \dots, \hat{R}_{in}]$. Thus, if we want to estimate the joint probability distribution $P(\hat{r}_i|CP_i)$, we can now estimate the product of all marginal probability distributions $P(\hat{R}_{ij}|CP_i)$. Combining Equation 3.4, we have the probability of having a set of readings $\hat{r}_i$ at checkpoint i :

$$P(\hat{r}_i|CP_i) = \prod_{j=1}^n P(R_{ij}|CP_i). \quad (3.5)$$

To calculate the probability for each checkpoint given a specific reading, we can use Bayesian formula

$$P(CP_i|\hat{r}_i) = \frac{P(\hat{r}_i|CP_i)P(CP_i)}{\sum_{i=1}^N P(\hat{r}_i|CP_i)P(CP_i)}. \quad (3.6)$$

In equation (6), $P(CP_i)$ is the prior probability. In the traditional Bayesian algorithm, this is uniformly distributed. After we have calculated $P(CP_i|\hat{r}_i)$ for all checkpoints, we can get a list of results.

The resultant location is averaged between the top three checkpoints (with the highest $P(CP_i|\hat{r}_i)$ values). $L^* = (x^*, y^*)$:

$$x^* := \sqrt{X_1^2 + X_2^2 + X_3^2}, \quad (3.7)$$

$$y^* := \sqrt{Y_1^2 + Y_2^2 + Y_3^2}, \quad (3.8)$$

where X_1, X_2, X_3 and Y_1, Y_2, Y_3 are the coordinates of the top three checkpoints.

3.3.3.2 Improvement with Gaussian Kernel

We notice that the prior probability of each checkpoint could be improved. Since positioning is a dynamic process, the previous location can help locate the current one. Suppose that the user's last known position is position $L^- = (x^-, y^-)$, and Δt time has passed. L^- is the result of L^* in the previous round. We also maintain $(q_0^*, q_1^*, q_2^*, q_3^*)$ to represent the current orientation vector, and θ^* to represent the current orientation angle. According to the differential equation for quaternion [139], we can use our readings from the gyroscope $\hat{g}_x, \hat{g}_y, \hat{g}_z$, which are the angular velocity on each axis, and our time shift Δt to update the quaternion:

$$\begin{bmatrix} q_0^* \\ q_1^* \\ q_2^* \\ q_3^* \end{bmatrix} := \begin{bmatrix} q_0^- \\ q_1^- \\ q_2^- \\ q_3^- \end{bmatrix} + \frac{1}{2} \cdot \Delta t \cdot \begin{bmatrix} -\hat{g}_x \cdot q_1 - \hat{g}_y \cdot q_2 - \hat{g}_z \cdot q_3 \\ \hat{g}_x \cdot q_0 - \hat{g}_y \cdot q_3 + \hat{g}_z \cdot q_2 \\ \hat{g}_x \cdot q_3 + \hat{g}_y \cdot q_0 - \hat{g}_z \cdot q_1 \\ -\hat{g}_x \cdot q_2 + \hat{g}_y \cdot q_1 + \hat{g}_z \cdot q_0 \end{bmatrix}, \quad (3.9)$$

$(q_0^-, q_1^-, q_2^-, q_3^-)$ are the values of $(q_0^*, q_1^*, q_2^*, q_3^*)$ in the last round. The initial values of $(q_0^-, q_1^-, q_2^-, q_3^-)$ are $(1, 0, 0, 0)$. Next, we derive the rotation angle $\Delta\theta$ around z axis with the updated $(q_0^*, q_1^*, q_2^*, q_3^*)$:

$$\Delta\theta := -\arctan\left(\frac{q_1^* q_2^* + q_0^* q_3^*}{q_0^{*2} - q_1^{*2} + q_2^{*2} - q_3^{*2}}\right), \quad (3.10)$$

$$\theta^* := \theta^- + \Delta\theta, \quad (3.11)$$

θ^- is the value of θ^* in the last round. The initial value of θ^- is 0.

Now that we have the rotation angle $\Delta\theta$, we need to know if the user is moving. To simplify our calculation, we set a threshold on the phone's acceleration. Once we receive a reading higher than that threshold, we assume that the user has moved one step. If so, we can use the previous positioning result: $L^- = (x^-, y^-)$, last orientation angle θ and a step length l to calculate the new position $L^* = (x^*, y^*)$:

$$x^* := x^- + l \cdot \cos(\theta^*), \quad (3.12)$$

$$y^* := y^- + l \cdot \sin(\theta^*), \quad (3.13)$$

l is the default step size, which is given in advance. After we have successfully estimated the position change in Δt , we can use the distance between them as our new step length.

We now introduce the method we use to reallocate the prior probability of each checkpoint in Section 3.3.3.1. In our design, we use the Gaussian kernel to achieve this.

The Gaussian kernel, or Radial basis function (RBF) kernel, is a useful kernel function that can map lower dimension points to higher dimension space. It describes a function of the squared Euclidean distance between two locations. Suppose we have a reference location (x', y') and a checkpoint at (x_i, y_i) , we calculate the kernel function $k(x_i, x', y_i, y')$ as follows

$$k(x_i, x', y_i, y') \triangleq \exp \left\{ -\frac{(x_i - x')^2 + (y_i - y')^2}{2\sigma^2} \right\}. \quad (3.14)$$

We use L^* as the reference point in Equation 3.3.3.2, we can have the kernel value of that checkpoint regarding point (x^*, y^*) . σ is the average step length that we have measured so far. To simplify the following calculation, we define the checkpoint CP_i 's kernel value as Z_i :

$$\begin{aligned} Z_i &:= k(x_i, x^*, y_i, y^*) \\ &= \exp \left\{ -\frac{(x_i - x^*)^2 + (y_i - y^*)^2}{2\sigma^2} \right\}. \end{aligned}$$

Once we have finished calculating from Z_1 to Z_N , we normalise the result of each checkpoint to get their new prior probability $P'(CP_i)$:

$$P'(CP_i) := \frac{Z_i}{\sum_{i=1}^N Z_i}. \quad (3.15)$$

Finally, by combining equation 3.6 and 3.3.3.2, our Bayesian formula $P(CP_i|\hat{r}_i)$ now becomes $P'(CP_i|\hat{r}_i)$

$$P'(CP_i|\hat{r}_i) = \frac{P(\hat{r}_i|CP_i)P'(CP_i)}{\sum_{i=1}^N P(\hat{r}_i|CP_i)P'(CP_i)}. \quad (3.16)$$

The $P(\hat{r}_i|CP_i)$ remains the same as in 3.3 and 3.4.

Once we have finished computing $P'(CP_i|\hat{r}_i)$ for each checkpoint, we can use equation 3.7 and 3.8 again to calculate the resultant location $L^{*'} = (x^{*'}, y^{*'})$, which we consider as L^- in the next Δt time.

$$x^* := \sqrt{X_1'^2 + X_2'^2 + X_3'^2}, \quad (3.17)$$

$$y^* := \sqrt{Y_1'^2 + Y_2'^2 + Y_3'^2}, \quad (3.18)$$

where X_1', X_2', X_3' and Y_1', Y_2', Y_3' are the coordinates of the top three checkpoints in the result list we calculated with equation 3.16.

3.3.4 Temporal Feature Extraction for IMU Orientation

With rapid advancements in deep learning, it becomes feasible to integrate it with our proposed Gaussian kernel-based fingerprinting method. As previously discussed, LSTM models excel at analysing sequential data. The user's movement change is determined by the quaternion equation (referenced by Eq. 3.9) offers accurate results with ideal data; its performance degrades significantly in the presence of noise. To address this limitation, we employ the LSTM model to process the IMU data.

Let us consider the local IMU sensor data, where the orientation information $o_t \in \mathbb{R}^O$ is governed by the quaternion equation. These IMU features, which are sequential in nature, are processed using our trained LSTM model, denoted as $\mathcal{M}$.

The LSTM model $\mathcal{M}$ processes the input orientation data sequence O_t to predict the change in orientation Δo_{t+1} of the user:

$$\Delta o_{t+1} = \mathcal{M}(O_t), \quad (3.19)$$

where $O_t = \{o_1, o_2, \dots, o_t\}$ represents the sequence of orientation data up to time t . The LSTM model, trained on this data, captures the temporal features of the sequence, ensuring consistent and accurate predictions. Consequently, the next predicted movement of the user, denoted as $p_{t+1} = (x_{t+1}, y_{t+1})$, can be expressed as:

$$x_{t+1} = x_t + l \cdot \cos(\Delta o_{t+1}), \quad (3.20)$$

$$y_{t+1} = y_t + l \cdot \sin(\Delta o_{t+1}), \quad (3.21)$$

This methodology effectively mitigates the noise introduced by the sensors, delivering robust and reliable prediction results. When the predicted location is aligned with the fingerprint stored in the database, further improvements in accuracy are observed.

3.3.5 Training Detail of the Temporal Feature Extraction Model

In this subsection, we delve into the specifics of the LSTM model structure and its training details, which plays a pivotal role in extracting temporal features from the IMU readings.

3.3.5.1 Model Architecture

The LSTM model is designed to capture the sequential patterns in the IMU data, making it apt for our application. The architecture comprises:

- **Input Layer:** Accepts the sequential IMU data, with each sequence having a length of $T = 150$ and feature dimension of $D = 4$. The time length is the maximum steps we store with the LSTM model, and in the experiment we choose 150 to be the best choice. The feature dimension is determined by the quaternion equation output, which is four.
- **Hidden LSTM Layers:** Comprising two layers of $N = 64$ LSTM units. These layers expand the initial input feature to a higher dimension to capture the temporal dependencies in the data.
- **Dense Layers:** Three fully connected layers that aid in refining the features for the final prediction and help eliminating the noise.
- **Output Layer:** Predicts the change in orientation or any other relevant output based on the processed features.

3.3.5.2 Training Procedure

- (1) **Data Preparation:** IMU data is segmented into sequences of fixed length $T = 150$ to serve as input to the LSTM model.
- (2) **Loss Function:** We employ the Mean Squared Error (MSE) as the loss function, given its efficacy in regression problems.
- (3) **Optimizer:** Adam optimiser is used, which dynamically adjusts the learning rate during training.
- (4) **Training Epochs:** The model is trained for 200 epochs at batch size of 256 to ensure efficient and stable training, with early stopping implemented to prevent overfitting.

3.3.5.3 Model Evaluation

The model and training script are implemented with PyTorch. We separate the experiment area into a training and validation set. After training, model performance is evaluated in the separate validation set. Key metrics such as the mean absolute error (MAE) and the root mean square error (RMSE) are computed to gauge the accuracy and reliability of the model.

3.4 System Implementation

In this section, we will briefly discuss the implementation of the algorithm. We will first introduce the mobile app we used to collect the fingerprint, then we will introduce the server that runs our algorithm, including the data structure on the back-end and the flow chart of our code.

3.4.1 Mobile App Implementation

Here we introduce the implementation of our mobile application.

3.4.1.1 App for Fingerprint Construction

This app is for service providers, for example the building manager, to collect the fingerprint and build the database. Before collecting the fingerprint, we need to store the data for the beacon and checkpoints first. Then we manually add checkpoints with our online fingerprint constructing application which connects directly with the fingerprint database and provides a user interface to display the radio map.

The workflow of offline fingerprint construction is shown in Figure 3.2. As we can see from the figure, once our application is started, it begins to detect the nearby Beacon signals. For each checkpoint we store about 100 set of data for the following calculation. Once the data are collected, we run a Python script to calculate their average value and standard deviation, form a fingerprint data structure, and upload it to the fingerprint database.

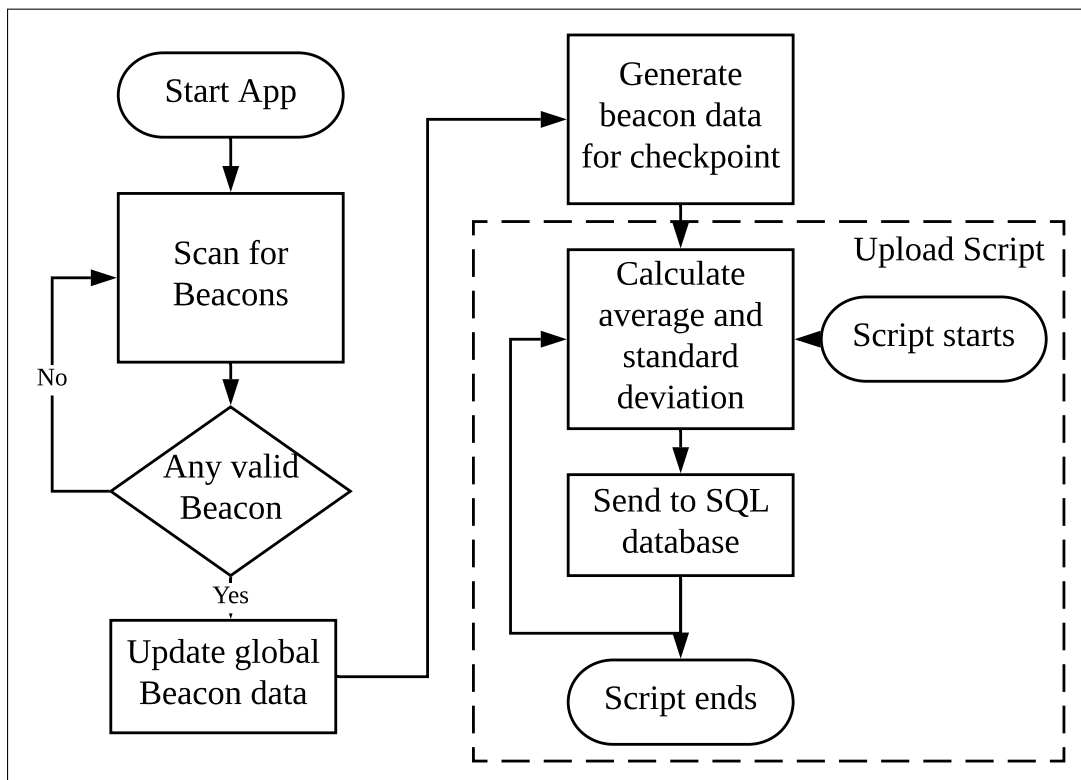


FIGURE 3.2. Workflow of fingerprint construction.

3.4.1.2 App for Positioning

This app is for general smartphone users to perform indoor positioning. It will scan the nearby Beacon signal and upload it to the real-time database. It will provide the raw data about our Bluetooth positioning. The information includes device serial number, timestamp, Beacon sniff, gyroscope, and accelerometer data. For Beacon, we use the CoreLocation framework provided by Apple to fetch the Bluetooth signal and sniff. For the gyroscope and accelerometer, we use the CoreMotion framework to collect information from our phone sensor.

As we have described in Section II-C2, we can use g_x, g_y, g_z to calculate the rotation angle with Equation (7) (8), and use a_x, a_y, a_z to decide whether the user is moving by calculating the sum of these three values and comparing it with our threshold.

3.4.2 Back-End Server Implementation

The back-end server is where we run our algorithm and find the positioning result. The server is written in Golang because it is open-source, fast, and concurrent, which is essential for a back-end server. Being open-source means that we have a lot of documents and help threads online to read. Being fast and concurrent means that our service will run efficiently since we expect more than five thousand people to use the system at the same time ideally. In this section, we will explain the structure and workflow in detail.

As shown in Figure 3.3, we generally have two phases: the cache phase and the positioning phase. The figure can be explained as follows: once we start our service, we first cache the Beacon sniff data and fingerprint we need; Then, we use the phone sensor data to estimate position, and calculate the components for our Bayesian formula; Finally, we calculate the result with Bayesian formula, sort the result list and get the final result.

For the cache phase, we need to load three tables in total and build the corresponding data storage structure. We mainly need two parts of data:

- **Beacon sniff** received from realtime database

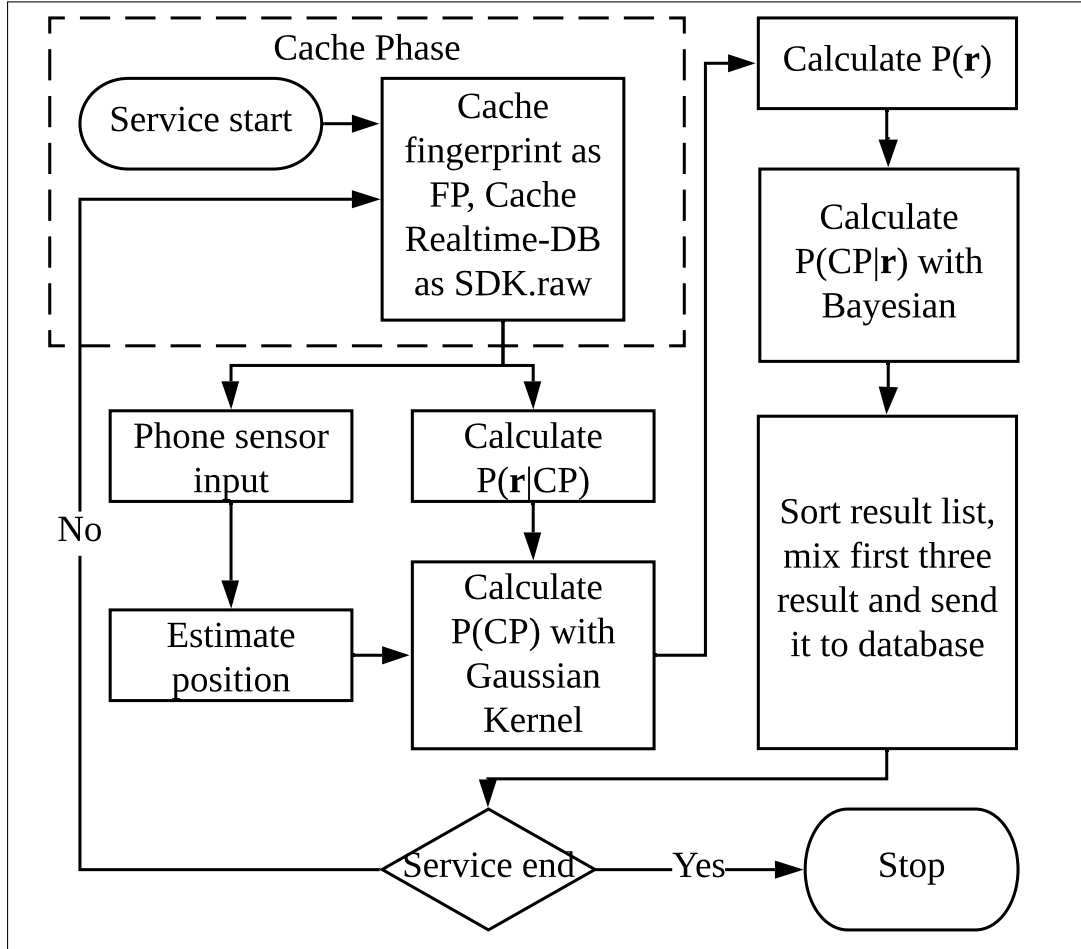


FIGURE 3.3. Workflow of back-end server

- **Beacon device** location and **checkpoint** information

So, we define data structure accordingly:

- **sdkRaw** - This data structure is to represent the real-time Beacon sniff uploaded by user. This is a key value structure. The first two map structures are the key, which stores user serial number and timestamp. The value is the beacon sniff, user's orientation, and acceleration.
- **CheckpointFP** - This data structure stores the fingerprint information. Similar to we have described above, the first two map structures are the key, representing checkpoint's ID and the beacon's ID. The value is the sniff data.

We use GORM¹, which is an object-relational mapping (ORM) library for dealing with relational databases, to connect the SQL database and cache their data. **sdkRaw** stores the observed Beacon sniff, **CheckpointFP** stores all checkpoints' locations and their average RSSI & standard deviation from each beam. Once we have all the cached data, we can proceed to the next phase.

To calculate the probability list for all checkpoints, we need three key components according to the Bayesian formula discussed in Section II:

- $P(\hat{r}_i|CP_i)$ - Calculate with Gaussian formula;
- $P(CP_i)$ - Value assigned according to Gaussian Kernel;
- $P(\hat{r}_i)$ - The sum of all $p(\hat{r}_i|CP_i)$;

The result list is a PairList (a structure in Golang) which contains a list of checkpoint IDs and their probability. As described in Section II, we sort the list by the probability values and choose the first three checkpoints, average their coordinates, and get our final result.

3.5 Evaluation

In this section, we will evaluate the performance of the algorithm we proposed and discuss the result obtained. First, we will introduce the experimental setup and the testing criteria for the indoor positioning system. Then we will run our system under different scenarios and analyse the result.

3.5.1 Experimental Setup

We deploy our system in a real indoor work space to conduct field experiments. The building we choose is 100 Harris St Sydney that has a complex indoor structure. Inside the building, we can find areas with not only multiple rooms and corridors, but also large empty halls. We deploy our Beacon devices on either the pillar or the ventilation duct to gain the best signal

¹<https://gorm.io/>

radiation. The first area we choose consists of 12 small rooms and a narrow corridor, with 6 Beacon APs deployed. We use this area to represent places like work or school. The second one is a large empty room with nine Beacon APs deployed. We use it to represent areas such as warehouses or train stations.

From Figure 3.4, we have a visual understanding of the testing areas. For the two testing areas we choose, they both have chairs, tables, plants, and other office supplies. Some Wi-Fi access points are installed inside the entire area, as well as Bluetooth access points so that other Wi-Fi or Bluetooth signals can interfere with the signal from our testing device, making the testing area closer to a real-life scenario where the area is always full of noise and obstacles.

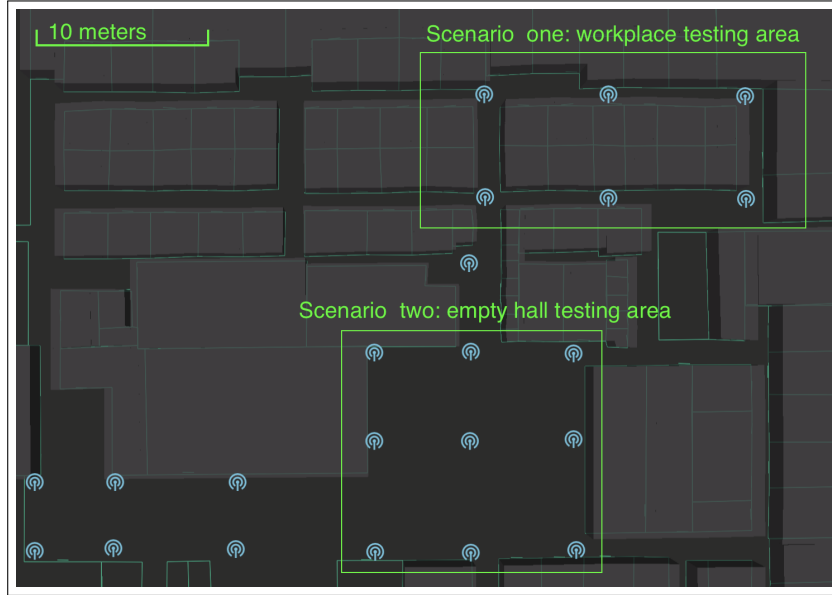


FIGURE 3.4. Testing areas.

3.5.2 Benchmark Approach and Testing Criteria

The benchmark approach that we select for our algorithm is the traditional Bayesian method. It is a probabilistic algorithm that was originally proposed by Youssef et al. [1]. The basic idea of the algorithm is to construct a radio map by collecting RSS, using the joint probability distribution as the checkpoint's fingerprint and using Bayesian algorithm to estimate the location of the user. In our GauPro algorithm, we use the Gaussian kernel to modify the prior probability in the Bayesian formula and implemented our own system for testing.

To evaluate an indoor positioning system based on Bluetooth technology, the most important properties of the system are error distance and precision:

- **Error distance** means the distance between our estimated position and the actual position. Before we run the test, at each location, we keep a record of the actual position. When we finish our positioning process, the algorithm will return an estimated position, that is, a pair of latitude and longitude. We then compare these two positions and calculate the distance between them, in metres.
- **Precision** measures the reliability and consistency of our algorithm. We can obtain precision by repeating multiple measurements under a persistent condition and see how many times the result has appeared. For example, during 100 experiments at one location, 50% of the time a result of less than 3m has appeared, 80% of the time a result of less than 5m has appeared, and 95% of the time a result of less than 10m has appeared.

To calculate the error distance and precision of our algorithm, we choose a start position and walk along a path consisting of testing positions. We repeat the experiment multiple times to make the result more reliable. Regarding the default orientation angle in Section II-C2, we set the east direction in Figure 3.4 as the angle of 0. Our algorithm will return a result by its latitude and longitude so that we can calculate the distance between the result and the testing points.

3.5.3 Positioning Result

We did separate experiments for both testing areas to ensure the performance of the system in different scenarios. We use tables and diagrams to analyse the result and compare our result with the traditional Bayesian positioning method to see if there is an improvement that improves the error distance and precision.

3.5.3.1 Scenario One - Empty Hall Area

This area has no walls and only contains a pillar, sofa, and tables. We performed positioning 12 times inside this area. The average accuracy in total is 2.543m, and the average standard deviation is 1.359. To make it easier for us to compare the results of both algorithms, we generate a chart (Figure 3.5).

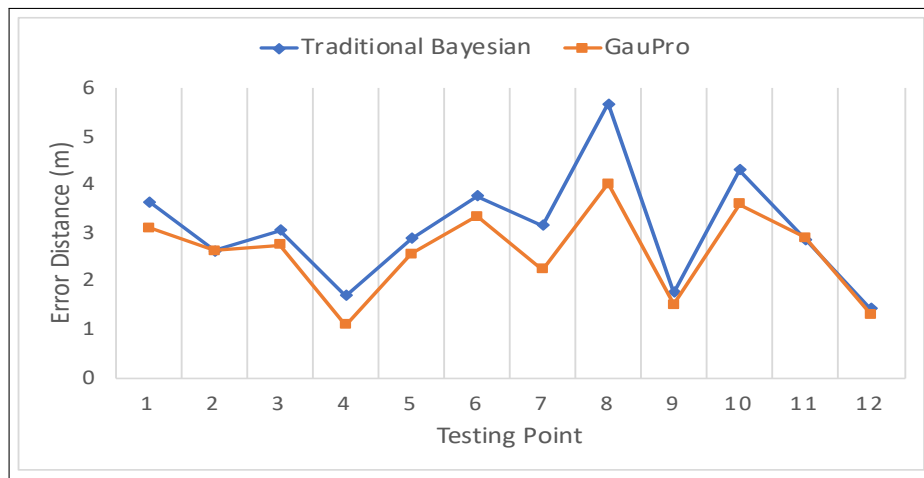


FIGURE 3.5. Comparison of error distances for the empty hall area.

We can tell from the chart that when implementing the improved algorithm, 10 of 12 test points can provide a better error distance. Compared to the traditional algorithm, our error distance has improved by 17.3%. We discover that the improved algorithm can help to eliminate some of the results with a large error distance. In other words, it can improve the worst scenarios of the original Bayesian algorithm. When we calculate the position with the original algorithm, every checkpoint in the testing area has the same prior probability, making the positioning results evenly distributed around the testing point. However, when we estimate a position and use the Gaussian kernel to assign the prior probabilities, checkpoints close to the estimated point will have higher priority. Thus, checkpoints far from the estimated position are not likely to be chosen by the algorithm. Hence, we can ignore the result that is not in the direction of the user's movement and usually has significant error distance.

We have another chart (Figure 3.6) to show the precision differences more directly. The improved algorithm can provide a better result in all three ranges because the results are more

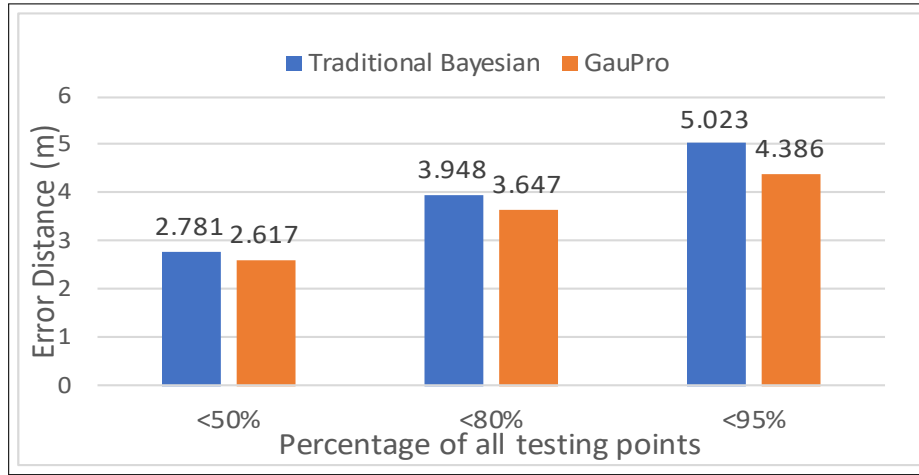


FIGURE 3.6. Precision comparison for empty hall area.

likely to gather around the predicted point rather than spreading across the map. Therefore, it is easier for our algorithm to get more consistent positioning results with higher accuracy.

3.5.3.2 Scenario Two - Workplace Area

Similar tests have also been performed in the workplace testing area. Inside this area, we have several glass walls and a pillar in each large room. We also have tables and chairs for people to use since it is a real work space. The average accuracy for this test area is 2.712 m, and the standard deviation is 1.125.

We run the same tests as with the original Bayesian algorithm. As shown in Figure 3.7, our algorithm has slightly improved 9 of the 12 results. Compared to the traditional algorithm, our error distance has improved by 8.1%. Figure 3.8 indicates that the precision for 50%, 80% and 95% of the time has been improved by 3.92%, 2.96% and 5.49% respectively.

3.6 Conclusion

In this chapter, we present the design, implementation and evaluation of a Beacon-based probabilistic fingerprint indoor positioning system. We estimate the user's relative position with the information from gyroscope and accelerometer and use the LSTM-based Gaussian kernel approach to improve the prior probability calculation in the traditional Bayesian method.

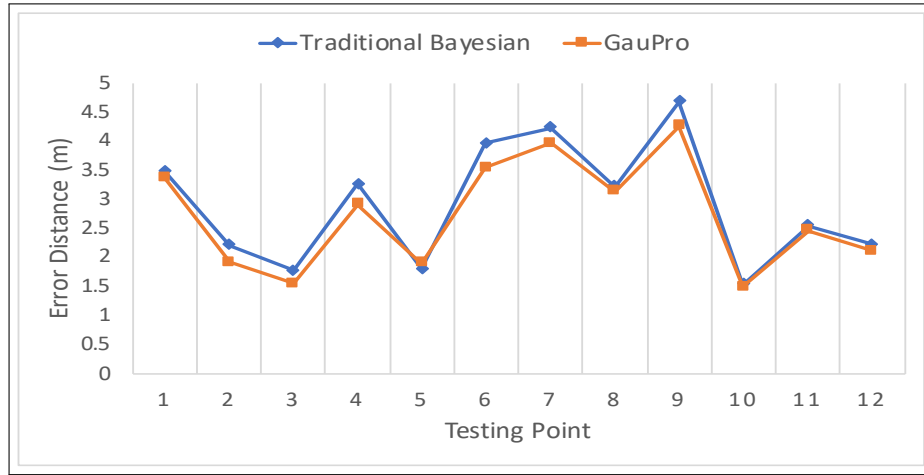


FIGURE 3.7. Comparison of error distances for the workplace area.

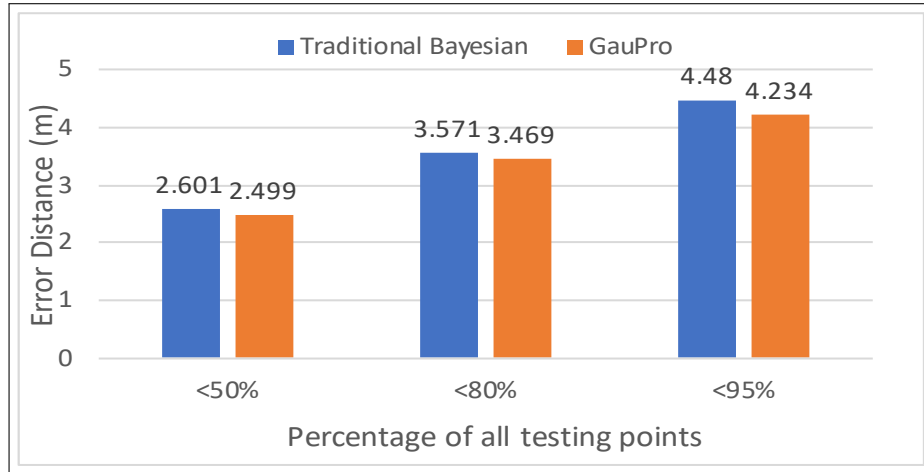


FIGURE 3.8. Precision comparison for the workplace area.

We set the traditional Bayesian method as the benchmark. According to the results of our experiments obtained, our system has fairly good error distance with an average of 2.543 and 2.712 metres for the two testing areas, and it improved the traditional algorithm by 17.3% and 8.1%. Our results appear to be more reliable and consistent compared to the original one. The experiments were done in a real workplace environment where we encounter various interference including Wi-Fi signals, other Bluetooth signals from people's phones and the signal fluctuation caused by multiple people walking inside the area at the same time.

Part 2

Computer Vision-Based Approach

With the proliferation of cameras in modern devices and advances in image processing, computer vision has become a viable method for indoor positioning. This approach uses images or video streams to identify pedestrians within indoor spaces and then calculate the user's position based on the camera projection matrix.

Deep learning, particularly CNNs, has revolutionised this domain. Instead of relying on handcrafted features, deep learning models can automatically extract and learn spatial hierarchies from raw image data. Techniques such as object detection, semantic segmentation, and multiple object tracking can be employed to locate and track specific objects or landmarks in the environment.

In this section, we address two primary challenges: improving object-tracking performance and improving object detection in densely crowded scenes. To tackle these challenges, we employ bipartite matching in conjunction with a Graph Attention Network (GAT). By treating the multiple-object tracking (MOT) problem as a graph-based challenge, we effectively identify the "break point" within tracklets. Furthermore, we have developed a customised YOLOX model capable of simultaneously detecting both a pedestrian's head and body. This is particularly advantageous for top-down view images from security cameras, where heads are often more discernible than bodies. Through our innovative approaches, we have achieved state-of-the-art detection and tracking performance on widely recognised MOT benchmarks.

Symbol Used in Part 2

Symbol	Description
c	tracked object
d	detection
e_{ij}	edge between nodes i and j
F_{ij}^a	cost function for appearance distance
F_{ij}^b	cost function for spatial distance
x_i	x coordinate of a detection
y_i	y coordinate of a detection
w_i	width of a detection
h_i	height of a detection
$\mathbf{r}_i$	the node features
$\mathbb{W}$	self attention weight matrix
$\mathbf{z}$	weight vector for the shared attention matrix
C_{ij}	pairwise matching degree
L_{ij}^{cls}	the classification loss
$L_{ij}^{hreg}, L_{ij}^{breg}$	regression loss for head and body

Graph Attention Network based Multiple Object Tracking

4.1 Introduction

Multiple object tracking (MOT) plays a pivotal role in transportation systems[156]. Tracking pedestrians & vehicles accurately is a huge topic in many fields such as surveillance, traffic management, and autonomous driving. With an accurate tracker, machines can automatically detect and track objects on the screen and record their movements without people being around.

To effectively solve the MOT problem, previous work has proposed various solutions, among which Tracking by detection (TBD) is the most popular approach for MOT in recent years. It divides the problem into four sub-problems: object detection[ren2015fasterm, 154], object feature extraction[29, 32], data similarity computing[15, 140] and data association[100]. These problems have been investigated jointly [159] or independently [120] to improve MOT performance. For example, the object detection problem alone can be considered a separate challenge that has formed a specific research field. In this work, we focus on the data association problem and build an effective object tracker for MOT.

The main challenge of the data association problem in MOT is the high complexity and unpredictability of human behaviours in the tracking scene[156]. More precisely, when multiple people move at the same time in a real scene, occlusion occurs. For example, when two or more people walk towards each other, one may be occluded by others or by obstacles. It then becomes invisible to the camera, and thus the tracking algorithm will lose the raw input

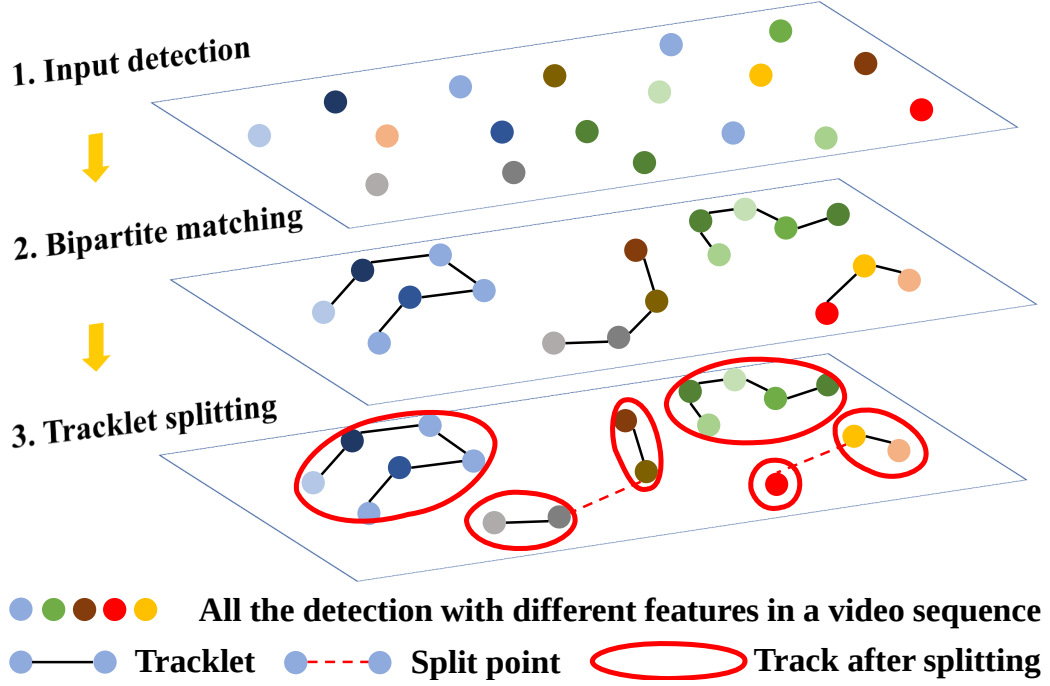


FIGURE 4.1. Concept of GATracker. 1) Given all the detection in a MOT video sequence, 2) we produce raw tracklets by clustering detection with similar features in time order with bipartite matching. 3) Then, we train a graph attention network model to separate mismatched tracklet.

from the detection. Occlusion is the most challenging problem in MOT, and many previous studies have focused on it.

[2] is one of the most successful and widely known approaches following this paradigm. Using a simple bipartite matching algorithm, it uses Kalman filter to estimate the movement of existing objects and match them to the next frame detection. [116] is the direct improvement to this approach where they leverage a pretrained deep appearance descriptor to help extract the visual features from the detection. These real-time approaches work fine in simple scenes, but do not perform well on very crowded scenes due to heavy occlusion.

Solving MOT offline by modelling it as a graph problem has a long history, and the [78] algorithm is one of the first studies to take advantage of a node-disjoint algorithm proposed in 1974 to tackle the problem. [14] proposed the additional lifted edges for the node disjoint path problem and has achieved satisfactory results. These methods treat MOT as a graph

problem and rely on highly hand-crafted mechanisms to solve the data association problem by finding node-disjoint paths.

Recently, graph neural networks (GNNs) have been proposed, and they are instrumental in treating problems where the input data is formulated as graphs. [25] models the tracking problem as an edge classification problem and solves the problem with a message passing network. [120] proposed a two-stage tracker in which detection is first clustered with their features and then used a graph convolution network (GCN) [114] to estimate the proposal score. The advantage of this approach is that the GCN can find high-score tracklets with similar features and thus keep the tracking precision high. However, it struggles to score long tracklets that consist of several different identities.

Inspired by [2] and [120], we propose GATracker, which is a two-stage object tracker based on the graph attention network (GAT). There are two main steps in this approach: 1) generation of tracklet proposal and 2) classification of nodes with a GAT [148]. As shown in Figure 4.1, given all the detection for an MOT tracking sequence, we first cluster the detection with similar features with a bipartite matching algorithm. Instead of matching the detection with the motion prediction of the last frame’s detection, we construct a list of tracked objects and match them. We define specific rules and parameters so that tracklets can be properly initialised or dynamically deactivated. The outputs of the bipartite matching are several independent tracklets with potentially a lot of mismatched clips. Each tracklet can be represented as a single undirected graph, with each node representing one single detection among the whole tracklet. Second, a pre-trained GAT is used to compute the hidden representations of each node. By specifying different weights to different nodes of their neighbourhood, the model finds out the split point where the identity switch happens (i.e., the nodes after the split point belong to a different object compared to the nodes before the split point). In this way, we split the tracklets generated in the bipartite matching step and produce more accurate association accuracy and identity preservation results.

The contribution of this work is threefold: 1) For the tracklet proposal, we apply a well-designed bipartite matching algorithm to associate detection with a tracked object list, which can ensure we cluster independent tracklet with high feature similarity while maintaining

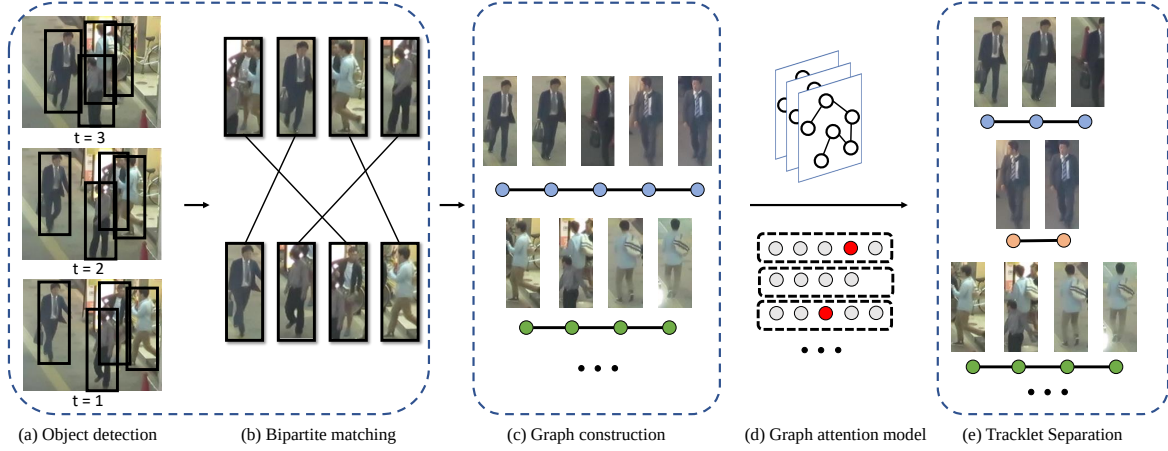


FIGURE 4.2. Overview of our proposed method. (a) We receive detection results from object detection model and extract the visual features with a pretrained ReID model. (b) We initialize the tracking by taking the detection result from the first frame as starting objects, and match new detection with the objects. The matching process consider both the spatial-temporal features and the appearance features. (c) We take the generated tracklet proposals, construct graphs according to their features. Each node represent one frame of detection in the whole tracklet. (d) We feed the graphs into our trained GAT model to find out the split point on each tracklet (i.e. to label every node in the graph). (e) We separate the tracklet according to the output of our GAT model to obtain the final tracking result.

low computing costs. 2) In the fashion of treating MOT as a graph problem, we first utilise the graph attention mechanism to model the nonlinearity and inner connection inside the tracklet proposal. We employ a GAT to classify the node of each tracklet by training the model to extract features from each node and their neighbourhood nodes. 3) We gain a clear improvement in HOTA and IDF1 score on two main MOT Challenge benchmarks.

4.2 Related Works

General solution for MOT. Tracking by detection has been the de facto solution for MOT for many years[156]. Since 2016 the object detection technique has evolved a lot and the detection accuracy has become higher and higher. Many state-of-the-art detectors have greatly contributed to MOT[58, 90, 117, 135, 165]. With the detection results provided, researchers have proposed many effective data association algorithms such as [14, 23, 53, 69, 82, 85, 105,

110, 132, 137, 140, 173]. These approaches can then be further classified into two fashions: online and offline. Online approaches deal with input detection and provide tracking results in real time. Off-line methods take the input as batches and leverage the spatial-temporal information across all frames to better identify the tracklets. For example, [2] use the Kalman filter to predict the movement of the last frame's objects and match them with the current frame's detection. [116] use the same approach but integrate the appearance information based on a CNN model. For offline approaches, [78] construct a graph based on the detection of all frames and find tracklets with a node disjoint path algorithm[152]. [14] extend the node disjoint path problem by introducing an additional elevated edge to the graph.

Object Detection in MOT. Many recent research pieces improve the performance of the tracker by training a good detector. [121] solves the tracking problem with a detector. [142, 160] takes two pairs of frames as input and predicts the location of the next frame of the object with a transformer model[17]. [144] considers the object permanence and builds a detector that can detect the object even when it is not shown directly in the scene. These approaches made great progress in the multi-object tracking accuracy (MOTA) but suffer a lot from the object identity tracking accuracy. This means that there is a lot of ID shift between targets. Often, one tracklet from their result will cover two or more individuals, which will greatly reduce association precision and eventually the IDF1 score, representing the overall preservation of identity of the tracking results.

Object Association. For the data association problem specifically, there has also been a lot of research done in the last decades. To handle the occlusion caused by the human crowd, [8] was proposed to estimate the movement of the crowd with an RNN. [4] combines the sequence prediction with a generative adversarial network to make a better prediction of crowded human movement. [12] use RNN to track objects in the long term by jointly deciding on multiple cues over a temporal window. [25] models the MOT problem as an undirected graph and solves the problem based on message-passing networks. The network learns to combine deep features into high-order information across the graph. [44] proposed an occlusion handling strategy by directly modelling the relation between occluding and occluded tracks without using a separate feature extraction model. [120] proposed a two-stage framework

that is similar to Faster RCNN. They first group detection into tracklet proposals and score them with a trainable GCN. Finally, a de-overlapping strategy is adopted to generate a track while avoiding reusing the same detection in multiple tracklets. This approach provides tracking results with high association accuracy, but has limited performance in many tracking sequences with a low camera height. Many detections with similar features are not clustered, leading to relatively low performance in those sequences. Qualitative results show that our approach clusters more detections with similar features, thus leading to a higher tracking rate.

Following the two-stage tracking framework introduced by [120], we propose our multiple object track, which uses a GAT to study the nature of each proposed tracklet. Our work shares a similar idea from [40]. They use a Siamese Bi-Gated Recurrent Unit to split the track and reconnect them with another Siamese GRU network. Compared to their work, our model utilises a different approach when splitting the tracklet and considers both spatial-temporal features and visual appearance features instead of visual features only. In addition, we improved the bipartite matching stage by averaging the features of the tracked objects. More details are explained in Section 4.3.2.

4.3 Methodology

4.3.1 Overview

Motivation for the proposed approach. Human can track single objects quite easily because our brain can distinguish the original tracking target and interference based on many facts: their visual appearance, their location compared to the last frame, their current state (e.g. walking, running, sitting) and the environment around them. As shown in Figure 4.2, after bipartite matching, one of the generated tracklets contains two persons who share similar appearance features, and their physical distance is also close, so the matching algorithm mismatched them together. We can handcraft an algorithm to analyse the tracklet based on the factors we have described above, but the better option is to train a proper deep neural

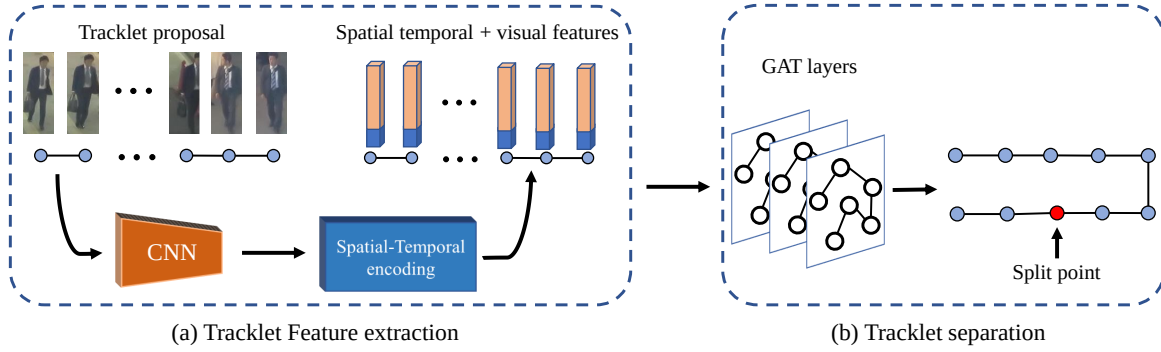


FIGURE 4.3. Example of tracklet visual extraction and processed with GAT to find the split point.

network (DNN) model. DNN can model complex non-linear relationships, suitable for our task since the factors impact each other, and it is not trivial to model them by hand.

Overview of the Method. Given all detection bounding boxes for a video sequence, our goal is to produce tracklets as tracking results and preserve the identity of the tracking object as long as we can. The first step is to extract visual features from the given detection. Then we group up detection with similar appearance features and spatial-temporal features with a simple bipartite matching algorithm to ensure each detection belongs to one unique tracklet. These tracklets are usually very long and consist of two or more objects (pedestrians). Next, we analyse each tracklet with a pre-trained graph attention network (GAT) and label each node in the proposed tracklets. Based on the information provided by the node itself and its neighbour nodes, the model judges whether there is an identity switch happening at this node. If there is, the model will label the node as a split point. After generating all the labels for the tracklets, we separate them according to the label and obtain the final results.

4.3.2 Feature Extraction and Bipartite Matching

Initialize Tracking. At the start frame of each tracking sequence, we initialise a list of the tracked objects according to their detection. Each time a new frame comes in, we calculate the feature distance between the detection and objects on the tracked list. Thus, each new detection will always be assigned to one existing object unless the number of new detections is more than the objects. In that case, the unassigned detection will be considered as a new

object and saved in the tracked object list. Then we check all assignments and update tracked objects if the appearance distance between the detection and the object is less than a threshold K . If it is larger than K , the detection will also be considered a new object. Also, to avoid stacking too many objects in the tracked list and save computing resources, we deactivate objects that are not updated for more than L frames. Those objects will be removed from the tracked list and never updated again.

Feature Extraction. During the bipartite matching step of our method, we extract and use two main features in our model - spatial-temporal features and visual appearance features. The spatial-temporal feature represents the size and location of the detected bounding box and its time frame location. With this feature alone, [2] can already solve the MOT problem. The visual appearance feature represents is very intuitive and represents the outward appearance of the object. This feature is used mainly in the research of human re-identification [68, 108, 65]. Usually, the feature vector is extracted with a CNN model directly from an RGB image of the detection bounding box.

Assignment Problem. To cluster detection with similar characteristics and ensure the independence of each tracklet, we use bipartite matching, more specifically the Hungarian algorithm [151]. Given the detection list at time t with k detection $D^t = \{d_1^t, d_2^t, \dots, d_k^t\}$ and a list of tracked objects $C^t = \{c_1^t, c_2^t, \dots, c_n^t\}$, we construct a bipartite graph $G = (S, Q; E)$ with n tracked objects as worker vertices S , k detection as job vertices Q and each edge E has a non-negative cost $F_{ij}(c_i^t, d_j^t)$. We want to match D^t to C^t at a minimum cost:

$$\operatorname{argmin} \sum_{e_{ij} \in E} F_{ij}(c_i^t, d_j^t) e_{ij} \quad (4.1)$$

where $e_{ij} \in \{0, 1\}$ indicates whether there is an edge between c_i^t and d_j^t . The cost consists of two sub-cost functions representing the cost of appearance feature and spatial-temporal feature, respectively. They are indicated as $F_{ij}^a(\cdot)$ and $F_{ij}^b(\cdot)$. We calculate the cosine distance between the visual features of the detected object and our tracked object to obtain $F_{ij}^a(\cdot)$ and the Euclidean distance of the bounding box distance to obtain $F_{ij}^b(\cdot)$.

Most tracking scene changes very fast, and so does the appearance features of the tracked objects. To make the appearance feature more robust when we do the matching, we average the visual feature vectors of the object's recent detection to make it more stable, and a sudden change of appearance will not affect too much of the overall features of the tracklet. More specifically, let the worker vertex (tracked object) $c_i^t = (\mathbf{a}_i^t, \mathbf{p}_i^t)$ where $\mathbf{a}_i^t$ is the averaged characteristic vector of the last R frames of the tracked object. $\mathbf{p}_i^t$ is simply the corresponding 2D detection coordinates denoted as $\mathbf{p}_i^t = [x_i^t, y_i^t, w_i^t, h_i^t]$ where x, y is the top left coordinate of the bounding box and w, h represents width and height. For the job vertex (new detection), the feature is defined as $d_j^t = (\mathbf{a}_j^t, \mathbf{p}_j^t)$. The only difference is that the new detection only has one frame, so we use their visual feature $\mathbf{a}_j^t$ directly. With the definition above, we can calculate the two sub-costs:

$$F_{ij}^a(\mathbf{a}_i, \mathbf{a}_j') = \frac{\mathbf{a}_i \cdot \mathbf{a}_j'}{|\mathbf{a}_i| \cdot |\mathbf{a}_j'|} \quad (4.2)$$

$$F_{ij}^b(\mathbf{p}_i, \mathbf{p}_j) = f \cdot (\mathbf{p}_i, \mathbf{p}_j) \quad (4.3)$$

where f represents the calculation of Euclidean distance. After having the appearance cost and the bounding box cost of each pair of detection and objects, we normalise the cost and add them together to formulate the final cost matrix:

$$F_{ij}(c_i^t, d_j^t) = ||\lambda F_{ij}^a(\mathbf{a}_i, \mathbf{a}_j') + (1 - \lambda) F_{ij}^b(\mathbf{p}_i, \mathbf{p}_j)|| \quad (4.4)$$

Combining formulas 4.1 and 4.4, we assign detection to our tracked object list frame by frame. The generated tracklets are used as input to our next step.

4.3.3 Classifying Nodes in Tracklet

Given tracking proposals generated from last step, we perform node classification with a GAT to identify the identity change in each tracklet. The input to the network is a tracklet which is also a graph $T_i = \{V_i, E_i\}$ where V_i represents the nodes and E_i represents the edges. The nodes are connected by time order in each tracklet. The GAT takes in a tracklet with N nodes

as input and outputs the probability set $P = \{p_i\}_{i=1}^N$ where each P_i is the probability that the corresponding node is a split point.

4.3.3.1 Feature Encoding

To obtain the feature for the input graph, we need to design an encoding that can represent the relative position, appearance, and time change of each pair of nodes. Both the spatial-temporal difference and the appearance difference are important. For a detection v_i at time t_i and its previous detection in the tracklet v_{i-1} at time t_{i-1} , their top left coordinate in the image and their width and height are represented as (x_i, y_i, w_i, h_i) and $(x_{i-1}, y_{i-1}, w_{i-1}, h_{i-1})$. We consider their relative position change as:

$$\left(\frac{2(x_i - x_{i-1})}{w_i + w_{i-1}}, \frac{2(y_i - y_{i-1})}{h_i + h_{i-1}}, \log \frac{w_i}{w_{i-1}}, \log \frac{h_i}{h_{i-1}} \right) \quad (4.5)$$

The extracted visual features $\mathbf{a}_{v_i}^{t_i}$ and $\mathbf{a}_{v_{i-1}}^{t_{i-1}}$ are already obtained in the previous step, so we can simply reuse them. We calculate their relative visual characteristic change $\|\mathbf{a}_{v_i}^{t_i} - \mathbf{a}_{v_{i-1}}^{t_{i-1}}\|$ and concatenate it with the relative position change we obtain above and also their time difference $t_i - t_{i-1}$ to produce the characteristic encoding $\mathbf{r}_i$ for each node in the graph. For the first object in each tracklet, $\mathbf{r}_i$ is set to all zeros.

4.3.3.2 Graph Attention Network

The graph attention network accepts the node features and their connectivity as input. As shown in Figure 4.3 the whole network is composed of M layers, the attention mechanism is performed by the following formula:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{z}^T [\mathbb{W}\mathbf{r}_i || \mathbb{W}\mathbf{r}_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\mathbf{z}^T [\mathbb{W}\mathbf{r}_i || \mathbb{W}\mathbf{r}_k]))} \quad (4.6)$$

$\mathbf{r}_i$ is the node features that each GAT layers gets, and $\mathbb{W}$ is the self attention weight matrix. $\mathbf{z}$ is the weight vector for the shared attention matrix. $||$ represents concatenation and T means transposition. $\mathcal{N}_i$ is a set of all neighbour nodes of node i in the graph. After applying the

LeakyReLU nonlinearity and the softmax operation with the results between node i and its other neighbours, we get the coefficients between node i and j .

After we have computed all the shared attention coefficients between node i and all its neighbours, the results are averaged to form $\mathbf{r}'_i$, which is the output for this node in the current layer and the input for the next layer in GAT.

Example of Node Classifying. GAT can help us model the non-linearity introduced by the object features. Taking the examples from Fig. 4.4, there are two tracklets generated by our bipartite matching step. $\Delta t, \Delta a, \Delta p$ represents the relative temporal, visual and spatial characteristics differences, respectively. They both have significant feature changes at frame 6, but we only need to separate the second tracklet according to the ground truth. Defining a heuristic function to summarise the pattern is not trivial since the tracking sequence is complicated. So, we separate the tracklets with the GAT model. We choose a node as the starting point and move in both temporal directions to explore further useful information. According to formula 4.6, the model aggregates the features of neighbouring nodes, and the attention mechanism learns the inner connection between the node and its neighbourhoods. Note that we do not need to go through the whole graph to determine the property of one node. The “receptive field” of the target node is enough for the model to make the judgment. The deeper we build our network, the further each node can explore. After gathering enough information, the model gives a prediction of each node about whether we should separate the tracklet at that point.

4.3.3.3 Training and Inference

The goal of GAT is to correctly classify the nodes in the tracklet. For a node i , we want to see whether the node is a split point or not. We can use binary cross-entropy loss for such classification. However, in practise, most of the node is not a split point; hence, the sample imbalance is a problem. Thus, we use focal loss with a parameter of $\alpha = 0.25$ and $\gamma = 2$ to address this issue.



FIGURE 4.4. Example of two tracklets with large feature changing.

During inference, after generating tracklet proposals from bipartite matching, we construct the graph and process it with the model. The model will produce numerical predictions. We binarize the output by choosing the indices of the maximal value for each prediction and separate the tracklets according to the output to generate our final track results.

We provide a pseudocode explanation of our GATracker algorithms in the supplementary materials.

4.4 Experiments

4.4.1 Datasets and Evaluation Method

To test the tracking algorithms under complicated scenes with frequent occlusion and various camera settings, the MOTChallenge provides several challenging tracking benchmarks. We choose the most recent MOT17 [16] and MOT20 [119] as our training and testing sequences.

Both sequences contain different sizes, speeds, and appearances of pedestrians. The camera frame rate, resolution, angle, and movement also vary much. To test our tracking algorithm, we pre-process the public detection with existing tracking algorithms. This strategy has been widely approved in many published methods [25, 102, 120]. Note that the performance results are generated on the MOTChallenge website for benchmark evaluation by uploading the tracking results. The ground truth file for test sequences is not provided to researchers. All ablation studies were conducted based on the MOT17 training dataset and detection and extraction of features were based on PermaTrack[144].

We use metrics provided by MOTChallenge [50, 89, 93] including Multiple Object Tracking Accuracy (MOTA), ID F1 score (IDF1), Higher Order Tracking Accuracy (HOTA), Mostly Track targets (MT), Mostly Lost targets (ML), False Positives (FP), False Negatives (FN), ID switches (IDSw), etc. They cover different aspects of the MOT problem and can indicate the performance of a tracking algorithm. Traditionally, MOTA and IDF1 are considered more important because MOTA measures the detection bounding box coverage and the other measures the preservation of object identity. As a recently proposed metric, HOTA explicitly measures both types of errors and combines these in a balanced way. It also measures aspects that are not present in MOTA or IDF1, such as localisation accuracy. Therefore, HOTA is also a preferred metric in the MOT problem.

4.4.2 Implementation Details

The ReID model follows the same structure of [120] which uses ResNet50 as the backbone and replace batch norm layer with instance-batch-norm (IBN) layer, and a batch norm layer and a classifier layer is added after the global average pooling layer. As a common practise, the model is pre-trained on Market1501 dataset and finetuned on MOT17 dataset to gain better performance.

For the bipartite matching part, the feature distance threshold K is set to 0.5 and the object deactivation parameter L is set to 200.

The GAT consists of 4 layers with the output feature nodes [512, 128, 32, 2] and the number of heads is set to 4, 4, 6 and 1. The drop out is set to 0 and no skip connection is added. The model is trained end to end with Adam optimiser with a learning rate of $5e^{-5}$ and with a weight decay of $5e^{-6}$ on the MOT17 dataset. We train for 500 epochs to get our results.

Data Augmentation. The MOTChallenge dataset is very small, so we need to do data augmentation to increase the size of the training data. We randomly clip the tracklet with a length between 1 and 30 frames from ground truth and concatenate two tracklet clips together to form one training tracklet. We also concatenate clips from the same ground truth to simulate long occlusion. Among all the training datasets, we choose 50% of them to be the training set, 30% of them to be the validation set and the rest of the test set.

Batch Processing. Batch processing for GAT is not as easy as in other neural networks. The GAT tensor manipulation framework for sparse matrix multiplication only supports rank-2 tensors, which limits the batching capabilities of the layer [148]. Hence, if we want to train multiple graphs together in an inductive setting, we need a different implementation of GAT since the original implementation is only suitable for a transductive setting. Inspired by [5] we merge the graphs into a single graph with multiple components during training and implement the layer in a way that is agnostic to this fact. This allows us to batch our input and significantly improve our training efficiency.

4.4.3 Ablation Study

4.4.3.1 Bipartite Matching

To test the influence of the bipartite matching step in the proposed approach, we skip the bipartite matching by taking the tracking results directly from [144] and generate the graph input. We compare the result with the one we add the bipartite matching. The results are summarised in Table 4.1. As we can see, for the one that does not use bipartite matching, the MOTA and IDF1 score is slightly improved compared to the original results, because the model can identify mismatched tracklets and correctly separate them. However, in the original result of PermaTrack, they have a lot of tracklet clips that belong to the same object that are

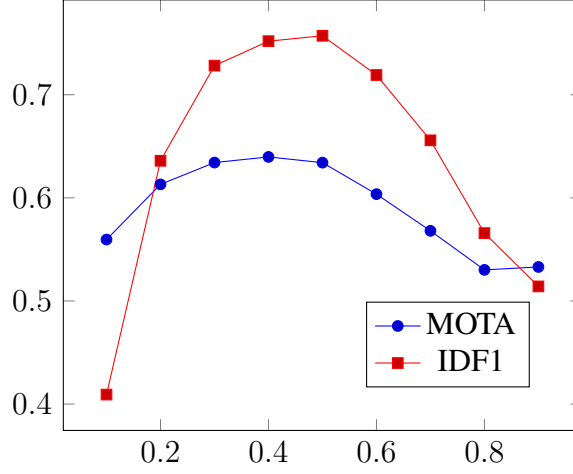


FIGURE 4.5. MOTA and IDF1 vs K values.

TABLE 4.1. Performance comparison with bipartite matching is not performed.

.	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓
Permatrack[144]	73.7	70.0	881	200	20887	65984	1843
GAT only	74.1	70.4	872	208	18840	66382	1921
Matching+GAT	73.9	78.2	777	224	9669	74450	3936

not clustered. When bipartite matching is added, the IDF1 score is significantly improved, as expected. The reason is that the bipartite matching step can help us cluster detection with similar features as much as possible and preserve the object identity.

The feature distance threshold K is critical in the bipartite matching step. We have carried out experiments with different K values, and the results are shown in Fig. 4.5. As we can see, the MOTA and IDF1 scores are both sensitive to the K value. We can see that when we gradually increase the value of K from 0.1 to 0.5, the MOTA and IDF1 score also increase. The performance starts to drop after $K > 0.5$ because Hence, we choose $K = 0.5$ to obtain the best object identity preservation percentage.

4.4.3.2 Graph Attention Network

The Need for GAT Model. To test the importance of GAT in our method, we remove the whole network and only use the results generated in step 1. The comparison is shown in Table

TABLE 4.2. Performance comparison when GAT model is not included (top: detection from [120]; bottom: detection from [144]).

Model.	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓
No Model	56.6	65.9	734	747	46725	196462	1550
With Model	58.9	68.7	728	763	30391	200308	1375
No Model	73.8	77.0	777	229	9672	74453	4166
With Model	73.9	78.2	777	224	9669	74450	3936

TABLE 4.3. Performance comparison with state-of-the-art on MOT17 and MOT20. Detection preprocess by Tracktor, PermaTrack and TransCtr.

Model.	HOTA↑	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	AssA↑	DetA↑	IDs↓
MOT17 - Preprocessed by [121]										
jCC[110]	42.5	51.2	54.5	20.9	37.0	25937	247822	44.2	41.2	1802
FWT[126]	39.0	51.3	47.6	21.4	35.2	24101	247921	36.8	41.9	2648
NOTA[104]	42.3	51.3	54.7	17.1	35.4	20148	252531	43.6	41.3	2285
FAMNet[111]	-	52.0	48.7	19.1	33.4	14138	253616	-	-	3072
Tracktor++v2[121]	44.8	56.3	55.1	21.1	35.3	8866	235449	45.1	44.9	1987
Lif_TsimInt[14]	50.7	60.5	65.6	27.0	33.6	14966	206619	54.9	47.1	1189
MPNTrack[25]	49.0	58.8	61.7	28.8	33.5	17413	213594	51.1	47.3	1185
LPC[120]	51.5	59.0	66.8	29.9	33.9	23102	206948	56.0	47.7	1122
Ours	52.5	58.9	68.7	31.0	32.4	30391	200308	57.4	48.2	1375
MOT17 - Preprocessed by [144]										
PermaTrack[144]	54.2	73.1	67.2	42.3	19.1	24577	123508	51.2	58.0	3571
Ours	56.1	72.6	71.2	41.3	19.1	25001	123872	55.2	57.6	5851
MOT20 - Preprocessed by [160]										
TransCtr[160]	43.5	61.0	49.8	48.4	15.5	49189	147890	36.1	52.9	4493
Ours	50.6	60.7	61.6	48.6	15.1	48356	147468	48.2	53.4	7548

4.2. We test with [120] and [144] detection separately. As we can see, both MOTA and IDF1 are worse when GAT is not used. The tracklets generated by bipartite matching contain many mismatched trajectories due to the close feature distance among the detected bounding boxes. Therefore, the result contains more ID Switch (IDSw), leading to lower MOTA and IDF1 scores. However, the result produced without the model contains higher MT and ML scores, so there is a trade-off between tracklet number and tracklet quality. From the result we can see that the GAT model can improve performance by correctly detecting identity changes in tracklets and separating them.

TABLE 4.4. Performance comparison with different feature encoding.

Feats.	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓
Spat-Temp	73.7	75.5	777	229	9672	74453	4613
App	72.9	73.4	780	222	9670	74451	7083
App and Spat-Temp	73.9	78.2	777	224	9669	74450	3936

TABLE 4.5. Performance comparison between GAT and GCN.

Model.	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓
GCN	73.8	76.8	777	227	9672	74453	4272
GAT	73.9	78.2	777	224	9669	74450	3936

TABLE 4.6. Performance comparison when tracklet reconnect is performed.

.	MOTA↑	IDF1↑	MT↑	ML↓	FP↓	FN↓	IDs↓
Reconnect	73.4	77.0	777	227	10915	74322	4522
No Reconnect	73.9	78.2	777	224	9669	74450	3936

Different Feature Encoding. We test the effectiveness of the spatial-temporal feature and the visual appearance feature separately. As shown in Table 4.4, with the appearance feature alone, we can achieve higher MT/ML scores. The reason could be that without considering other features, the appearance features help to preserve more object identities to track more objects and avoid losing track of some challenging tracklet. The spatial-temporal features help to reduce the ID switch, and thus improve the MOTA score. When we concatenate them together, the model can learn the inner connection between these two kinds of feature and provide better performance. In [40] the GRU network only takes in visual features and thus lacks a vital part of the information.

GAT vs GCN comparison. We conduct experiments to compare the performance of GCN and GAT in our proposed method, since both can process graph input. As shown in Table 4.5, GAT achieves better performance in both MOTA and IDF1 score. This verifies that GAT performs better in our node classification problem. The reason is that GAT is more expressive and allows the model to learn the neighbourhood aggregation coefficients where GCN uses constants instead of learnable weights.

4.4.4 Benchmark Evaluation

We compare our results on MOT17 datasets with several methods published on the MOTChallenge site, and the results are shown in Table 4.3. Our approach achieves state-of-the-art results by achieving the best Higher Order Tracking Accuracy (HOTA), IDF1, Mostly tracked (MT), Mostly lost (ML) and False Negative (FN). We use the same public detection with LPC [120] and MPNTrack [25] and perform the same post-processing (i.e., bilinear interpolation). We have improved the HOTA score by a small margin of 1.0 percentage points and the best IDF1 score by 1.9 percentage points. The improvement in MT, ML score, and FN is based on our approach to successfully clustering objects with long occlusion. And by adding an interpolated bounding box, we can track the undetected object, reducing the FN and improving the MT/ML. Our approach also performs better in Association Accuracy (AssA), proving that our method preserves more object identity.

We also take several other state-of-the-art tracking methods and use their detection results as input to produce our tracking results on both MOT17 and MOT20. Note that we only train our model on MOT17 and use it directly on MOT20. We do not add the same post-processing since the original methods do not use them as well. The improvements are based on purely detection re-associating. The results are also shown in Table 4.3. Compared to the original method, while maintaining similar MOTA scores, our approach can significantly improve the IDF1 score by 4.0 and 11.8, respectively. The HOTA score is also 1.9 and 7.1 percentage better, demonstrating overall better performance since it acts as a single unified metric for ranking trackers. The AssA improved by a large margin by 4.0 and 12.1 percentage scores as well.

Compared to the original method, our approach can preserve more object identities and achieve better performance than the detection provider methods. One problem with our method is that our result produces a higher ID switch (IDSw) than the original methods, reducing our MOTA score. A possible reason is that the detection bounding box is tiny in some scenarios, so both the visual and spatial-temporal features vary a lot constantly. Thus, the model failed to recognise those tracklets and produced incorrect predictions. Improving the

performance of such tracklets while maintaining the ability to separate other kinds of tracklet correctly is a future research direction for us. Given a fixed number of public detections, a key performance issue in the MOT challenge is the trade-off between MOTA and IDF1. As we have discussed above, our approach achieves a higher IDF1 score with the cost of a relatively smaller number of FP and IDs increasing, thus increasing the HOTA score.

4.5 Conclusions

In this work, we propose GATracker, a two-stage tracker for the MOT problem. We cluster detection with similar spatial-temporal and visual appearance features with a simple bipartite matching process and generate tracklet proposals. Then we construct a graph for each tracklet proposal and use a GAT model to search for split points. The model leverages the attention mechanism to specify different weights to different nodes in the graph. Attention is obtained by stacking features of a node’s neighbourhoods. The model is trained in an inductive setting so that it can operate on any other tracking sequence. We extensively studied the impact of each component of our method in the ablation study. Our method achieves state-of-the-art results on the MOT17 and MOT20 benchmarks compared to existing public detection approaches.

Joint Head Body Detection for Crowded Environment

5.1 Introduction

Tracking by detection is one of the most critical frameworks for the MOT challenge. Since the modern MOT challenge is now prone to happen in more and more crowded scenes, the tracker performance is limited by the design of the object detectors. With a new detector, it is possible to achieve near state-of-the-art performance in terms of tracking accuracy with a traditional tracking algorithm. In the meantime, many approaches with novel architectures have been proposed for the pedestrian detection task. Earlier works like Faster-RCNN [135], Mask-RCNN [92], SSD [154] use anchor-based model framework, while recent research on anchor-free detection has started to attract more and more attention [136].

The key challenge of pedestrian detection is the occlusion of the scene. In crowded scenes, people overlap with each other most of the time. The challenge of occlusion makes it difficult for the detection model to extract useful features to identify the person effectively. It also causes a problem with the Re-ID functions with some models because the visual features required to identify the person constantly changes due to occlusion. Multiple previous researches have stated that exploiting multiple parts of a pedestrian helps to improve the performance of pedestrian detection [34, 35, 37].

According to previous research [35], human body detection is often missing due to large overlaps in extremely crowded scenes. The model failed to identify enough features for the occluded person or eliminate them during the non-maximum suppression (NMS) postprocess procedure. Compared to body detection, the occlusion between heads is less likely to occur.

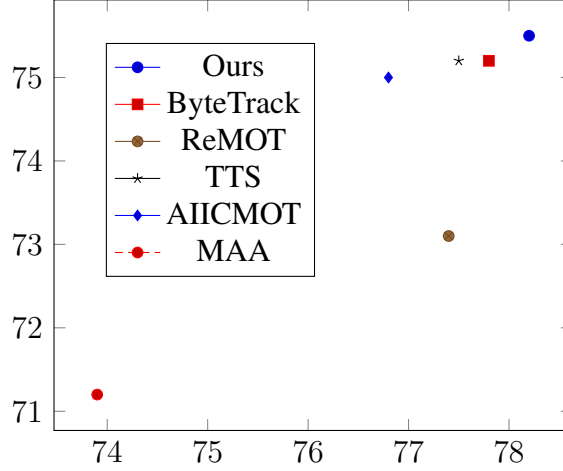


FIGURE 5.1. MOTA-IDF1 comparisons of different trackers on the test set of MOT20. With the help of our supplementary body detector, the proposed tracking framework achieves 78.2 MOTA and 75.5 IDF1 scores, outperforming all previous trackers.

With this consideration, the feature map can become more distinguishable and consistent for training. By comparing the head detection results and body detection results on the same MOT challenge sequence, we find that the head detector generally identifies more people than the body detector (except for those people not showing full bodies in the scene). Recently, [123] introduced a new head tracking data set to further assist humans in efficiently tracking in densely crowded environments. Therefore, by combining head tracking with body tracking detectors, we should be able to improve the performance of the MOT challenge.

However, adopting head-to-body tracking is not a trivial task. First of all, many pedestrians do not show their heads in the camera view. The head detector alone cannot fully replace the body detector in the context of MOT. Also, in a traditional detector, human heads are detected separately. Manually generating body bounding boxes from head detection introduces lots of false positives as people have different poses. Hence, detecting both the head and body simultaneously and then linking them together is a challenge. Bi-Box [37] proposed a model to estimate both the visible part and the whole body at the same time. JointDet [35] and HBAN[128] are two anchor-based solutions that rely on a fixed head-body ratio to generate the head proposal from the regular body proposal. The problem with these methods is that they still rely on anchors or external method to generate head and body proposals, leading to

low recall rate and inaccurate body Region of Interest (RoI). Although most standing humans follow a static head-body ratio, it still varies a lot due to different human poses and camera positions. Using a fixed head-body proposal to predict pedestrians in complex scenes leads to a less satisfying result.

Motivated by the findings, we propose JointTrack as a novel solution. The contributions of this work are threefold: 1). We adopt the joint head and body detector in an anchor-free style to further boost its detection performance in extremely crowded scenes, and a supplementary body detector is leveraged to overcome its limitation in the MOT challenge; 2). Instead of relying on a fixed head-body ratio, a new module to include both the head and body prediction based on SimOTA is proposed, which learns the relationship dynamically during training; 3). The proposed model achieves state-of-the-art head detection performance on CrowdHuman dataset, and the joint head and body tracking framework achieves state-of-the-art performance in the MOT20, HT21 challenges. It is worth noting that this work focusses on the image detection part, which can be used together with the GATracker proposed in Chapter 4.

5.2 Related Work

5.2.1 Object Detection and Occlusion Handling in Pedestrian Tracking

Pedestrian detection has been widely studied for its significance in many real-world applications such as indoor localisation, crowd control, and autonomous driving. Since Convolution Neural Networks (CNNs) are introduced into the field, many successful two-stage benchmarks have been proposed [58, 135]. They first use a Region Proposal Network (RPN) to find possible regions for pedestrians and then use CNN to predict their location. Another popular approach is the one-stage detector represented by YOLO [90]. They divide the image into $S \times S$ grids and directly predict pedestrians from the predefined anchors on those grids. Recent research suggests that removing anchors from the detection process helps improve accuracy and significantly reduce inference time [158, 178]. Despite the development of new detection methods, occlusion has been a big challenge in further improving performance.

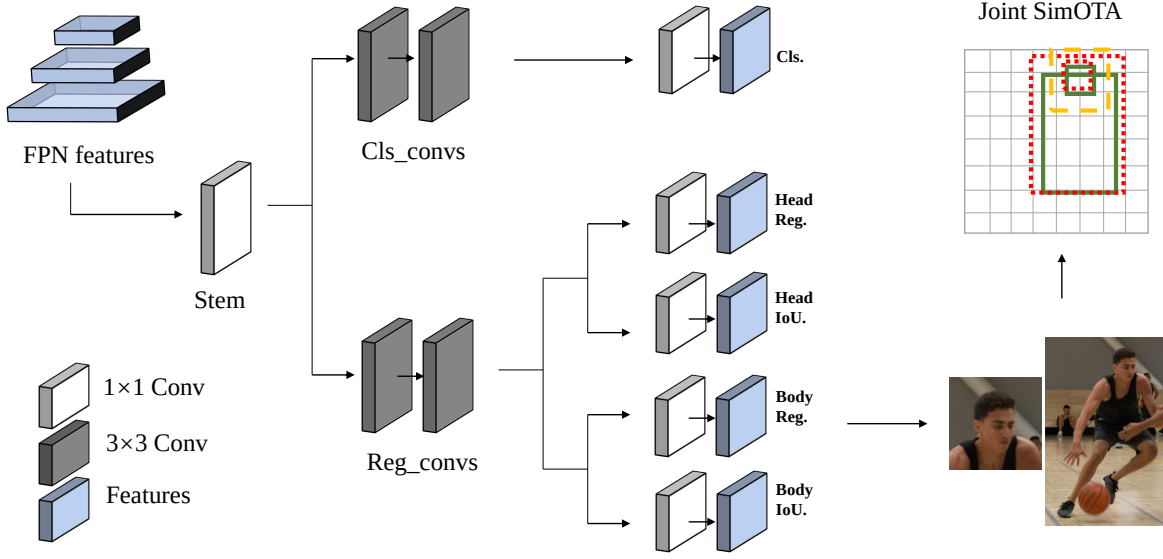


FIGURE 5.2. Overview of the proposed model. The image is first processed by a FPN structure to extract features of different scales. Then, a 1×1 stem conv layer followed by two 3×3 branches extract the classification and regression features of the image. For the regression branch, we add two additional prediction heads to predict the head and body simultaneously. The predictions are evaluated by a Joint SimOTA module to filter out the grids that have either bad head prediction or bad body prediction. We calculate losses for the rest of the grids.

People overlapping with each other, hiding behind obstacles, and making different poses cause a lot of miss detection and inaccuracy.

To handle a variety of occlusions, a general framework is to integrate the body detector with the pedestrian detector. Early research such as DeepParts [172] handles occlusion with a part pool and trains extensive part detectors to improve the pedestrian recall rate. Zhou and Yuan [38] categorise different types of occlusion patterns and use its own occlusion-specific detectors that are trained simultaneously to obtain the final classifier. The Bi-box [37] uses two separate detection branches in the model to detect the visible part and the full body at the same time. Chu et al. [161] propose a new framework to predict multiple instances instead of one and introduce a new EMD Loss and Set NMS.

When applying occlusion handling techniques to MOT, more features can be explored. SORT [2] uses the motion factor to handle occlusions by predicting pedestrian locations in the

next frame and matching them with the incoming detection. The visual characteristic is another important factor. DeepSort [116] explores the ReID features as part of the matching metric. FairMOT [168] proposes a joint detect-tracking framework that can extract the ReID features while detecting pedestrians. MTrack [49] improves FairMOT by extracting visual features from multiple body parts instead of the centre of the body. However, all of these approaches have limited performance in heavily occluded scenes because the ReID features for pedestrians in such scenes are sometimes inconsistent and unreliable. Therefore, it is considered that visual features have little impact on today's tracker [170] as the tracker just relies on exploiting the potential of low-score detection and matching them with high-score detection with a traditional Kalman filter, which after all could still achieve state-of-the-art performance.

5.2.2 Head and Body Detection in Dense Crowd Tracking

Head detection is difficult due to the small size of the head compared to the body of the pedestrian, so it is first used for crowd counting [41, 171] to estimate the density of the crowd. The development of the feature pyramid network (FPN) [145] enables a more accurate detection of objects on different scales. Since head detection scenes are generally more challenging than body detection scenes in terms of crowd density, many researchers have discovered that combining head detection with body detection usually helps to improve the detector performance.

Lu et al. [128] propose a head-body alignment net to jointly detect the human head and body. They use two parallel RPN branches to propose the head body RoIs and use an additional Alignment Loss to enforce body boxes to locate compactly around the head region. PedHunter [34] trains the model to predict a head mask while predicting pedestrian bodies. It serves as an attention mechanism to assist in feature learning in CNN. It helps reduce false positive by learning more distinguishable pedestrian features. However, the method does not eliminate false negatives, as the RPN in the model is trained just like the traditional body detector. Chi et al. propose JointDet[35] to use pedestrian heads to aid in body detection, which shares a similar idea to this work. However, they use a static body-head ratio to generate body

proposals from head proposals to reduce the computational workload when tiling the anchors for body proposals. In our work, we abandon the previous anchor-based design and propose a much simpler and more efficient solution to the joint head-body detection problem.

5.3 Methodology

5.3.1 Framework Overview

The overall framework is shown in Figure 5.2. For the object detection part, we train a head detector adopting the YOLOX[62] structure. We add an additional regression branch to the decoupled head and the classification head remains the same. The detector aims to classify and regress human heads in the scene. It also generates a body prediction along with each head detection. To combine the main detector with the supplementary body detector in the process of multiple object tracking, we first do a bipartite matching to pair our head detection with the external body detection, then the unmatched head detection can contribute to the tracking process.

A heat map comparison is shown in Fig. 5.3. For the input image Figure 5.3a, we demonstrate two heat maps: one from the baseline body detector and the other from our joint head-body detector. The two detectors are built with an identical backbone network and are trained with the same training dataset & equal epochs for a fair comparison. From the figure we can see that the head detection model can extract more distinguishable features from the image than the body detection model. The head heat map also contains features that are missed by the baseline body detector (e.g. the man sitting on the right side and the man standing in the middle). Therefore, it should perform better on extremely crowded scenes with a proper architecture design.

The advantages of the proposed approach are: 1). To address the head and body proposal relationship problem existing in previous work. Since a person can have various postures, generating the body proposal according to the common standing posture ratio is not always reliable. A bad proposal often leads to a bad prediction. Hence, we solve this issue by

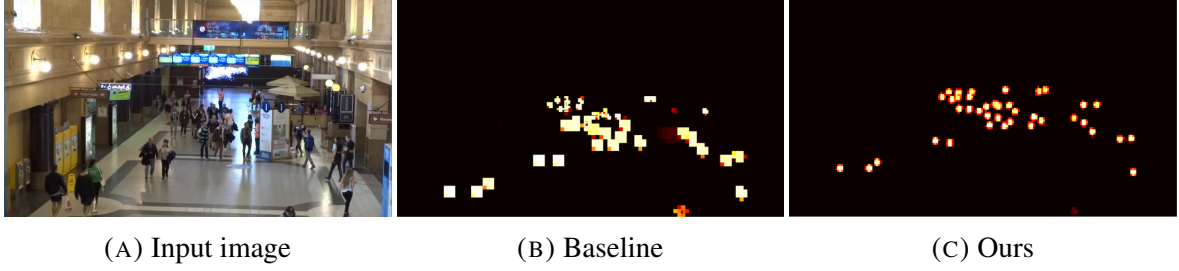


FIGURE 5.3. Heatmap comparison between head detection and body detection model under the same model structure. (a) is the input image from the MOT20 data set. (b) is the heatmap from the baseline. (c) Heatmap of our model. Compared to the baseline, our heat map is more distinguishable and can detect more pedestrians in the scene.

training an anchor-free detector and predicting the head and body from the same feature maps simultaneously. 2). In crowded scenes, human heads are usually easier to detect. In some heavily occluded scenes, the head of a person can be easily observed, while the rest of the body is hidden. However, only detecting heads is not sufficient for MOT problems, since sometimes heads are not visible. Thus, we use an additional body detector to serve as a complementary model.

5.3.2 Anchor-Free Head-Body Detection

In the previous design of JointDet[35], the head and body of the pedestrian are predicted from two separate RoIs following a specific head-body ratio. This causes two problems: 1). The ratio is statistically obtained based on all human head-body pairs in the CrowdHuman dataset, which are dominantly from a horizontal camera angle, resulting in a standing posture head-body ratio. The body proposal generated from this ratio is not suitable for the top-down camera angle sequences in MOT. 2). The extra hyperparameters introduced by the anchor boxes heavily influence the performance, impacting the overall recall and precision score.

Therefore, we adopt the design of the recent YOLOX [62] and propose an anchor-free style joint head-body detector. As shown in Figure 5.2, for each location, the regression value (top left x and y , width and height) of the head and body are directly predicted by two parallel CNN branches. An illustration is shown in Figure 5.4.

With the anchor-free design, our model can directly predict the pedestrian body box without using a predefined body anchor box, while greatly reducing the requested efforts to heuristically tune the parameters for training. Supported by the training advantage, we also eliminate the performance impact of the body anchors.

5.3.3 Joint SimOTA

We use both the head prediction and the body prediction to assign labels during training. Specifically, the degree of pairwise matching between a ground truth g_i and a prediction p_j is calculated as:

$$C_{ij} = L_{ij}^{cls} + \lambda_1 L_{ij}^{hreg} + \lambda_2 L_{ij}^{breg} \quad (5.1)$$

where λ_1 and λ_2 are the balancing coefficient. L_{ij}^{cls} is the classification loss between ground truth g_i and prediction p_j . L_{ij}^{hreg} and L_{ij}^{breg} are the regression losses for the prediction of the head and body, respectively. For a ground truth g_i , we select the best predictions k with the least cost within a fixed centre region of the head ground truth box as its positive samples. The value k is dynamically determined according to the IoUs of the predicted boxes [176]. The grids that contain these positive samples are considered as positive grids and all other grids are considered negative. Only positive grids are saved for loss computing to reduce the computing resource cost.

The Joint SimOTA makes sure that during the grid sample assignment process, only those with a good head prediction and a good body prediction are selected as positive samples. It serves as a function similar to the combination of the head-body relationship discriminating module (RDM) and the statistical head-body ratio proposal generation method introduced in [35]. Instead of generating the body proposals from heads based on a statistical ratio, all head-body prediction pairs are generated based on the features extracted by the stem network and dynamically evaluated by the joint simOTA process. This helps to simplify the prediction process by reducing the extra hyper-parameters introduced by the RDM module and boost the average detection precision.



FIGURE 5.4. Illustration of our anchor-free approach vs. traditional static head-body ratio anchor-based approach. a). Anchor-based approach, where head and body boxes are regressed from separate RoIs. b). Our anchor-free approach, where head and body boxes are predicted simultaneously from the same grid box, achieving a faster inference speed and fewer parameters for training.

5.3.4 Tracking Framework

We use a simple yet efficient two-step tracking framework inspired by ByteTrack [170]. First, we need to combine the detection results from both detectors to eliminate the duplicated detection and split the detection set into two parts: first class detection and second class detection. We do a bipartite matching between the body prediction from the traditional body detector and our joint head-body detector. This creates three different parts of the detection:

- Matched detection. This represents the people detected by both the main detector and the supplementary detector. Their confidence scores are set to which one is higher in either detector. This helps to improve weak detection scores.

- Unmatched body. This represents the people detected by the supplementary body detector only, usually happening with pedestrians that are walking out of the screen's top edge since their heads are not visible.
- Unmatched head. This represents pedestrians that are only detected by our joint head-body detector. Usually happens with heavily overlapped crowd whose bodies are highly occluded but heads are clearly visible.

We use the Hungarian Algorithm for the bipartite matching step. The matched head and body are all classified as first class detection. For the unmatched head and unmatched body, we further classify them according to their confidence score. Once we have the two sets of detections, we can perform the tracking. The detailed tracking algorithm is shown in Algorithm 1.

Algorithm 1 Tracking algorithm with both detectors

Input: head detection h_Det , body detection b_Det

Output: tracklets τ

Initialisation : $\tau \leftarrow \emptyset$

```

1: for frame  $i$  in video length do
2:   match  $h\_Det_i$  with  $b\_Det_i$ 
3:    $\mathcal{D}_{high} \leftarrow$  matched heads and bodies, high score heads, high score bodies
4:    $\mathcal{D}_{low} \leftarrow$  low score heads, low score bodies
   ———Update all tracklets with Kalman filter———
5:   for  $t$  in  $\tau$  do
6:      $t \leftarrow KalmanFilter(t)$ 
7:   end for
   ———First class detection association———
8:   match  $\tau$  with  $\mathcal{D}_{high}$ 
9:    $\tau_{remain} \leftarrow$  unmatched tracklets
10:   $\mathcal{D}_{remain} \leftarrow$  unmatched high score detection
   ———Second class detection association———
11:  match  $\tau_{remain}$  with  $\mathcal{D}_{low}$ 
   ———Update and initialize tracklets———
12:   $\tau \leftarrow$  matched detection
13:   $\tau \leftarrow \tau + \mathcal{D}_{remain}$ 
14: end for
15: return  $\tau$ 

```

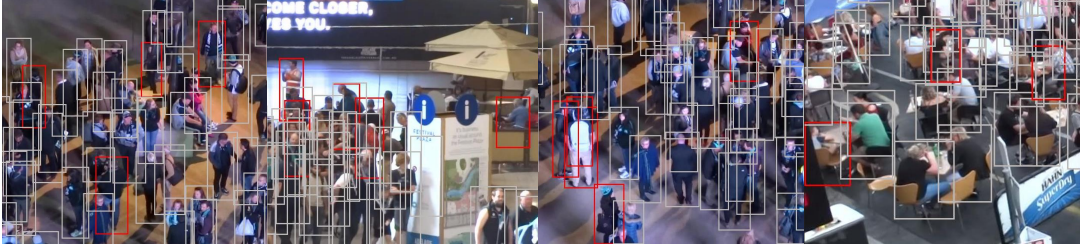


FIGURE 5.5. Qualitative results on MOT20 dataset. Pedestrians detected by the baseline are shown in grey boxes. Pedestrians detected by our method which are not detected by the baseline are shown in red boxes. Best viewed in colour.

5.3.5 Loss Function

The loss function of the purposed method consists of four parts and is defined as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{obj} + \alpha_1 \mathcal{L}_{head} + \alpha_2 \mathcal{L}_{body} \quad (5.2)$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{obj}$ is the binary cross entropy loss for head classification and confidence score prediction. $\mathcal{L}_{head}$ and $\mathcal{L}_{body}$ are the regression loss for the predicted head and body bounding boxes. α_1 and α_2 are the balancing coefficients and we set them at 5 during our training and experiments. We use an additional L1 loss for head and body detections for the last 10 epochs of training.

5.3.6 Training Details

5.3.6.1 Dataset Construction and Data Augmentation

Since there is a few datasets that contain head and body labels at the same time, we use existing datasets to build our own training data. We combine the ground truth for MOT20 and HT21 since they have two identical training sequences. We also use the Crowdhuman dataset during training since they explicitly label the pedestrian heads and bodies. Following the work of [62], we use Mosaic and MixUp strategies to improve detecting performance. They are commonly used in YOLOv4 [9], YOLOv5 [86], and other detectors.

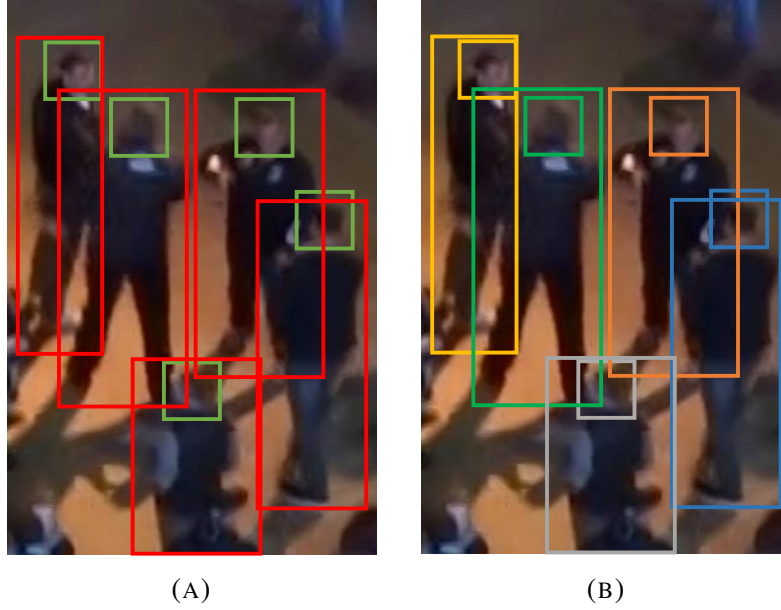


FIGURE 5.6. Demo of creation of our mixed training dataset. We use bipartite matching to assign head labels to body labels. a): Red boxes represents the body labels from MOT20 dataset, green boxes represents the head labels from HT21 dataset; b): after bipartite matching, each head label is assigned a body label similar to Crowdhuman dataset. Same colour represents the label box belong to the same person.

5.3.6.2 Training Parameters Setting

We used a YOLOX-x model pre-trained on the COCO dataset to initialise the training. The model was trained on two RTX3090 graphic cards with a batch size of 8 and epoch number of 80. Following the design of ByteTrack[170], the optimiser is SGD with a momentum ratio of 0.9. The learning rate is set at 0.0001 with a 1 epoch warm-up and using the cosine annealing strategy introduced in [75]. The input of the training images is set to 896×1600 . The whole training takes about 42 hours. For post-processing, we use an NMS of threshold of 0.45 for the head prediction and 0.7 for the body prediction to eliminate overlapped detection.

5.4 Experiments

Since our approach focusses on extremely crowded scenes, the experiments are conducted mainly on the MOT20 dataset. Experiments are also conducted on Crowdhuman and HT21 to demonstrate the robustness of the proposed approach.

5.4.1 MOT Challenge

MOTChallenge is a human tracking benchmark that provides carefully annotated data sets and clear metrics to evaluate the performance of tracking algorithms and pedestrian detectors. An earlier benchmark MOT17 provides over $29k$ annotations in a total of 14 sequences. MOT20 is a more recent benchmark consisting of 8 different sequences depicting very crowded challenging scenes.

5.4.1.1 MOT20

For the MOT20 challenge, we perform two parts of experiments. The first is the detection performance, and the second is the tracking performance. Similar to the “Pedestrian Detection in Complex and Crowded Events” in HiEve, we evaluated the detection performance on the MOT20 dataset. Since the ground truth label for the MOT20 test data is not available, we conducted the experiment on the training set and showed qualitative results for the test set. The training dataset for our experiment is the Crowdhuman dataset and the mixed dataset we created for our model. We choose two sequences from the HT21 dataset and the MOT20 dataset and combine their ground-truth label. We add an offset to the label to match the image size differences. Figure 5.6 is a sample of the data set.

For the detection performance metric, we choose the logarithmic average miss rate over 9 points ranging from 10^{-2} to 10^0 FPPI (MR^{-2}) to evaluate the performance of our detector. Performance comparison is shown in Table 5.1. We use the current top performance detector used in ByteTrack [170] as our baseline. As shown in the table, our method achieves 92.99% AP, which is 7.32% higher than the baseline. For the log-average missing rate, we also

TABLE 5.1. AP and MR^{-2} performance of different detection methods on MOT20.

Method	AP $\uparrow$	$MR^{-2}\downarrow$
Baseline	85.67	14.0
Ours w/o Joint SimOTA	91.16	9.0
Ours	92.99	7.0

TABLE 5.2. Performance comparison on the test set with state-of-the-art on MOT20.

.	MOTA $\uparrow$	IDF1 $\uparrow$	HOTA $\uparrow$	FP $\downarrow$	FN $\downarrow$	IDs $\downarrow$
FairMOT	61.8	67.3	54.6	103440	88901	5243
TransCenter	61.9	50.4	43.5	45895	146347	4653
TransTrack	65.0	59.4	48.5	27197	150197	3608
CSTrack	66.6	68.6	54.0	25404	144358	3196
SOTMOT	68.6	71.4	57.4	57064	101154	4209
MAA	73.9	71.2	57.3	24942	108744	1331
ReMOT	77.4	73.1	61.2	28351	86659	2121
ByteTrack	77.8	75.2	61.3	26249	87594	1223
Ours	78.2	75.5	61.9	30187	81119	1325

achieves 7% MR^{-2} which is 7% better than the base line. The result has shown that our method reaches the supreme pedestrian detection performance by letting the model focus on the heads instead of the bodies.

For tracking performance, the results are uploaded to the official website for evaluation. The main evaluation metrics are Multiple Object Tracking Accuracy (MOTA), IDF1, and Higher Order Tracking Accuracy (HOTA). The result is compared to FairMOT [168], TransCenter [160], TransTrack [142], CSTrack [30], SOTMOT [101], MAA [43], ReMOT [59] and ByteTrack [170]. Despite the fact that many detections are ignored in the evaluation process [119], we still achieve a state-of-the-art performance of 78.2% MOTA and 75.5% IDF1.

5.4.2 Qualitative Result on MOT20

In this section, we demonstrate the effectiveness of our model. The results are shown in Figure 5.5. We have randomly selected 3 tracking sequences from the MOT20 test set and performed detection with both the baseline approach and our proposed approach. It is clear from Figure 5.5, our approach has a stronger ability to detect pedestrians in crowded scenes. In Figure 5.5, the red boxes are pedestrians only detected by our method and not from the

TABLE 5.3. Performance comparison on the test set with state-of-the-art on HT21.

.	MOTA $\uparrow$	IDF1 $\uparrow$	IDEuc1 $\uparrow$	HOTA $\downarrow$	FP $\downarrow$	FN $\downarrow$
HeadHunter	57.8	53.9	54.2	36.8	51840	299459
FairMOT	60.8	62.8	69.9	43.0	118109	198896
FM_OCSORT	67.9	62.9	62.1	44.1	102050	164090
THT	70.7	68.4	63.5	47.3	33545	211162
pptracking	72.6	61.8	59.7	44.6	71235	154139
Ours	77.9	70.8	68.4	50.2	42,205	140867

TABLE 5.4. Head and body AP performance of different detection methods on Crowdhuman.

Method	head	body
Baseline-Head	54.2	-
Baseline-Body	-	57.8
Ours	55.0	57.9

baseline. It usually happens to people who overlap heavily with others or only show part of their body due to different postures. Since our model focusses on detecting the heads, it can extract the corresponding head features from the image and accurately predict the body boxes.

5.4.3 Ablation Study on Joint SimOTA

Our Joint SimOTA module helps the model to learn head-body relationship dynamically, which can improve the detection performance. We conduct the ablation study by training our proposed model with the same settings except the Joint SimOTA module. According to Table 5.1, adding the Joint SimOTA module can increase AP by 1.83% and reduce the logarithmic average missing rate by 2%. Performance improvement indicates that body regression that contributes to the training process helps the model to predict more accurate body bounding boxes. Otherwise, the model learns the features for the head and body separately, leading to more false positives.

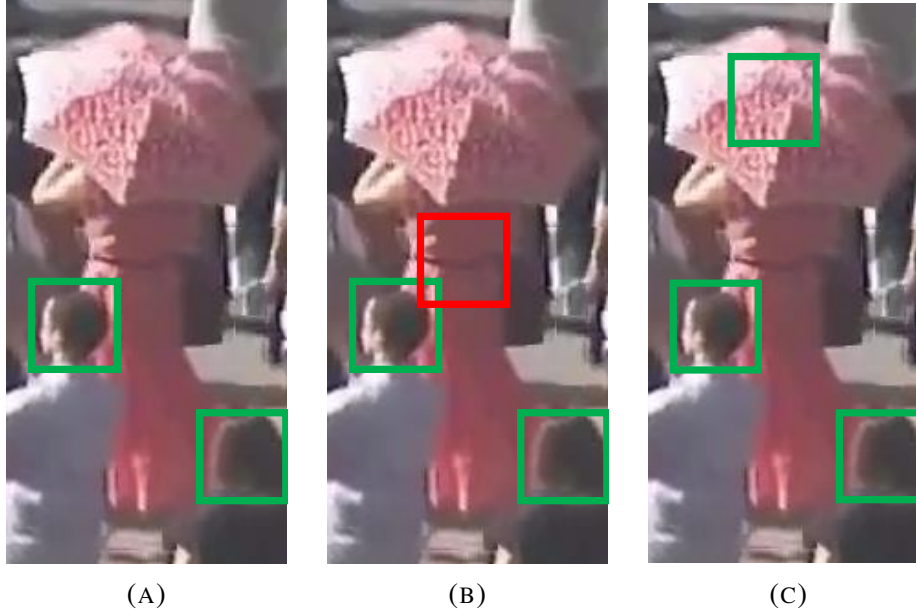


FIGURE 5.7. Comparison of head detection between our model and the baseline model. The person dressed in pink covered their head with an umbrella, making the features of the head invisible to CNN. a). One of the baseline models only detects two heads in the scene; b). Another baseline model predicts the wrong location of the head, represented with a red box; c). Our model can correctly predict the three locations of the head of the person.

5.4.4 Crowdhuman

Crowdhuman is a public pedestrian detection dataset that contains various crowded scenes. It provides over 470k pedestrian labels in a total of 15000 training set and 4370 validation set. For each pedestrian, the label contains the head, visible body part, and full body annotation. To demonstrate the robustness of our method, we performed experiments on the validation set with both the head and body detection. We retrain the YOLOX model for the two tasks separately as our baseline. According to Table 5.4, our approach improves the AP of the head and body by 0.8% and 0.1%, respectively. The result shows that combining head and body detection helps to boost the performance of both tasks.

5.4.5 Head Detection and Tracking

Since our model detects the head and body simultaneously, the model can also be adopted in head tracking datasets. Because of the Joint SimOTA module we added in the training phase, the body part of a pedestrian can contribute to the head prediction as well. As shown in Figure 5.7, the head of the pedestrian is covered by an umbrella. The baseline models search for the head features in the figure and end up with missing or wrong detection. Our proposed model can correctly predict the head location of the pedestrian. Compared to HeadHunter [123], FairMOT [168], FM_OCSORT [27], THT and pptracking [18], our joint head-body detection model achieves state-of-the-art results in the HT21 head tracking challenge without need for additional training data as shown in Table 5.3.

5.5 Conclusion

In this work, we propose an anchor-free style joint head-body detection model to detect the head and body of pedestrians simultaneously. By focussing on the heads, the model can detect pedestrians more effectively than the baseline body detectors due to high occlusion. With our proposed model, we can significantly reduce the false negative detected in extremely crowded scenes. We have conducted extensive experiments on MOT20, HT21, and Crowdhuman datasets. Our approach achieves state-of-the-art detection and tracking performance and is robust to various scenarios.

Part 3

Multimodal Approach

Previous chapters contain single-source IPS, but they can be combined to further improve the system performance. The multimodal approach to indoor positioning integrates data from multiple sources, harnessing the strengths of each modality to achieve higher accuracy and robustness. By combining information from wireless signals, computer vision, inertial sensors, and even acoustic signals, a more comprehensive understanding of the user's position and orientation can be achieved.

Deep learning plays a crucial role in fusing data from these diverse sources. Feature fusion methods are designed to handle heterogeneous data, ensuring that the information from each modality is appropriately weighted and integrated. For example, when visual data might be unreliable due to poor lighting, the system could rely more on wireless signals or inertial measurements. This adaptability makes the multimodal approach particularly resilient to varied indoor scenarios and challenges.

In this section, we present our innovative deep reinforcement learning (DRL) based multimodal approach tailored for indoor positioning, with a specific focus on the indoor pedestrian identification matching challenge. We commence by detailing the digital twin platform employed for prototyping our algorithm, followed by an elucidation of the foundational concepts of reinforcement learning (RL). Subsequently, we delineate the design of our RL environment, encompassing the state and action spaces, as well as the reward functions. Our RL agent adeptly extracts motion features from diverse modalities and seamlessly fuses them. Notably, the proposed RL agent demonstrates a markedly enhanced matching success rate in comparison to existing methodologies.

Symbol Used in Part 3

Symbol	Description
P_{3D}	a 3D point in UE5
V	camera view matrix
P	projection matrix
Δv	velocity difference
Δt	time difference
ω	object rotation speed
s	state
a	action
r	reward
π	policy of a DRL agent
$V^*(s)$	optimal value function for s
k	DRL update index
δ	temporal difference
γ	decay factor
B	batch of transitions
τ	a trajectory in DRL
$P(\tau \theta)$	the probability of a trajectory
A	advantage
ϵ	“trust region” for the policy update
$\mathcal{L}(\theta_k, \theta)$	the surrogate advantage
μ_t	local RSSI
ϕ_t	local IMU sensor data
v_t	global visual tracking result

A Deep Reinforcement Learning Based Multimodal IPS Approach

In this section, we combine the above wireless technologies and vision-based IPS technologies together to prototype a multimodal approach **PediFuse**. By combining the robustness and range capabilities of wireless methods such as Wi-Fi, Bluetooth, and RFID with the high-resolution spatial information provided by vision-based IPS, our goal is to overcome the limitations inherent to each individual technology.

6.1 Introduction

Many indoor environments such as warehouses and factories benefit from indoor localisation and navigation services. Applications such as inventory management and asset tracking require high-precision indoor positioning methods that are often difficult to achieve with one specific technology. The use of multimodal data is essential for indoor localisation and navigation, as it significantly enhances the precision and reliability of these services [57]. Incorporating diverse data modalities, like visual data from surveillance cameras and sensor data from users' mobile devices, enables a more comprehensive understanding of the environment and the user's context within it. State-of-the-art computer vision technologies ensure that live surveillance cameras can provide visual updates in real time [36], while sensors on mobile devices capture movement and orientation data [45]. Fusion of these data modalities can offer precise positioning and orientation, enabling efficient localisation. Consequently, the use of multimodal data can significantly improve the quality of indoor location-based services, ensuring accurate and real-time navigation guidance for users. This synergy is vital in indoor settings where GPS data is often unavailable or unreliable.

Existing solutions: Most existing solutions for data fusion are analytical approaches. In particular, heterogeneous motion features need to be extracted for different sources of input. Earlier studies such as EV-loc [24] calculated the user’s distance between different wireless transmitters from two types of input: Wi-Fi and camera detection. Then a cost matrix is formulated to find the best match between wireless vision and camera vision. IdentityLink [143] connects users to mobile devices using video and Radio Frequency (RF) signal data with Boolean motion characteristics (moving or not moving) and received signal strength fluctuation (RSS). More recent work like TAR [157] proposed a way to find the match between Bluetooth proximity information and camera detections based on estimated physical distance between pedestrians in a retail store and the pre-installed security cameras in the store. PHADE [141] utilises two types of motion features called “IsWalking” and “Relative Walking Direction” to match the detection results of smartphone’s Inertial Measurement Unit (IMU) sensors and surveillance cameras. RFCam can be considered as an improved method which utilises the Channel State Information (CSI) of the multi-antenna Wi-Fi radio, which contains both the Angle of Arrival (AoA) and the distance information. However, the system still recognises only three basic types of activity (walking, standing, and turning around).

These traditional methodologies rely on defining pedestrian motion patterns through the extraction of heterogeneous embedding features from various sources of input such as angle, distance, orientation, and basic activities (including walking, standing, turning, etc.) are usually limited in their effectiveness. While they can create detailed pedestrian profiles, the process is usually computationally intensive and time-consuming, as the components need to be designed and tested separately. Additionally, they may not effectively account for the dynamic and unpredictable nature of crowded environments. In such settings, basic motion types often fail to fully encapsulate a subject’s movement pattern, leading to uncertainties during the matching process. To compensate for these uncertainties, additional modules, such as uncertainty-aware and temporal weighting modules, are required, which further increase the computational load on the system.

Our Approach: In this study, Deep Reinforcement Learning (DRL) is used to address the optimal matching problem associated with multimodal data fusion. DRL typically involves

an agent, which is a neural network model, interacting with a specific environment. The agent seeks to optimise its strategy to achieve the best possible performance. Various DRL approaches have been proposed over the years. For example, DeepMind introduced the Deep Q-Network [150] (DQN) method, a value-based approach, which was successful in playing Atari games. Policy-based methods [129], such as the REINFORCE algorithm [76], have also shown success in tackling problems that are challenging to model using conventional methods. Additionally, the Actor-Critic method [149], which combines aspects of both value-based and policy-based approaches, has been proven to be effective in various applications.

DRL distinguishes itself from traditional supervised learning methods, especially in analysing crowded scenes, by its inherent adaptability and capability to learn from trial and error without the need for labelled datasets. Unlike supervised learning, which relies on predefined input-output pairs and may struggle with the dynamic and unpredictable nature of crowded environments, DRL autonomously identifies relevant features and optimises actions based on a reward system. This approach enables DRL to better handle the variability and complexity of human behaviours and interactions, offering advantages in adaptability, feature extraction, long-term planning, scalability, and generalisation to new environments. Thus, DRL provides a more flexible and robust framework for understanding and managing crowded spaces.

DRL is a more flexible and robust approach with its ability to learn optimal policies end-to-end. DRL can adaptively learn to associate specific actions (i.e. matches between different modality data) with states (i.e., the observed features from the data) to maximise a cumulative reward signal. This signal is based on the accuracy of the matches, which provides a strong incentive for the agent to find the best match. DRL has inherent robustness to noise and is capable of handling uncertainty, which is a common characteristic in real-world data. This makes it a particularly suitable technique for our problem, as it can learn to optimise over a distribution of possible states, allowing it to find reliable matches even in challenging scenarios with high levels of noise and variability. By shifting to a DRL approach, we can move away from manual rules-based methods toward more adaptive, robust, and efficient solutions for multimodal data fusion in complex environments.

Using DRL for multimodal data fusion is not a trivial task. Chen et al. [112] use reinforcement learning for multimodal sentiment analysis. However, RL here is only used to train the sentiment prediction component, which does not involve fusing inputs between modalities. Guo et al. [79] also introduced a way to combine multimodal data for human action recognition by letting the agent adjust the weight ratio of different modalities, which does not suit our problem. In this article, we assume that an DRL agent is installed on each pedestrian's smartphone. The agent which takes three types of input: local RSSIs, local IMU readings, and global visual detection results from surveillance cameras. For every timestamp, the agent predicts the probabilities of matching its installed device to one of the global visual detection (or none if the confidence score is lower than a threshold). This problem is similar to a multiarmed bandit (MAB) problem [6], except that the observation involves strong temporal dependencies which common MAB algorithms cannot handle. The DRL methods that use techniques to handle sequential data (such as RNN [122] or LSTM[134]) offer more advantages in this scenario.

The main contributions of this work are as follows.

1. We have innovatively formulated the multimodal identity matching problem as a DRL problem. We have proposed solutions to challenges such as agent policy updates and reward function design, enabling end-to-end agent training. To the best of our knowledge, this is the first study to use DRL to directly address the indoor identity matching problem with a unified model.
2. We demonstrate that our approach is capable of extracting heterogeneous motion characteristics from pedestrians in a noisy environment. A DRL-based pedestrian matching framework is proposed to achieve fast real-time pedestrian tracking for crowded indoor scenarios.
3. Experiments and case studies are provided to demonstrate the effectiveness of the proposed approach.

This chapter is structured as follows: we first outline the platform selected for development, elucidating the rationale behind this choice. Subsequently, we provide an introduction to the fundamentals of DRL, which serves as the cornerstone of our multimodal approach.

Following that, we delve into the intricacies of our proposed method, detailing its architecture and key components. Finally, we assess the effectiveness of our approach through a series of meticulously designed experiments.

6.2 Background Technology and Preliminaries

To address the challenges of data acquisition for IPS research, we turned to a digital platform that serves as a virtual mirror of the real world. As highlighted in Chapter 1, one of the predominant hurdles in contemporary IPS research is the daunting task of accumulating large-scale experimental datasets. Investigating densely populated environments usually requires reliance on public datasets like MOT17 and MOT20 [119] or the adoption of simulated or analysis-driven results.

6.2.1 Real-World Experiment vs. Emulator

Sourcing an appropriate public dataset that encompasses all the requisite data modalities is an uphill battle. A comprehensive multimodal IPS approach demands a plethora of data sources, including wireless signal data, camera-based detection or tracking data, and potentially other sensor data such as IMU, LIDAR, or geomagnetic readings. Current methodologies often resort to conducting experiments in select indoor environments, such as university campuses or research labs. These datasets, tailored to specific research needs, are seldom made public, leaving a gaping void in accessible data for broader research endeavours.

Consequently, we explored alternative avenues, such as data simulators like Matlab. Given our focus on wireless signals, it was imperative for our chosen simulator to have the capability to generate simulated wireless signals. Additionally, visual signals were crucial to our research, so we needed a simulator that could seamlessly integrate both. While it is feasible to simulate visual signals by directly generating x and y coordinates, accurately replicating the nuances of heavy occlusion and human movement patterns becomes a complex endeavour. Manual fine-tuning of these aspects is not only tedious, but also lacks precision.



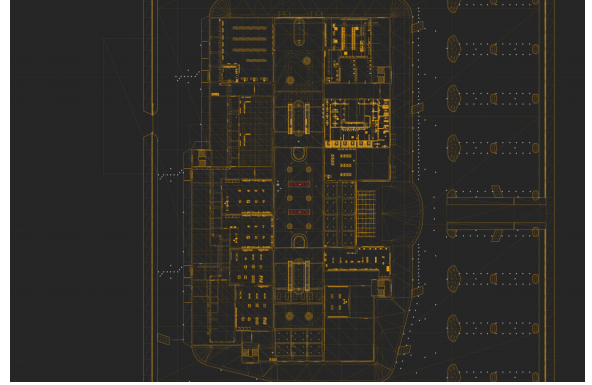
(A) Exterior of the shopping mall environment.



(B) Interior of the shopping mall environment.



(C) Example Layout map of the shopping mall environment (Layer 1).



(D) Grid layout of the shopping mall environment (Layer 1).

FIGURE 6.1. Illustration of the testing ground for our proposed multimodal IPS algorithm. (A) Overview of the shopping mall: A comprehensive structure mainly constructed of concrete and glass, surrounded by trees and roads. The total area is approximately 20 km^2 . (B) Interior of the mall: A diverse assortment of establishments, such as food courts, entertainment areas, retail stores, and bookshops, are present. To mimic real-world scenarios, various furnishings, such as sofas and indoor plants, are strategically placed. (C) Ground-level layout: This provides a clear representation of the interior architecture and arrangement. (D) Detailed grid layout: Offers a more precise visualisation, highlighting structural elements such as walls and pillars.

Existing Matlab-based signal emulators fell short of our requirements, primarily due to their inability to handle the high computational demands of our project. In contrast, modern gaming engines such as Unreal Engine and Unity emerged as promising alternatives. They offer realistic visual rendering capabilities and allow for the creation of tailored indoor or outdoor

environments. In this context, the digital twin platform, WirelessDT, equipped with Real-Time Ray-Tracing techniques, emerged as the ideal solution for our experimental needs. This platform not only met our technical specifications, but also provided a holistic environment that closely mirrored real-world scenarios, making it an invaluable tool for our research.

The platform provides real-time simulation and GPU acceleration for the task. When choosing this platform for our experiments, there are several advantages:

- **Control Over Variables:** Using Unreal Engine 5 (UE5), we can precisely control all variables in a virtual environment. In real-world settings, controlling the exact number of people in a given space is a logistical challenge. In UE5, researchers can specify the exact number of virtual pedestrians in a scene, meaning that they can simulate various crowd densities, from a nearly empty room to a packed stadium, and observe how the indoor localisation system performs under each condition. The unpredictable movement patterns of real people can also be programmed in the emulator. Particular behaviours, such as sudden stops or changes in speeds or directions, can be easily achieved. Other factors such as signal strength and environmental factors can also be configured. For example, the parameters of a receiving device can be changed to simulate different brands of mobile phones.

The conditions, architectural structures, and even the materials used in the construction of an indoor environment can be modified at will. For example, we can change the material of the walls to see how different surfaces affect the signal reflexion. This level of control is challenging to achieve in real-world scenarios. This allows us to rigorously test our approach for indoor localisation in crowded settings by adjusting parameters to identify strengths and weaknesses, thereby enabling effective model optimisation.

- **Scalability and Reproducibility:** Conducting experiments in a simulated environment allows for easy scaling of test scenarios and ensures that experiments are reproducible. It allows us to simulate different pedestrian densities without needing physical resources. For instance, if we want to test how a system performs with 100 pedestrians versus 1,000, we can simply adjust the number of virtual pedestrians on

the platform. The number of test targets can be easily scaled up with less increased costs compared to real-world experiments. Also, in the simulated environment, the lighting, the time of the day, and even the indoor objects' placement remain consistent across experiments unless deliberately altered, ensuring that any changes in outcomes are due to the variables being studied and not external factors.

Moreover, we can run the same experiment multiple times under the same conditions, ensuring data reliability. The ability to reproduce the same conditions consistently means that we can engage in iterative testing. If a potential solution or improvement is identified, it can be tested under the same conditions as the original experiment to accurately measure its impact.

- **Cost-Effective:** Real-world testing can be time-consuming and expensive. Recruiting human subjects, securing indoor spaces, setting up equipment, and ensuring privacy and consent for data collection can all add up. For example, physical experiments might require specialised equipment, from sensors to cameras and even setup crews. Making changes to the experimental setup can be costly and time-consuming. In addition, manual data collection and processing can be labour intensive and prone to errors.

In comparison, virtual experiments in UE5 can be set up and run at a much lower cost and can be done more quickly, facilitating rapid and iterative prototyping, since real-world experiments may require substantial logistical adjustments for similar changes. In a simulated environment, virtual agents replace human subjects, eliminating the cost mentioned above. It allows rapid iterations; for example, if we want to change a parameter or test a new hypothesis, we can do so immediately without incurring additional costs.

- **Safety and Privacy:** In real-world experiments, there are potential safety and privacy issues, especially when dealing with crowded indoor scenarios. There are always risks in collecting data from real people. For example, there might be data breaches, data misuse, or unauthorised access. In addition, the consent management process can be time-consuming and requires careful management to ensure participants fully understand the research and their rights.

Using a virtual environment like UE5 can eliminate these concerns since all ‘subjects’ are artificially intelligent robot actors. On the digital platform, the generated data is entirely synthetic, ensuring that no real personal data is ever at risk. It sidesteps many of these ethical dilemmas, as no real individuals are involved.

In essence, using a digital platform like WirelessDT brings benefits in many aspects: The sandbox environment ensures that every element can be controlled and adjusted; the scalability and reproducibility can provide consistent and reliable data; it can save long-term costs such as infrastructure, manpower, and logistical expenses; Compared to real-world experiments, it provides a controlled, risk-free alternative that allows us to conduct experiments without compromising on the safety and privacy challenges.

In the subsequent section, we first give a brief overview of the testing environment we created in WirelessDT. We then dive into the three primary sensors employed in our RL-based multimodal pedestrian identification approach: the wireless transceiver, the IMU sensor, and the surveillance camera. We provide a concise overview of how we successfully emulated these real-world sensors in a digital environment.

6.2.2 Shopping Mall Environment

We used an environment derived from a public UE5 map, representing a shopping mall complete with interior decorations to more accurately mimic real-world conditions. A visual depiction of this setting can be found in Figure 6.1. The mall spans three distinct indoor levels, interconnected by stairs and elevators. Within this structure, our custom-programmed artificial intelligence robot roams freely, offering dynamic indoor positioning scenarios essential for our study.

We opted for a mall environment for our IPS simulations for several compelling reasons. Primarily, a shopping mall represents a typical setting for indoor positioning services, characterised by its multilevel structure, array of shops, and diverse zones. Such a complexity challenges the robustness and precision of an IPS. The wide variety of materials, from metal

and glass to concrete and wood, found in malls impacts signal propagation in unique ways, making it an ideal testing ground for the system.

High foot traffic, a common feature in malls, aligns with the crowd density parameters of our study. Furthermore, the architectural diversity of malls, with their expansive open areas like atriums juxtaposed against tighter spaces like individual shops, provides a range of scenarios from clear line-of-sight to obstructed conditions. Adding to this complexity, the omnipresence of electronic devices, including mobile phones, Wi-Fi networks, and security systems, tests the IPS's resilience against interference.

Beyond the technical considerations, there is a strong commercial rationale. An effective IPS can transform the shopping experience by guiding customers to specific outlets, facilitating location-centric promotions, and streamlined inventory management for retailers. Conducting research in a setting that mirrors real-world applications not only enhances its relevance but also underscores its potential for practical commercial adoption.

6.2.3 Involved Sensors

6.2.3.1 Wireless Signal Transceiver

Within the building, we strategically placed wireless transmitters and surveillance cameras, replicating real-life setups. Initially, we integrated the wireless transceivers. These transmitters, located indoors, emit signals that can either permeate through structures and materials or be reflected and scattered. This behaviour is governed by the bidirectional reflectance distribution function (BRDF) [47] and the multiple importance sampling algorithm (MIS) [51]. A representative area with deployed wireless transmitters is illustrated in Figure 6.2.

The figure depicts eight transmitters, denoted by red icons, strategically positioned near the room ceiling to optimise signal coverage. In the accompanying figure, the distribution of the simulated wireless signal within the room is illustrated by varying colours, with brighter hues signifying stronger signals. In particular, from this example, it becomes evident that the

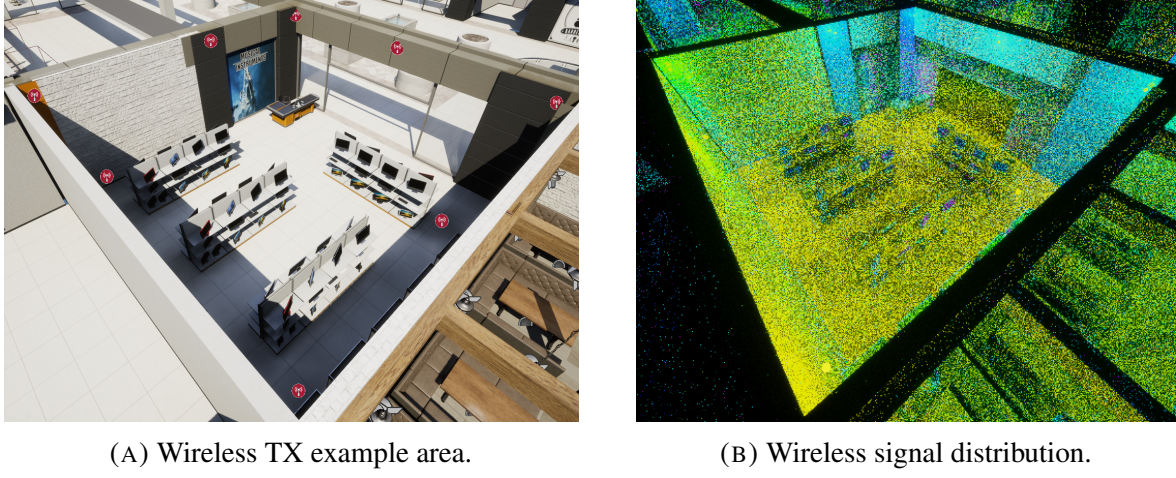


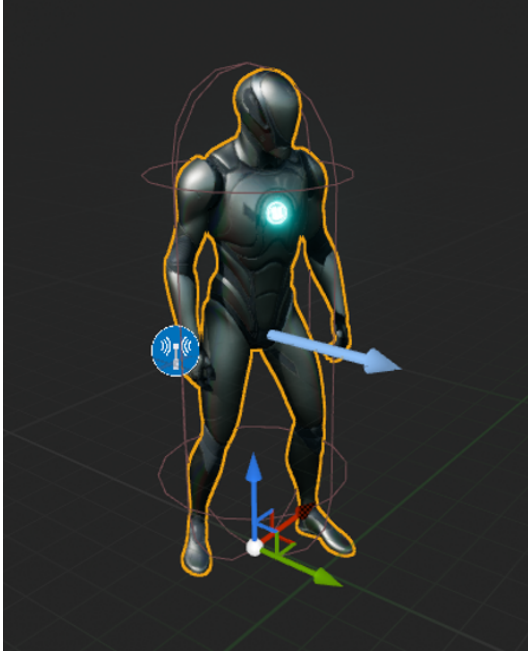
FIGURE 6.2. Wireless transmitter sensor example in the shopping mall environment.

type of building material has a pronounced effect on signal reflexion, which subsequently influences the overall signal strength.

For the receiving end, we have designed AIs equipped with receivers, mirroring pedestrians using smartphones in reality. These receivers capture signals from the proximate transmitters, as exemplified in Figure 6.3. These AI entities can navigate unrestricted within a designated zone or adhere to a preset movement pattern. Combined, the AI receiver and strategically placed digital signal transmitters constitute the wireless signal sensing framework in our virtual environment. The system is designed to emulate the behaviour of Wi-Fi or Bluetooth beacons commonly found in public spaces.

6.2.3.2 Camera Sensor

The camera sensor can be instantiated using the built-in camera actor of UE5. A significant challenge is to accurately simulate the detection results that contemporary computer vision algorithms would produce and subsequently collect. As shown in equation 6.1, given a 3D point $P_{3D} = (X, Y, Z)$, our goal is to discern its corresponding 2D coordinates on a pinhole camera. With UE5, one can readily access the camera's view matrix V and its projection matrix P . By employing the "Transform Vector by Matrix" operation, we can derive the homogeneous coordinates. The transformed X and Y values are then normalised by their



(A) Example of AI with a receiver holding in their right hand.



(B) Example of AI with more human-like appearance.

FIGURE 6.3. Wireless receiver examples for the shopping mall environment. We combine the default UE5 AI and our programmed wireless receiver object to simulate pedestrians in the real world.

respective weights. A subsequent transformation ensures that the resulting coordinates lie within the $[0, 1]$ range. A comprehensive blueprint of this process is shown in Figure 6.4.

$$\begin{aligned}
 P' &= P \times V \times P_{3D} \\
 P'' &= \left(\frac{X'}{W'}, \frac{Y'}{W'}, \frac{Z'}{W'} \right) \\
 x &= \frac{\text{width} \times (X'' + 1)}{2} \\
 y &= \frac{\text{height} \times (1 - Y'')}{2}
 \end{aligned} \tag{6.1}$$

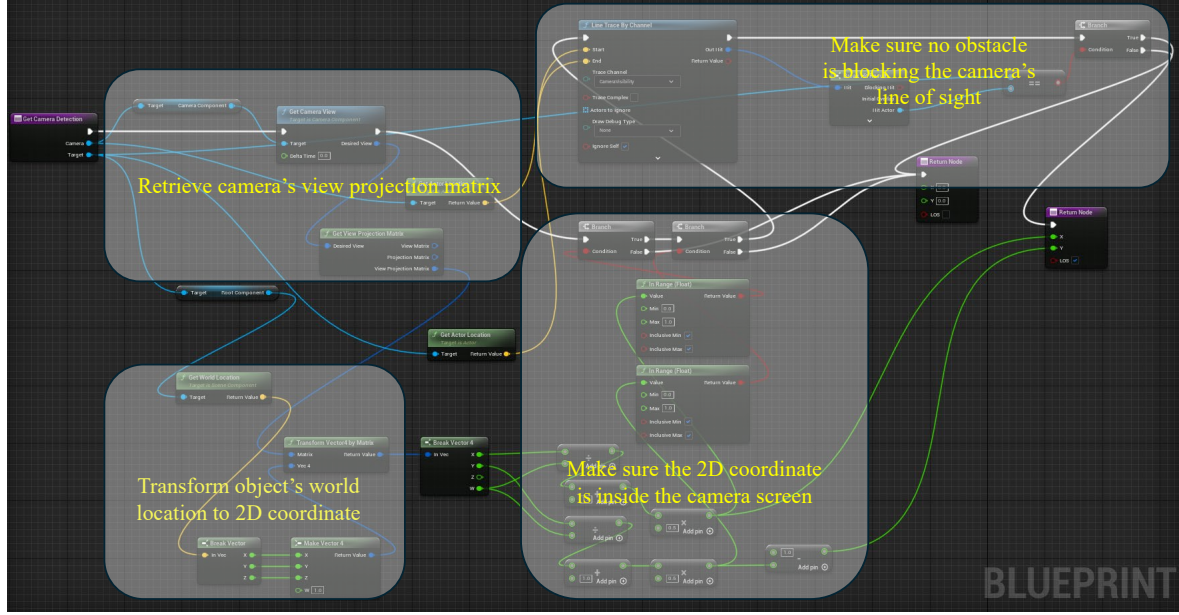


FIGURE 6.4. A detailed blueprint implementation of the camera sensor for our simulated environment.

6.2.3.3 IMU Sensor

The IMU sensor is relatively simple compared to the wireless sensor and the camera sensor. When an actor in UE5 is moving, the velocity and orientation are updated for each frame. To calculate the acceleration a of an object given two velocities v_1 and v_2 at two different time intervals Δt_1 and Δt_2 , we can use the formula:

$$a = \frac{\Delta v}{\Delta t}$$

Here, the change in velocity Δv is $v_2 - v_1$, and the change in time Δt is $\Delta t_2 - \Delta t_1$. To calculate the rotation speed ω of an object given two orientations θ_1 and θ_2 at two different time intervals Δt_1 and Δt_2 (in UE5, we use delta seconds), we can use the formula:

$$\omega = \frac{\Delta \theta}{\Delta t}$$

In this equation, the change in angular displacement $\Delta\theta$ is $\theta_2 - \theta_1$, and the change in time Δt is $\Delta t_2 - \Delta t_1$.

In real-world settings, designing IMU sensor-based IPS presents unique challenges, particularly due to factors like sensor drift and the random movements of people holding the devices. Sensor drift is a phenomenon in which the sensor's measurements gradually deviate from the actual values over time, even when the device is stationary. This drift can accumulate and result in significant errors that may compromise the accuracy and reliability of the system. Similarly, random hand movements, such as shaking, tilting, or rotating the device, can introduce errors that are difficult to account for in the algorithm. These erratic motions can particularly affect the readings of acceleration and angular velocity, making it challenging to obtain a stable and accurate positioning estimate. Therefore, addressing these issues is critical to developing robust and reliable IMU sensor-based IPS solutions. To better simulate real-world conditions, we introduced Gaussian noise into the calculations for acceleration and rotation speed. The acceleration a and angular velocity ω with added Gaussian noise can be modelled as:

$$a' = a + \mathcal{N}(0, \sigma_a^2)$$

$$\omega' = \omega + \mathcal{N}(0, \sigma_\omega^2)$$

where $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with a zero mean and a variance of σ^2 . σ_a^2 and σ_ω^2 are the noise variances for acceleration and angular velocity, respectively. A detailed implementation of the IMU sensor simulation is presented in Figure 6.5.

The simulated IMU sensor is designed to generate data that mimics the behaviour of a hand-held device's motion and orientation in the (x, y, z) coordinate system. For orientation, the simulation yields values for yaw, pitch, and roll. It is important to note that these orientation values are not the same as the raw data collected from modern smartphones' accelerometers and gyroscopes. However, raw data from real-world sensors can be algorithmically fused to

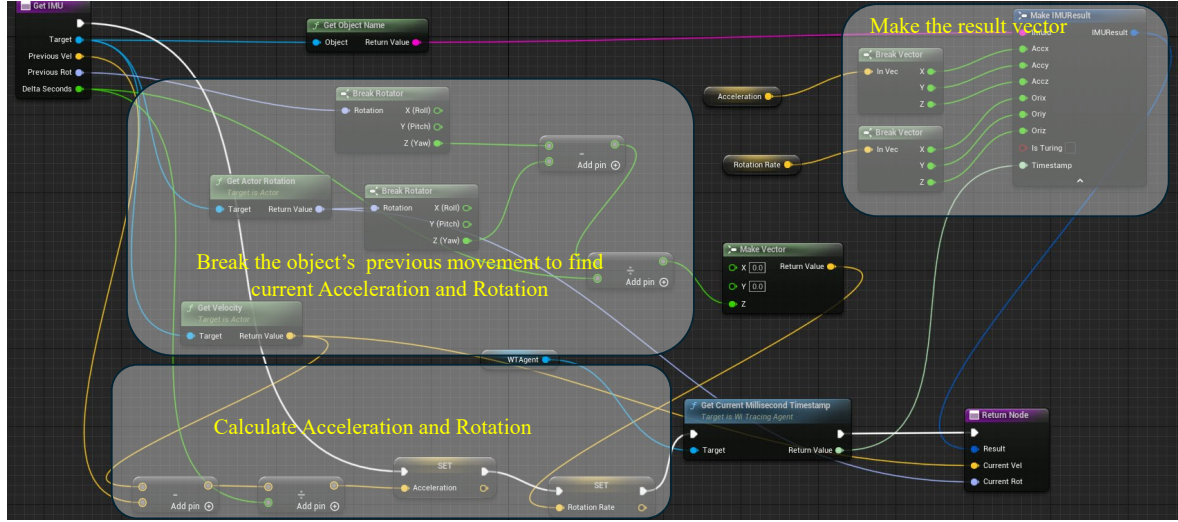


FIGURE 6.5. A detailed blueprint implementation of the IMU sensor for our simulated environment.

derive these orientation angles. As described above, in order to more accurately represent a real-world scenario, noise is introduced into the data collection process. Therefore, we consider this approach to be a proper simulation of an actual IMU sensor.

It's important to acknowledge that the simulated environment does not fully replicate the complexities of the real world. Our simulation approach, which includes modeling wireless signals using ray tracing algorithms, represents a simplified approximation of actual wireless signal behaviors due to technical constraints. We cannot incorporate every channel characteristic within the signal simulation. Additionally, the materials present in indoor environments significantly impact signal propagation, as signals reflect off floors and walls, and these interactions cannot be precisely duplicated in our model. Regarding the IMU sensors, they are configured to be held in the right hand of pedestrians, resulting in a pattern of movement that is relatively consistent.

6.2.4 Background of Deep Reinforcement Learning

In this section, we will introduce some background knowledge about Reinforcement Learning (RL) since this is the main algorithm we used to build our multimodal agent. RL is a subfield of machine learning that draws inspiration from behaviourist psychology and is rooted in the

ideas of learning from interaction with the environment to achieve a goal. It is one of three basic machine learning paradigms, along with supervised learning and unsupervised learning.

6.2.4.1 Markov Decision Process

For reinforcement learning, the problem is usually stated as a Markov Decision Process (MDP). A MDP is a mathematical model commonly used in the fields of operations research and decision theory to describe an environment for sequential decision-making of a dynamic system. Figure 6.6 shows a graphical representation of MDP. An MDP is typically defined by the following components:

- **States (S):** A set of states in which the system should be.
- **Actions (A):** A set of actions that the agent can take.
- **Transition probabilities (P):** A probabilistic model $P(s'|s, a)$ that describes how a state can transfer to another state when an action is taken
- **Rewards (R):** A function that assigns a real number to each state action pair, representing the immediate reward received after transitioning from state to state.

The goal of an MDP is to find a policy π that maximises the expected sum of rewards over time. Solving an MDP usually involves finding the optimal policy that yields the highest reward given any input state. We demonstrate an example of MDP for agent training in Figure 6.7.

Reinforcement learning can be broadly divided into two categories according to Figure 6.8: model-based RL and model-free RL. The primary distinction between the two lies in the agent's access to a model of the environment. In model-based RL, the agent uses this model to plan ahead by simulating future states and rewards. One of the seminal contributions in this category is AlphaZero [138], which uses Monte Carlo Tree Search (MCTS) to optimise move selection for both players in various games. The agent, trained through self-play, achieves superhuman performance in multiple challenging domains.

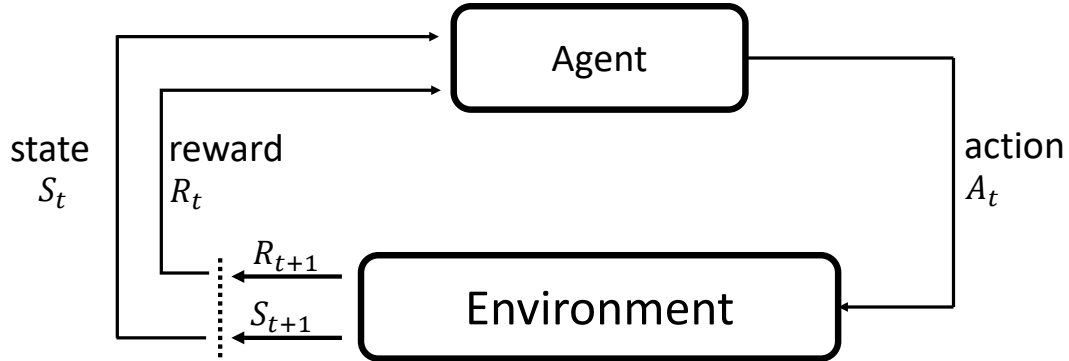


FIGURE 6.6. A example of the Markov Decision Process (MDP).

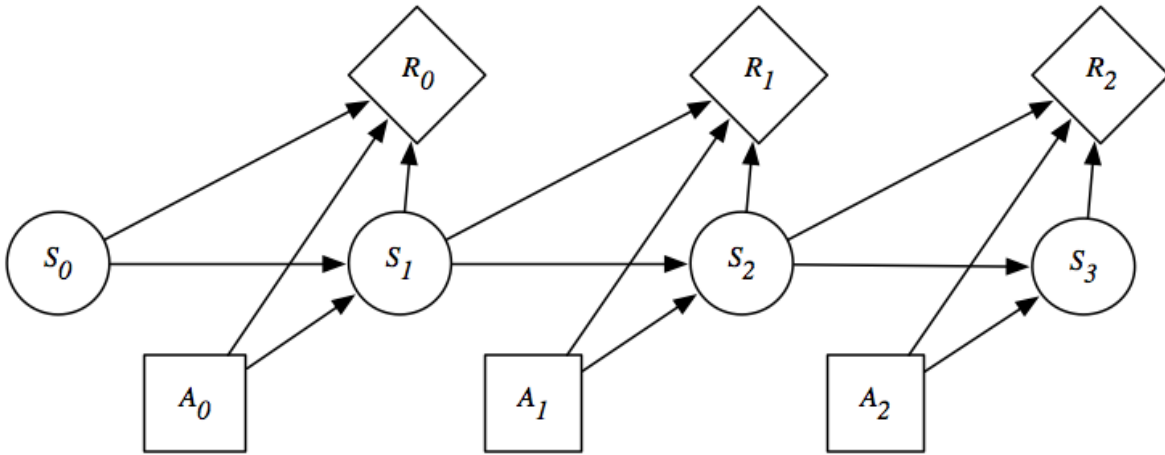


FIGURE 6.7. An example of the Markov Decision Process for agent training. Given every state s_t , an action a_t is applied to the environment, causing the state to transit to another state s_{t+1} . The reward for this action r_t is also gained in this process. We collect the reward and proceed with the training procedure.

However, implementing and fine-tuning model-based RL can be complex due to its inherent complexity. Despite these challenges, model-based methods have attracted considerable attention and have been rigorously explored by researchers. In contrast, this paper adheres to the model-free RL paradigm for problem modelling.

Model-free RL can also be categorised into two types: Policy Optimisation and Q-Learning Algorithms. Q-Learning focusses on learning the value for each pair of state actions. Theoretically, we can construct a Q table for the state-action pair to store all the values. However,

many RL problems have very large state spaces or even infinite state spaces. It is not possible or too complex and time-consuming to create such a large value table to solve those problems. Deep Q Network (DQN) [150] uses a neural network to simulate the Q values for each action based on the state. Policy optimisation methods allow the agent to learn directly the policy function that maps state to action. The policy is determined without using a value function. Methods like Policy Gradient (PG) [106] train an agent that produces a probability distribution of actions given an observation.

6.2.4.2 Bellman Equation

Richard Bellman proposed the Bellman Equation [125, 124] to solve dynamic programming. In the context of RL and MDP, the Bellman equation is often used to characterise the value function or the Q-function, which represents the expected sum of rewards when starting from a particular state or a particular state-action pair, respectively. Bellman equation plays a foundational role in both value-based RL and policy-based RL. For value-based RL, the goal is to find the optimal value function $V^*(s)$. The Bellman equation for this can be defined as:

$$V^*(s) = \max_a \mathbb{E} [r + \gamma V^*(s')]$$

where $\max_a$ is the maximisation over all possible actions, r is the immediate reward when state s transits to s' under action a , and γ is the decay factor for the reward. The Bellman Equation in Policy-Based RL, denoted as $V^\pi(s)$, can be defined as:

$$V^\pi(s) = \mathbb{E}_{a \sim \pi} [r + \gamma V^\pi(s') | S_t = s]$$

We can see that both value-based RL and policy-based RL make use of the Bellman Equation to break down the problem of finding optimal solutions into more manageable subproblems. In summary, the Bellman equation provides a mathematical structure that enables the systematic calculation of optimal policies or value functions in RL. It is the foundation for many RL algorithms.

Here, we introduce some popular RL algorithms under the model-free RL category.

6.2.4.3 Deep Q Network

As mentioned above, **Deep Q Network(DQN)** is one of the first studies to use deep learning in value-based algorithms. It extends traditional Q-Learning to high-dimensional state spaces by employing deep neural networks as function approximators for the Q-function. The objective is to find the optimal Q-function $Q^*(s, a)$ that satisfies the Bellman equation:

$$Q^*(s, a) = \mathbb{E} \left[r + \gamma \max_{a'} Q^*(s', a') \right]$$

In DQN, a neural network parameterized by θ is used to approximate this optimal Q function, since we do not have access to Q^* :

$$Q(s, a|\theta) \approx Q^*(s, a)$$

Since the Q function for the policy obeys the Bellman equation, the difference between two Q values of the two consecutive states s and s' , also known as the temporal difference error δ is defined as:

$$\delta = Q(s, a) - (r + \gamma \max_a Q(s', a))$$

We can use Mean Squared Loss Huber loss to minimise the temporal difference error. We calculate this over a batch of transitions B , which are sampled from the replay memory:

$$\mathcal{L} = \frac{1}{|B|} \sum_{(s,a,s',r) \in B} \mathcal{L}(\delta)$$

$$\text{where } \mathcal{L}(\delta) = \begin{cases} \frac{1}{2}\delta^2 & \text{for } |\delta| \leq 1, \\ |\delta| - \frac{1}{2} & \text{otherwise.} \end{cases}$$

Deep neural networks are used to process input and output actions. Simple networks like feed-forward networks are enough for most of the RL tasks. In order to train the network and stabilise the process, experience replay and target networks are commonly used. For replay of the experience, the transitions (*Transition, state, next_state, reward*) are stored

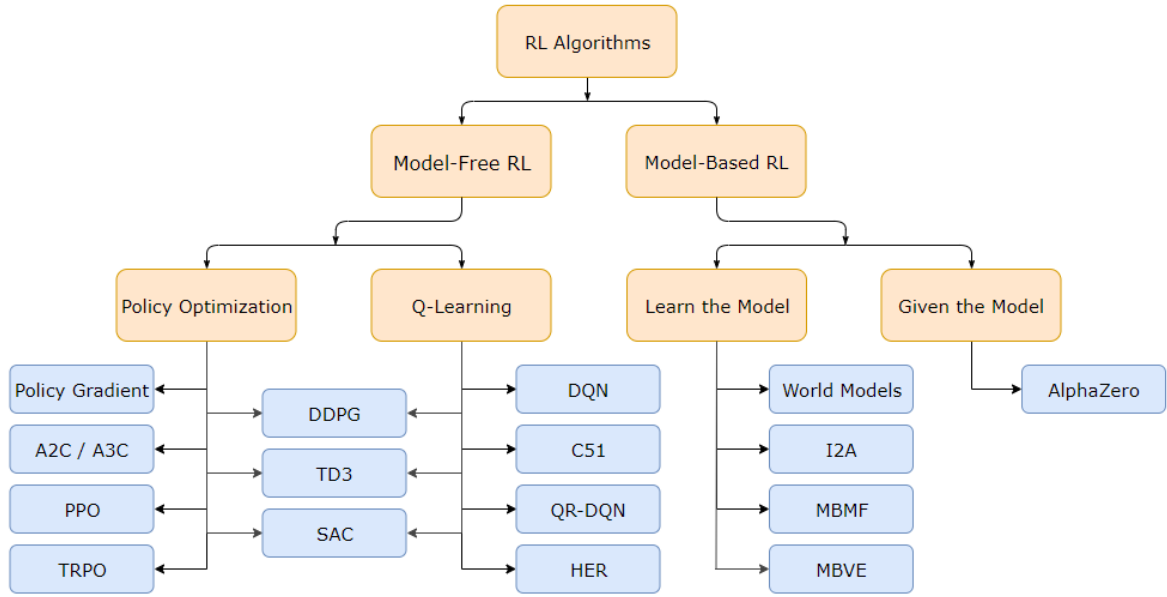


FIGURE 6.8. A Taxonomy of RL Algorithms according to OpenAI.

in a replay buffer. Then, a sampling method is implemented for selecting a random batch from the transitions for training. For target network, a separate network with parameters θ^- is maintained for calculating the target y , reducing the correlation between the target and the parameters being updated.

6.2.4.4 Policy gradient Algorithm

Policy gradient is a classic policy-based RL algorithm that directly optimises the policy. The policy parameters are adjusted to increase the expected return. For a stochastic and parameterized policy π_θ , we expect to maximise the return $J(\pi_\theta)$:

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$$

where τ is the trajectory $\tau = s_0, a_0, s_1, a_1, \dots, s_T, a_T$ and $R(\tau)$ is the total reward for the whole trajectory. The policy is updated through gradient ascent:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_\theta)|_{\theta_k}$$

where k represents the updating step, α is the learning rate, and $\nabla_\theta J(\pi_\theta)$ represents the gradient of the objective function $J(\pi_\theta)$. The gradient is a vector pointing to the direction of

the highest ascent of the function $J(\pi_\theta)$. Algorithms that optimise the policy in this way are called policy gradient algorithms.

We can derive the basic policy gradient with the following equation:

$$\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} [R(\tau)] \\
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \int_{\tau} P(\tau|\theta) [R(\tau)] \\
\nabla_\theta J(\pi_\theta) &= \int_{\tau} \nabla_\theta P(\tau|\theta) [R(\tau)] \\
\nabla_\theta J(\pi_\theta) &= \int_{\tau} P(\tau|\theta) \nabla_\theta \log P(\tau|\theta) [R(\tau)] \\
\nabla_\theta J(\pi_\theta) &= \mathbb{E}_{\tau \sim \pi_\theta} [\nabla_\theta \log P(\tau|\theta) R(\tau)]
\end{aligned}$$

In this equation, $P(\tau|\theta)$ represents the probability of a trajectory τ occurring under the policy π_θ :

$$P(\tau|\theta) = p(s_0) \prod_{t=0}^T \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t)$$

where $p(s_0)$ is the initial state distribution. All steps are iterated over each time step to get $P(\tau|\theta)$. The Log-probability of a trajectory can be expressed as:

$$\log P(\tau|\theta) = \log p(s_0) + \sum_{t=0}^T (\log p(s_{t+1}|s_t, a_t) + \log \pi_\theta(a_t|s_t))$$

Hence, the basic policy gradient derivation can be expressed as:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^T \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau) \right]$$

We collect a set of (N) trajectories $\mathcal{D} = \{\tau_i\}_{i=1, \dots, N}$ by letting the agent act freely in the environment with its policy π_θ . The policy gradient can be estimated with the following

equation:

$$\hat{g} = \frac{1}{N} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) R(\tau)$$

This equation offers a practical way to approximate the true policy gradient using sampled data. The gradient of the logarithmic probability of taking action a_t at state s_t indicates how much we should adjust the parameter θ to increase the probability of action.

For the RL algorithms introduced above, DQN requires fewer samples for effective learning, as it makes better use of each sample by storing it in a replay buffer and reusing it multiple times. It is also more stable and less sensitive due to the choice of hyperparameters. However, if the complexity of the action space is too high, DQN might struggle a lot.

Compared to DQN, Policy Gradients can be applied to a wider range of problems, particularly those with complex action spaces, including continuous actions. For example, choosing the best visual detections that match given a set of wireless received signals and tracking results. Normally, Policy Gradients converge faster, but they often suffer from getting stuck in local optima. Therefore, it is very important to design the reward function to address this issue.

6.2.4.5 Actor-Critic Method

The choice between DQN and Policy Gradients often boils down to the specifics of the problems at hand. Figure 6.9 shows the relationship between policy-based algorithms and value-based algorithms. There are also some hybrid approaches that can be used to combine these two algorithms to provide better results. Actor-Critic methods [149] is a popular hybrid algorithm that uses both a policy (actor) and a value function (critic) when training the agent.

Figure 6.10 provides an overview of a standard Actor-Critic algorithm. There are two agents instead of one used in this method: the actor and the critic agent:

- **Actor** learns the policy of the agent. Given a state s , the actor outputs a probability distribution over the available actions a ;
- **Critic** learns to evaluate the value function $V(s)$ of a given state s ;

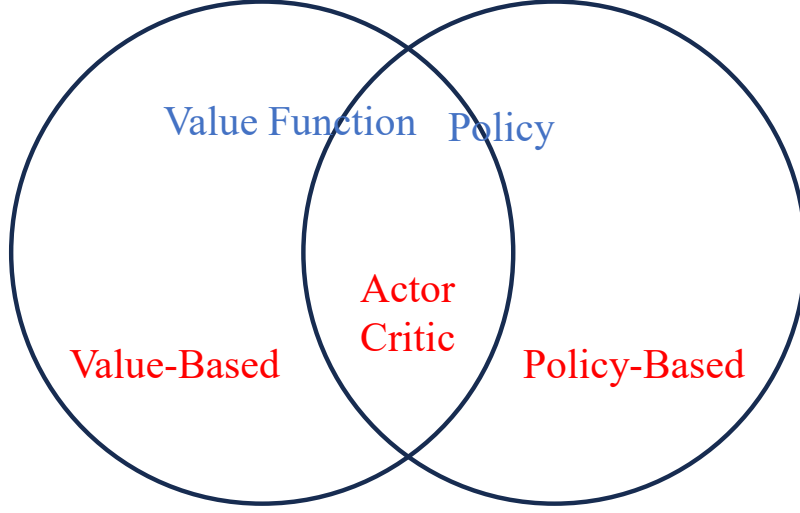


FIGURE 6.9. Value-Based RL and Policy-Based RL.

At each time step, the environment generates a state s that is observable by both the actor and the critic agents. The actor takes an action a , and the critic provides an evaluation $V(s)$ of the current state. Upon receiving the action of the actor, the environment generates a reward r and transitions to a new state s' . The critic then updates its evaluation to $V(s')$.

Subsequently, the advantage A is calculated to gauge the quality of the action a relative to the expected average reward in state s . A common formula for computing the advantage is:

$$A = r + \gamma V(s') - V(s)$$

Once the advantage A has been computed, the critic's parameters are updated to minimise the discrepancy between its current value estimate $V(s)$ and the observed return $r + \gamma V(s')$. This is typically achieved through a mean-square loss optimisation process.

Finally, the actor's policy is updated to maximise the expected return, guided by the advantage A . Actions that result in a higher advantage are thereby given increased probabilities of being selected in future time steps.

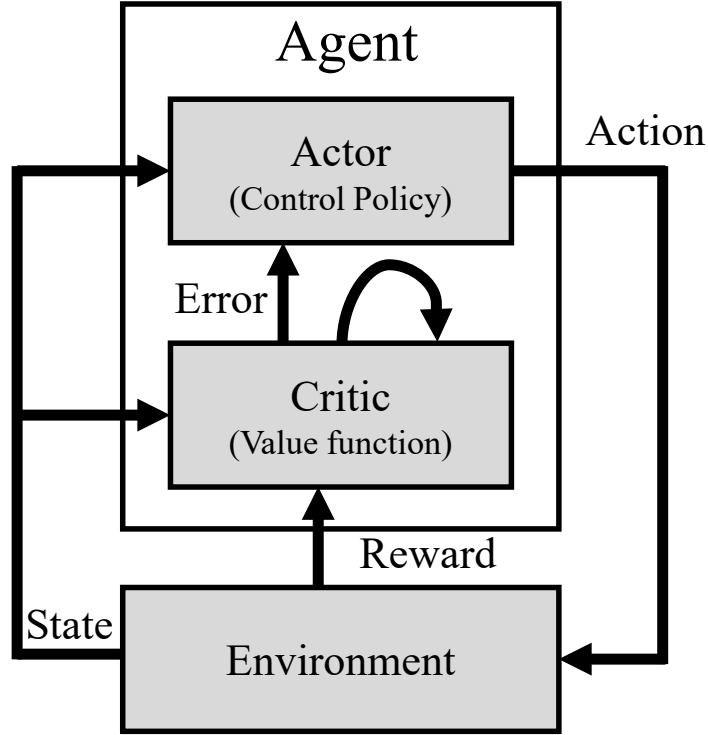


FIGURE 6.10. An overview of the standard Actor-Critic framework.

$$\nabla_{\theta} J(\pi_{\theta}) \approx \frac{1}{N} \sum_{\tau \in \mathcal{D}} \sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A$$

6.2.4.6 Trust Region Policy Optimisation

To avoid parameter updates that change the policy too much in one step and improve training stability, many advanced Actor-Critic algorithms are proposed. Trust Region Policy Optimization (TRPO) [88] proves that minimising a certain surrogate objective function guarantees policy improvement with nontrivial step sizes. A KL divergence constraint on the size of the policy update at each iteration to ensure that the policy update process is stable.

The theoretical TRPO update can be described as:

$$\theta_{k+1} = \operatorname{argmax}_{\theta} \mathcal{L}(\theta_k, \theta)$$

subject to

$$\bar{D}_{KL}(\theta||\theta_k) \leq \delta$$

where $\mathcal{L}(\theta_k, \theta)$ is the surrogate advantage. The surrogate advantage is a measure of how the policy performs relative to its older version π_{θ_k} :

$$\mathcal{L}(\theta_k, \theta) = \mathbb{E}_{s \sim \pi_\theta} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a) \right]$$

and $\bar{D}_{KL}(\theta||\theta_k)$ is the average Kullback-Leibler (KL) Divergence between policies that across different states:

$$\bar{D}_{KL}(\theta||\theta_k) = \mathbb{E}_{s \sim \pi_\theta} [KL(\pi_\theta(\cdot|s)||\pi_{\theta_k}(\cdot|s))] \leq \delta$$

The KL-Divergence score quantifies the similarity between two probability distributions. Given two probability distributions $p(x)$ and $q(x)$ over a variable X , the KL-Divergence can be defined as follows:

$$\bar{D}_{KL}(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

The primary goal of TRPO is to maximise the objective function $\mathcal{L}(\theta_k, \theta)$, subject to a trust region restriction. This constraint, measured by the KL-Divergence between the old and new policies, ensures that the new policy does not diverge significantly from the old policy. Specifically, the constraint enforces that the divergence is within a threshold parameter δ . This mechanism offers a safety net by mitigating overly aggressive updates that could otherwise jeopardise the learning process or lead to overfitting.

6.2.4.7 Proximal Policy Optimisation

However, implementing TRPO can be computationally challenging due to the complexity involved in handling the trust region constraint. Although approximations can be used to reduce computational overhead, TRPO remains relatively inaccessible for many applications.

In contrast, Proximal Policy Optimisation (PPO) [87] offers a more practical alternative while retaining the core benefits of TRPO. PPO simplifies the trust region mechanism by replacing the constrained optimisation problem with a clipped objective function. This change eliminates the need for second-order optimisation methods commonly used in TRPO, making PPO easier to implement and computationally more efficient.

The PPO algorithm updates the policy via:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{s \sim \pi_{\theta_k}} [L(s, a, \theta_k, \theta)],$$

where L is given by

$$L(s, a, \theta_k, \theta) = \min \left(\frac{\pi_{\theta}(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a), \operatorname{clip} \left(\frac{\pi_{\theta}(a|s)}{\pi_{\theta_k}(a|s)}, 1 - \epsilon, 1 + \epsilon \right) \right) A^{\pi_{\theta_k}}(s, a),$$

where ϵ is a hyperparameter defining the “trust region” for the policy update. A simpler version of the equation can be derived in the following way:

Given that $\epsilon \in (0, 1)$, define

$$F(r, A, \epsilon) \doteq \min(rA, \operatorname{clip}(r, 1 - \epsilon, 1 + \epsilon)A).$$

When $A \geq 0$,

$$\begin{aligned}
F(r, A, \epsilon) &= \min(rA, \text{clip}(r, 1 - \epsilon, 1 + \epsilon)A) \\
&= A \min(r, \text{clip}(r, 1 - \epsilon, 1 + \epsilon)) \\
&= A \min \left(r, \left\{ \begin{array}{ll} 1 + \epsilon & r \geq 1 + \epsilon \\ r & r \in (1 - \epsilon, 1 + \epsilon) \\ 1 - \epsilon & r \leq 1 - \epsilon \end{array} \right\} \right) \\
&= A \left\{ \begin{array}{ll} \min(r, 1 + \epsilon) & r \geq 1 + \epsilon \\ \min(r, r) & r \in (1 - \epsilon, 1 + \epsilon) \\ \min(r, 1 - \epsilon) & r \leq 1 - \epsilon \end{array} \right\} \\
&= A \left\{ \begin{array}{ll} 1 + \epsilon & r \geq 1 + \epsilon \\ r & r \in (1 - \epsilon, 1 + \epsilon) \\ r & r \leq 1 - \epsilon \end{array} \right\} \\
&= A \min(r, (1 + \epsilon)) \\
\therefore F(r, A, \epsilon) &= \min(rA, (1 + \epsilon)A)
\end{aligned}$$

When $A < 0$,

$$\begin{aligned}
F(r, A, \epsilon) &= \min(rA, \text{clip}(r, 1 - \epsilon, 1 + \epsilon)A) \\
&= A \max(r, \text{clip}(r, 1 - \epsilon, 1 + \epsilon)) \\
&= A \max \left(\begin{array}{ll} 1 + \epsilon & r \geq 1 + \epsilon \\ r, & r \in (1 - \epsilon, 1 + \epsilon) \\ 1 - \epsilon & r \leq 1 - \epsilon \end{array} \right) \\
&= A \left\{ \begin{array}{ll} \max(r, 1 + \epsilon) & r \geq 1 + \epsilon \\ \max(r, r) & r \in (1 - \epsilon, 1 + \epsilon) \\ \max(r, 1 - \epsilon) & r \leq 1 - \epsilon \end{array} \right\} \\
&= A \left\{ \begin{array}{ll} r & r \geq 1 + \epsilon \\ r & r \in (1 - \epsilon, 1 + \epsilon) \\ 1 - \epsilon & r \leq 1 - \epsilon \end{array} \right\} \\
&= A \max(r, (1 - \epsilon)) \\
\therefore F(r, A, \epsilon) &= \min(rA, (1 - \epsilon)A)
\end{aligned}$$

Summarizing, for all cases,

$$F(r, A, \epsilon) = \min(rA, g(\epsilon, A)),$$

where $g(\epsilon, A) = (1 + \epsilon)A$ for $A \geq 0$, and $g(\epsilon, A) = (1 - \epsilon)A$ for $A < 0$.

Hence, the simple version of the PPO policy update equation can be expressed as:

$$L(s, a, \theta_k, \theta) = \min \left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}, g(\epsilon, A^{\pi_{\theta_k}}) \right)$$

where

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A & A \geq 0 \\ (1 - \epsilon)A & A < 0. \end{cases}$$

An intuitive explanation is that we set a boundary for the policy update with the parameter ϵ . When the advantage is positive, meaning that the action becomes more likely, the objective

can increase, but not more than $(1 + \epsilon)A^{\pi_{\theta_k}}$. When the advantage is negative, meaning the action becomes less likely, the objective can decrease but also not more than $(1 - \epsilon)A^{\pi_{\theta_k}}$. The trust region here is set manually rather than determined by the KL divergence in TRPO. This is called the PPO-Clip.

In our study, we chose the PPO algorithm as our solution for training the agent. We carefully designed the environment, the action set, and the reward function so that the agent can find the best match between the local IMU and RSSI to the global tracklets. We adapted the algorithm to use vectorised state, action, and reward so that it can explore the temporal information in the environment.

6.3 DRL Environment of Pedestrian Identification

6.3.1 Problem Formulation

To use Deep Reinforcement Learning (DRL) in fusion of multimodal information, the problem must first be structured as a Markov Decision Process (MDP). Given a stochastic environment, at each timestamp t , the agent perceives a state s_t . Based on this state, the agent selects an action a_t from its available action space. This decision transitions the agent to a new state s_{t+1} , during which it receives a reward $R(s_t, a_t)$. The selection of the action a_t is dictated by a policy network $\pi(a_t|s_t)$, which subsequently influences the probability of state transition $P(s_{t+1}|s_t, a_t)$. The aim of our DRL algorithm is to identify a policy that maximises the cumulative reward over time.

Consequently, we formulate the identity matching process as a Reinforcement Learning (RL) problem. In this setup, different input modalities are treated as the agent's observations, and the agent is tasked with determining the most suitable visual detection for the device at each timestamp. A higher success rate of the match results in a more substantial reward, motivating the agent to identify and integrate the most distinguishable matching features. In this context, the state-transition function is dictated by the natural progression of time. As a result, unlike traditional MDP, each agent's step number is restricted to one (a similar setup can be found

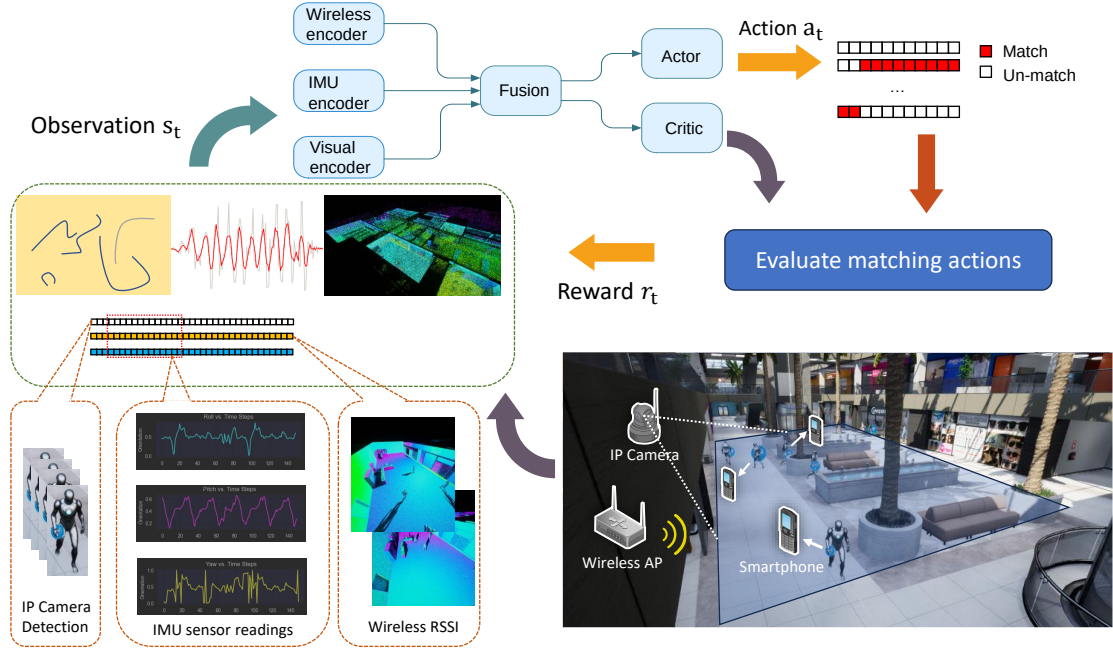


FIGURE 6.11. Overview of our proposed DRL training process for pedestrian identity matching. Three types of input are extracted from the emulated environment: visual tracking results from IP Cameras, acceleration and orientation of the device, and received wireless signal strength. These inputs are processed as observation and fed to the agent to perform identity matching. At each timestamp, the agent maps the device to a visual detection. The predictions are evaluated by the reward generated by the environment.

in [109]). However, we provide the agent with various lengths of observation time series, enabling it to effectively use temporal information.

In the following section, we introduce the state space, action space, and reward function that we used in our DRL environment to train the identity matching agent.

6.3.2 State Space

In our proposed system, each agent corresponds to a unique pedestrian within the scene. At each timestamp t , we assume that the pedestrian receives M wireless signals and visual detection results for N people in the same scene. The pedestrian's device also collects its local acceleration and orientation with the IMU sensors pre-installed on the device. Therefore,

the observation space s_t for a pedestrian navigating in a dynamic environment can be defined as follows:

$$s_t = (\mu_t, \phi_t, v_t)$$

where

$$\mu_t = \{\text{TX}_{m,t} : \text{RSSI}_{m,t} | m \in \{1, \dots, M\}\} \quad (6.2)$$

$$\psi_t = \{l_t, o_t\}$$

$$v_t = \{(x_{n,t}, y_{n,t}) | n \in \{1, \dots, N\}\}$$

The state space s_t is thus composed of three components: the RSSI received locally μ_t , the local IMU sensor data ϕ_t and the global visual tracking result v_t .

For each detected RSSI, the corresponding transmitter's name and the RSSI reading are logged. This procedure results in a total of M pairs, each consisting of a transmitter name and an associated RSSI value. $TX \in \mathbb{R}^C$ is the transmitter name transferred to an embedding vector, where C is the size of the characteristic of the TX name. $RSSI$ is the received signal strength value for that transmitter.

For the IMU sensor, the acceleration feature $l_t \in \mathbb{R}^L$ and the orientation feature $o_t \in \mathbb{R}^O$ are calculated from the smartphone coordinate system (SCS) to the earth coordinate system (ECS) with a quaternion [42]. In our simulated environment, it is calculated with UE5 built-in function.

For visual detection, we simulate the detection results of pedestrian heads and use the centre point coordinate to represent its location on the camera view. The x and y coordinates of the camera view are calculated with the world location of the pedestrian's head and the camera's view projection matrix.

6.3.3 Action Space

The agent performs two key tasks: Feature extraction and Feature Fusion. We denote the feature extraction module as $\mathcal{F}$ and the feature fusion module as $\mathcal{G}$. The feature fusion module effectively takes advantage of complementary information from the different modalities and encapsulates them into a single coherent representation, resulting in a set of feature vectors $f_t = \mathcal{F}(s_t)$. Next, these feature vectors can be fused using the feature fusion module, resulting in a final feature vector $g_t = \mathcal{G}(f_t)$. Now, the matching probabilities can be calculated on the basis of g_t . We denote the function to compute these probabilities by $\mathcal{H}$, so the matching probabilities are $a_t = \mathcal{H}(g_t)$.

Therefore, the overall operation of the agent can be represented as a composition of these functions:

$$a_t = \mathcal{H}(\mathcal{G}(\mathcal{F}(s_t))) \quad (6.3)$$

The action space for time t can be determined as follows:

$$\begin{aligned} a_t = \pi_t &= [a_{1,t}, a_{2,t}, \dots, a_{N,t}] \\ \text{where } a_{t,n} &\in [0, 1] \text{ for } n \in \{1, \dots, N\} \\ \text{and } \sum_{n=1}^N a_{n,t} &= 1 \end{aligned} \quad (6.4)$$

At each time step t , the agent selects an action a_t from the action space $\mathcal{A}$, which determines the matching probabilities for each visual detection. It is important to note that these probabilities can be interpreted as the confidence of the agent that each visual detection matches its identity. As such, a higher probability means that the agent is more confident that the visual detection matches its identity.

6.3.4 Reward

A matching reward $r_{match}(s_t, a_t)$ should indicate whether a successful match has been made:

$$r_{match}(s_t, a_t) = \begin{cases} 1, & \text{if match} \\ 0, & \text{if unmatch} \end{cases} \quad (6.5)$$

This function assigns a reward of 1 for a match and 0 for a nonmatch.

To effectively leverage temporal information, the environmental model should provide the agent with multi-step observations (observation for a short trajectory). For a trajectory starting at t_0 and extending for τ frames, we can adapt our observation space to accommodate a sequence of states s_t , extending from t_0 to t_τ . This modification enables the agent to make decisions based on a continuous stream of information, enhancing the performance of our model in dynamic scenarios.

$$\mathbf{s}_t = \{s_t | s_t = (\mu_t, \psi_t, v_t)\} \quad t \in \{0, \dots, \tau\} \quad (6.6)$$

Given the observation of the selected trajectory, the agent takes action of assigning the probability of the identity matching:

$$\mathbf{a}_t = \pi_t = \{a_0, a_1, \dots, a_\tau\} \quad (6.7)$$

An identity matching prediction is made for every time step in the trajectory. The reward function thus becomes:

$$\mathbf{r}_t^{match}(s_t, a_t) = \{r(s_0, a_0), \dots, r(s_\tau, a_\tau)\} \quad (6.8)$$

For each time step during the selected trajectory, each correct identity assignment suggests that the agent has taken the right actions and should thus be awarded. On the other hand, an incorrect prediction implies that the agent has failed to extract relevant motion features or failed to fuse the extracted features; hence, it should be punished. To encourage the agent to

learn more evenly distributed matching probabilities rather than focussing on a few samples in the batch, we add log-based penalty entropy to the reward function based on the overall matching probability during τ frames.

$$\begin{aligned}\bar{p}_t &= \frac{1}{\tau} \sum_{i=t}^{\tau} r_{match}(s_i, a_i) \\ r_t^{pen} &= \min(\lambda, \log(\bar{p}_t + 1 \times 10^{-9})) \\ r_t &= r_t^{match} + r_t^{pen}\end{aligned}\tag{6.9}$$

6.4 Training RL Agent for Identification Matching

We illustrate our training procedure in Figure 6.11. To train our agent, we employ Proximal Policy Optimisation (PPO)[87], an on-policy DRL algorithm. This methodology involves the random sampling of fixed-length trajectory segments from the environment and the concurrent training of both the actor and the critic networks. The actor network predicts the action at each timestamp, whereas the critic network evaluates the resultant state to ascertain the relative success or failure of the chosen action. These networks are refined through the optimisation of a surrogate objective function using stochastic gradient ascent. The use of PPO ensures an effective balance between exploration and exploitation, thereby facilitating a robust learning process.

The main objective function is defined as follows.

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(\rho_t(\theta) \hat{\mathbf{A}}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{\mathbf{A}}_t) \right]\tag{6.10}$$

where $\rho_t(\theta)$ is the vectorized probability ratio:

$$\rho_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}\tag{6.11}$$

for current policy and the previous policy at k -th iteration. $\hat{\mathbf{A}}_t$ is the vectorized advantage estimates [133], which is calculated by the following formula:

$$\mathbf{A}^\pi(s, a) = \mathbf{Q}^\pi(s, a) - \mathbf{V}_{\phi_k}(s) \quad (6.12)$$

where $\mathbf{Q}^\pi$ are the Q-values of the state-action pairs (s, a) and $\mathbf{V}_{\phi_k}$ are the value determined by the critic network based on the current observation on the k -th iteration. To calculate $\mathbf{Q}^\pi$, we first compute the rewards-to-go value for one sampled trajectory:

$$G(s_t) = \sum_{i=t}^{\tau} \gamma_1^{i-t} r(s_i, a_i) \quad (6.13)$$

where $G(s_t)$ is the cumulative reward function within one trajectory. s_t is the observation at timestamp t , containing a length of τ timestamps' input features, γ is the discount factor and $r(s_i, a_i)$ is the reward given by the action and the observation according to the equation 6.9.

For a trajectory that starts at time t , the agent generates the corresponding predictions for all τ timestamps. Rewards associated with future timestamp predictions are multiplied by a decay parameter γ , and then added to the prediction of the current timestamp. This approach is crucial as it encourages the model to incorporate temporal information from the input data. Consequently, predictions made for future timestamps contribute to the initial predictions, rather than the model merely concentrating on local feature comparisons. The rewards-to-go for a single episode is then written as:

$$\hat{\mathbf{R}}_t = \{G(s_i) | i \in \{0, \dots, \tau\}\} \quad (6.14)$$

All cumulative reward vectors in one training episode are represented as follows:

$$\hat{\mathbf{R}}^{\pi_{\theta_k}} = \{\hat{\mathbf{R}}_i | i \in \{0, \dots, T\}\} \quad (6.15)$$

where T represents the number of time steps in a single training episode. Contrary to the original PPO, rewards from different time steps are not aggregated because the temporal information is already encapsulated within the training of a single trajectory.

The clip function, $\text{clip}(\mathbf{r}_t(\theta), 1 - \epsilon, 1 + \epsilon)$, constrains the probability ratio to the interval $[1 - \epsilon, 1 + \epsilon]$. This function was introduced in the original PPO algorithm as a safeguard to prevent the agent from making too large updates during the gradient ascent process, thus ensuring stability and robustness in learning.

To update the policy, we maximise the updated PPO-Clip objective using the simplified PPO-Clip:

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \frac{1}{|\mathcal{D}_k|T} \sum_{d \in \mathcal{D}_k} \sum_{t=0}^T \min\left(\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}\right) \hat{\mathbf{A}}_t, g(\epsilon, \hat{\mathbf{A}}_t) \quad (6.16)$$

where

$$\hat{\mathbf{A}}_t = \mathbf{A}^{\pi_{\theta_k}}(s_t, a_t) \quad (6.17)$$

$$g(\epsilon, A) = \begin{cases} (1 + \epsilon)A, & A \geq 0 \\ (1 - \epsilon)A, & A < 0. \end{cases} \quad (6.18)$$

Given the calculated cumulative rewards, the value function is updated using the MSE loss:

$$\phi_{k+1} = \underset{\phi}{\operatorname{argmin}} \frac{1}{|\mathcal{D}_k|T} \sum_{d \in \mathcal{D}_k} \sum_{t=0}^T (\mathbf{V}_{\phi_k}(s) - \hat{\mathbf{R}}_t)^2 \quad (6.19)$$

For each iteration during the training, we first collect a set of trajectories $\mathcal{D}_k = \{d_i\}$ by running policy π . At each time step, the agent assigns matching probabilities between the user and the global visual detections. After collecting the trajectories, the cumulative rewards are calculated in batches and the prediction value is also generated by the critic model simultaneously. Therefore, the actor policy and the critic policy are updated according to equations 6.16 and 6.19. We update the policy multiple times for each iteration. The details of our training process are summarised in algorithm 2.

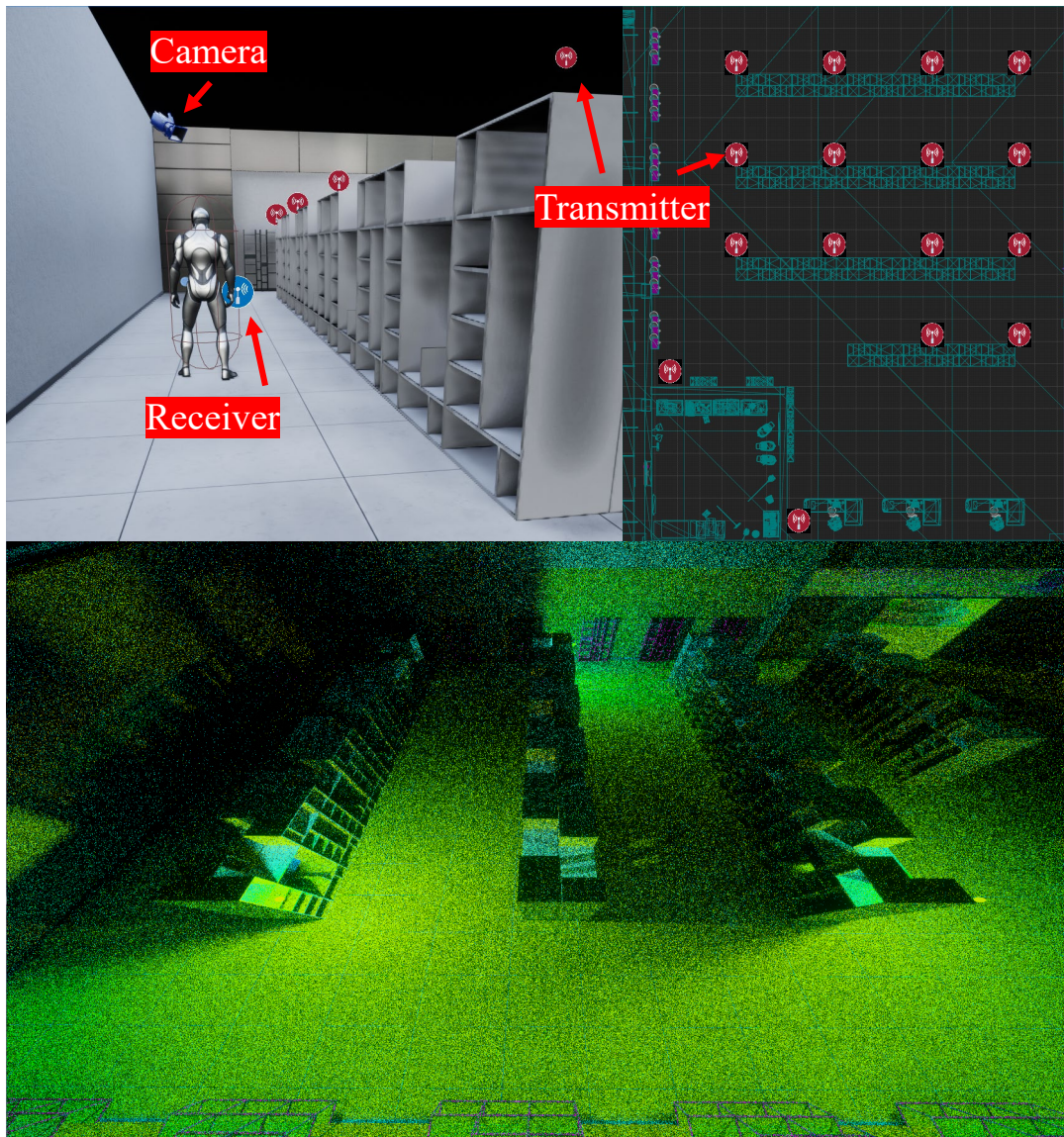


FIGURE 6.12. The provided illustration presents a comprehensive setup of a warehouse - one of the most classic indoor positioning application environments. The top left section provides a camera view of the indoor environment, the detailed floor plan which delineates the warehouse's architectural blueprint is shown in the top right, and the RSSI distribution of the environment is shown on the bottom. Wireless transmitters are positioned near the shelves. The strength of the wireless signal is intuitively represented through a gradient of colours; brighter shades correspond to a more robust signal. We disable some of the transmitters for better visualization.

Algorithm 2 Our agent training process using PPO-Clip.

Input: Initial actor policy θ_0 and initial value function parameter ϕ_0 , the tracking environment

```

while  $k = 0, 1, 2, \dots$  do
  For each episode, collect trajectory set  $\mathcal{D}_k$ , each trajectory, at time  $t$ , contains length of  $\tau$ 
  time frames of observation denoted as  $s_t$ , the correspond action  $\mathbf{a}_t$  is determined by the
  current policy  $\pi_k = \pi(\theta_k)$ 
  Compute cumulative reward  $\hat{\mathbf{R}}_k$ 
  Compute estimated advantage  $\hat{\mathbf{A}}_t$  and obtain the current value function  $\mathbf{V}_{\phi_k}(s)$ 
  while  $i = 0, 1, 2, \dots, n$  do
    Update actor policy with Adam optimizer
    Update critic policy with Adam optimizer
  end while
end while

```

6.5 Evaluation

6.5.1 Experiment Setup

Previous research efforts have typically deployed and tested their methods in various indoor building environments, a process that can often prove labour intensive, time-consuming, and fraught with constraints. For example, obtaining the exact location of the true location of the ground of pedestrians can be challenging. There are also ethical considerations surrounding data safety and privacy. Creating real-world experimental scenes with extremely high pedestrian density is not only difficult, but it may also present additional ethical and logistical issues.

To avoid these challenges, we constructed and tested our methods in custom indoor environments using a wireless signal emulator developed on the Unreal Engine 5 (UE5) platform. This approach offers several significant advantages:

- (1) **Control Over Variables:** Using Unreal Engine 5 (UE5), we can precisely control all variables in a virtual environment, including pedestrian density, movement patterns, signal strength, and environmental factors. This level of control is challenging to achieve in real-world scenarios. This allows us to rigorously test our approach for

indoor localisation in crowded settings by adjusting parameters to identify strengths and weaknesses, enabling effective model optimisation.

- (2) **Scalability and Reproducibility:** Conducting experiments in a simulated environment allows easy scaling of test scenarios and ensures that experiments are reproducible. It allows us to simulate different pedestrian densities without needing physical resources. Moreover, we can run the same experiment multiple times under the same conditions, ensuring data reliability.
- (3) **Cost-Effective:** Real-world testing can be time-consuming and expensive. Recruiting human subjects, securing indoor spaces, setting up equipment, and ensuring privacy and consent for data collection can all add up. Compared to virtual experiments in UE5, they can be set up and run at a much lower cost and can be done more quickly, facilitating rapid and iterative prototyping, as real-world experiments may require substantial logistical adjustments for similar changes.
- (4) **Safety and Privacy:** In real-world experiments, there are potential safety and privacy issues, especially when dealing with crowded indoor scenarios. Using a virtual environment like UE5 can eliminate these concerns as all 'subjects' are artificially intelligent robot actors.

In our simulation, we placed multiple wireless transmitters throughout the environment and gathered the corresponding data. We made the assumption that every pedestrian in the scene is carrying a smartphone equipped with a wireless signal receiver and an Inertial Measurement Unit (IMU) sensor, as these features are standard on most modern smartphones. In a further effort to enhance the realism of our simulations, we attached phones to the hands of pedestrian robot actors. This approach more accurately reflects real-world scenarios where pedestrians typically carry their phones in hand, allowing us to collect more precise and applicable data.

To further increase the realism of our simulation, we introduced several camera objects into the scene to emulate the presence of security cameras in real world settings. We collected visual detection data using built-in blueprint functions provided by UE5. These functions allow for calculations involving the camera object's view and projection matrices as well as the world locations of the pedestrians. This detailed approach allowed us to develop a

comprehensive and realistic simulation of a crowded indoor environment. The visual detection results are produced by the camera’s view and projection matrix to simulate the detection results with state-of-the-art visual detection algorithms like YOLO [36] or Swin [174]:

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Q \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (6.20)$$

where (u, v) are the x and y coordinates of the pixel in the camera, z_c is the z-coordinate of the camera relative to the world origin, (x_w, y_w, z_w) are the world coordinate of the source of the light ray which hits the camera object and Q is the camera’s view-project matrix consists of both the intrinsic and extrinsic parameters. Only the objects that have directly line trace contact with the camera are sent to the environment for training & inferencing. Gaussian noise is added to all sources of inputs for better simulation purposes.

We demonstrate one of the testing areas in Figure 6.12. The area is a warehouse with several lines of shelves inside, which is one of the best use cases for indoor localisation and navigation applications. On the upper left side of the figure, we offer a camera view of the warehouse, providing a comprehensive understanding of the spatial layout. We demonstrate the sensors involved in the environment. The layout is placed in the top right. This top-down view allows for a clear understanding of the room’s configuration, including any obstructions that could potentially interfere with signal strength or distribution. The bottom of the figure presents a detailed visualisation of the distribution of wireless signals throughout the area. In this visualisation, a lighter colour represents a stronger wireless signal. We disable some of the transmitters and leave only three of them for better visualisation. From the figure we can see that the signal is stronger near the location of the deployed transmitters.

6.5.2 Evaluation of Different Modalities

We conduct an evaluation of our proposed method using various combinations of characteristics and varying levels of people density. We deconstruct our policy to assess the impact

of different modalities. In the figures, the X-axis represents different time steps taken by the agent to match identities, while the Y-axis signifies the overall matching success probabilities for a batch of testing data. The time steps are determined by the system's delta time for frame generation, which is approximately 0.03 s. Each frame of data is sent to the DRL environment; therefore, a window of 30 time steps can be considered equivalent to 1 second in real life.

As evidenced in Figure 6.13, the performance of the agent using only IMU features is inadequate. Particularly in scenes with high person density, motion features alone cannot effectively distinguish individuals from the crowd. For a scene that contains ten people, the matching success rate begins at 11%, which approximates a random assignment. As the number of steps increases, the matching success rate gradually improves to 41%, but this performance is still unsatisfactory for our task.

In contrast, an agent using only wireless signals exhibits a significantly higher overall matching success rate compared to the agent using only IMU features. The rate starts at 62%, since the wireless signal comprises both distance and angle information, approximately using a fingerprint approach. However, it can be observed that as the trajectory exceeds 10 seconds, the performance begins to decline. This is likely due to fluctuation in the signal distribution, which disrupts the agent's prediction and results in less accurate features.

When we combine all the modalities, the performance is significantly improved and gradually increases the successive rate with time growth.

6.5.3 Evaluation of User Identification Matching

We evaluate our proposed approach against two baselines (1) RFCam [73] which uses wireless signal distance information, angle of arrival (AoA) information, and motion recognition, and (2) PHADE [141] which exploit the motion feature extraction of the smartphone's IMU sensor. Note that part of the RFCam method requires a multi-antenna wireless transmitter to obtain the AoA and distance information, which cannot be gathered on our emulation platform due to current system constraints, we replace it with a fingerprint method, which also gives the angle and distance of the devices relative to the camera. We used the same training data to

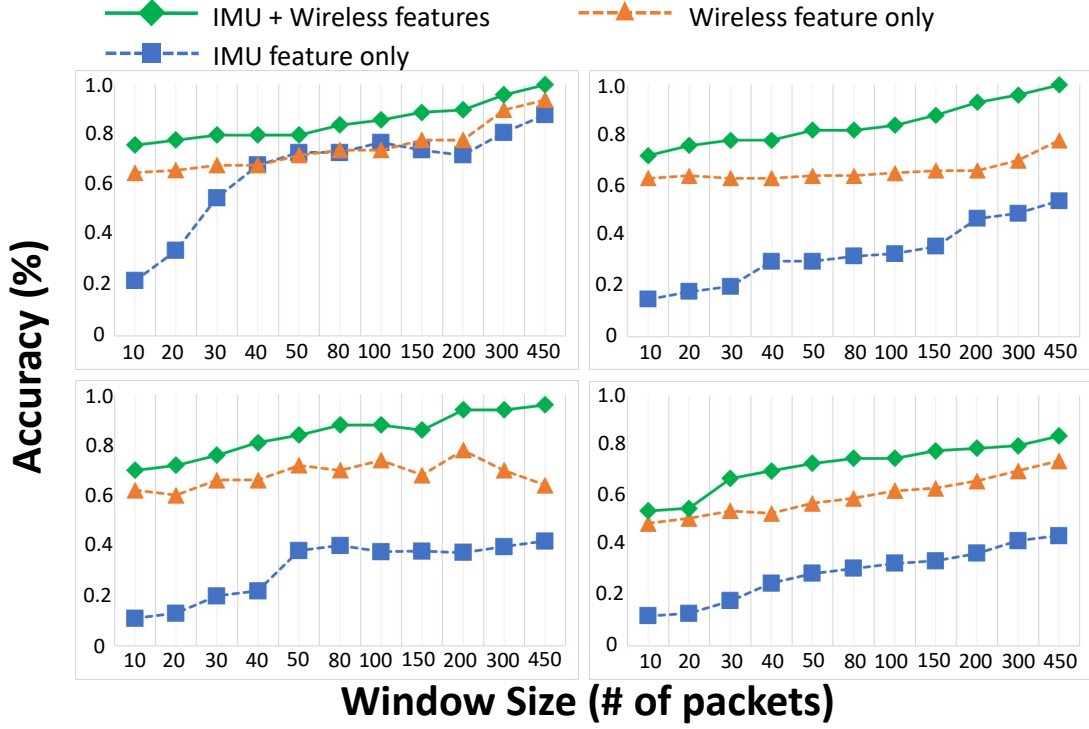


FIGURE 6.13. Ablation study on different modalities with three (top left), five (top right), ten (bottom left), and twenty (bottom right) pedestrians simultaneously present in the scene.

train the baseline method and our proposed method. From the result we can observe that our proposed method consistently outperforms the existing methods, especially for challenging scenes such as short time windows and crowded scenes.

6.5.4 Influence of Accumulating Reward

We also investigate the necessity of accumulating future rewards within a single tracklet. As illustrated in Figure 6.15, we compare two training logs with identical configurations except for reward shaping. The X-axis denotes the training steps, while the Y-axis represents both the reward and batch successful matching rate (due to their positive correlation). When contrasting the two figures, we observe the crucial role of the reward shaping during agent training. Without considering the temporal correlation of predictions within a trajectory, the agent tends to overfit the features rapidly, leading to poor performance. On the other hand, incorporating the accumulation of rewards encourages the agent to leverage the temporal

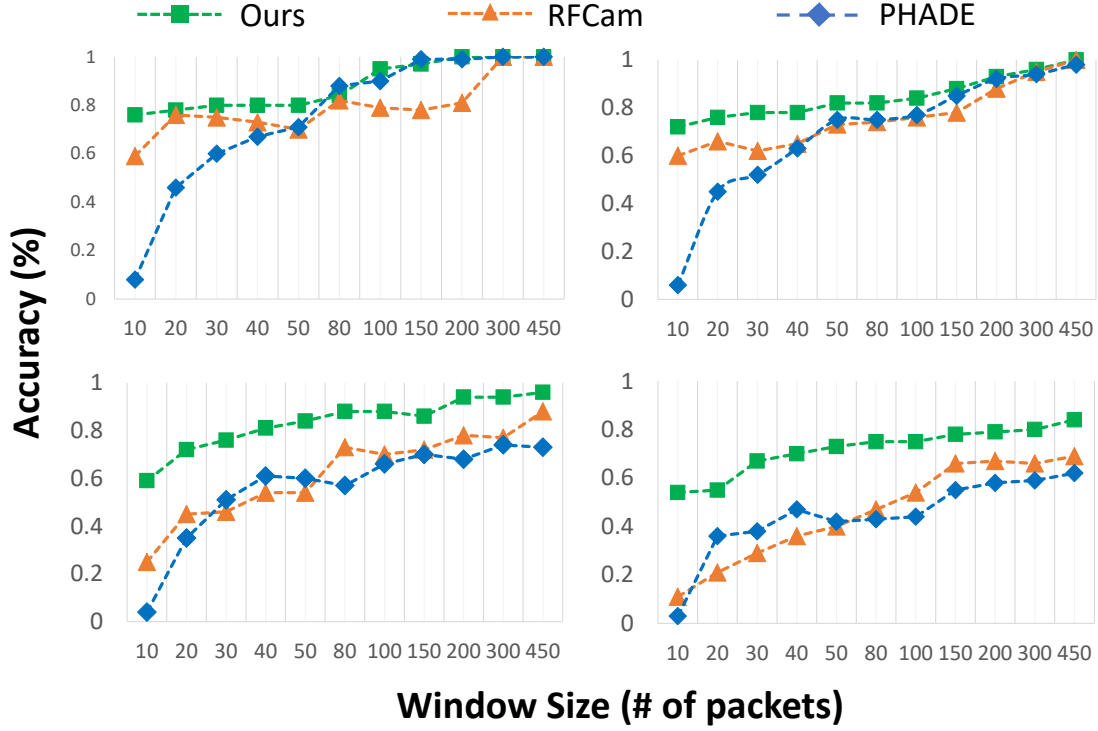
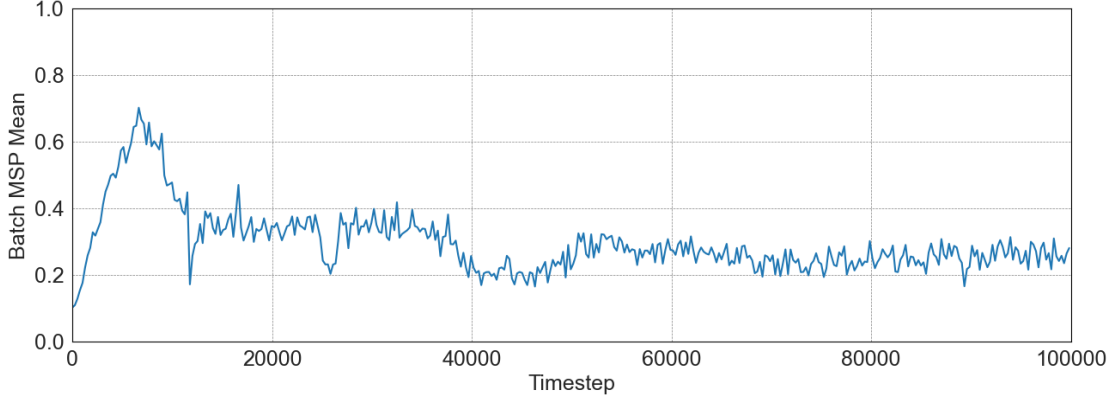


FIGURE 6.14. Benchmark evaluation against RFCam and PHADE for three (top left), five (top right), ten (bottom left), and twenty (bottom right) people scene.

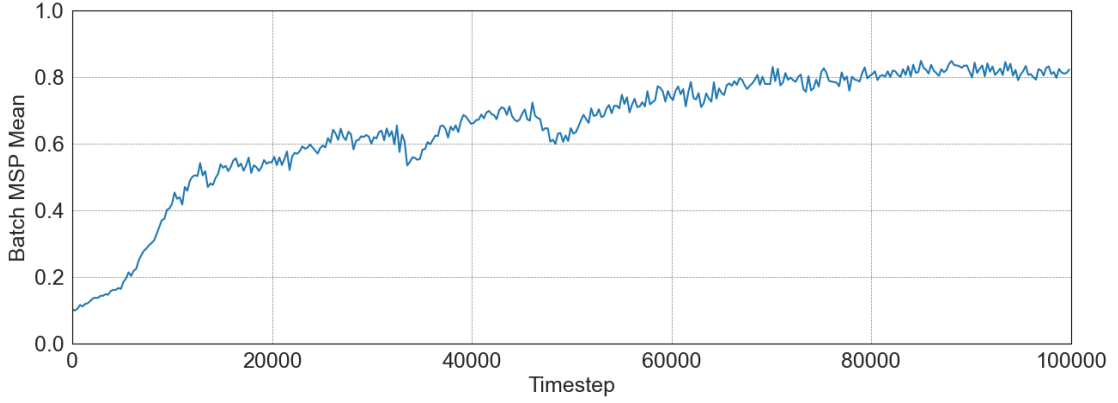
information of different input modalities, thereby facilitating the learning of relationships between the features.

6.5.5 Analysis with Case Study

To investigate the effect of the learnt policy on the extraction of motion patterns, we performed a case study to show the extracted feature. We choose one tracklet to show the example of the IMU feature encoder in Figure 6.16. On the top half figure there shows the ground truth motion trajectory captured by the camera. The same as in the previous example, the motion features are summed up and visualised on the bottom half. We directly observe that the two “turning” actions are successfully identified by the feature. The small wiggles at the end of the trace are also captured by the feature, as they are clearly identified with the low feature values.



(A) Training log with non-cumulative reward



(B) Training log with cumulative reward

FIGURE 6.15. Ablation study for training w/o cumulative reward.

We select another trajectory and compare it with the trajectory selected above to describe the visual feature extraction module of our agent in Figure 6.17. The feature maps are of shape $[150, 128]$ where 150 is the time length and 128 is the dimension of the feature. We sum up the feature dimension for visualisation. According to the figure, these two trajectories have very different visual features. This example shows that the learnt visual feature encoder effectively extracts time, distance, and motion features from the pedestrian tracking results.

In the process of associating IMU and wireless characteristics with visual detection outcomes, rewards are assigned to the agent based on formula 6.5. Figure 6.18 illustrates the reward distribution across the 150 time steps. Initial observations of the figure indicate uncertainty in the agent's visual detection assignments during the first half of the observation period. This uncertainty arises primarily because many pedestrians are clustered around the central

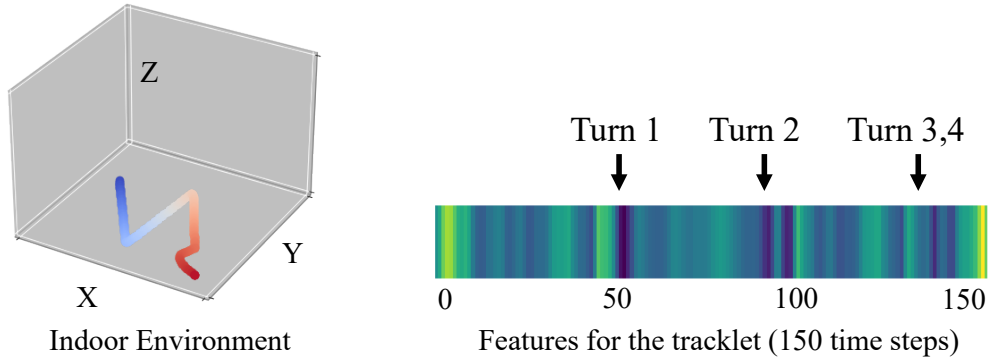


FIGURE 6.16. Visual trajectory vs. motion feature for one tracklet, time length is 150. The X and Y axes represent the coordinates of normalised objects. The tracklet is depicted through a colour gradient that transitions from blue to red. This colour transition signifies the progression of movement from the starting point (blue) to the ending point (red). Best viewed in colour.

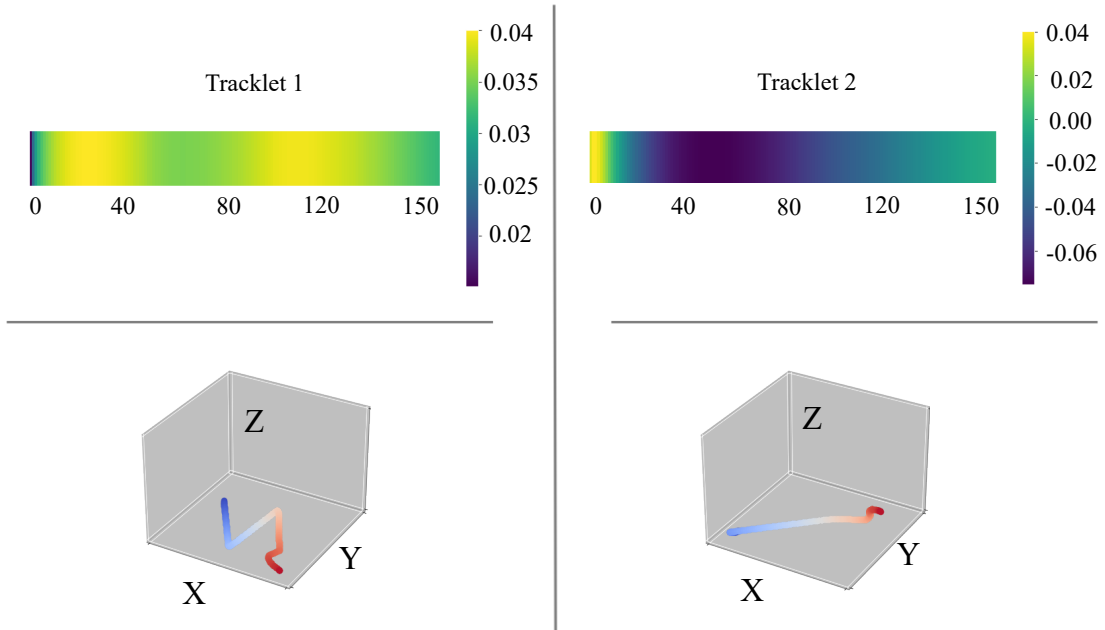


FIGURE 6.17. Visual trajectory vs Visual feature extracted by the agent. Different trajectories shows very different features.

area and have just begun to move, providing limited motion features for reference. As more multimodal information accumulates, the accuracy of the agent's identity matching improves, culminating in a nearly perfect success rate in the later stages.

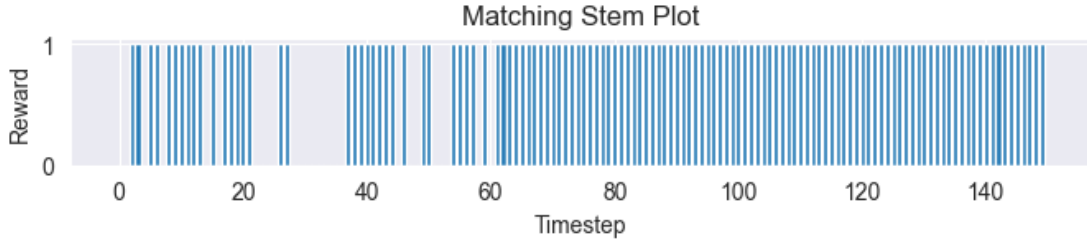


FIGURE 6.18. Matching reward example for a trajectory of 150 time steps, 1 indicates correct matching and 0 otherwise.

The case study shows that our DRL policy effectively learns crucial features from various modalities. These features, traditionally extracted manually on the basis of prior research, are integral to the fusion process.

6.6 Conclusion

Indoor environments, such as shopping malls and factory warehouses, are significantly benefitting from advances in indoor positioning technologies. In this context, our work introduces a DRL-based approach for multimodal identification of indoor pedestrians. We meticulously designed an RL environment and used the PPO algorithm to tackle the associated challenges. Our findings reveal that a well-designed RL environment and reward function are instrumental in solving the complexities of multimodal fusion. Furthermore, an ablation study confirms the effectiveness of our RL agent in capturing motion patterns from diverse input sources and autonomously aligning them. When evaluated against contemporary benchmarks, our agent shows robust efficiency in resolving identity matching problems.

CHAPTER 7

Conclusion

Indoor pedestrian positioning has been a huge challenge for the past two decades. To overcome the disadvantages of GPS for indoor scenarios, many technologies are explored. Among those technologies, we select three of the most common ones: wireless signal, IMU data, and vision tracking data. We make key contributions to the wireless fingerprint approach and both the detection and the tracking algorithms for the vision-based algorithms. We also developed a multimodal approach based on deep reinforcement learning to fuse the above chosen data. The proposed approach achieves state-of-the-art performance in highly crowded indoor settings.

We designed a beacon-based probabilistic fingerprint indoor positioning system. The data from the gyroscope and accelerometer is first preprocessed with an LSTM model to obtain the user's relative position. Then it is combined with the Gaussian kernel to improve the prior probability calculation in the Bayesian method. We use Bluetooth beacon devices as signal transmitters and the smartphone as receiver. We conduct our experiments in a real workplace environment where multiple indoor obstacles are present that can cause interference to the bluetooth signal. Experiment results show that we improved the error distance of the baseline method by 17.3%.

In recent years, the rapid development of computer vision has brought significant improvements to indoor positioning approaches. Since accurate vision tracking can provide accurate localisation results, we improved the vision tracking performance by using an overly aggressive detection and association approach and use GAT to find the proper split point for the tracklet. This method achieves state-of-the-art results on the public multiple object tracking dataset MOT17 and MOT20 compared to existing public detection approaches.

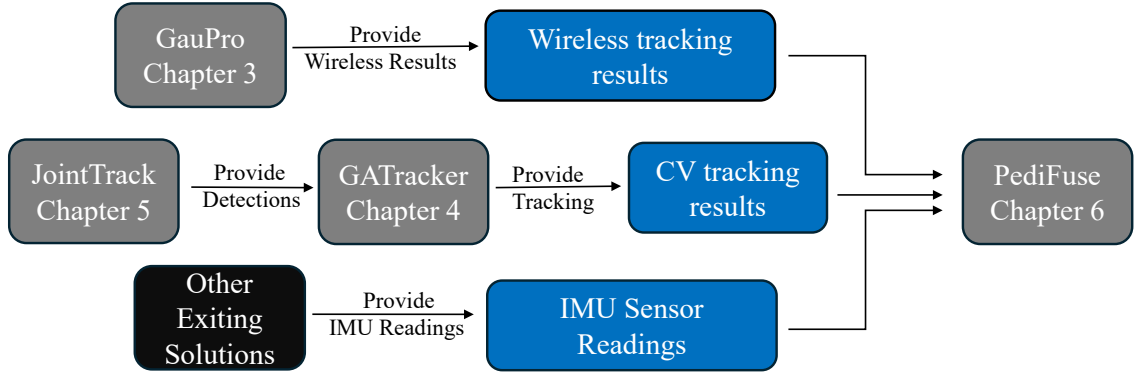


FIGURE 7.1. Intrinsic connections of the algorithms proposed in this thesis.

Another significant contribution of this paper lies in the development of an anchor-free algorithm for simultaneous head and body detection in pedestrians. Contemporary multiple object tracking datasets have increased the complexity by substantially increasing pedestrian density, which in turn exacerbates the challenges associated with heavy occlusions. To address this, we devised a pedestrian detection algorithm that builds on the YOLOX model. This novel approach is capable of detecting both the head and body of pedestrians in a unified manner. Such an enhancement is particularly beneficial for identifying small heads within densely populated scenes, thereby reducing false positives. Empirical tests confirm the efficacy of our model, which secured the top rank in the private detector category of the MOT20 challenge.

Our work synthesises vision and wireless technologies to introduce a multimodal DRL-based solution for indoor positioning. Previous research on multimodal IPS highlights the inherent difficulties in extracting and correlating pedestrian motion patterns. To address these challenges, we meticulously designed a reinforcement learning environment complete with well-defined states, actions, and reward functions. Using a customised version of the PPO algorithm, our agent is trained to automatically extract and integrate motion patterns from various data streams. Our experiments, conducted on a specialised digital twin platform known as WirelessDT, validate the effectiveness of our approach. Compared to existing studies, our DRL agent shows robust efficiency and significant advances in solving the indoor positioning problem.

In this thesis, we demonstrate that the algorithms we propose are capable of operating in parallel, yet under certain conditions they may also function collaboratively. This synergistic relationship is illustrated in Figure 7.1. Specifically, GauPro, discussed in Chapter 3, is responsible for producing results related to wireless tracking. Meanwhile, JointTrack, introduced in Chapter 5, is adept at generating accurate detection results, which are subsequently utilised by GATracker, as detailed in Chapter 4. By integrating the results derived from wireless and computer vision (CV) signals with the IMU sensor readings, as proposed by existing methodologies, PediFuse, which we propose in Chapter 6, is able to identify the optimal alignment between these signals. This, in turn, facilitates effective pedestrian localisation.

The algorithms proposed in this thesis have significant potential for application in various industries, leveraging their capabilities to improve operational efficiency, customer experience, and safety standards. For example, in the transportation sector, they can revolutionise the way traffic flows are managed and monitored. By accurately locating pedestrians and vehicles, JointTrack and GATracker contribute to the development of smarter traffic control systems that reduce congestion and enhance road safety. In public transportation systems, they could optimise passenger flow in stations, minimising bottlenecks and improving the overall efficiency of transit operations.

In the retail industry, PediFuse can be used to track customer movements within stores or shopping centres. These data can provide invaluable information on shopper behaviour, allowing retailers to optimise store layouts, product placements, and staff levels to enhance the shopping experience and increase sales. Additionally, by understanding the flow of customers in real time, retailers can implement more effective crowd management strategies, especially during peak shopping periods, ensuring a safer and more enjoyable shopping environment.

For tourism, these algorithms offer opportunities to enhance visitor experiences at attractions, park, and cities. By tracking the flow and density of tourists, operators can manage crowds more effectively. For healthcare, these algorithms can optimise patient flow and staff allocation, reducing waiting times and improving the delivery of care. For manufacturing and warehousing, by tracking the movement of goods and personnel in real-time, these algorithms can enhance operational efficiency. Smart cities, urban planning, surveillance, and even

schools and universities can benefit from these algorithms to improve user experience. Their versatility and adaptability make them a valuable tool in a wide range of applications.

7.1 Future Outlook

The indoor positioning problem is still a huge challenge in the industry. Many potential improvements can still be made. For example, research on algorithms that require minimal computational power for IoT environments where energy is a concern. We can investigate ways to use ambient energy to power the hardware. In addition, we need to explore strategies to ensure that localisation data is anonymous to safeguard user privacy. Simulation environments also have great potential, as advanced simulation that closely mimics real-world conditions is also a challenging task.

Bibliography

- [1] Y. Moustafa A, A. Ashok, S. A Udaya, and N. Sam H. “A Probabilistic Clustering-based Indoor Location Determination System.” In: *Technical Reports from UMIACS, UMIACS-TR-2002-30* (2002).
- [2] B. A., G. Z., O. L., R. F., and U. B. “Simple Online and Realtime Tracking.” In: *IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 3464–3468.
- [3] N. Abdelmoumen. “An Algebraic Solution to the Multilateration Problem.” In: *Proceedings of the 15th International Conference on Indoor Positioning and Indoor Navigation, Sydney, Australia*. Vol. 1315. 2012.
- [4] G. Agrim, J. Justin, F. Li, S. Silvio, and A. Alexandre. “Social Gan: Socially Acceptable Trajectories With Generative Adversarial Networks.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2255–2264.
- [5] G. Aleksa. *Pytorch-gat*. <https://github.com/gordicaleksa/pytorch-gat>. 2020.
- [6] S. Aleksandrs et al. “Introduction to Multi-armed Bandits.” In: *Foundations and Trends* 12.1-2 (2019), pp. 1–286.
- [7] K. Alexander, M. Eric, R. Nikhila, M. Hanzi, R. Chloe, G. Laura, X. Tete, W. Spencer, B. Alexander C, L. Wan-Yen, et al. “Segment Anything.” In: *Arxiv Preprint Arxiv:2304.02643* (2023).
- [8] A. Alexandre, G. Kratarth, R. Vignesh, R. Alexandre, F. Li, and S. Silvio. “Social Lstm: Human Trajectory Prediction in Crowded Spaces.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 961–971.
- [9] B. Alexey, W. Chien-Yao, and L. Hong-Yuan Mark. “Yolov4: Optimal Speed and Accuracy of Object Detection.” In: *Arxiv Preprint Arxiv:2004.10934* (2020).

- [10] K. Ali, G. Nikolaos, and A. David. “Modern Wlan Fingerprinting Indoor Positioning Methods and Deployment Challenges.” In: *IEEE Communications Surveys & Tutorials* 19.3 (2017), pp. 1974–2002.
- [11] M. Ali and M. Osama. “RFID Indoor Location Identification for Construction Projects.” In: *Automation in Construction* 39 (2014), pp. 167–179.
- [12] S. Amir, A. Alexandre, and S. Silvio. “Tracking the Untrackable: Learning to Track Multiple Cues With Long-term Dependencies.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 300–311.
- [13] M. Andr. “A Comprehensive Survey of Visual Slam Algorithms.” In: *Robotics* 11.1 (2022), p. 24.
- [14] H. Andrea, H. Roberto, R. Bodo, and S. Paul. “Lifted Disjoint Paths With Application in Multiple Object Tracking.” In: *International Conference on Machine Learning*. 2020, pp. 4364–4375.
- [15] M. Anton, R. S Hamid, D. Anthony, R. Ian, and S. Konrad. “Online Multi-target Tracking Using Recurrent Neural Networks.” In: *Thirty-first AAAI Conference on Artificial Intelligence*. 2017.
- [16] M. Anton and Leal-Taix. “Mot16: A Benchmark for Multi-object Tracking.” In: *Arxiv Preprint Arxiv:1603.00831* (2016).
- [17] V. Ashish, S. Noam, P. Niki, U. Jakob, J. Llion, G. Aidan N, and K. “Attention Is All You Need.” In: *Advances in Neural Information Processing Systems* 30 (2017).
- [18] PaddlePaddle Authors. *Paddledetection, Object Detection and Instance Segmentation Toolkit Based on Paddlepaddle*. <https://github.com/PaddlePaddle/PaddleDetection>. 2019.
- [19] P Bahl, Vn Padmanabhan, and A Balachandran. “Enhancements to the RADAR user location and tracking system.” In: *Microsoft Research* (2000).
- [20] Paramvir Bahl and Venkata N. Padmanabhan. “RADAR: An in-building RF-based user location and tracking system.” In: *Proceedings - IEEE INFOCOM*. 2000.
- [21] Anja Bekkelien, Michel Deriaz, and Stéphane Marchand-Maillet. “Bluetooth indoor positioning.” In: *Master’s thesis, University of Geneva* (2012).

- [22] W. Bo and N. Ram. “Tracking of Multiple, Partially Occluded Humans Based on Static Body Part Detection.” In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 1. 2006, pp. 951–958.
- [23] Y. Bo and N. Ram. “Multi-target Tracking By Online Learning of Non-linear Motion Patterns and Robust Appearance Models.” In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1918–1925.
- [24] Z. Boying, T. Jin, Z. Junda, L. Xinfeng, X. Dong, and Z. Yuan F. “Ev-loc: Integrating Electronic and Visual Signals for Accurate Localization.” In: *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2012, pp. 25–34.
- [25] Bras. “Learning A Neural Solver for Multiple Object Tracking.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 6247–6257.
- [26] K. Bumgon, B. Wonsun, and K. Yong C. “Indoor Localization for Wi-fi Devices By Cross-monitoring Ap and Weighted Triangulation.” In: *2011 IEEE Consumer Communications and Networking Conference (CCNC)*. 2011, pp. 933–936.
- [27] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. “Observation-centric sort: Rethinking sort for robust multi-object tracking.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 9686–9696.
- [28] Jose Luis V. Carrera, Zhongliang Zhao, Torsten Braun, Haiyong Luo, and Fang Zhao. “Discriminative Learning-based Smartphone Indoor Localization.” In: *CoRR* abs/1804.03961 (2018). arXiv: 1804.03961. URL: <http://arxiv.org/abs/1804.03961>.
- [29] K. Chanh, L. Fuxin, C. Arridhana, and R. James M. “Multiple Hypothesis Tracking Revisited.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4696–4704.
- [30] L. Chao, Z. Zhipeng, Z. Xue, L. Bing, Z. Shuyuan, and H. Weiming. “Rethinking the Competition Between Detection and Reid in Multiobject Tracking.” In: *IEEE Transactions on Image Processing* 31 (2022), pp. 3182–3196.

- [31] X. Chao, Y. Daiqin, C. Zhenzhong, and T. Guang. “3-d Ble Indoor Localization Based on Denoising Autoencoder.” In: *IEEE Access* 5 (2017), pp. 12751–12760.
- [32] C. Charles and O. Bruno A. “Learning Transformational Invariants from Natural Movies.” In: *Advances in Neural Information Processing Systems*. 2009, pp. 209–216.
- [33] H. Chaur-Heh, C. Jen-Yang, and N. Bo-Hong. “Deep Learning-based Indoor Localization Using Received Signal Strength and Channel State Information.” In: *IEEE Access* 7 (2019), pp. 33256–33267.
- [34] C. Cheng, Z. Shifeng, X. Junliang, L. Zhen, L. Stan Z, and Z. Xudong. “Pedhunter: Occlusion Robust Pedestrian Detector in Crowded Scenes.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 10639–10646.
- [35] C. Cheng, Z. Shifeng, X. Junliang, L. Zhen, L. Stan Z, and Z. Xudong. “Relational Learning for Joint Head and Human Detection.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 10647–10654.
- [36] W. Chien-Yao, B. Alexey, and L. Hong-Yuan Mark. “Yolov7: Trainable Bag-of-freebies Sets New State-of-the-art for Real-time Object Detectors.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7464–7475.
- [37] Z. Chunluan and Y. Junsong. “Bi-box Regression for Pedestrian Detection and Occlusion Estimation.” In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 135–151.
- [38] Z. Chunluan and Y. Junsong. “Learning to Integrate Occlusion-specific Detectors for Heavily Occluded Pedestrian Detection.” In: *Asian Conference on Computer Vision*. 2016, pp. 305–320.
- [39] Saturn Cloud. *A Comprehensive Guide to Convolutional Neural Networks - the Eli5 Way*. <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>. Accessed: 2018-12-25. 2018.
- [40] M. Cong, Y. Changshui, Y. Fan, Z. Yueqing, Z. Ziwei, J. Huizhu, and X. Xiaodong. “Trajectory Factory: Tracklet Cleaving and Re-connection By Deep Siamese Bi-gru

- for Multiple Object Tracking.” In: *2018 IEEE International Conference on Multimedia and Expo (ICME)*. 2018, pp. 1–6.
- [41] Z. Cong, L. Hongsheng, W. Xiaogang, and Y. Xiaokang. “Cross-scene Crowd Counting Via Deep Convolutional Neural Networks.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 833–841.
- [42] John J Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Educacion, 2005, pp. 19–62.
- [43] S. Daniel and B. J. “Modelling Ambiguous Assignments for Multi-person Tracking in Crowds.” In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 133–142.
- [44] S. Daniel and B. Jurgen. “Improving Multiple Pedestrian Tracking By Track Management and Occlusion Handling.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 10958–10967.
- [45] Doll. “Behavior Recognition Via Sparse Spatio-temporal Features.” In: *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. 2005, pp. 65–72.
- [46] V. Dominik, S. Georg, H. Robert, H. Sebastian, G. Adrian, and S. Eckehard. “Camera-based Indoor Positioning Using Scalable Streaming of Compressed Binary Image Signatures.” In: *2014 IEEE International Conference on Image Processing (ICIP)*. 2014, pp. 2804–2808.
- [47] N. Fred E. “Directional Reflectance and Emissivity of An Opaque Surface.” In: *Applied Optics* 4.7 (1965), pp. 767–775.
- [48] R. David E, H. Geoffrey E, W. Ronald J, et al. *Learning Internal Representations By Error Propagation*. 1985.
- [49] Y. En, L. Zhuoling, and H. Shoudong. “Towards Discriminative Representation: Multi-view Trajectory Contrastive Learning for Online Multi-object Tracking.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8834–8843.

- [50] R. Ergys, S. Francesco, Z. Roger, C. Rita, and T. Carlo. "Performance Measures and A Data Set for Multi-target, Multi-camera Tracking." In: *European Conference on Computer Vision*. 2016, pp. 17–35.
- [51] V. Eric and G. Leonidas J. "Optimally Combining Sampling Techniques for Monte Carlo Rendering." In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*. 1995, pp. 419–428.
- [52] S. Erick, I. Devasena, M. Rahul, and A. David. "Sdr-fi: Deep-learning-based Indoor Positioning Via Software-defined Radio." In: *IEEE Access* 7 (2019), pp. 145784–145797.
- [53] B. Erik, E. Volker, and S. Thomas. "High-speed Tracking-by-detection Without Using Image Information." In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. 2017, pp. 1–6.
- [54] F.Ramsey and H.Robert. "Location Fingerprinting with Bluetooth Low Energy Beacons." In: *IEEE Journal on Selected Areas in Communications* (2015). ISSN: 07338716.
- [55] F.Silke, K.Kyandoghere, Z.Ana, and L.Zighuo. "An indoor Bluetooth-based positioning system: Concept, implementation and experimental evaluation." In: *Proceedings of the International Conference on Wireless Networks*. 2003. ISBN: 1932415041.
- [56] Z. Faheem, G. Athanasios, and L. Kin K. "A Survey of Indoor Localization Systems and Technologies." In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2568–2599.
- [57] Z. Faheem, G. Athanasios, and L. Kin K. "A Survey of Indoor Localization Systems and Technologies." In: *IEEE Communications Surveys & Tutorials* 21.3 (2019), pp. 2568–2599.
- [58] Y. Fan, C. Wongun, and L. Yuanqing. "Exploit All the Layers: Fast and Accurate Cnn Object Detector With Scale Dependent Pooling and Cascaded Rejection Classifiers." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2129–2137.
- [59] Y. Fan, C. Xin, S. Sakriani, W. Yang, and N. Satoshi. "Remot: A Model-agnostic Refinement for Multiple Object Tracking." In: *Image and Vision Computing* 106 (2021), p. 104091.

- [60] L. David G. “Object Recognition from Local Scale-invariant Features.” In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, pp. 1150–1157.
- [61] S. Gaoyang, P. Byeong-hun, N. Song-hyeon, K. Bosung, R. Byeong-hee, and K. Young-Bae. “A 3-dimensional Triangulation Scheme to Improve the Accuracy of Indoor Localization for Iot Services.” In: *2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. 2015, pp. 359–363.
- [62] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. “YOLOX: Exceeding YOLO Series in 2021.” In: *arXiv preprint arXiv:2107.08430* (2021).
- [63] Sara A. van de Geer. “Least Squares Estimation.” In: *Encyclopedia of Statistics in Behavioral Science*. 2005.
- [64] J. Guang-yao, L. Xiao-yi, and P. Myong-Soon. “An Indoor Localization Mechanism Using Active RFID Tag.” In: *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC’06)*. Vol. 1. 2006, 4–pp.
- [65] W. Guanshuo, Y. Yufeng, C. Xiong, L. Jiwei, and Z. Xi. “Learning Discriminative Features With Multiple Granularities for Person Re-identification.” In: *Proceedings of the 26th ACM International Conference on Multimedia*. 2018, pp. 274–282.
- [66] X. Han, Y. Zheng, Z. Zimu, S. Longfei, Y. Ke, and L. Yunhao. “Enhancing Wifi-based Localization With Visual Clues.” In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015, pp. 963–974.
- [67] C. Hao, Z. Yifan, L. Wei, T. Xiaofeng, and Z. Ping. “Confi: Convolutional Neural Networks Based Indoor Wi-fi Localization Using Channel State Information.” In: *IEEE Access* 5 (2017), pp. 18066–18074.
- [68] L. Hao, G. Youzhi, L. Xingyu, L. Shenqi, and J. Wei. “Bag of Tricks and A Strong Baseline for Deep Person Re-identification.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.
- [69] S. Hao, Z. Yang, C. Jiahui, X. Zhang, and Z. Jun. “Heterogeneous Association Graph Fusion for Target Association in Multiple Object Tracking.” In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.11 (2018), pp. 3269–3280.

- [70] H. Hendrik, N. Abdelmoumen, and B. J. “An Imu/magnetometer-based Indoor Positioning System Using Kalman Filtering.” In: *International Conference on Indoor Positioning and Indoor Navigation*. 2013, pp. 1–9.
- [71] B. Herbert, T. Tinne, and V. Luc. “Surf: Speeded Up Robust Features.” In: *Computer Vision—ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I* 9. 2006, pp. 404–417.
- [72] B. HL. “Radio Propagation for Modern Wireless Systems” Prentice-hall.” In: *Englewood Cliffs, Nj* (2000).
- [73] C. Hongkai, M. Sirajum, and L. Shan. “Rfcam: Uncertainty-aware Fusion of Camera and Wi-fi for Real-time Human Identification With Mobile Devices.” In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6.2 (2022), pp. 1–29.
- [74] D. Hugh and B. Tim. “Simultaneous Localization and Mapping: Part I.” In: *IEEE Robotics & Automation Magazine* 13.2 (2006), pp. 99–110.
- [75] L. Ilya and H. Frank. “SGDR: Stochastic Gradient Descent With Warm Restarts.” In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.
- [76] W. Ronald J. “Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning.” In: *Reinforcement Learning* (1992), pp. 5–32.
- [77] H. Jeffrey and B. Gaetano. “Location Systems for Ubiquitous Computing.” In: *Computer* 34.8 (2001), pp. 57–66.
- [78] B. Jerome, F. Francois, T. Engin, and F. Pascal. “Multiple Object Tracking Using K-shortest Paths Optimization.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.9 (2011), pp. 1806–1819.
- [79] G. Jiale, L. Qiang, and C. Enqing. “A Deep Reinforcement Learning Method for Multimodal Data Fusion in Action Recognition.” In: *IEEE Signal Processing Letters* 29 (2021), pp. 120–124.
- [80] D. Jiang, X. Yu, N. Marius, O. Zhonghong, and Yl. “Imoon: Using Smartphones for Image-based Indoor Navigation.” In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. 2015, pp. 85–97.

- [81] D. Jiang, X. Yu, O. Zhonghong, C. Yong, and Y. Antti. “Indoor Tracking Using Crowdsourced Maps.” In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2016, pp. 1–6.
- [82] X. Jiarui, C. Yue, Z. Zheng, and H. Han. “Spatial-temporal Relation Networks for Multi-object Tracking.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3988–3998.
- [83] N. Jiazhi, Z. Fusang, X. Jie, H. Qiang, C. Zhaoxin, M. Junqi, X. BinBin, W. Pengsen, B. Guangyu, L. Xin, et al. “Experience: Pushing Indoor Localization from Laboratory to the Wild.” In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking (MOBICOM)*. 2022, pp. 147–157.
- [84] X. Jingao, C. Hengjie, Q. Kun, D. Erqun, S. Min, W. Chenshu, Z. Li, and Y. Zheng. “Ivr: Integrated Vision and Radio Localization With Zero Human Effort.” In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.3 (2019), pp. 1–22.
- [85] P. Jinlong, W. Tao, L. Weiyao, W. Jian, S. John, W. Shilei, and D. Erui. “Tpm: Multiple Object Tracking With Tracklet-plane Matching.” In: *Pattern Recognition* 107 (2020), p. 107480.
- [86] Glenn Jocher. <https://github.com/ultralytics/yolov5>. Version v3.1. Oct. 2020. URL: <https://doi.org/10.5281/zenodo.4154370>.
- [87] S. John, W. Filip, D. Prafulla, R. Alec, and K. Oleg. “Proximal Policy Optimization Algorithms.” In: *Arxiv Preprint Arxiv:1707.06347* (2017).
- [88] S. John, L. Sergey, A. Pieter, J. Michael, and M. Philipp. “Trust Region Policy Optimization.” In: *International Conference on Machine Learning*. 2015, pp. 1889–1897.
- [89] L. Jonathon, O. Aljosa, D. Patrick, T. Philip, G. Andreas, and Leal-Taix. “Hota: A Higher Order Metric for Evaluating Multi-object Tracking.” In: *International Journal of Computer Vision* 129.2 (2021), pp. 548–578.
- [90] R. Joseph, D. Santosh, G. Ross, and F. Ali. “You Only Look Once: Unified, Real-time Object Detection.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 779–788.

- [91] K.Manikanta, P.Zhang, and K.Sachin. "Localizing low-power backscatter tags using commodity WiFi." In: *CoNEXT 2017 - Proceedings of the 2017 13th International Conference on emerging Networking EXperiments and Technologies*. 2017. ISBN: 9781450354226.
- [92] H. Kaiming, G. Georgia, and Doll. "Mask R-cnn." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969.
- [93] B. Keni and S. Rainer. "Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics." In: *Eurasip Journal on Image and Video Processing* 2008 (2008), pp. 1–10.
- [94] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. "SpotFi: Decimeter level localization using WiFi." In: *SIGCOMM 2015 - Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 2015. ISBN: 9781450335423.
- [95] M. Krystian, S. Cordelia, and Z. Andrew. "Human Detection Based on A Probabilistic Assembly of Robust Part Detectors." In: *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I* 8. 2004, pp. 69–82.
- [96] F. Kunihiro. "Neocognitron: A Self-organizing Neural Network Model for A Mechanism of Pattern Recognition Unaffected By Shift in Position." In: *Biological Cybernetics* 36.4 (1980), pp. 193–202.
- [97] H. Kurt, S. Maxwell, and W. Halbert. "Multilayer Feedforward Networks Are Universal Approximators." In: *Neural Networks* 2.5 (1989), pp. 359–366.
- [98] L.Konrad and W.Matt. "MoteTrack: A robust, decentralized approach to RF-based location tracking." In: *Lecture Notes in Computer Science*. 2005.
- [99] D. Li. "The Mnist Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]." In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.
- [100] R. Liangliang, L. Jiwen, W. Zifeng, T. Qi, and Z. Jie. "Collaborative Deep Reinforcement Learning for Multi-object Tracking." In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 586–602.

- [101] Z. Linyu, T. Ming, C. Yingying, Z. Guibo, W. Jinqiao, and L. Hanqing. “Improving Multiple Object Tracking With Single Object Tracking.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2453–2462.
- [102] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. “GSM: Graph Similarity Model for Multi-Object Tracking.” In: *IJCAI*. 2020, pp. 530–536.
- [103] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge. “Summary of ChatGPT-Related research and perspective towards the future of large language models.” In: *Meta-Radiology* 1.2 (2023), p. 100017. ISSN: 2950-1628. URL: <https://www.sciencedirect.com/science/article/pii/S2950162823000176>.
- [104] C. Long, A. Haizhou, C. Rui, and Z. Zijie. “Aggregate Tracklet Appearance Features for Multi-object Tracking.” In: *IEEE Signal Processing Letters* 26.11 (2019), pp. 1613–1617.
- [105] W. Longyin, D. Dawei, L. Shengkun, B. Xiao, and L. Siwei. “Learning Non-uniform Hypergraph for Multi-object Tracking.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8981–8988.
- [106] K. Sham M. “A Natural Policy Gradient.” In: *Advances in Neural Information Processing Systems* 14 (2001).
- [107] Segura. M, Mut. V, and Sisterna. C. “Ultra Wideband Indoor Navigation System.” In: *Iet Radar, Sonar & Navigation* 6.5 (2012), pp. 402–411.
- [108] Y. Mang, S. Jianbing, L. Gaojie, X. Tao, S. Ling, and H. Steven CH. “Deep Learning for Person Re-identification: A Survey and Outlook.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 44.6 (2021), pp. 2872–2893.
- [109] A. Manoj, W. Yang, and M. Vashisht. “Decore: Deep Compression With Reinforcement Learning.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12349–12359.
- [110] K. Margret, T. Siyu, A. Bjoern, B. Thomas, and S. Bernt. “Motion Segmentation & Multiple Object Tracking By Correlation Co-clustering.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.1 (2018), pp. 140–153.

- [111] D. Micha. “Convolutional Neural Networks on Graphs With Fast Localized Spectral Filtering.” In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 3844–3852.
- [112] C. Minghai, W. Sen, L. Paul Pu, and Baltru. “Multimodal Sentiment Analysis With Word-level Fusion and Reinforcement Learning.” In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 2017, pp. 163–171.
- [113] L. Mingkuan, L. Ning, N. Qun, L. Chang, C. S-H Gary, and G. Chengying. “Sweeploc: Automatic Video-based Indoor Localization By Camera Sweeping.” In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.3 (2018), pp. 1–25.
- [114] K. Thomas N and W. Max. “Semi-supervised Classification With Graph Convolutional Networks.” In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.
- [115] D. Navneet and T. Bill. “Histograms of Oriented Gradients for Human Detection.” In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, pp. 886–893.
- [116] W. Nicolai, B. Alex, and P. Dietrich. “Simple Online and Realtime Tracking With A Deep Association Metric.” In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 3645–3649.
- [117] C. Nicolas, M. Francisco, S. Gabriel, U. Nicolas, K. Alexander, and Z. Sergey. “End-to-end Object Detection With Transformers.” In: *European Conference on Computer Vision*. 2020, pp. 213–229.
- [118] P.Zhang, B.Dinesh, J.Kiran, and K.Sachin. “HitchHike: Practical backscatter using commodity WiFi.” In: *Proceedings of the 14th ACM Conference on Embedded Networked Sensor Systems, SenSys 2016*. 2016. ISBN: 9781450342636.
- [119] D. Patrick, R. Hamid, M. Anton, S. Javen, C. Daniel, R. Ian, R. Stefan, S. Konrad, and Leal-Taix. “Mot20: A Benchmark for Multi Object Tracking in Crowded Scenes.” In: *Arxiv Preprint Arxiv:2003.09003* (2020).
- [120] D. Peng, W. Renliang, C. Wongun, Z. Changshui, H. Zhangping, and D. Wei. “Learning A Proposal Classifier for Multiple Object Tracking.” In: *Proceedings of the*

- IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 2443–2452.
- [121] B. Philipp, M. Tim, and L. Laura. “Tracking Without Bells and Whistles.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 941–951.
 - [122] M. Larry R and J. LC. “Recurrent Neural Networks.” In: *Design and Applications 5* (2001), pp. 64–67.
 - [123] S. Ramana, D. Cedric, M. Eric, and P. Julien. “Tracking Pedestrian Heads in Dense Crowd.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3865–3875.
 - [124] B. Richard. “A Markovian Decision Process.” In: *Journal of Mathematics and Mechanics* (1957), pp. 679–684.
 - [125] B. Richard. “Dynamic Programming.” In: *Science* 153.3731 (1966), pp. 34–37.
 - [126] H. Roberto and Leal-Taix. “Fusion of Head and Full-body Detectors for Multi-object Tracking.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 1428–1437.
 - [127] W. Roy, H. Andy, F. Veronica, and G. Jonathan. “The Active Badge Location System.” In: *ACM Transactions on Information Systems (tois)* 10.1 (1992), pp. 91–102.
 - [128] L. Ruiqi, M. Huimin, and W. Yu. “Semantic Head Enhanced Pedestrian Detection in A Crowd.” In: *Neurocomputing* 400 (2020), pp. 343–351.
 - [129] S. Richard S, M. David, S. Satinder, and M. Yishay. “Policy Gradient Methods for Reinforcement Learning With Function Approximation.” In: *Advances in Neural Information Processing Systems* 12 (1999).
 - [130] C.Sudarshan S. “Beacon placement for indoor localization using Bluetooth.” In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. 2008.
 - [131] S.Nattapong and K.Prashant. “Location fingerprint analyses toward efficient indoor positioning.” In: *6th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2008*. 2008. ISBN: 076953113X.

- [132] S. Samuel, V. Paul, C. Wongun, and C. Manmohan. “Deep Network Flow for Multi-object Tracking.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6951–6960.
- [133] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. “High-Dimensional Continuous Control Using Generalized Advantage Estimation.” In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2016.
- [134] H. Sepp and S. J. “Long Short-term Memory.” In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [135] R. Shaoqing, H. Kaiming, G. Ross, and S. Jian. “Faster R-cnn: Towards Real-time Object Detection With Region Proposal Networks.” In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 91–99.
- [136] Z. Shifeng, C. Cheng, Y. Yongqiang, L. Zhen, and L. Stan Z. “Bridging the Gap Between Anchor-based and Anchor-free Detection Via Adaptive Training Sample Selection.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9759–9768.
- [137] S. ShiJie, A. Naveed, S. HuanSheng, M. Ajmal, and S. Mubarak. “Deep Affinity Network for Multiple Object Tracking.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.1 (2019), pp. 104–119.
- [138] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, et al. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play.” In: *Science* 362.6419 (2018), pp. 1140–1144.
- [139] S. Simone, G. Marco, F. Luca, and R. Alessandro. “A Double-stage Kalman Filter for Orientation Tracking With An Integrated Processor in 9-d Imu.” In: *IEEE Transactions on Instrumentation and Measurement* 62.3 (2012), pp. 590–598.
- [140] T. Siyu, A. Mykhaylo, A. Bjoern, and S. Bernt. “Multiple People Tracking By Lifted Multicut and Person Re-identification.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3539–3548.

- [141] C. Siyuan and W. He. “Enabling Public Cameras to Talk to the Public.” In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.2 (2018), pp. 1–20.
- [142] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. “TransTrack: Multiple-Object Tracking with Transformer.” In: *arXiv preprint arXiv: 2012.15460* (2020).
- [143] N. Le T, K. Yu Seung, T. Patrick, and Z. Joy. “Identitylink: User-device Linking Through Visual and Rf-signal Cues.” In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2014, pp. 529–539.
- [144] Pavel Tokmakov, Jie Li, Wolfram Burgard, and Adrien Gaidon. “Learning to track with object permanence.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10860–10869.
- [145] L. Tsung-Yi and Doll. “Feature Pyramid Networks for Object Detection.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2117–2125.
- [146] V.Deepak, K.Swarun, and K.Dina. “Decimeter-level localization with a single WiFi access point.” In: *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016*. 2016. ISBN: 9781931971294.
- [147] Vanderbrug and Rosenfeld. “Two-stage Template Matching.” In: *IEEE Transactions on Computers* 100.4 (1977), pp. 384–393.
- [148] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. “Graph attention networks.” In: *Stat* 1050.20 (2017), pp. 10–48550.
- [149] K. Vijay and T. John. “Actor-critic Algorithms.” In: *Advances in Neural Information Processing Systems* 12 (1999).
- [150] M. Volodymyr, K. Koray, S. David, G. Alex, A. Ioannis, W. Daan, and R. Martin. “Playing Atari With Deep Reinforcement Learning.” In: *Proceedings of the Neural Information Processing Systems (NIPS) Deep Learning Workshop*. 2013.
- [151] K. Harold W. “The Hungarian Method for the Assignment Problem.” In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97.

- [152] S. John W. “Disjoint Paths in A Network.” In: *Networks* 4.2 (1974), pp. 125–145.
- [153] Yapeng Wang, Xu Yang, Yutian Zhao, Yue Liu, and Laurie Cuthbert. “Bluetooth positioning using RSSI and triangulation methods.” In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. IEEE. 2013, pp. 837–842.
- [154] L. Wei, A. Dragomir, E. Dumitru, S. Christian, R. Scott, F. Cheng-Yang, and B. Alexander C. “Ssd: Single Shot Multibox Detector.” In: *European Conference on Computer Vision*. 2016, pp. 21–37.
- [155] Z. Weixun, N. Shawn, L. Congmin, and S. Zhenfeng. “Patternnet: A Benchmark Dataset for Performance Evaluation of Remote Sensing Image Retrieval.” In: *Isprs Journal of Photogrammetry and Remote Sensing* 145 (2018), pp. 197–209.
- [156] L. Wenhan, X. Junliang, M. Anton, Z. Xiaoqin, L. Wei, and K. Tae-Kyun. “Multiple Object Tracking: A Literature Review.” In: *Artificial Intelligence* 293 (2021), p. 103448.
- [157] L. Xiaochen, J. Yurong, J. Puneet, and K. Kyu-Han. “Tar: Enabling Fine-grained Targeted Advertising in Retail Stores.” In: *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 2018, pp. 323–336.
- [158] Z. Xingyi, W. Dequan, and Kr. “Objects as Points.” In: *Arxiv Preprint Arxiv:1904.07850* (2019).
- [159] Z. Xingyi, K. Vladlen, and Kr. “Tracking Objects as Points.” In: *European Conference on Computer Vision*. 2020, pp. 474–490.
- [160] Yihong Xu, Yutong Ban, Guillaume Delorme, Chuang Gan, Daniela Rus, and Xavier Alameda-Pineda. “TransCenter: Transformers with Dense Queries for Multiple-Object Tracking.” In: *CoRR* abs/2103.15145 (2021). arXiv: 2103.15145. URL: <https://arxiv.org/abs/2103.15145>.
- [161] C. Xuangeng, Z. Anlin, Z. Xiangyu, and S. Jian. “Detection in Crowded Scenes: One Proposal, Multiple Predictions.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12214–12223.
- [162] W. Xuyu, G. Lingjun, and M. Shiwen. “Phasefi: Phase Fingerprinting for Indoor Localization With A Deep Learning Approach.” In: *2015 IEEE Global Communications Conference (globecom)*. 2015, pp. 1–6.

- [163] W. Xuyu, G. Lingjun, M. Shiwen, and P. Santosh. “Csi-based Fingerprinting for Indoor Localization: A Deep Learning Approach.” In: *IEEE Transactions on Vehicular Technology* 66.1 (2016), pp. 763–776.
- [164] H. Yan, S. Hao, Z. Yanwei, and X. Zhang. “Deepdiff: Learning Deep Difference Features on Human Body Parts for Person Re-identification.” In: *Neurocomputing* 241 (2017), pp. 191–203.
- [165] L. Yan, Z. Chongyang, Z. Muming, Z. Hao, and S. Jun. “Where, What, Whether: Multi-modal Learning Meets Pedestrian Detection.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 14065–14073.
- [166] L. Yann and B. L. “Gradient-based Learning Applied to Document Recognition.” In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [167] W. Yapeng, Y. Xu, Z. Yutian, L. Yue, and C. Laurie. “Bluetooth Positioning Using Rssi and Triangulation Methods.” In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. 2013, pp. 837–842.
- [168] Z. Yifu, W. Chunyu, W. Xinggang, Z. Wenjun, and L. Wenyu. “Fairmot: On the Fairness of Detection and Re-identification in Multiple Object Tracking.” In: *International Journal of Computer Vision* 129 (2021), pp. 3069–3087.
- [169] Z. Yifu, S. Peize, J. Yi, Y. Dongdong, W. Fucheng, Y. Zehuan, L. Ping, L. Wenyu, and W. Xinggang. “Bytetrack: Multi-object Tracking By Associating Every Detection Box.” In: *European Conference on Computer Vision*. 2022, pp. 1–21.
- [170] Z. Yifu, S. Peize, J. Yi, Y. Dongdong, W. Fucheng, Y. Zehuan, L. Ping, L. Wenyu, and W. Xinggang. “Bytetrack: Multi-object tracking by associating every detection box.” In: *European Conference on Computer Vision*. Springer. 2022, pp. 1–21.
- [171] Z. Yingying, Z. Desen, C. Siqin, G. Shenghua, and M. Yi. “Single-image Crowd Counting Via Multi-column Convolutional Neural Network.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 589–597.
- [172] T. Yonglong, L. Ping, W. Xiaogang, and T. Xiaou. “Deep Learning Strong Parts for Pedestrian Detection.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1904–1912.

- [173] X. Yu, A. Alexandre, and S. Silvio. “Learning to Track: Online Multi-object Tracking By Decision Making.” In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4705–4713.
- [174] L. Ze, H. Han, L. Yutong, Y. Zhuliang, X. Zhenda, W. Yixuan, N. Jia, C. Yue, Z. Zheng, D. Li, et al. “Swin Transformer V2: Scaling Up Capacity and Resolution.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 12009–12019.
- [175] L. Ze, L. Yutong, C. Yue, H. Han, W. Yixuan, Z. Zheng, L. Stephen, and G. Baining. “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.
- [176] G. Zheng, L. Songtao, L. Zeming, Y. Osamu, and S. Jian. “Ota: Optimal Transport Assignment for Object Detection.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 303–312.
- [177] Y. Zheng, W. Chenshu, and L. Yunhao. “Locating in Fingerprint Space: Wireless Indoor Localization With Little Human Intervention.” In: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. 2012, pp. 269–280.
- [178] T. Zhi, S. Chunhua, C. Hao, and H. Tong. “Fcos: Fully Convolutional One-stage Object Detection.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9627–9636.
- [179] L. Zhongzheng, Y. Dong, C. Huaming, Z. Yu, and B. Wei. “Wirelessdt: A Digital Twin Platform for Real-time Evaluation of Wireless Software Applications.” In: *2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-companion)*. 2023, pp. 146–150.