IMPERIAL

Imperial College London
Department of Computing

Real-Time Critical Moving Object Identification and Dynamic Path Planning for Assisting Visually Impaired People

Hadeel Ridha Surougi

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London, April 2024

Statement of Originality

I hereby declare that this thesis entitled '*Real-Time Critical Moving Object Identification and Dynamic Path Planning for Assisting Visually Impaired People*' is my own original work, except where acknowledged, and that all sources used or referred to have been acknowledged. I also confirm that the content of this thesis has not been previously submitted for a degree or qualification at this or any other university.

Hadeel Surougi

April 2024

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Autonomous objects like cars, motorcycles, bicycles, and pedestrians pose risks to visually impaired people (VIPs). Existing camera-based assistive solutions fail to identify fast-moving threats, namely Critical Moving Objects (CMOs), and cannot plan optimal collision-free local paths, essential for guiding VIPs around CMOs.

This thesis presents AR-AwareAvoid, a novel AI-based assistance scheme to assist VIPs in safely navigating outdoor environments, including different classes of CMOs moving at varying speeds. AR-AwareAvoid comprises two components: ARAware, which effectively identifies CMOs and gives the VIP more time to avoid danger through simultaneously addressing real-time CMO identification, CMO risk level classification, and a risk-aware prioritised CMO warning notification, and MinD, which plans the shortest path required to avoid CMOs while considering the VIP's mobility constraints, CMO types and movement patterns, and predicted collision times. MinD also conducts a look-ahead safety prediction of the optimal path to ensure more safety for VIPs.

Real-world experiments demonstrate that ARAware accurately identifies CMOs (97.26% mAR and 88.20% mAP), precisely classifies CMOs according to their risk levels (100% mAR and 91.69% mAP), timely warns about high-risk CMOs while effectively reducing false alarms by postponing low-risk CMOs' notifications, and operates in real-time (only takes 31ms processing time on an NVIDIA RTX 2080 GPU. Compared to the DEEP-SEE, ARAware achieves significantly higher CMO identification accuracy (by 42.62% in mAR and 10.88% in mAP), with 93% faster end-to-end processing speed. Additionally, simulation results demonstrate that MinD outperforms the Artificial Potential Field (APF) approach in effectively planning a short collision-free route, requiring only 2.75m of movement on average, shorter than APF by 84.47%, with a 0% collision rate, adapting to the VIP's mobility limitations, and provides a high safe time separation ($\geq 4.80s$ on average compared to APF). MinD also shows real-time performance, with decisions taking only $1.20ms$ processing time on the same GPU.

Acknowledgements

I am deeply grateful to my supervisor, Professor Julie A. McCann, for her invaluable guidance, unwavering support, and insightful feedback throughout my doctoral research journey. Her expertise, encouragement, and mentorship have been instrumental in shaping the direction and quality of this thesis.

I am immensely thankful to my academic colleagues, Dr Dokhyl Alqhtani and Dr Cong Zhao, for their extended discussions, valuable suggestions, and encouragement at various stages of my research, which have contributed significantly to improving this thesis.

I am grateful to Imperial College London for providing the necessary resources, facilities, and opportunities to facilitate the completion of this research endeavour. The supportive environment fostered by the institution has been instrumental in my academic growth and success.

I owe a debt of gratitude to my dearest and most fantastic husband, Ismail Mohdar and my lovely family, especially my incredible mother, for their unwavering love, understanding, and encouragement throughout this challenging yet rewarding process. Their patience, support, and belief in me have strengthened and motivated me.

Warm appreciation and heartfelt thanks go to my dear friends, Abeer, Heba, Jihan, and Khlat, as well as their amazing sons, Ahmed, Abdullah and Mohammed, and beautiful daughters, Fatima, Remas, Amira, Renad, and Mona, for their presence in my life and throughout my studies in London. Studying abroad can be a daunting experience, but their unwavering support, camaraderie, and encouragement made me feel like a home away from home. Their friendship and companionship have not only made the journey enjoyable but also filled it with cherished memories. Special respect also goes to my friends' husbands, Abbas and Samer, who have stood by my husband's side like brothers.

Lastly, I express my heartfelt appreciation to all individuals who have directly or indirectly contributed to this thesis, whether through their assistance, encouragement, or inspiration. Your contributions have played an integral role in shaping this work, and I am genuinely thankful for that.

Dedication

This thesis is dedicated to my beloved family, whose unwavering love, support, and encouragement have been the driving force behind my academic journey.

To my parents, Hayat Sharaf and Ridha Surougi, your sacrifices and guidance have shaped me into the person I am today.

To my dearest husband, Ismail Mohdar, your unwavering support, love, and understanding have been my rock throughout this journey. Your presence by my side has made the challenges seem surmountable and the victories all the sweeter.

To my wonderful brothers, Adnan, Emad, and Maher, and my lovely sisters, Ghyda and Manar, your unwavering belief in me has been a constant source of strength.

To my dear friends, Abeer, Heba, Jihan, and Khlat, along with their devoted husbands, Abbas and Samer, and amazing sons, Ahmed, Abdullah and Mohammed, and beautiful daughters, Fatima, Remas, Amira, Renad, and Mona, whose presence and support have made studying abroad feel like home. Your camaraderie and encouragement have filled my days with warmth and motivation.

Finally, to all those who have supported and inspired me along this journey, whether near or far, this thesis is a testament to your belief in me.

Thank you for being my pillars of strength.

Contents

Abstract **iii**

Acknowledgements **iv**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 3

1.3 Research Aims 7

1.4 Contributions 7

1.4.1 ARAware: CMOs Identification 7

1.4.2 MinD: Local Path Planning 9

1.4.3 Performance Comparisons: YOLOv8 vs YOLOv3 10

1.4.4 Performance Evaluation Under Different Settings 11

1.5 Hardware, Software, and Dataset Setup 11

1.5.1 ARAware Scheme 11

1.5.2 MinD Scheme 13

1.6 Hardware and Software Requirements 14

1.6.1 Hardware Requirements: . . . 14
1.6.2 Software Requirements: . . . 14
1.7 Publications . . . 15
1.8 Thesis Outline . . . 16

2 Background **18**

2.1 Vision and Visual Impairment . . . 18
2.1.1 Types, Prevalence, and Challenges of Visual Impairment . . . 19
2.1.2 Cognitive Mapping of VIPs . . . 20
2.2 Computer Vision and Deep Learning . . . 23
2.2.1 Object Detection . . . 25
2.2.2 Object Tracking . . . 28
2.2.3 Distance Estimation . . . 30
2.3 Path Planning . . . 33
2.4 Optimisation . . . 36
2.4.1 Classification of Optimisation Problems . . . 37
2.4.2 Solving Methods for Non-Convex Optimization Problems . . . 38

3 Related Work **41**

3.1 Mobiltiy Assistant Aids for VIPs . . . 41
3.1.1 Non-Vision-Based Mobility Aids . . . 42
3.1.2 Vision-Based Mobility Aids . . . 45
3.2 Local Path Planning . . . 48

4 Schematic Design **55**

4.1 AR-AwareAvoid Scheme 55

4.1.1 Wearable Vision Module (WVM) 56

4.1.2 ARAware: CMOs Identification Scheme 57

4.1.3 MinD: Local Path Planning Scheme 78

5 Implementation and Evaluation **90**

5.1 ARAware Implementation 90

5.1.1 ARAware's Prototype Implementation 90

5.1.2 Evaluation Methodology 92

5.1.3 Experimental Dataset 94

5.1.4 Evaluation Metrics 96

5.2 MinD Implementation 98

5.2.1 Evaluation Methodology 101

5.2.2 Evaluation Metrics 102

6 Results and Discussions **104**

6.1 ARAware Results and Discussions 104

6.1.1 MOT Validation 104

6.1.2 Validation of Distance and Speed Estimation 106

6.1.3 Performance of ARAware 113

6.1.4 ARAware versus DEEP-SEE 118

6.2 MinD Results and Discussions . . . 122
6.2.1 Travelling Distance . . . 122
6.2.2 Time Separation . . . 130
6.2.3 Collision Rate . . . 134
6.2.4 Computation Time and Convergence Iteration . . . 136

7 Alternative Approaches and Limitations 139

7.1 Performance Comparisons: YOLOv8 vs YOLOv3 . . . 139
7.1.1 Object Detection . . . 140
7.1.2 Object Tracking . . . 142
7.1.3 E2E CMO Identification . . . 142
7.2 Impact of Frame-Skipping on ARAware . . . 144
7.3 Impact of Moving Angle Constraint on MinD . . . 148
7.4 Limitations . . . 152
7.4.1 Limitation of ARAware . . . 153
7.4.2 Limitations of MinD . . . 154

8 Conclusion and Future Work 157

8.1 Conclusion . . . 157
8.2 Future Work . . . 166

Bibliography 171

.1 Notations . . . 202

.2 Derivation of the α_i' angle 206

.3 Ethical Approval 207

List of Tables

5.1 The Parameter Settings of MinD's Simulation Experiments for Each Class. . . . 100

5.2 The Parameter Settings of MinD's Simulation Experiments for All Classes. . . . 100

6.1 The Performance of YOLOv8-Based Object Detector on VMOT With Two Different Sizes of YOLOv8. 105

6.2 The Performance of Deep SORT-Based Object Tracker on VMOT. 105

6.3 Performance of Distance and Speed Estimation on VCRP. 109

6.4 Overall Performance of ARAware. 114

6.5 Performance of ARAware's CMO Identification on VMOT. 114

6.6 Performance of ARAware's CMO classification On VCRP. 116

6.7 Performance of ARAware's Warning Emission. 116

6.8 Warning Emission Deadline Setups. 117

6.9 Object Tracking Performance Evaluation: ARAware's Deep SORT-Based Tracker vs. DEEP-SEE's GOTURN-Based Tracker. 119

6.10 CMO Identification Performance Evaluation: ARAware vs. DEEP-SEE. 120

6.11 Average Travelling Distances Achieved for Each CMO Class. 129

6.12 Average Time Separation Achieved for Each CMO Class. 136

6.13 Collision Rate Among MinD Scheme and Other Methods. 137

6.14 The Average Computation Time and Convergence Iteration. 137

7.1 Object Detection Performance on VMOT: YOLOv8 vs. YOLOv3. 141

7.2 Object Tracking Performance on VMOT: YOLOv8 vs. YOLOv3. 142

7.3 Overall Performance of ARAware's CMO Identification: YOLOv8 vs. YOLOv3. 144

7.4 Results of ARAware's CMO Identification Under Different Frame-Skipping Settings. 145

7.5 Performance of ARAware's CMO Identification on VMOT Under Different Frame-Skipping Settings for Each Class of CMOs. 147

8.1 VIP Assistive Mobility Solution Requirements 161

1 Notations Used Throughout the Thesis. 202

List of Figures

2.1 Human Visual System [1]. 19

2.2 A Timeline of the Most Active Computer Vision Research (Reproduced From [2]). 23

2.3 The Taxonomy of AI (Reproduced From [3]). 24

2.4 A Timeline of YOLO family (Reproduced From [4]). 26

2.5 A Visualization of the YOLOv8 Architecture (Reproduced From [4]). 27

2.6 Tracking-By-Detection (TBD) Paradigm [5]. 29

2.7 Examples of Framework Space Approach [6]. 34

2.8 The Scheme of Artificial Potential Field (APF) [6]. 35

4.1 The General Schematic Architecture of AR-AwareAvoid. 56

4.2 The General Schematic Architecture of ARAware. 58

4.3 Moving Object Trajectory Update. Note: When the trajectory is updated at t_j, the estimated distance and estimated moving speed are also updated to calculate the latest expected collision time. 60

4.4 Demonstration of the Reference Camera Coordinate System. Note: The camera center is at the origin (0,0,0), X-axis and Y-axis are the horizontal and vertical axes that are parallel to the image plane axes, and Z-axis is the principal axis that is orthogonal to the image plane. 62

4.5 Distance Estimation Based on DisNet (Reproduced From [7]). 62

4.6 Demonstration of the 2D XZ-Plane Considered in Collision Prediction. Note: The 2D XZ-plane is the ground plane shared by the VIP and the detected object. 65

4.7 Critical Zone Estimation Based on Geometric Principles. 66

4.8 Critical Zone Estimation Based on Geometric Principles for Multiple objects. . . 68

4.9 CMO Risk Level Classification Based on Potential Damages. 72

4.10 VIP Local Path Planning Problem. 79

4.11 CMO Collision Avoidance. 83

4.12 Collision Avoidance for Multiple CMOs. Note that the VIP is directed based on the combined angles of all CMOs. 85

5.1 The ARAware's Real-World Prototype. 91

5.2 Illustration of Video Clip Collection and Experimental Environment Setting-up. 96

6.1 The Validation of Distance Estimation of All Four Classes in VCRP (44 Samples in Total). (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Samples, Y-axis (left): Distance, Y-axis (right): Error. 106

6.2 The Validation of Speed Estimation of All Four Classes in VCRP With Static Camera (21 out of 44 Samples, in Total). (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Samples, Y-axis (left): Speed, Y-axis (right): Error. 111

6.3 The Validation of Speed Estimation of All Four Classes in VCRP With Moving Camera (23 out of 44 Samples, in Total). (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Samples, Y-axis (left): Speed, Y-axis (right): Error. 112

6.4 Illustration of CMO Identification on VMOT. 114

6.5 Illustration of CMO classification On VCRP. 115

6.6 The Travelling Distance of Each CMO Class Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF). . . . 123

6.7 The Travelling Distance of Two CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF). 125

6.8 The Travelling Distance of Three CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF). . . . 126

6.9 The Travelling Distance of Five CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF). 127

6.10 The Travelling Distance of Seven CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF). . . . 128

6.11 The Time Separation of Each CMO Class Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF). 130

6.12 The Time Separation of Two CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF). 132

6.13 The Time Separation of Three CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF). 133

6.14 The Time Separation of Five CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF). 134

6.15 The Time Separation of Seven CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF). 135

7.1 The performance of ARAware's CMO Identification With Different Inter-Frame Delay. 146

7.2 The Travelling Distance and Time Separation of Two CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time. 149

7.3 The Travelling Distance and Time Separation of Three CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time. 150

7.4 The Travelling Distance and Time Separation of Five CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time. 151

7.5 The Travelling Distance and Time Separation of Seven CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time. 152

1 Ethical Approval Form. 207

List of Acronyms

APF Artificial Potential Field.

AR-AwareAvoid Ambient Risk Awareness and Avoidance.

ARAware Ambient Risk Awareness.

CCS Camera Coordinate System.

CMOs Critical Moving Objects.

CRP Collision and Risk Predictor.

E2E End-to-End.

EmFPS End-to-End mean Frame Per Second.

GT Ground Truth.

HCI Human-Computer Interaction.

IOU Intersection Over Union.

mAP mean Average Precision.

mAR mean Average Recall.

MinD Minimising Distance.

MOT Moving Object Tracker.

MOTA Multiple Object Tracking Accuracy.

MOTP Multiple Object Tracking Precision.

NBRD Notification-Based Random Decision.

PCS Pixel Coordinate System.

PM 2D-to-3D Projection Module.

SCP Sequential Convex Programming.

SETREC Science Engineering and Technology Research Ethics Committee.

SORT Simple Online Real-Time.

VCRP Video-Based Collision and Risk Prediction.

VIPs Visually Impaired People.

VMOT Video-Based Multiple Objects Tracking.

WCS World Coordinate System.

WEM Warning Emission Module.

WVM Wearable Vision Module.

YOLO You Only Look Once.

Chapter 1

Introduction

This chapter sets the stage for this thesis by introducing the research topic of interest, assisting the mobility of visually impaired people (VIPs), and the motivation behind this research. It begins by providing a brief overview of the daily challenges faced by VIPs, highlighting the global prevalence of visual impairments. Next, it delves into existing assistive technologies available to assist VIPs' mobility, critically examining their strengths and limitations. Then, it outlines the research objectives and contributions. It also covers the hardware and software setups used in this research, along with the datasets employed for evaluation. Finally, it provides an overview of the resulting publications and a roadmap for the thesis structure.

1.1 Motivation

Due to the limited visual perception, visually impaired people (VIPs) face significant challenges in their daily lives. These challenges can range from difficulties with mobility and orientation to social isolation and reduced quality of life. According to the World Health Organization [8], approximately 258 million individuals globally live with visual impairment; 39 million are completely blind and 219 million experience different vision problems. This vast number highlights the global scope and the urgency of addressing the challenges faced by this population. Safe navigation of outdoor environments constitutes a considerable challenge for VIPs as vari-

ous moving obstacles like cars, motorcycles, bicycles, and pedestrians are present. The diverse characteristics of moving obstacles, including different sizes, varying movements and velocities, and differing approaching times toward a VIP, present complex challenges and pose various potential risks to VIPs 's physical safety or likely death [9]. For example, cars are more hazardous for VIPs than bicycles due to their larger size, higher speeds, and resulting impact force [10], and fast-moving cars with speeds $> 50km/h$ pose a greater risk of death and have a shorter arrival time than slow-moving cars with speeds $< 15km/h$ [11]. As stated in [12], in India, over 30,000 VIPs die annually due to mobility troubles, and around 400 are hospitalised as the result of pedestrian-vehicle accidents. This emphasises the significance of identifying such ambient threats while considering the aforementioned characteristics to ensure VIPs' safety and independence mobility.

VIP Mobility assistance approaches can be generally classified into conventional tools, such as guide dogs and white canes, and technological solutions, including non-vision-based (rely on sensors like ultrasonic) and vision-based (use cameras like RGB) approaches [13]. With conventional tools, moving objects are difficult to predict as they only help with obstacle avoidance in the immediate area of the VIP [14]. Specifically, white canes provide limited tactile information confined by the obstacle size [15], while guide dogs can identify serious situations but cannot provide information about the precise nature of the situation [16]. Technological solutions offer more advantages than traditional tools in detecting obstacles at a longer distance and can provide more comprehensive and precise information about potential threats without the need to train animals [17, 18]. VIPs' mobility can be significantly improved by providing VIPs with sufficient spatial and contextual information about potential threats in their surroundings [19].

To detect obstacles, non-vision-based technological solutions use active sensors like laser [20], infrared [21], and ultrasonic [22], while vision-based solutions use different cameras types like stereo [23], RGB [24], and RGB-D [25]. Current research has preferred vision-based solutions for their potential to provide richer spatial and contextual information. Cameras can function similarly to the human eye, enabling vision-based methods to effectively detect and classify objects, understand the surrounding environment, estimate the distance between VIPs and obstacles, and predict potential risks.

1.2 Problem Statement

Considerable vision-based solutions have been proposed to assist VIPs' outdoor mobility by identifying obstacles along VIPs' paths [13]. However, several practical drawbacks exist, which are classified into two groups and discussed in detail below.

Threat Identification Limitation. In practice, these solutions significantly fail to ensure the VIPs' safety as more serious threats, *e.g.,* faster-moving vehicles, cannot be effectively identified. This is because of a practical deficiency in their object identification methods, such as a lack of object tracking [26] and/or a restricted detection area of only up to $10m$ [26, 27, 28, 29]. For example, in [26], a smartphone-based VIP navigation system is proposed. It detects static and moving objects and estimates objects' distance to the VIP and approaching direction (*i.e.,* front, right, or left). Based on a Convolutional Neural Network (CNN) object classifier, the system can balance detection accuracy and processing speed by switching between fast and stable modes. However, this system cannot precisely predict possible collisions with VIPs due to the lack of tracking of an object's movement. Object tracking is crucial for obtaining critical information (*e.g.,* object velocity and moving direction) and constructing object's moving trajectory, thus effectively predicting collisions. The system's ability to estimate object-VIP distance is also practically limited; only $10m$ can be roughly approximated. This distance is insufficient for providing early warning of fast-moving objects. Besides, the system's recognition rate is relatively low, only 55%, which increases to 60% when certain obstacle types like potted plants are excluded. Studies on assisting VIPs, e.g., [30, 31], consider an accuracy above 85% very good for obstacle recognition.

In [28], a CNN-based navigation assistant, DEEP-SEE, for VIPs is introduced. It jointly detects, tracks, and recognises obstacles VIPs encounter in urban areas, such as cars, motorcycles, bicycles, pedestrians, and obstructions. It warns about these threats based on their relative position to the VIP with different priorities, *e.g.*, cars have the highest priority. However, DEEP-SEE's collision identification range (the trapezium of interest) is limited to approximately $6m$, with an average collision time of only $1 \sim 2s$, even for slow-moving objects ($< 22km/h$). Such a short time is generally inadequate for the VIP to react. Further, since DEEP-SEE does not

differentiate between objects moving at varying speeds, the proposed fixed $6m$ safety zone cannot effectively protect VIPs from fast-moving objects like cars and motorcycles. For example, if a car travels at $35km/h$, within this fixed range, the VIP will have only $0.62s$ to react, which is a significantly short time. To ensure VIP safety, the safety zone should adapt according to the speed of approaching objects, allowing ample time for avoidance. Besides, the DEEP-SEE's threat priority mechanism is not comprehensive; it does not account for situations where a distant fast-moving low-threat object could collide with the VIP before a close slow-moving high-threat object. Finally, DEEP-SEE cannot work in real-time (using the equivalent hardware) as it can only process 20 frames per second (fps) of the $30fps$ input video on an NVIDIA GeForce GTX 1050 GPU. *Real-time* video processing is defined as the capability to process all captured frames immediately, in which each frame's processing time should be $\leq 0.033s$ ($33ms$), assuming that the camera captures $30fps$ [32]. The *'real-time'* notion is a crucial to prevent the loss of critical spatial, temporal and semantic information from the input video frames, which is essential for *accurate* and *in-time* collision prediction. According to [33], based on specific application requirements, a frame rate of $10fps$ or higher can suffice for detecting fast-moving objects, while a lower frame rate of $2 \sim 3fps$ might be enough for detecting slow-moving ones. However, if the object detection rate lags behind the camera's capture rate, it could pose a serious problem for real-time processing; it could lead to accumulating delays over time, significantly impacting real-time performance [33]. Research on VIP assistance suggests a minimum frame rate of $30fps$ is essential for real-time performance [34, 35].

In [27], a stereo camera-based system, Bbeep, is proposed to aid VIPs in navigating crowded environments such as airports by alerting both the VIP and nearby pedestrians to clear the VIP's path. Bbeep tracks the movement of sighted pedestrians and predicts their future trajectories and potential collisions with the VIP in real-time. Bbeep will trigger audible alarms if a pedestrian is predicted to be located within a $1m$-wide VIP safety path, and three warning signals can be emitted: a low-urgency beep (a collision within $5s$), an intermediate-urgency beep (a collision within $2.5s$), and a stop sound (an imminent risk within $70cm$). However, Bbeep is specifically designed for pedestrians; it cannot predict the future trajectories of other objects, such as cars moving at higher speeds or objects with less predictable movement pat-

terns. Moreover, Bbeep's limited safety zone ($1m$-wide) and passive warning emission method, which rely on simple beeps to alert nearby pedestrians to actively avoid collisions with the VIP, cannot effectively prevent collisions with more critical objects like cars, motorcycles, and bicycles. Such objects are more challenging to detect and react to the beeping alarms.

Path Planning Limitation. The majority of vision-based solutions cannot aid VIPs in effectively avoiding collisions as they do not carry out *optimal collision-free local path planning*. Path planning generally involves determining a route from a starting point to a final destination while avoiding obstacles in between [36]. Path planning is essential for efficient VIP navigation as it evaluates all navigation possibilities, which helps avoid unnecessary or unsafe routing while minimising the VIP's movement time, effort, and computer resources [37]. Accordingly, numerous efforts have been made to enhance VIP mobility. These efforts include adapting existing robot global path planning algorithms (*e.g.,* A* [38], Dijkstra [39], Ant Colony Optimisation (ACO) [37], and Particle Swarm Optimization (PSO) [40]) and developing novel ones [41, 42]. Nonetheless, global path planning algorithms are incapable of prevention collisions between VIPs and dynamic obstacles, since they necessitate a static representation of the environment, which is impractical for real-world scenarios.

For example, in [41], a VIP multi-objective path planning algorithm, OPTIPATH, is proposed to minimise the total distance and the number of turns that a path can take from the source to the final destination (i.e., achieving the shortest path with the minimum number of turns) in an indoor environment, i.e., supermarket. The algorithm can re-route the VIP when an obstacle is detected. However, it is impractical for two reasons. First, it assumes that all obstacles are static. Second, it first plans the optimal route and then detects obstacles between two successive nodes of the whole path; for instance, this may result in a path suggestion that is blocked by a stationary object. Thus, the number of misdirections increases, which leads to longer distances for the VIP to travel. A look-ahead collision-free estimation (that ensures no obstacle exists in the next time slot) can overcome such problems. In [42], DynaPATH is proposed to plan the safest paths in indoor environments with dynamic changes. It exploits data generated by the IoT-embedded device to adapt to such changes by dynamically building a global obstacle model (defining obstacle positions), which is combined with a grid map (containing an environment's

structure, *e.g.,* walls). However, when objects are moving at high speed, *e.g.,* cars, it is unsafe to update the obstacle model based on the objects' current relative positions only; the objects' own movement patterns should be considered also.

Local path planning is more promising than global path planning, as it can promptly adapt to real-time changes in the VIP's surroundings, such as the sudden appearance of obstacles. This is achieved by dynamically calculating local collision-free routes without requiring a full predefined environmental map. Existing local path planning algorithms have primarily been developed for robotics [43]. As robot mobility constraints differ from those of VIPs, such algorithms cannot be directly applied to VIPs. Robots are fully controllable; their avoidance reactions, like stop or move, and movements, like acceleration and velocity, are completely and automatically controlled and adjusted by planning algorithms, which is not the case for VIPs [44]. This makes these algorithms more complex. Besides, ensuring VIP safety is a top priority, and mistakes cannot be tolerated; this requires a shift in perspective that considers VIPs' physical capabilities in practice. As far as the researcher knows, no study has developed a local path planning algorithm dedicated to VIPs that can be applied to an unknown outdoor environment containing moving objects of various classes and speeds.

To summarise, the state-of-the-art vision-based VIP mobility assistant solutions are promising but cannot effectively prevent collisions between a VIP and different types of dangerous objects moving at low-to-high speeds (*i.e.*, Critical Moving Objects, CMOs). These solutions are not capable of practically assisting VIPs and providing them with sufficient time to successfully avoid such threats. A more practical VIP assistance scheme should accurately identify and effectively warn the VIP about CMOs, generate optimal collision-free paths, and ensure sufficient time for VIPs to avoid CMOs in-time. Further, to ensure dynamic timely VIP safety, all processes involved in the scheme should be conducted in near real-time, without missing any critical information in the input video frames.

1.3 Research Aims

This research presents AR-AwareAvoid (or Ambient Risk Awareness and Avoidance) AI-based assistance scheme to assist VIPs in safely navigating outdoor environments in which different classes of CMOs, *i.e.,* cars, motorcycles, bicycles, and pedestrians, moving at varying speeds, are present. As far as the researcher knows, AR-AwareAvoid is the first real-time practical camera-based VIP mobility assistant scheme that (1) effectively and efficiently identifies CMOs and provides longer VIP-reaction times for CMO avoidance by addressing the combined issues of *CMO identification, risk level-based CMO classification, and prioritised CMO warning* simultaneously, and (2) plans the shortest path required to avoid CMOs considering CMOs' motion constrains and VIPs limited physical abilities, such as walking speed and auditory perception, and is able to provide look-ahead safety prediction on the planned path. To bring these together effectively requires different thinking and novel algorithms. This is to address the limitations of existing VIP mobility assistant solutions and enhance VIP safety. To evaluate the AR-AwareAvoid scheme in a highly practical setting, this research also implements a prototype and conducts extensive real-world scenarios in outdoor environments. This research also develops an empirically parameterised simulator that enables safe 'what-if' experimentation, allowing for a thorough and ethical evaluation of the proposed scheme.

1.4 Contributions

The core contributions of this thesis lie in developing and evaluating the key components that constitute the framework of AR-AwareAvoid, and can be summarised as follows:

1.4.1 ARAware: CMOs Identification

- The research provides an end-to-end camera-based CMO avoidance scheme, ARAware (or Ambient Risk Awareness), from video capture to message delivery, where the essential properties of the VIP's perception in practice and CMOs' types and kinematics

are comprehensively considered. In particular, it designs a Deep Learning-based moving object identification capable of achieving real-time processing without missing critical information in the input video. Further, it develops a novel object-VIP collision prediction method based on objects' moving direction for effective collision detection. It also constructs a 2D to 3D coordinate projection method for more accurate object localisation and object-VIP distance estimation, which significantly improves the performance of object-VIP collision prediction and, thus, CMO identification. The ARAware scheme is customisable, designed for future-proofing, so that other object detectors, trackers, and distance estimators can be integrated for either higher accuracy or faster processing speed according to the VIP's requirements.

- It provides a systematic CMO risk level estimation scheme derived from comprehensive empirical studies that provides an indicator of the damage caused to the VIP via collisions of different types of objects moving at different speeds in practical settings. It classifies the identified CMOs with different risk levels, and develops a novel prioritised CMO warning strategy based on CMOs's risk levels and predicted collision times. This is to immediately warn about high-risk threats for VIP's safety and to postpone the warning of low-risk threats until an acceptable deadline (sufficient for VIP to react) to reduce potential false alarms.

- It implements a real-world prototype of ARAware using commercially available devices (*i.e.*, an RGB camera, a laptop, and bone-conduction headphones), and validates its effectiveness and efficiency based on extensive real-world field trials. Experimental results demonstrate that, consuming $30fps$ input video, ARAware manages to accurately identify CMOs with 97.26% overall mean Average Recall (mAR) and 88.20% overall mean Average Precision (mAP), precisely classify their risk levels with 100% overall mAR and 91.69% overall mAP, and achieve real-time processing with an average E2E processing speed of $32fps$ ($31ms$). Furthermore, ARAware's warning emission strategy manages to effectively alert the VIP about identified CMOs with different risk levels, where high-risk CMOs are alerted immediately (*e.g.*, cars moving at $60km/h$ are alerted at least $6s$ in advance), and low-risk CMOs are not alerted until the deadline is reached (*e.g.*,

pedestrians moving at $3km/h$ are alerted $5s$ ahead, instead of $15.21s$ in advance when they are first identified), which effectively reduce potential false alarms, considering the accuracy of detecting proximate objects is higher than that of distant objects. ARAware achieves significantly higher CMO identification accuracy than the closest state-of-the-art approach, DEEP-SEE [28], by 42.62% in mAR and 10.88% in mAP, with 93% faster end-to-end processing speed.

1.4.2 MinD: Local Path Planning

- The research develops a collision-free local path planning scheme, MinD, consistent with VIPs' limited capabilities, which plans the shortest path required by the VIP to avoid CMOs, where the VIP's walking speed, auditory perception, various CMO types, and CMOs' movement patterns are comprehensively accounted for to ensure that the VIP can avoid threats effectively and within a safe time. In particular, it formulates MinD to be conditional on the estimated collision times between the VIP and the CMOs and an adaptive threshold set based on the CMO's risk level, *i.e.,* a large value for high-risk CMOs (cars and motorcycles) to ensure VIP safety, and an acceptable value for low-risk CMOs (bicycles and pedestrians) to avoid unnecessary VIP transitions. It also conducts a look-ahead safety prediction trajectory analysis of the optimal path to ensure more safety for VIPs. This predicts the future positions and movement patterns of the VIP and the CMOs and estimates the collision-free and safe time interval between them along the optimal path according to an adaptive threshold set based on their speeds.

- The VIP-local path planning problem is challenged by being non-convex by nature. Therefore, this research re-formulates the non-convex problem into a solvable convex local path planning sub-problem using *Sequential Convex Programming* (SCP) to find a tolerable sub-optimal solution and reduce the computational complexity of the original non-convex problem.

- To evaluate the MinD scheme in a highly practical setting, the research implements an empirically parameterised simulator, where its parameters are derived from the extensive

real-world experiments conducted by ARAware's implemented prototype: people walking, cycling and driving in outdoor spaces, while the researcher mimics the VIP. Note, simulation allows safe 'what-if' experimentation to fully evaluate MinD against the state-of-the-art, and in an ethically safe, near-to-realistic way. Experimental results show that MinD outperforms the state-of-the-art *Artificial Potential Field* (APF) [45] approach and the baseline Notification-Based Random Decision (NBRD) method in effectively planning free-collision paths suitable for VIP's mobility and perception conditions, ensuring short avoidance distances. MinD requires only $2.75m$ of avoidance distance on average for all CMO classes (up to 7 CMOs) moving at varying speeds from $3.6km/h$ to $108km/h$, which is considerably shorter than APF and NBRD by 84.47% and 75.94%, respectively, and ensures a 0% collision rate. With this distance, the VIP can avoid up to 7 CMOs moving at speeds of up to $108km/h$ with only $1.98s \sim 6.55s$ reaction time when walking at speeds of $1.5km/h \sim 5km/h$ (note, with APF and NBRD, the VIP needs $\geq 7.96s$ reaction time). MinD also achieves a large average safe time separation ($11.00s$ for cars, $10.78s$ for motorcycles, $8.23s$ for bicycles and $6.43s$ for pedestrians, which is $\geq 4.80s$ and $\geq 1.24s$ compared to APF and NBRD, respectively). This is achieved with a negligible average processing time of $1.20ms$, running on an NVIDIA GeForce RTX 2080 GPU.

1.4.3 Performance Comparisons: YOLOv8 vs YOLOv3

- This research provides an in-depth comparison of YOLOv3 and YOLOv8, evaluating their effectiveness in identifying CMOs with a focus on both accuracy and efficiency. Given the detector's critical role in the proposed scheme's core functionalities, this comparison aims to identify the detector that effectively meets the scheme requirements, i.e., accuracy and real-time performance. This evaluation encompasses three key components: object detection, YOLO-based object tracking, and end-to-end CMO identification.

- By utilising YOLOv8, which provides both improved accuracy and reduced computational requirements, ARAware's object detection module significantly enhances the scheme's overall performance. i.e., in accurately identifying CMOs and providing real-time VIPs

mobility assistance.

1.4.4 Performance Evaluation Under Different Settings

- This research conducts additional evaluation, specifically evaluating YOLOv8-based CMO identification under six k-frame skipping settings (e.g., k= 1, 3, 5, 15, 30, and 50 frames) to cope with limited computational resources.
- It also evaluates MinD's performance under two settings regarding CMOs' moving angle: *nearest neighbour focus* and *comprehensive awareness*. This is to identify the setting that best enables effective free-collision path planning with larger safety buffers and minimal travel distances.

1.5 Hardware, Software, and Dataset Setup

The following subsections detail the hardware, software, and dataset configurations used in this research, focusing on implementing and evaluating the ARAware and MinD Schemes.

1.5.1 ARAware Scheme

To effectively evaluate ARAware, a real-world prototype was implemented using commercially available devices. Specifically, an RGB camera is used to capture video streams at a resolution of 640×480 and a frame rate of $30 fps$. The camera was mounted on a stabiliser to reduce shake and capture smooth and clear footage for accurate analysis. The stabiliser is securely attached to a chest belt mount strap, positioning the camera at the VIP's eye level (approximately $1.52m$ high). This setup frees the VIP's hands and provides a close, natural human field of view. A laptop equipped with an Intel Core i9-9900K CPU, an NVIDIA GeForce RTX 2080 GPU, and 64GB RAM was used to host ARAware's components and process the video streams (received from the camera via USB) in real-time. The primary processing unit is the GPU, which

provides the necessary computational power for deep learning-based tasks and real-time end-to-end CMO identification. The operating system used was Ubuntu 18.04.4 LTS, chosen for its robustness, security features, and compatibility with different development tools and libraries used in ARAware. Bone conduction headphones were utilised to deliver audio feedback to VIPs. These headphones allow the VIP to receive auditory information via Bluetooth without blocking ambient sounds, ensuring the VIP's safety and situational awareness.

To implement all of ARAware's components, Python 3.7.7 was used due to its compatibility with and support for a wide range of machine learning libraries and deep learning frameworks, such as TensorFlow and Keras. In particular, the Ultralytics Python library 8.0.1453[1] was used to implement ARAware's YOLOv8-based object detector. For ARAware's Deep SORT-based object tracker and DisNet-based distance estimator, public implementations[2,3] based on TensorFlow-gpu 2.1.0 and Keras 2.3.1 were adapted and employed. Camera calibration based on OpenCV 4.8.1.78[4] was used to estimate the camera focal length as $f_x = 644.83px$ and $f_y = 644.07px$, with the principal point determined to be $(p_x, p_y) = (322.97px, 238.56px)$. The CUDA Toolkit 11.0.182 and cuDNN 8.5.0.96 were used to leverage the GPU acceleration for ARAware's deep learning-based components, significantly enhancing the ARAware's computational efficiency and performance.

Using the ARAware prototype discussed above, two datasets, VMOT (Video-Based Multiple Object Tracking) and VCRP (Video-Based Collision and Risk Prediction), with a combined total duration of $43min$ (totaling 74.207K frames and 397.7 MB in size), were collected for evaluation. These datasets captured diverse real-world scenarios in private outdoor environments.[5] Up to five volunteers, following pre-defined scripts, performed actions like driving vehicles (i.e., up to two cars and one motorcycle) and bicycles (up to three) or walking (up to five pedestrians) at varying speeds (up to $60km/h$) and located at different distances (up to $100m$). These experiments took place during both daytime and nighttime, with lighting conditions exclud-

[1] https://docs.ultralytics.com/modes/predict/#inference-sources
[2] https://github.com/theAIGuysCode/yolov3_deepsort
[3] https://github.com/guanjianyu/DisNet
[4] https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html
[5] This study is approved by the Science Engineering and Technology Research Ethics Committee (SETREC) in the Imperial College London, see Appendix .3.

ing overly bright or extremely low-light situations, and the experimental results were averaged across both daytime and nighttime conditions. Weather conditions were not considered in this stage and are planned for future investigation. The VMOT and VCRP datasets included specific ground truth data (manually identified using the MATLAB Ground Truth Labeler[6]) for objects moving angles, distances to the camera, speeds, and risk levels, which are not provided in existing public datasets (like COCO and KITTI). More details on each dataset are provided in Chapter 5.

1.5.2 MinD Scheme

Aligned with ARAware for consistency, MinD is also implemented in Python 3.7.7. This implementation leverages CVXPY 1.1.18 library for convex optimization, ensuring efficient and accurate computation of optimal paths. A 2D Python-based empirically parameterised simulator was developed to effectively evaluate MinD while prioritising health and safety. The parameters of this simulator were derived from extensive real-world experimentation conducted with ARAware. The simulator was executed on the same machine hosting ARAware, which is equipped with Ubuntu 18.04.4 LTS, an Intel Core i9-9900K CPU, an NVIDIA GeForce RTX 2080 GPU, and 64GB of RAM. It uses fundamental geometric algorithms to simulate the CMOs' movements and the VIP manoeuvres in 2D environments, assuming that both the VIP and CMOs have Y-axis coordinates of zero, as the Y-axis is perpendicular to the ground plane. These environments include car parks, driveways, pedestrian crossings, and walking footpaths. The Monte Carlo [46] simulation was utilised to generate a diverse set of random experimental scenarios (3,200 in total) for MinD evaluation. For each class of CMO, 800 scenarios were created, with CMOs moving towards the VIP at eight different speeds (100 scenarios for each speed): $11km/h$ to $108km/h$ for cars and motorcycles, $7km/h$ to $54km/h$ for bicycles, and $3.4km/h$ to $36km/h$ for pedestrians. For simplicity, the following assumptions were made during the experiment: CMOs are assumed to move directly towards the VIP without changing their direction or speed. The VIP is expected to manoeuvre around these threats by changing

[6]https://uk.mathworks.com/help/driving/ground-truth-labeling.html

direction. Evaluating MinD with CMOs adapting their speed and direction is left for future work. Additionally, the VIP is assumed to walk at a constant speed of $5km/h$ towards a pre-defined, known final destination. Future work will also include evaluating MinD when the VIP adapts their speed as well. More details on the simulator's parameters setup are provided in Chapter 5.

1.6 Hardware and Software Requirements

To run the proposed models of this thesis and ensure real-time processing capabilities, the following hardware and software configurations are recommended, based on the researcher's implementation.

1.6.1 Hardware Requirements:

- An RGB camera: Used for capturing real-time video data.
- Stabilser: Recommended to ensure smooth video capturing.
- Bone-conduction headphones: Used for delivering auditory alerts to the VIP without blocking ambient sounds.
- High-End Computing Portable Device: Equipped with a high-end GPU to receive images from the camera, perform processing, and trigger alerts. A laptop equipped with an Intel Core i9-9900K CPU, NVIDIA GeForce RTX 2080 GPU, and 64GB of RAM is recommended for real-time performance.

1.6.2 Software Requirements:

While the listed software versions are those used in the implementation, newer versions may also work, though they have not been investigated by the researcher.

- Ubuntu 18.04.4 LTS.
- Python 3.7.7.
- CUDA Toolkit 11.0.182
- CuDNN 8.5.0.96.
- TensorFlow-gpu 2.1.0.
- Keras 2.3.1.
- Ultralytics 8.0.1453.
- OpenCV 4.8.1.78.
- Numpy 1.19.1.
- CVXPY 1.1.18.
- lxml 4.5.2.
- tqdm 4.66.1.
- seaborn 0.12.2.
- Pillow 7.2.0.

Note that the camera needs to be calibrated to obtain the focal length and principal point. This calibration process is essential for the 2D-to-3D projection module of ARAware, enabling the recovery of the 3D trajectory of tracked objects for accurate collision prediction and CMO identification.

1.7 Publications

The work described in this thesis resulted in the following publications:

- **Hadeel Surougi**, Cong Zhao, and Julie A. McCann, "ARAware: Assisting Visually Impaired People with Real-Time Critical Moving Object Identification," in *Sensors*, 2024, 24(13):4282, pp. 1-33 (published). Note that this paper includes the contributions mentioned in Section 1.4.1

- **Hadeel Surougi**, and Julie A. McCann, "Real-Time Optimisation-Based Path Planning for Visually Impaired People in Dynamic Environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1839-1848, (published). Note that this paper includes the contributions mentioned in Section 1.4.2

In all of the above papers, Hadeel Surougi has been responsible for conducting all of the research, developing, analysis, experiments, simulations, and evaluations of results. She also obtained all the graphs and did all the writing of manuscripts.

1.8 Thesis Outline

This thesis is structured as follows:

Chapter 1. This chapter provides an overview of the research, including its motivation, problem statement, research objectives and contributions, and a summary of the resulting publications.

Chapter 2. To provide readers with a solid foundation, this chapter delves into the necessary background information and explores the fundamental concepts that form the basis of this research.

Chapter 3. This chapter critically reviews relevant literature and their key findings, including research on developing mobility assistant aids for VIPs and local path-planning approaches.

Chapter 4. This chapter delves into the schematic design of the proposed AR-AwareAvoid scheme, which comprises two key schemes: the CMO identification scheme, i.e., ARAware, and

the local path planning scheme, i.e., MinD. It provides a comprehensive overview of the methodology employed and a detailed description of the path-planning problem and its mathematical formulation.

Chapter 5. This chapter discusses the technical aspects of the proposed schemes' implementation, empirical evaluation, real-world experiments, and simulations.

Chapter 6. This chapter presents and analyses the results obtained from this work, providing valuable insights into the effectiveness of the proposed approachs.

Chapter 7. This chapter explores the performance of the proposed schemes under different settings and discusses their limitations and potential areas for improvement.

Chapter 8. This chapter concludes this work with a reflection on the research objectives and the overall contribution of the research to the field.

In conclusion, this chapter has highlighted the critical need for improved mobility assistive solutions for VIPs by outlining the daily challenges VIPs face and the limitations of existing assistive technologies. Having identified this gap, the chapter then presented the specific objectives of this thesis and the unique contributions it has achieved. Finally, it provided an overview of the resulting publications and a roadmap for the thesis structure. The next chapter will lay a foundation for the research outlined in the subsequent chapters by delving into essential foundational knowledge needed to comprehend the impact of visual impairment and the crucial areas in computer vision and deep learning for developing computer vision-based assistive solutions.

Chapter 2

Background

The previous introduction chapter explored the research motivation, defined the problem the research addresses and outlined the research goals, contributions, and resulting publications. To establish a clear foundation for the research detailed later in the following chapters, this chapter delves into the relevant foundational knowledge required for this thesis. It overviews visual impairment, its prevalence and impact on daily life, and VIPs cognitive mapping. It also offers a concise overview of essential areas in computer vision and deep learning, which are crucial for developing computer vision-based assistive solutions.

2.1 Vision and Visual Impairment

This section provides a general overview of blindness, shows the urgent need to pay attention to VIPs, and the factors that drive the motivation behind this thesis.

Our sensory systems, *i.e.,* sight, hearing, touch, smell, and taste, connect us to the outside world. Without them, we would be isolated from our environment and each other [1]. Sight (or vision) is crucial for many aspects of daily life, such as mobility, communication, depth perception, and experiencing the world's beauty through colours and visual stimuli. Vision is the ability of the eyes to detect and interpret light, resulting in the formation of images in

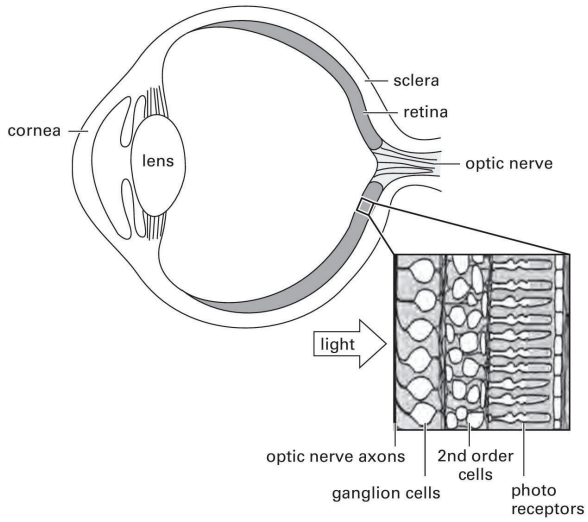


Figure 2.1: Human Visual System [1].

the brain. It is a complex process that involves several parts of the eye, including the cornea, lens, retina, and optic nerve, see Fig. 2.1. Light enters the eye through the cornea, which is refracted or bent, and then passes through the lens, further focusing the light onto the retina at the back of the eye. The retina contains specialised cells called photoreceptors that convert light into electrical signals. These signals are then transmitted through the optic nerve to the brain, which interprets them as images. Vision is typically compromised when defects or diseases occur in virtually any part of the visual system [1]. For instance, vision is debilitated if an opaque lens obstructs light entry into the eye or if the image fails to focus sharply on the retina. Various conditions or diseases, such as cataracts, glaucoma, age-related macular degeneration, diabetic retinopathy, and corneal opacity, can impair sight [47].

2.1.1 Types, Prevalence, and Challenges of Visual Impairment

Blindness has several different levels of severity, ranging from mild to complete or total blindness [48]. Some common terms used to describe the severity of blindness include 'partial blindness', 'low vision', 'legal blindness', and 'total blindness' [48]. Partial blindness describes someone who has some vision but not enough to perform daily tasks without assistance. Low vision describes someone who has a significant loss of vision but still has some usable vision that

can be improved with visual aids. Legal blindness describes someone having a visual acuity of 20/200 or worse in the better eye with corrective lenses or a visual field that is restricted to 20 degrees or less. Total blindness describes someone who has no light perception and cannot see anything. Surveys consistently report that total blindness is the most feared disability among people [1]. In the UK, it is approximated that almost two million individuals are impacted by sight loss, and by the year 2050, this number is expected to increase twofold [49]. The prevalence of vision impairment increases with age, where most affected individuals are over 50 years old [50]. The rising prevalence of vision impairment emphasises the urgent need to pay great attention to this population. Understanding VIPs' daily life challenges and actively seeking assistive solutions can significantly improve their quality of life, which is the primary motivation driving this thesis.

Visual impairment poses significant challenges to individuals in their daily lives. Depending on the severity of the blindness condition, VIPs may experience difficulty in performing routine tasks such as reading, cooking, travelling independently, distinguishing colours, perceiving depth, and identifying facial expressions and currency notes [51, 52]. Even simple tasks like crossing the street or navigating unfamiliar environments can be challenging and potentially hazardous [51]. VIPs may also experience social isolation [53] and difficulty in finding employment and accessing education and healthcare services [51, 54]. These challenges can significantly impact VIPs' quality of life and mental health, such as feelings of depression and anxiety [55]. Despite all the challenges VIPs face, outdoor navigation remains one of the most challenging tasks. According to [56], more than 63% of VIPs have experienced an injury while navigating outdoors at least once in the past. This significant challenge is the driving force behind this thesis, which focuses on developing a vision-based solution to assist VIPs in achieving greater independence and safety during outdoor mobility.

2.1.2 Cognitive Mapping of VIPs

Understanding how VIPs construct their spatial awareness through non-visual senses is essential when developing VIP assistive solutions [57]. This is to provide VIPs with the necessary

information (i.e., spatial and contextual) to effectively perceive their surroundings and ultimately enhance their mobility and independence. Therefore, this section provides an overview of VIP cognitive mapping.

Cognitive mapping (or spatial understanding) is a mental representation of physical space or environmental features [58], which is vital for developing adequate orientation and mobility skills [59]. Vision plays a crucial role in constructing individuals' cognitive maps of their surroundings and thus guiding their actions [58]. Several studies have investigated the ability of VIPs to construct cognitive maps as sighted people and three distinct theories have been developed: *deficiency*, *inefficiency*, and *difference* theories [60]. *Deficiency* theory believes that individuals who are congenitally blind are unable to construct spatial knowledge of their surroundings as they have never had the opportunity to experience environmental features visually [60]. *Inefficiency* theory states that VIPs can understand spatial concepts at a lower stage than sighted individuals due to their reliance on auditory and haptic cues alone. *Difference* theory suggests that VIPs possess the same fundamental abilities to process and understand spatial concepts as sighted people; while functionally equivalent, these processes might be carried out differently and often in a slower manner [61]. The researchers in [62] further argue that any differences in VIPs' mental representations of space are attributed to their lack of gathered information, experiences, or stress. For example, in [63], it is found that exposure to a simulation of a familiar urban area improved the knowledge of spatial configuration in adults with visual impairments. Additionally, in [64], it is stated that while non-visual senses may be deemed 'inferior' in encoding spatial relational information, VIPs possess no less potential than sighted individuals for developing a comprehensive representation of space.

To improve VIPs' spatial perception, it is essential to understand how they develop their mental representation and the environmental features (elements) consisting their mental map. The author in [65] claims that the cognitive maps of VIPs tend to contain basic features of environments (streets, buildings, parks, etc). In [58], it is stated that the formation of a cognitive map depends on the information received by the VIP. For example, if VIPs rely on a tactile map to identify their environment, their mental representation is formed based on the tactile map's elements. However, if they explore an area by walking, their cognitive map is constructed

based on environmental cues such as olfactory, auditory, and haptic [66, 67, 68, 69]. Generally, two methods exist for environmental space representation and spatial coding: *egocentric* and *allocentric* [70]. Egocentric method involves a self-to-object representation, which means the mental representation in which the location of an object is encoded relative to the self-body [70]. Allocentric method is considered object-to-object representation, where an object is spatially identified according to the locations of other objects [71]. Although the allocentric strategy may support better spatial representation [70], VIPs typically tend to use the egocentric strategy to recognise the spatial information of ambient objects [72, 73]. The egocentric strategy aligns well with how VIPs navigate their surroundings. They rely heavily on their body movements and how their senses interact with the environment as they move through it. This makes egocentric coding more natural for processing spatial information.

Given that VIPs, like sighted individuals, can generally understand spatial information, but potentially through slower processing due to reliance on non-visual cues, knowing that gathering enough information about the surroundings can enhance VIP spatial awareness and recognising that VIPs often rely on egocentric strategy to aware the spatial information of ambient objects, this thesis aims to take these into account. It aims to prioritise providing VIPs with sufficient time for conveyed messages to be understood and allows ample reaction time to navigate around obstacles identified relative to the VIP position. Additionally, it aims to furnish detailed audio descriptions of obstacles (e.g., type, risk level, direction, and arrival time) to enhance the VIP's perception of their surrounding threats and, thus, safe mobility. Voice feedback has received more attention in VIP assistive aids than haptic feedback [74]. This is because voice offers a natural and intuitive way to communicate and allows for richer and more descriptive information to be conveyed. Note that although constructing a complete mental representation of VIPs' surroundings falls outside the scope of this research, it aims to assist VIPs in identifying and avoiding ambient threats within their surroundings (critical parts of their surroundings), and this potentially contributes to enhancing their cognitive mapping and thus allowing for more effective and timely reactions.

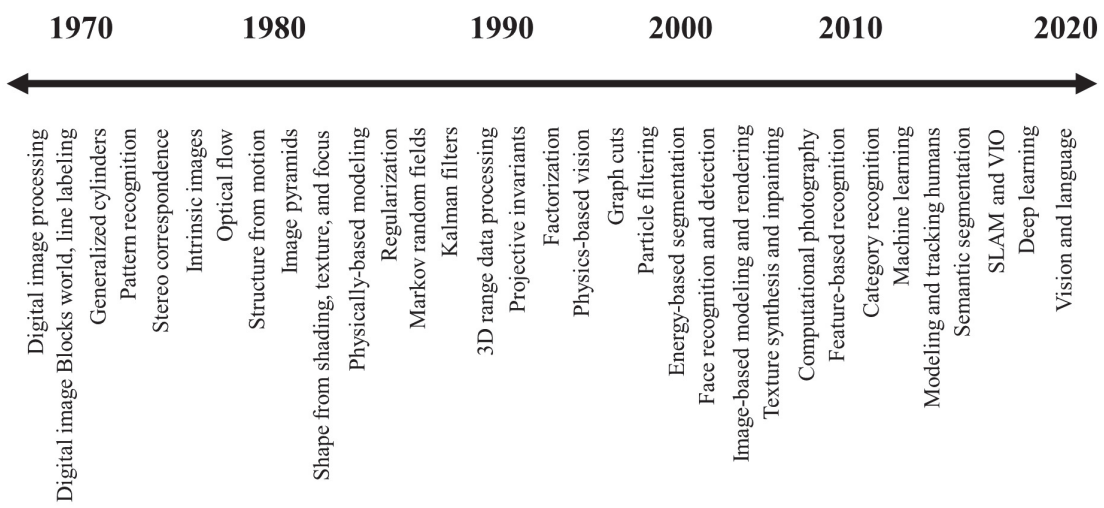


Figure 2.2: A Timeline of the Most Active Computer Vision Research (Reproduced From [2]).

2.2 Computer Vision and Deep Learning

This section delves into the core aspects of Computer Vision and Deep Learning that are essential for crafting vision-based assistive solutions for VIPs.

Computer Vision (CV) is a field of computer science that enables computers to interpret and analyse digital images and videos to mimic some aspects of human vision and perception [2]. It encompasses various techniques and algorithms to identify and classify objects, track motion, recognise patterns and shapes, estimate depth, reconstruct environments in 3D, and perform multiple other image and video analysis tasks. Such functionalities are generally valuable in developing assistive technologies for VIPs. For instance, CV can be used to identify obstacles like curbs, potholes, or uneven surfaces and can help VIPs navigate their surroundings more safely. A study published in 2022 [2] contends that despite the remarkable progress made in CV, the aspiration of attaining a computer capable of describing an image with the same level of detail and causality as a two-year-old child remains unfulfilled. This is primarily due to the limited understanding of human vision and the mechanisms of the human brain.

CV has been an extremely active research field for several decades, with experts dedicating their efforts to addressing significant and complex problems. Fig 2.2 presents a timeline of some research areas in CV that have been particularly active over the last 50 years. Deep

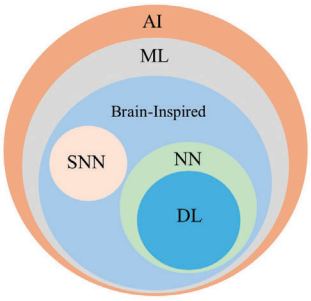


Figure 2.3: The Taxonomy of AI (Reproduced From [3]).

Learning (DL) has recently emerged as a transformative power in CV, achieving tremendous progress and impressive results in solving challenging problems such as object identification, natural language processing, and speech recognition [75]. DL is a branch of Machine Learning (ML). ML is a subset of Artificial Intelligence (AI), see Fig. 2.3, that strives to mimic the learning process of humans by allowing computer systems to enhance their performance on a given task without explicit programming [76]. DL algorithms are a mathematically sophisticated and advanced form of ML algorithms; they aim to imitate the structure and information-processing capabilities of the human brain [77]. Within the human brain, billions of interconnected neurons operate in unison to perform tasks such as processing information, pattern recognition, and decision-making [78]. Similarly, DL heavily relies on *neural networks* (NNs) simulating biological neurons, also known as artificial neural networks (ANNs) [77]. NNs comprise multiple layers (three or more) of interconnected nodes, known as *artificial neurons* [3]. These artificial neurons receive input from other neurons, conduct computations, and transmit output to other neurons in the next layer. By leveraging NNs, DL can effectively process large amounts of complex data and identify patterns and relationships in that data. Consequently, DL has achieved remarkable results in many fields, including science, healthcare, finance, and government [75, 77]. A part of this achievement is due to the availability of expansive datasets, powerful computing resources, and advances in algorithms and architectures [79]. Building upon DL's remarkable performance in CV, this thesis employs state-of-the-art DL-based methods for object detection, object tracking, and VIP-object distance estimation to develop an accurate and efficient vision-based assistive solution for enhancing VIP outdoor mobility.

2.2.1 Object Detection

In CV, object detection is a crucial task that identifies (classification problem) and localises (regression problem) objects of a specific class (e.g., cars, trees, or humans) in digital images or videos [80]. It serves as a foundation for numerous other CV tasks like object tracking [81], image captioning [82], and instance segmentation [83]. Recent advancements in DL techniques [84] have propelled object detection to the forefront of research, leading to remarkable breakthroughs and unprecedented attention. Object detection has become widely utilised in many applications like autonomous driving, robot vision, and video surveillance [80].

Existing DL-based object detectors can be broadly categorised into two-stage detectors and one-stage detectors [85]. Two-stage detectors employ a sequential approach and consist of two main stages: region proposal and object classification [80, 85]. In the first stage, a set of potential object locations (regions of interest, RoIs) is generated using various algorithms such as Selective Search or Region Proposal Networks (RPNs). In the second stage, the RoIs are further refined and classified into different object categories using convolutional neural networks (CNNs) like AlexNet [86]. CNNs, a specialized form of ANNs, are well-suited for image and visual data processing tasks. Their unique architecture, comprising convolutional, pooling, and fully connected layers, makes them highly effective for object detection and other tasks like image classification and segmentation [87]. Examples of two-stage detectors are Faster R-CNN, R-FCN, and Cascade R-CNN [88]. One-stage detectors combine both tasks (classification and localisation) in a single step. This eliminates the computational overhead and accelerates inference, making them suitable for real-time applications [85]. As mentioned in Chapter 1, achieving a minimum processing rate of $30fps$ (each frame processing time should be $\leq 0.033s$) is necessary for real-time VIP assistance [34, 35]. Popular one-stage detectors include You Only Look Once (YOLO) [89] and Single-Shot Multibox Detector (SSD) [90].

The YOLO has stood out among various object detection algorithms due to its exceptional balance between speed and accuracy, which enables rapid and reliable object identification [91]. The YOLO family has undergone successive iterations, each addressing limitations and enhancing performance from the previous version [91], see Fig. 2.4. For example, in 2018,

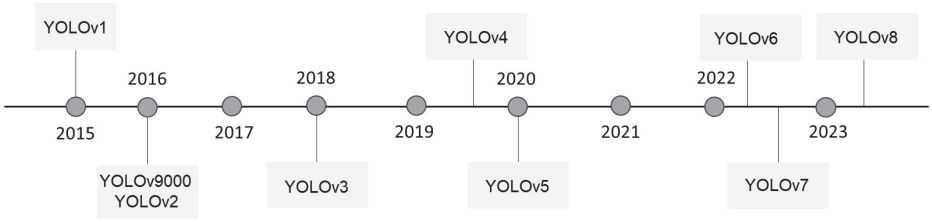


Figure 2.4: A Timeline of YOLO family (Reproduced From [4]).

YOLOv3 is introduced by [92] to improve object detection accuracy and speed over previous versions, *i.e.,* YOLOv2. It utilises a feature pyramid network to effectively predict object categories and bounding boxes at three different scales, enabling better handling of objects of varying sizes. Additionally, YOLOv3 introduces novel techniques like multi-scale training and dynamic anchor assignment, further improving its object detection accuracy [4]. Further significant improvements over YOLOv3 are YOLOv4 [93] and YOLOv5 [94], incorporating novel architectures, techniques, and training methodologies to achieve even better performance.

YOLOv8 stands as the latest version (at the time of thesis writing) of the YOLO series [95]. YOLOv8 was developed in 2023 by Ultralytics, the same team that develops YOLOv5. YOLOv8 builds upon the success of YOLOv5, incorporating a modified backbone, anchor-free head, and novel loss function for enhanced performance and user-friendliness across diverse computer vision tasks. YOLOv8 achieves unmatched speed and accuracy while maintaining a streamlined design suitable for various applications and adaptable to different hardware platforms [4]. Fig. 2.5 depicts the architecture of YOLOv8. It consists of two main components: the backbone and the head, both employing a fully convolutional neural network, CNN. In particular, YOLOv8 employs a novel backbone network, a modified version of CSPDarknet53 [93], comprising 53 convolutional (Conv) layers and incorporating cross-stage partial connections to improve information transmission across different network levels. In this backbone, the convolutional layers are stacked sequentially to progressively extract relevant features from the input image, and C2f (a new convolution layer replaces the YOLOv5's C3 layer) layers are introduced to integrate high-level features with contextual information for enhanced detection accuracy. The SPPF [96] (Spatial Pyramid Pooling Faster) module and subsequent convolutional layers process features at various scales [4, 95]. The head receives the feature maps from the backbone

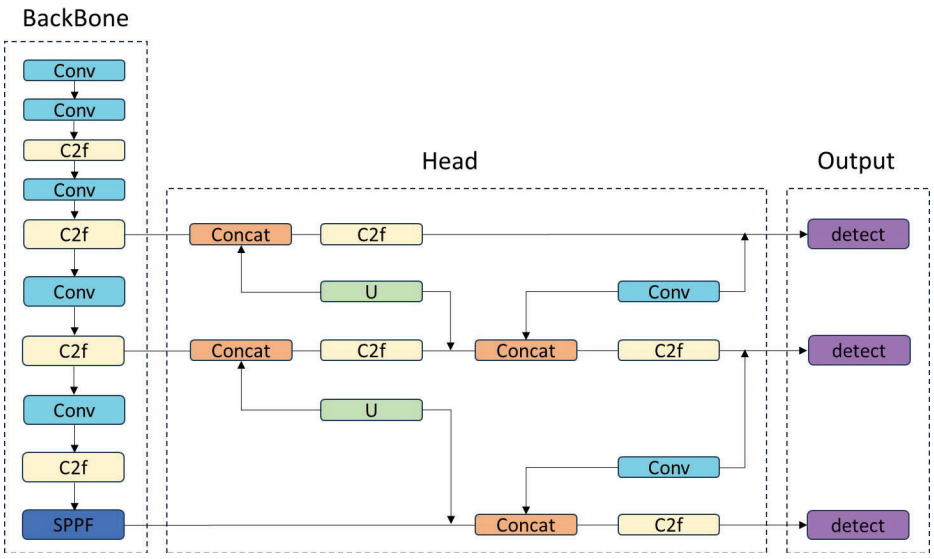


Figure 2.5: A Visualization of the YOLOv8 Architecture (Reproduced From [4]).

and performs further processing to produce the model's final output, *i.e.,* bounding boxes and object classes. YOLOv8 uses a detachable head to independently handle detection confidence scores, classification, and regression tasks. This allows each branch to focus on its specific task, enhancing the model's overall accuracy. The U (Upsample) layers depicted in Fig. 2.5 are used to increase the feature maps' resolution. The head also employs a sequence of convolutional layers for feature analysis, followed by a linear layer to predict bounding boxes and class probabilities. This head's design prioritises both speed and accuracy, carefully considering the number of channels and kernel sizes in each layer to optimise YOLOv8 performance [4, 95].

Moreover, YOLOv8 supports anchor-free detection, which means it directly predicts object centres, unlike earlier YOLO models (YOLOv5 and earlier) that relied on anchor boxes (a well-known challenge) [4]. This reduces the number of box predictions, therefore accelerating the post-processing step (Non-Maximum Suppression, NMS) for faster inference [97]. Real-time obstacle detection is vital for VIP outdoor mobility, and faster inference with YOLOv8 makes this possible. Real-time processing enables timely and accurate assistance, ensuring VIP safety. Due to its fast and accurate object detection capabilities, YOLOv8 is the chosen object detector for the scheme proposed by this thesis (discussed later in Chapter 4).

2.2.2 Object Tracking

Object tracking is a critical task in computer vision that detects objects and estimates their trajectories as they move in a scene [98]. Unlike object detection, object tracking maintains objects' identities and continuously updates their positions over time (set of a video's frames). It is vital for applications like visual surveillance, human-computer interaction, and vehicle navigation [99]. Object tracking is a mid-level task that supports high-level computer vision tasks like pose estimation [100], action recognition [101], and behaviour analysis [102]. There are two main types of object tracking techniques: single-object tracking (SOT), which focuses on tracking a specific object of interest, and multi-object tracking (MOT), which involves simultaneously tracking multiple objects in a given scene [103]. MOT has gained growing attention due to its ability to track and analyse multiple moving objects, a crucial capability for various computer vision tasks. MOT presents greater challenges than SOT due to the presence of occlusions, object interactions, appearance variations, and complex motion patterns. This requires robust algorithms capable of effectively handling such challenges [103]. Another challenge is that *online* tracking is crucial in many applications, where tracking is performed in real-time as the video streams. This means that the tracker should not rely on future frames to determine the current position of an object [99].

MOT generally consists of two main components: the *observation model* and the *tracking process* [104]. The observation model serves as a descriptor of objects, incorporating unique features that enable the distinction, detection, and tracking of individuals. Examples of such features are colour, shape, texture, and motion. The tracking process involves linking the observations to trajectories ('tracklets'). It can be divided into Bayesian theory-based tracking, *e.g.,* Kalman Filter, Particle Filter, Bayesian framework, and PHD Filter, and data association-based tracking, *e.g.,* Hungarian algorithm, Network Flow (NF), Maximum Weight Independent Set (MWIS), and Conditional Random Field (CRF)[104]. Based on how objects are initialised for tracking, MOT approaches can be divided into Detection-Based Tracking (DBT), also known as Tracking-By-Detection (TBD), and Detection-Free Tracking (DFT) [103]. In TBD, the objects are first detected, and then their trajectories are constructed. In DFT, a fixed number of

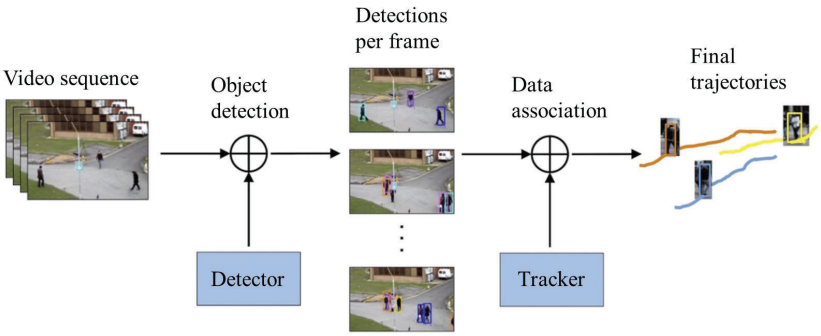


Figure 2.6: Tracking-By-Detection (TBD) Paradigm [5].

objects are manually initialised in the first frame, and then they are localised in the subsequent frames.

Due to advancements in object detection, Tracking-By-Detection (TBD) has become a popular paradigm for object tracking, achieving significant success [104]. TBD leverages DL-based object detectors, association methods, and appearance modelling to achieve accurate and robust multi-object tracking. In TBD, two primary steps are followed: object detection and data association [104], see Fig. 2.6. In object detection, a pre-trained CNN-based object detector, *e.g.,* YOLO, is employed to identify potential object locations in each frame of an input video. In data association, the detected object candidates are represented using an observation model, and then, the tracking process links observations with their objects and forms their trajectories. TBD algorithms for object tracking can be classified into SORT-based object tracking (e.g., SORT [81], Deep SORT [105], and Strong SORT [106]) and LSTM-based object tracking (e.g., RNN_LSTM [107], MHT_bLSTM [108], and BLSTM_MTP [109]) [110]. LSTM-based trackers excel at handling occlusions but sacrifice speed compared to SORT-based trackers, which prioritise real-time performance [110]. Real-time is an essential requirement for VIP assistance.

SORT [81] is a widely used algorithm that combines object detection and tracking to achieve real-time multi-object tracking. It comprises four components: detection, estimation model, data association, and creation and deletion of track identities [81]. For detection, SORT utilises the Faster R-CNN two-stage object detector [88] to initialise its tracking process; however, it is adaptable to various object detectors. The estimation model receives detected objects and predicts their positions (targets) in the next frame based on a linear constant velocity model. If

detection is associated with a target (tracked object), the bounding box of the detected object is utilised to update the target state, where the velocity components are optimally solved using the Kalman Filter [111]; otherwise, no corrections are applied. Data association involves associating detections with existing tracks using the Hungarian algorithm [112] and rejecting associations below a certain IOU (Intersection Over Union) threshold. This contributes to solving the occlusion problem and maintaining track IDs. The creation and deletion of track identities manage the creation of new IDs and termination of existing IDs based on detection associations and the duration of track loss. Despite good overall precision and accuracy, SORT suffers from highly frequent identity switches. These highly switches can lead to inaccurate tracking and compromised performance, especially in the presence of occlusion [110]. Deep SORT extends the SORT algorithm by introducing a deep appearance descriptor and a more sophisticated association metric that combines both motion and appearance information. This allows Deep SORT to achieve a more reliable and precise multi-object tracking approach than SORT; it successfully tracks objects through longer occlusion periods and significantly reduces the number of identity switches by 45% [105]. Strong SORT [106] optimises Deep SORT by incorporating AFlink and GSI modules to further improve tracking accuracy (reducing identity switches by 20.59%); however, it comes at the cost of processing speed. It experiences a significantly decrease in speed of around 46% compared to Deep SORT, posing a challenge in meeting the requirements of real-time applications [110]. Deep SORT strikes a balance between accuracy and speed, making it a suitable choice for assisting VIPs. Therefore, this thesis employs Deep SORT for object tracking.

2.2.3 Distance Estimation

In CV, distance estimation refers to estimating the physical distance between a sensing method (such as a camera) and objects in a scene. Distance (depth) information plays a crucial role in inferring the 3D structure of an environment, making it essential for various tasks in robotics and autonomous systems [113]. Several methods are employed to estimate distance, depending on the available data and the application's requirements. These methods can be classified into

active and passive methods [114]. Active distance methods directly acquire depth information from a scene by emitting a signal, like light or sound, towards obstacles. The reflected signal is then detected and used to calculate the distance based on the *time-of-flight* principle. This principle measures the time it takes for the signal to travel to an obstacle and reflect back, which is directly proportional to the distance [115]. Examples of active methods are laser [116], ultrasound [117], radio signals [118], infrared light [119], and RGB-D cameras [120]. However, active methods are constrained by their limited range and susceptibility to environmental interference [113].

Passive methods leverage computer vision techniques to infer distance by analysing existing environmental information, *e.g.,* object size and motion, in images captured using RGB cameras. These methods can be categorised into mono-vision (monocular vision), which relies on a single camera [121] and stereo-vision, which utilises dual cameras separated by a small distance [122]. Stereo-vision, also known as binocular vision, mimics the human visual system's depth perception by utilising two cameras placed at a small distance from each other. These cameras capture images from slightly different perspectives. Stereo-vision compares disparities in pixel positions between corresponding points in the two images using techniques such as Block Matching algorithms [123] and then computes the depth of objects in the scene using the formula: $f\frac{b}{(x_l - x_r)}$, where x_l and x_r are corresponding points in the left and right images, b is a stereo camera's baseline (the distance between the two cameras), and f is the focal length of a camera [124]. Depth estimation from stereo images is a common technique, with many methods (e.g., [125, 126, 127]) relying on stereo matching. However, in stereo-vision, the depth estimation accuracy diminishes when the distance between the stereo cameras and an obstacle increases [113]. This limitation arises from the inherent challenge of matching corresponding points on distant objects due to the reduced parallax effect (the apparent shift in position) caused by the limited baseline distance between the two cameras. Moreover, collecting stereo images requires complicated procedures for alignment and calibration [113].

The advancement in computer vision has made mono-vision more convenient than stereo-vision [113]. Monocular depth estimation algorithms can be categorised based on their underlying principles [113]: Structure from Motion (SfM) [128, 129], traditional handcrafted feature-based

techniques [130, 131], and DL-based methods [132, 133, 134, 135]. DL-based methods demonstrate prominent performance in recovering depth information from single images. This advantage can be attributed to the superior capability of NNs in modelling complex relationships within monocular cues compared to other methods [113]. In [132], a Structure-Aware Residual Pyramid Network (SARPN) is proposed for accurate monocular depth estimation. SARPN incorporates a residual pyramid decoder to predict multiple-scale depth maps while considering the underlying hierarchical scene structures. It progressively adds finer structures at each specific scale while maintaining the coarser layout predicted at the higher levels. SARPN also uses an adaptive dense feature fusion module. This module adaptively fuses the image features from all scales when predicting the residual depth map for each level, contributing to a more accurate prediction. Similar efforts have been made in [133, 134] to improve the accuracy of depth map prediction. While these methods [132, 133, 134] achieve a high-depth prediction accuracy, their complexity translates to high power consumption, making them unsuitable for real-time applications [136].

To address this, research has proposed various methods prioritising both speed and accuracy. In [137], a novel lightweight architecture, PyD-Net, is designed for fast and accurate depth map inference by leveraging a pyramid of features extracted from a single input image. At an image resolution of 128×256, PyD-Net achieves real-time inference with a processing time of only $0.020s$ ($\leq 0.033s$, see the real-time requirement in Chapter 1) running on NVIDIA GeForce GTX Titan X (Maxwell) GPU and a mean absolute error of $0.146m$ (evaluated on KITTI dataset [138]). Another real-time approach is proposed in [139]. Running on an NVIDIA GeForce GTX 1080 Ti, this method achieves real-time performance ($44fps$) with an average processing time of $0.023s$ with a mean absolute error of $0.110m$ on KITTI [138] and $0.423m$ on Make3D [140]. Limited by the KITTI dataset's $80m$ maximum depth, network predictions in [137] and [139] are capped at this value during training. However, in [137], network performance is evaluated only up to $50m$, while in [139], evaluation errors are computed when the depth is under $70m$. Although both methods achieved high accuracy at distances shorter than $70m$, their performance at larger distances needs to be verified. In [7], a novel lightweight neural network, DisNet, is proposed for long-range distance estimation. It is specifically designed to enhance

long-range obstacle detection in railway applications. DisNet has three hidden layers with 100 neurons per layer; it takes in six features through its input layer and predicts the distance to the object in the image using a single output neuron. While DisNet performs the core distance estimation using its NN architecture, it relies on YOLO as a preprocessing step, i.e., the six input features (the ratios of the object bounding box dimensions to the image dimensions) are calculated based on YOLO's generated bounding boxes. The evaluation results of DisNet demonstrate its effectiveness and accuracy in long-range distance estimation (*e.g.,* achieving an accuracy of 99.44% and 87.16% with GT distances of $100m$ and $300m$, respectively). This long-range and accurate capability makes DisNet a particularly suitable choice for this thesis, which aims to identify different classes of CMOs (cars, motorcycles, bicycles, and pedestrians) moving at low-to-high speeds while providing enough time for VIPs to avoid such threats safely and in-time.

2.3 Path Planning

Path planning is a fundamental problem in various domains, including robotics, autonomous vehicles, and VIP navigation assistance [141]. Generally, path planning involves finding an optimal or near-optimal path from a starting point to a destination while considering constraints such as obstacles, terrain, or dynamic environmental conditions. It aims to facilitate safely navigating through obstacles, minimising travel time, energy consumption, or other defined metrics (*e.g.,* path smoothness [142]). Based on the availability of environmental information, path planning can be divided into global path planning and local path planning [6].

Global Path Planning. Global path planning requires a complete knowledge of the environment (including obstacles) before initiating the search for an optimal path. It typically includes a prior phase known as environmental modelling [6]. In this phase, a comprehensive representation of the environment is constructed using various forms such as grids, graphs, polygons, or points. This is to facilitate understanding of environmental characteristics, thus reducing unnecessary planning and computational burden [6]. Common methods utilised for en-

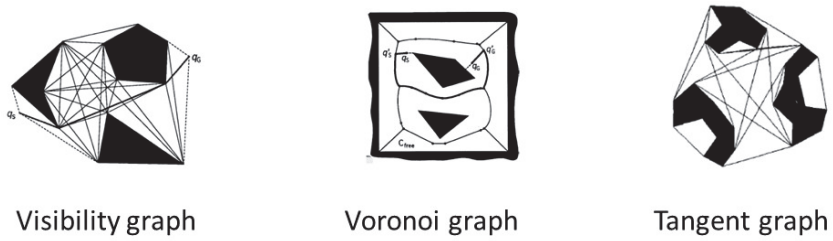


Figure 2.7: Examples of Framework Space Approach [6].

vironmental modelling are the framework space approach (*e.g.,* visibility graph, voronoi graph, tangent graph, see Fig. 2.7), free space approach, cell decomposition approach, topological method, and probabilistic roadmap method [6]. Global path planning algorithms generally fall into two categories [6]: heuristic approaches [143, 144] and artificial intelligence (AI) algorithms [145, 146, 147, 148, 149]. Dijkstra [143] and A* [144] algorithms are the most common heuristic approaches to finding the shortest path in graphs representing the environments. The graph is a mathematical representation of a set of nodes and edges. Nodes represent entities, such as locations or obstacles, while edges represent the connections between these entities. AI-based global path planning utilises various methods, including ANNs [145], Genetic Algorithm (GA) [146], Ant Colony Optimisation algorithm (ACO) [147], Particle Swarm Optimisation (PSO) [148] and Simulated Annealing (SA) [149]. While global path planning is beneficial, it fails in dynamic environments as it assumes a static and pre-known environment, but real-world scenarios often involve changes in obstacles and conditions [6]. This reliance on static information can lead to planning an entirely unusable path. Furthermore, global path planning can be computationally expensive, especially in complex environments [6]. As the environment grows larger, the number of possible paths and the calculations required to find the optimal path increase exponentially, leading to significant delays in planning and resource limitations. This becomes a critical issue for real-time applications like VIP mobility assistance, where a quick response is essential. When navigating dynamic environments, adaptable and efficient real-time path planning is crucial for VIP safety.

Local Path Planning. Local path planning offers greater flexibility than global path planning. In local path planning, the agent has limited or no prior information about the environment before beginning the planning process [150]. Local path planning involves dynamic environ-

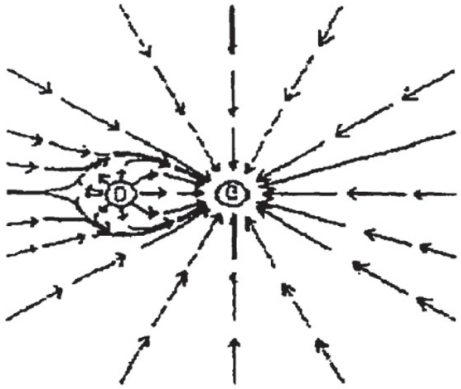


Figure 2.8: The Scheme of Artificial Potential Field (APF) [6].

mental modelling, often using sensors to gather the necessary real-time information within the surroundings. This enables real-time adaptive path planning in response to dynamic environmental changes. An example of local path planning is Artificial Potential Field (APF) [151], a popular algorithm in robotics and autonomous systems inspired by the concept of potential field in physics. APF assigns "potentials" to each point in the environment, attracting the agent towards the goal and repelling it from obstacles, see Fig. 2.8 ('G' represents the goal and 'O' represent an obstacle). This effectively guides the agent to its destination while avoiding collisions. The agent moves by following the gradient of the potential field; it navigates through low-potential areas and avoids high-potential regions. While APF offers an effective approach in robotics, its traditional form can struggle in highly dynamic environments due to local minima (have areas of low potential that trap the agent, preventing it from reaching the goal) and oscillations (the agent might oscillate between potential values, leading to inefficient movement). Several advancements (e.g., [152, 153]) have been made to adapt APF and address such limitations. However, the inherent repulsive force in APF typically increases as the robot nears obstacles (neglecting obstacles velocities and only reacting to their location). This can cause sharp movements (or 'shocks') near obstacles, potentially leading to navigation issues [6]. In [154], a Velocity Obstacles (VO) local path planning algorithm is proposed for robots. VO is specifically designed to ensure collision-free velocities for a moving robot operating in dynamic environments. By considering the velocities and predicted future positions of both the robot and other moving obstacles, VO calculates feasible velocities that enable the robot to avoid po-

tential collisions effectively. More examples of robot local path planning algorithms include the Behavior Decomposition method [155], the Cased-Based Learning method [156] and the Rolling Windows algorithm [157]. While local path-planning algorithms have succeeded in robotics, their direct application to VIP navigation presents significant challenges. Unlike robots, VIPs possess varying abilities (e.g., auditory perception) and preferences (e.g., prioritising physical safety and requiring ample time to perceive their surroundings and navigate safely), making it challenging to directly apply a one-size-fits-all approach borrowed from robotics. Therefore, this thesis aims to develop a novel local path planning scheme that complies with VIPs' mobility and perception constraints and prioritises VIPs' safety by considering CMOs' types, CMO's movement pattern, and predicted collision times and conducting a look-ahead safety prediction of the optimal path.

2.4 Optimisation

The VIP local path planning proposed in this thesis is an optimisation problem; it aims to find the shortest collision-free paths for VIPs to navigate around obstacles (CMOs) safely. Therefore, this section provides a brief overview of the optimisation concept, explores different classifications of optimisation problems, and discusses available methods for solving them, particularly focusing on non-convex problems as the thesis's path planning problem is non-convex.

Optimisation is a fundamental concept in mathematics, computer science, and engineering that aims to find the best possible solution for a given problem [158]. Typically, it involves maximising or minimising an objective function while satisfying a set of constraints, known as *constrained* optimisation. The objective function represents the quantity to be optimised, such as profit, efficiency, or accuracy, while the constraints define the limitations or requirements that must be adhered to. Not all optimisation problems have constraints, known as *unconstrained* optimisation; in those cases, the focus is solely on optimising the objective function without any restrictions. Optimisation problems arise in various fields, including operations research, finance, and machine learning [158]. A general formula for a basic optimisation problem can

be expressed as follows [158]:

$$\begin{aligned} &\min_{x} \ f(x) && (2.1)\\ &\text{Subject to} \\ &g_i(x) \leq 0. \qquad \text{for } i = 1, 2, ..., m \\ &h_j(x) = 0, \qquad \text{for } j = 1, 2, ..., n, \end{aligned}$$

where $f(x)$ represents the quantity to be minimised or maximised ('min' can be replaced with 'max' in the above equation), it can be any mathematical expression that depends on one or more decision variables x. The functions $g_i(x)$ and $h_j(x)$ represent the inequality and equality constraints on the values that the variables x can take, respectively.

2.4.1 Classification of Optimisation Problems

Optimisation problems can be categorised based on the properties of their objective functions and constraints. The following section explores two key classifications of optimisation problems.

Linear vs. Non-Linear Optimisation Problems. Optimisation problems can be classified into linear and non-linear problems [158]. In linear problems, the objective function, $f(x)$, and the constraints, $g_i(x)$ and $h_j(x)$, are all linear functions of the decision variables x, see Eq. (2.1). Such problems can be efficiently solved using a variety of techniques, including the well-established Simplex method [159]. For large-scale linear problems with numerous variables and constraints, powerful software solvers such as CPLEX or Gurobi offer a more efficient solution approach [160]. A key advantage of linear optimisation problems is that they are guaranteed a unique optimal solution (assuming a feasible region exists) [158]. Conversely, optimisation problems become non-linear when at least one of the functions $f(x)$, $g_i(x)$ and $h_j(x)$ is a non-linear function of x. Compared to linear optimisation problems, solving non-linear problems can be more challenging due to the complexity introduced by non-linearities, and they often lack a straightforward solution method [158].

Convex vs. Non-Convex Optimisation Problems. Convexity is a broader property related to a problem's shape and structure. Mathematically, it implies that the function lies above its tangent lines at any point within the feasible region (the set of valid solutions) [158]. This translates geometrically to a function that curves inward, like a bowl. Convexity is a desirable property as it guarantees the existence of a globally optimal solution and enables the use of efficient solution methods [161, 158]. Based on convexity, optimisation problems can be categorised into convex and non-convex problems [158]. In convex optimisation problems, all the functions $f(x)$, $g_i(x)$ and $h_j(x)$ in Eq. (2.1) are convex. These problems can be efficiently solved using various algorithms, such as Newton methods [162] and Interior-Point methods [163], and guarantee a single global optimum. Non-convex optimisation problems arise when any of the functions $f(x)$, $g_i(x)$ or $h_j(x)$ are non-convex. In such problems, the feasible region defined by these functions can have shapes that deviate from convexity. Non-convexity introduces complexities to such problems, e.g., multiple local optima and saddle points [164], making it difficult to guarantee finding the global optimum. Therefore, algorithms developed for solving convex problems are not directly applicable to non-convex problems [158].

Linear problems are a special case of convex optimisation problems. This means that all linear optimisation problems are inherently convex. Non-linear optimisation problems, on the other hand, can be either convex or non-convex based on the nature of the objective function and constraints [158]. Understanding the classifications of optimisation problems enables the suitable selection of tailored solution methods, leading to desirable results. Many real-world problems are inherently non-linear in nature [165], i.e., their solutions cannot be found with simple linear relationships. The VIP local path planning that the thesis tries to address is a non-linear (i.e., non-convex) problem, discussed later in Chapter 4. The following sections will discuss common methods used to address non-convex optimisation problems.

2.4.2 Solving Methods for Non-Convex Optimization Problems

One particularly effective methodology for tackling non-convex optimisation problems is approximating the original problem with a series of solvable sub-problems, where these sub-problems

are formulated to closely resemble the original problem within a specific region of the parameter space [158]. By iteratively solving these sub-problems and leveraging the solutions as starting points for subsequent ones, a refined solution can be gradually achieved, potentially converging towards an optimal solution for the original problem. Common approximation approaches that are widely used for solving non-convex problems are Sequential Linear Programming (SLP), Sequential Quadratic Programming (SQP), and Sequential Convex Programming (SCP) [166].

In SLP, the non-convex problem is approximated by a sequence of linear subproblems. SLP employs Taylor expansions, typically first-order, to linearise the non-convex objective function and constraints around an initial guess (starting point). While SLP simplifies the computational complexity of non-convex problems, it can suffer from slow convergence. This may necessitate many iterations to converge to an optimal solution for the original non-convex problem [167]. SQP refines SLP's approximation approach by employing quadratic subproblems. Unlike SLP's linear approximations, SQP leverages second-order Taylor expansions for the objective function, capturing both its slope and curvature. While constraints in SQP might be approximated with linear expansions, focusing on a higher-order objective function provides a more accurate representation of the original problem than SLP, potentially leading to solutions closer to the optimal solution. SQP algorithms have proven successful in solving various nonlinear problems [168, 169, 170]. However, they are highly sensitive to initial guesses (a poor initial guess can significantly impact the solution or even lead to convergence failure) and can be computationally expensive when solving complex problems [171]. SCP has become increasingly popular due to its efficiency in handling diverse non-convex problems, such as trajectory optimisation [172], rocket landing [173] and spacecraft control [174]. The SCP approximates an original non-convex problem into a sequence of solvable convex sub-problems using approximation methods like Taylor expansions and Particle method [158]. Compared to SQP, SCP is more efficient; it requires fewer iterations to find a solution and is less sensitive to initial guesses [175]. SCP's ability to efficiently handle non-convex problems by finding reliable solutions with fast convergence makes it suitable for solving the non-convexity of the VIP local path planning proposed in this thesis.

In conclusion, this chapter has established a clear foundation for understanding the challenges

faced by VIPs and available opportunities to develop assistive solutions. Specifically, it explored visual impairment's prevalence and impact on daily life and discussed cognitive mapping for VIPs to provide a deeper understanding of how they perceive their environment. Additionally, it provided an overview of fundamental computer vision and deep learning techniques essential for crafting vision-based VIP assistive solutions. The following chapter will critically assess research concerning VIP mobility assistance and robot local path planning.

Chapter 3

Related Work

After establishing the necessary foundational knowledge of this thesis in the previous chapter, this chapter delves into a critical review of existing research in two key areas: VIP mobility assistance and robot local path planning. This review aims to identify current solutions, their strengths and limitations, and potential gaps that inform the development of the the approach proposed by this thesis.

3.1 Mobiltiy Assistant Aids for VIPs

Vision, along with other senses, enables individuals to understand their surroundings comprehensively and to act accordingly. Vision loss can profoundly impact daily life by hindering mobility, communication, and access to information [13, 176]. Generally, studies related to VIPs have focused on three key domains: transmission of information, computer accessibility, and mobility assistance [176]. Compared to the other domains, assisting VIPs with mobility is more challenging. It involves three difficult related issues: obstacle avoidance, navigation and way-finding (orientation), and space perception [176]. In the past few decades, numerous electronic travel aids (ETAs) have been proposed to surpass the limitations of conventional tools, *i.e.,* white canes and dogs, and facilitate safe and independent mobility for VIPs. These EATs utilise various sensors to detect and locate obstacles in VIPs' surroundings, providing

them with a sense of the external environment [177]. Despite the sensors used, they all operate on the same principle: scanning the environment and relaying the gathered information to other VIPs' senses, primarily hearing, touch or both, as a substitute for vision. ETAs can be generally classified into non-vision-based and vision-based mobility aids. This research focuses on works related to VIPs' mobility assistance domain.

3.1.1 Non-Vision-Based Mobility Aids

To collect environmental information, non-vision-based mobility aids utilise *active* sensors like ultrasound [178, 179, 180, 181, 182, 183], infrared (IR) [184, 185], laser [185, 186], radar [187] and radio frequency identification (RFID) [188, 189, 190, 191].

RFID uses radio waves to identify and track objects or people automatically. It has been widely used to facilitate indoor navigation and assist VIPs in various scenarios. For example, in [188], a robotic shopping assistant, RobotCar, is developed to help VIPs navigate a typical grocery store and carry purchased items. It utilises RFID tags for localisation purposes, which are deployed at different locations within the store, and laser range finding for navigation. Similarly, in [189], a mobile solution, BlindShopping, utilising RFID and QR-code technology, is presented to enable accessible shopping for VIPs. In [190], RFID technology has also been utilised for self-guided museum tours tailored to the needs of VIPs. However, RFID systems have significant limitations that restrict their widespread adoption for VIPs' mobility assistance. One key limitation is the need for prior knowledge of a building's architecture [192], which restricts their adaptability to different environments, especially outdoor environments, as they are vast and dynamic. Further, RFID tags must be installed extensively throughout the surroundings to ensure effective assistance, which can be challenging, costly and time-consuming. Finally, the restricted read range of RFID systems requires VIPs to be close to the reader for successful communication, limiting their mobility and independence.

Many alternative technologies have been used to assist VIPs in effectively avoiding obstacles during their mobility. In [184], an IR-based intelligent assistance system was developed to aid VIPs in avoiding front aerial obstacles and detect cases of VIP falls. The system includes

wearable glasses, an intelligent walking stick, and a mobile app. Equipped with an IR sensor module, the glasses detect obstacles and transmit a Bluetooth signal to the walking stick, which alerts the VIP through vibrations. Additionally, a LoRa-based network is used to notify the VIP's family members in the event of a collision or fall. However, the system's limitation is the inability to detect typical obstacles moving on the ground, which can be a significant safety hazard for VIPs. In [185], the IR-based device, Tome Pouce, is proposed as a complementary tool to assist VIPs alongside white canes. It enhances VIPs' safety by detecting obstacles within a protected field extending from the knees to the top of the head, utilising collimated LEDs and detectors to sense IR beams. The device can detect obstacles up to $3m$ away with a maximum error of 20% in distance estimation. However, external sources of IR radiation, such as the sun or other IR sensors, can interfere with the IR sensors, resulting in weakened or distorted signals. This hampers the effectiveness of obstacle detection and distance estimation in IR-based devices.

In [185], the authors improved IR-based obstacle detection by integrating a laser module that emits a focused beam of light; this results in the Teletact system. Teletact extends the detection range to up to $15m$ and provides precise proximity information within the $10cm$ to $10m$ range with a maximum error of only 1% [185]. A similar improvement was made in [186] to the conventional Tome Pouce [185], resulting in Tome Pouce III. However, the highly directional nature of the laser beam (maintains a narrow path without significant divergence) requires VIPs to actively scan their surroundings to detect obstacles effectively [193]. Neglecting this scanning process may lead to overlooking potential hazards outside the too-narrow direct path ($< 0.5m$ wide [194]) of the laser beam. In [187], a radar-based tool is developed to assist VIPs in unfamiliar environments. It uses radar technology to perceive the surroundings and provides acoustic feedback based on distance measurements. The system incorporates an 80 GHz FMCW (Frequency Modulated Continuous Wave) radar with a 24 GHz bandwidth. While effective for wayfinding, the bulky design of the antenna presents a significant challenge [177].

Ultrasonic sensors are widely preferred in VIP mobility assistance solutions due to their superior accuracy compared to other sensors [177, 180]. They have been demonstrated to reliably measure distance and angle with minimal errors and variances [195]. In [178], conventional

eyeglasses were equipped with a pair of integrated ultrasonic sensors. These sensors emit a down-swept frequency-modulated (FM) ultrasound signal ($40 \sim 70kHz$ within $1ms$) to detect obstacles VIPs encounter. The reflected signals are processed and converted into stereo-audible sound, providing VIPs with the spatial perception of obstacles through headphones. The FM ultrasound-based system successfully detects even small obstacles, such as a 1-mm-diameter wire. However, background noise interference can significantly affect the system's performance in noisy outdoor environments. Additionally, the downward sweeping of ultrasound frequency limits the system's range, making it less effective for long-distance object detection.

Further, to enhance the capabilities of traditional white canes for VIP navigation, electronic canes like GuideCane [179], SmartCane [180], UltraCane (BatCane) [181, 182], and Necklace cane [183] were introduced. These devices utilise ultrasonic sensors to detect obstacles at greater distances than white canes. GuideCane [179] is a robotic cane with a thumb-operated joystick, wheels, a steering mechanism, and multiple ultrasonic sensors positioned in a semi-circular pattern covering a wide sector of 120°. When an obstacle is detected, the cane's obstacle avoidance algorithm automatically steers the wheels to avoid it. SmartCane [180] integrates a built-in computer and uses ultrasonic sensors to detect obstacles. It provides feedback to the VIP through different types of beeps that indicate the distance to the obstacles: far ($> 1.2m$), medium ($1.2 \sim 0.9m$), and close ($< 0.9m$). Ultracane [181, 182] incorporates an integrated ultrasonic sensor capable of detecting obstacles in both short-range (up to $2m$) and long-range (up to $4m$) modes. It can also detect obstacles above the wrist at approximately $1.5m$. Feedback is conveyed to the VIP through vibrations. The Necklace [183] is a neck-worn ultrasound-based device designed to complement other canes by detecting overhead obstacles (in the area directly in front and above the VIP's head). However, one limitation of these electronic canes is their reliance on ultrasonic sensors, which can be affected by environmental factors, *i.e.,* noise and acoustic interference. In noisy environments, the performance of the ultrasonic-based systems can be significantly degraded, leading to inaccurate obstacle detection. Besides, ultrasonic sensors have a limited range; this can pose serious risks for VIPs, as such sensors cannot detect serious threats (*e.g.,* a car moving at high speed) located beyond their detection range.

In conclusion, non-vision methods have demonstrated notable potential in enhancing the mo-

bility of VIPs. However, these methods have a common fundamental limitation. They can only detect the presence of obstacles and do not provide any additional information about the obstacle type or, more importantly, whether the object is moving or stationary. A better solution would allow VIPs to fully perceive their surroundings and identify threats. This would involve differentiating between obstacles of different risk levels and various approaching moving objects. The solution would also indicate where VIPs need to move for self-protection.

3.1.2 Vision-Based Mobility Aids

Current research on VIP mobility assistance focuses on vision-based solutions, which can capture richer information than non-vision-based methods. These solutions typically use computer vision algorithms, such as CNN-based classifiers, to analyze images and videos captured by different types of cameras, such as stereo [196], RGB [197, 198], and RGB-D cameras [199, 200].

In [199], a VIP navigation assistance system called NAVI is proposed. It uses a low-cost RGB-D camera to capture range information from a built-in IR-based depth sensor and visual information from the camera's RGB sensor. These two types of information are fused to detect major structural elements, such as walls and floors. The system can accurately segment floors with 99% precision. However, it only works for static structural obstacles in indoor environments. In [200], a CNN-based road barrier recogniser called KrNet is proposed to aid VIPs in outdoor environments. KrNet is designed to detect road barriers, such as those found at the entrance and exit of construction areas and car parks. KrNet integrates a smart glass with an RGB-D sensor to capture images with depth information. When barriers are detected in a specific image within a range of $20m$, the VIP is alerted through headphones and guided towards an intended destination. Despite running on resource-limited mobile devices, KrNet's capability is restricted to detecting only a single class of objects.

In [197], an obstacle detection method called Deformable Grid (DG) is proposed to prevent collision for VIPs. DG is a grid that can be deformed gradually according to the movement of objects in a scene. When there are no objects present, DG returns to its regular shape. DG is integrated into a glasses-type wearable device equipped with an RGB camera and Bluetooth

headphones. The risk of collision is detected by measuring the degree of deformation, and corresponding audio feedback is generated. In [198], the authors improved the accuracy and processing speed of DG by using a vertex deformation function. However, both methods cannot distinguish between different types of obstacles and their risk level. They only emit a signal to indicate the presence of an incoming object, i.e., the signal's temporal frequency increases as the object gets closer to the VIP. This can lead to false alarms, which needlessly unnerve VIPs.

In [201], a deep learning-based object recognition system is presented for assisting VIPs in outdoor environments. The system involves capturing video frames using a smartphone; the frames are then uploaded to a CNN-based object classifier on a Cloud. The classifier identifies the objects in the scene and sends the information, such as the object class, to the VIP's headphones. The system can accurately identify 11 common object classes, such as cars, motorcycles, bicycles, and pedestrians, with a 96.4% recognition rate. However, it is limited to object classification only and does not provide depth information (the proximity of the objects to the VIP) on the detected objects. This means the system cannot predict potential collisions that threaten the VIP's safety. In [202], a smartphone-based system is proposed to detect and classify obstacles for VIPs in both indoor and outdoor environments. The system uses the Scale Invariant Feature Transformation (SIFT) and the Speed-Up Robust Feature Extractor (SURF) methods to extract points of interest in the captured frames. These extracted points are then tracked using the Lucas-Kanade algorithm and classified into their corresponding categories, such as vehicles, bicycles, pedestrians, and static obstacles, using the Support Vector Machine (SVM) and the Bag of Visual Words (BoVW) models. The system also provides two levels of obstacle risk: urgent and normal. The risk level is determined based on the obstacle's relative position and movement direction (*i.e.,* whether it is approaching or departing from the camera focus). However, the complexity of this system results in a limited processing speed of approximately $7fps$ (based on the equivalent hardware). When processing video clips with a typical frame rate of $30fps$, this method misses a significant number of essential frames, potentially compromising VIP's safety. For example, when a car approaches a VIP at $54km/h$, it travels $15m$ in one second. If the detection system only operates at $7fps$, crucial frames showing the car's proximity to the VIP might be missed due to frame gaps. This delay in detection

significantly reduces the VIP's reaction time to avoid the approaching hazard.

In [203], a real-time assistive solution is proposed to detect the presence of moving objects along with their trajectories, *i.e.,* upward, downward, forward, backward, left-to-right, and right-to-left. However, this solution is limited to identifying the direction of objects' motion and cannot recognise such moving objects. This can hinder the effectiveness of potential object-VIP collision prediction. In [204], a simple mobile application for real-time object identification is introduced to improve VIPs' perception of ambient objects. By capturing photos of the surroundings, the application informs the VIP about the class and absolute location of the recognised objects in the captured frame. However, the application does not estimate the proximity of the objects relative to the VIP and, thus, cannot predict collisions, posing a significant risk to the VIP's safety.

In [205], a deep learning-based obstacle detection system is proposed to improve VIPs' free-collision mobility in outdoor environments. The system uses a smartphone camera to capture images of the surrounding environment continuously. The images are then fed to a DeepLabV3 semantic segmentation model for real-time identification of various objects. The system also estimates the distance to the recognised objects through triangulation. It offers two modes: stable mode and fast mode. In stable mode, the navigation system prioritises accuracy and provides detailed voice instructions to help VIPs familiarise themselves with unfamiliar surroundings. Conversely, in fast mode, designed for crowded environments with a high collision risk, the navigation system emphasises speed and reports more real-time detected objects. For comprehensive perception, voice feedback is delivered to the VIPs, including information about the obstacles' types, directions, and distances. However, the system's distance estimation based on triangulation is inaccurate and highly error-prone [206, 207]. Triangulation-based distance estimation becomes unreliable, especially for large distances, because small triangulation errors translate into large distance errors [208]. In [209], a portable real-time system is proposed to assist VIPs in perceiving objects and people around them and estimating their distance. The system utilises a CNN-based object detector, YOLOv3, with a single camera mounted on a Raspberry Pi board. The system can detect people and predict their distance with a 98.8% accuracy rate. However, the system's distance estimation is limited to a maximum range of

$10m$ and is only designed for pedestrians. This means the system may not accurately estimate distance for objects other than pedestrians or distances beyond the specified limit.

In conclusion, current research on VIP mobility assistance has emphasised vision-based solutions, which use computer vision algorithms to analyse images and videos captured by various cameras. While these approaches have shown potential in assisting VIPs, they also have certain practical limitations. These limitations vary among limited object identification accuracy, limited functionality (e.g., single object class detection, lack of proximity estimation or shortcomings in distance estimation, or inability to recognise risk levels), or failure to operate in real-time. Such limitations make the proposed VIP solutions fail to effectively detect high-risk moving objects such as fast cars or motorcycles. There is a need for further advancements in VIP mobility assistance systems to address these challenges and provide more practical solutions. Developing systems that can effectively and efficiently detect moving objects, classify their types, estimate their proximity, predict potential collisions, and assess objects' risk levels is crucial. By doing so, the safety and independence of VIPs can be enhanced, empowering them to navigate their surroundings confidently.

3.2 Local Path Planning

This section reviews previous work on local path planning, a well-studied topic in other fields, such as robotics and autonomous vehicles. Local path planning allows for quick and effective responses to dynamic changes and obstacles, making it an essential component of any mobile system that needs to navigate in a dynamic environment. This thesis discusses robotics and autonomous vehicles' local path planning, as current research on VIP mobility assistance, to the best of the researcher's knowledge, lacks local path planning algorithms designed explicitly for VIPs.

In [210], a collision avoidance approach for Unmanned Aerial Vehicles (UAVs) is introduced. The approach combines Simulated Annealing (SA) with a simple threat-avoidance method to plan a near-optimal path within a 2D radar environment constrained by regular circular threats.

The SA algorithm is employed to find the best solutions, which are then filtered by the threat-avoidance method to remove forbidden paths that may result in collisions. The resulting path is the shortest collision-free route between a source and target position. The study demonstrates that SA-based solutions are effective in escaping local minima and generating threat-free paths. However, the algorithm exhibits slow convergence, leading to high computation times. For example, using an Intel Core2 Duo P8600 2.4GHz CPU, it requires $921,030$ iterations and $9.218s$ to find the best solution in a 2D radar map with 10 threats. Additionally, when producing a threat-free path, the approach solely considers the circular areas occupied by the threats, ignoring their motion characteristics such as speed and moving direction.

The Velocity Obstacle (VO) algorithm, presented in [211], is designed for motion planning of robots operating in dynamic environments. This algorithm defines a set of robot velocities that would result in a collision with an object moving at a known speed. Through filtering, the algorithm identifies feasible velocities that allow the robot to avoid collisions with the obstacle, considering the robot's acceleration constraints. VO has been shown to be effective in avoiding dynamic obstacles, and several improved versions have been proposed. These versions include Reciprocal Velocity Obstacle (RVO) [212], Hybrid Reciprocal Velocity Obstacle (HRVO) [213], Optimal Reciprocal Collision Avoidance (ORCA) [214], and ClearPath [215]. However, applying VO-based methods to VIPs poses challenges due to the inability to control their speeds definitively. The unpredictable and variable nature of VIPs' speeds limits the application of such solutions. In [216], researchers propose an algorithm based on the Artificial Potential Field (APF) for planning a safe path in a dynamic environment for robot soccer. A similar approach is presented in [217] for vehicles. The APF algorithm is inspired by the physical concept of a potential field, where the robot is influenced by two forces: the attractive force towards the destination and the repulsive force from obstacles. This algorithm is widely used in robotics due to its simplicity and real-time capabilities. However, a limitation of the APF algorithm is its susceptibility to failure when the magnitudes of the repulsive and attractive forces become equal, causing the autonomous agent never to reach its intended destination.

In [218], a human-centred X-Y-T space path planning approach is presented to assist robots in safely navigating dynamic environments while also ensuring free space for pedestrians nearby.

The method utilises a 2D grid-based representation of the surroundings to generate collision-free motion paths, considering the robot's dynamic constraints and humans' personal space and directional area. The path is continuously updated in real-time to adapt to environmental changes and human motion. However, one drawback of this approach is the storage space required to store the environmental information in the grid-based map, which can significantly impact the algorithm's efficiency and scalability when applied to more extensive or diverse environments. In [219], a path planning algorithm called GA3C-CADRL is introduced, utilising Deep Reinforcement Learning (DRL) to aid robots in safely navigating through crowded areas with pedestrians. This algorithm extends the solutions introduced in [220] and [221] to learn collision avoidance strategies without making assumptions about specific behaviour rules followed by pedestrians. However, a limitation of the DRL-based approach is the requirement for a large-scale dataset during the training phase. Generating such datasets can be challenging as it involves extensive random simulation scenarios, and the training phase itself consumes considerable resources and time. Additionally, while the algorithm performs well on training data, it cannot guarantee the same level of effectiveness when applied to real-world data that differs from the training data.

Typically, algorithms designed for autonomous vehicles or robots are customised to accommodate their specific physical constraints, kinematics, limited movement capabilities, and lack of cognition, increasing their complexity. Consequently, applying these algorithms directly to VIPs becomes challenging as they have different mobility and perception constraints. There is a need for a specialised approach that complies with the distinct VIPs' mobility and perception constraints. Developing tailored algorithms is essential to enhance VIPs' mobility, independence, and safety.

Together with chapters 1 and 3, this thesis is now able to list the requirements of its proposed VIP mobility assistive scheme, which are needed to overcome the limitations identified in the current state-of-the-art approaches. These requirements include:

1. **Detection and Recognition of Common Obstacles in Daytime and Nighttime Conditions:** The solution must detect and recognise common obstacles encountered

in everyday outdoor environments, such as vehicles, under different lighting conditions, including both daytime and nighttime, to ensure safe VIP navigation at all times.

2. **Recognising and Tracking Moving Objects:** The solution should recognise and track moving objects, given their higher risk compared to static ones, to perceive and respond to dynamic threats accurately, reducing collision risks.

3. **Prediction of Potential Collisions With Dynamic Obstacles:** The solution should predict potential collisions with moving obstacles to prevent accidents and injuries for VIPs navigating dynamic environments.

4. **Accurate Estimation of VIP-Object Distance:** The solution must accurately calculate the distance between obstacles and VIPs to facilitate timely warnings and thus enable effective avoidance reactions.

5. **Extended Collision Identification Area:** The scheme should effectively identify potential collisions beyond the limited $10m$-safety zone established in previous literature studies (e.g., [26, 27, 28, 29]) to enhance threat detection capabilities and ensure VIP safety.

6. **Extended Distance Estimation for Early Warning:** The scheme must provide precise long-range distance estimation to cope with fast-moving threats. This enables early warnings and helps VIPs to safely avoid high-risk objects.

7. **Extended Collision Prediction Window:** The solution should offer a larger window for collision prediction, extending beyond the current limitation of $1 \sim 2s$ [28], to improve the ability to predict and avoid potential collisions effectively.

8. **Identifying Threats Moving at Varying Speeds and Prioritising Threats According to Risk Level:** The scheme should differentiate between objects moving at varying speeds and prioritise their warnings based on their potential collision time and risk level. This ensures timely warnings and effective manoeuvres and addresses the limitation of solely distance-based risk classification that potentially leads to scenarios in

which a fast-moving object (classified as low-risk) collides with the VIP before a slow-moving object (classified as high-risk), as found, for example, in [28].

9. **Detection of Faster-Moving Objects:** The solution must detect and respond to fast-moving threats, overcoming the limitations of current solutions that are restricted to slow-moving objects (e.g., $< 22km/h$ as identified in [28]), to reduce the risk of collisions with high-speed objects, thus improve VIP safety.

10. **Providing Detailed Information About Detected Threat:** The solution must provide details about detected obstacles, such as their type, location, distance, and whether they are in motion or static, to assist VIPs in fully perceiving threats and thus enabling effective response.

11. **Reduction of False Alarms:** The solution should reduce potential false alarms to enhance scheme reliability and VIPs' satisfaction.

12. **Real-Time Performance:** The scheme must run in real-time (i.e., processing frame rate should be $\geq 30fps$, which means each frame's processing time should be $\leq 33ms$; see Chapter 1) to ensure prompt threat detection and in-time alarm emission, enhancing VIP safety.

13. **Resilience to Infrared Radiation Interference:** The solution should remain unaffected by external sources of infrared radiation, such as sunlight, to maintain accurate threat detection capabilities.

14. **Resilience to Background Noise and Environmental Acoustic Interference:** The solution should remain resilient to background noise and environmental acoustic interference to ensure accurate threat detection.

15. **Reduced Scanning Burden Through Expanded Field of View:** The solution should offer a wider field of view compared to active sensors (whose range is limited to a narrow band, e.g., laser beam provides a field of view $< 0.5m$ [194]) to eliminate the VIP's need to scan their surroundings actively, thus preventing the risk of overlooking potential hazards resulting from neglecting this scanning process.

16. **Optimal Collision-Free Local Path Planning:** The solution should perform optimal collision-free local path planning (not requiring a full predefined environmental map of the surroundings) to guarantee safe navigation for VIPs within dynamic and unknown environments.

17. **Dynamic Collision-Free Path Adaptation:** The scheme must dynamically calculate collision-free local routes in response to real-time environmental changes to ensure continuous VIP safety.

18. **Generating Unblocked Paths in the Near Future:** The solution should plan routes that are not blocked by other obstacles in the near future (the next time slot) to avoid unnecessary misdirections and longer travel distances for VIPs.

19. **Incorporation of Object Movement Patterns in Path Planning:** The solution should consider objects' movement patterns when planning paths to avoid collisions.

20. **Consideration of VIP Walking Speeds in Path Planning:** The solution should account for VIPs' walking speeds when generating navigation paths to maintain their safety.

21. **Efficient Path Planning:** The solution must achieve fast convergence and low computation time for path planning to ensure real-time performance.

22. **Avoiding Reliance on Artificial Force-Based Path Planning:** The scheme's path planning algorithm must not rely on the artificial force concept to prevent impractical scenarios, such as when the destination's attractive force and obstacle's repulsive force, as encountered in Artificial Potential Field (APF) based approaches [216], reach equilibrium. This equilibrium may result in unfeasible navigation decisions that compromise VIP safety.

23. **Reduced Dependency on Extensive Data Collection and Training:** The solution should reduce reliance on extensive data collection and training for effective path planning. Excessive data requirements can lead to time-consuming setup processes and limit the solution's applicability to new or unforeseen environments.

In conclusion, this chapter has comprehensively reviewed existing research in the critical areas of VIP mobility assistance and robot local path planning. This review identified current solutions, their strengths, limitations, and potential gaps that will inform the development of the proposed solution of this thesis. The next chapter will delve into the details of the proposed AR-AwareAvoid scheme, focusing on its two main components: ARAware and MinD schemes.

Chapter 4

Schematic Design

Building upon the critical review of VIP mobility assistance and robot local path planning in Chapter 3, this chapter delves into the schematic design of the proposed AR-AwareAvoid scheme which comprises two main schemes, ARAware and MinD. It provides a detailed description of the VIP local path-planning problem and its mathematical formulation. This chapter serves as a roadmap for the subsequent implementation and evaluation phases of this innovative scheme.

4.1 AR-AwareAvoid Scheme

The general architecture of AR-AwareAvoid, depicted in Fig. 4.1, comprises **three** key components: *Wearable Vision Module* (WVM) to capture real-time video of VIPs' surroundings, *CMOs Identification Scheme* (ARAware) to identify and notify VIPs of *Critical Moving Objects* (CMOs) that pose a potential collision risk, and *Local Path Planning Scheme* (MinD) to plan the shortest safe path required by VIP to avoid CMOs successfully. For VIPs mobility assistance, AR-AwareAvoid offers two distinct modes: *Advance warning* and *Advance planning* modes. Leveraging ARAware, *Advance warning* mode provides early notifications about potential threats (CMOs), allowing VIPs ample reaction time for self-safe avoidance. Enabled by MinD, *Advance planning* mode dynamically calculates collision-free paths around CMOs to

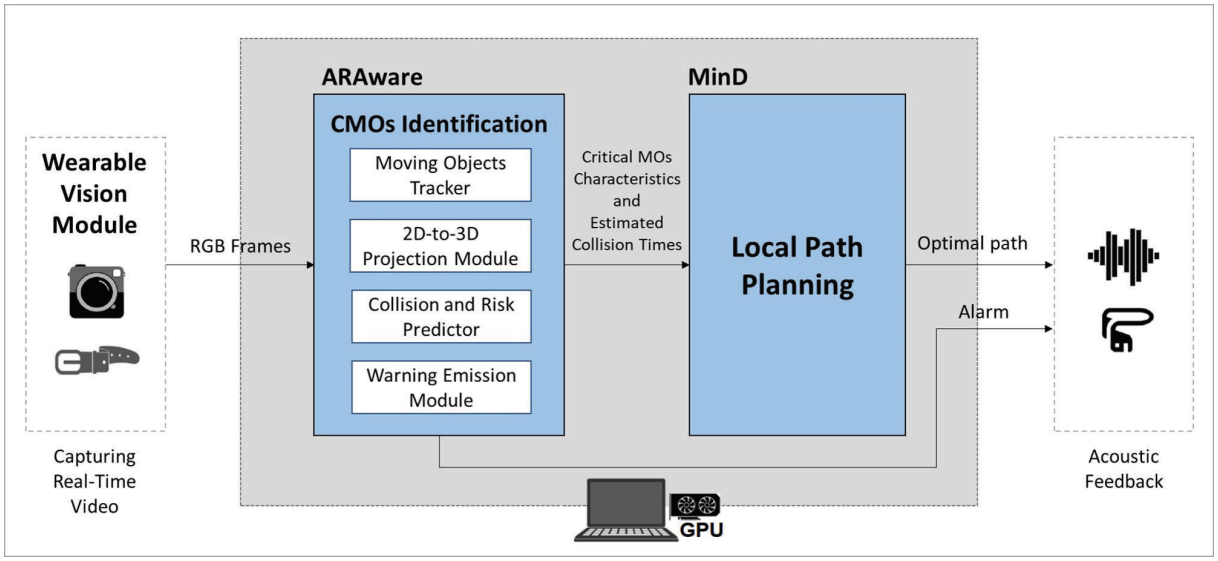


Figure 4.1: The General Schematic Architecture of AR-AwareAvoid.

guide VIPs to avoid CMOs with minimal effort and without a decision-making burden and to ensure timely and effective threat avoidance. The MinD scheme is built upon the core functionalities of the ARAware scheme. A detailed description of the AR-AwareAvoid's components is presented below. Note that Table 1 in Appendix .1 summarises all the symbols used in the development and formulation of the AR-AwareAvoid scheme.

4.1.1 Wearable Vision Module (WVM)

The WVM module utilises an RGB camera with a fixed focal length for video capture, chosen for its cost-effectiveness and potential to provide rich visual information essential for comprehensive analysis. To avoid potential jarring and disorientation caused by natural head movements and walking-induced camera shake, the camera is mounted on a stabilisation stick attached to the VIP's chest strap, positioned at eye level to provide a human-like field of view. Mounting the camera on a stabiliser ensures a stable and consistent environmental capture, allowing for effective and accurate analysis of recorded scenes. Positioning the camera at eye level provides a human-like field of view, enhancing the realism of the captured footage. The chest-mounted configuration avoids any discomfort or restrictions on head movement, allowing the VIP to navigate and interact with the environment freely. The camera's height depends on the

VIP's height. The camera captures video at a specific *frame rate* (specified later in Chapter 5), indicating the number of video frames captured per second, and operates at a specific resolution. All the captured frames are then transmitted to ARAware via a USB cable.

4.1.2 ARAware: CMOs Identification Scheme

This section delves into the design of the ARAware scheme, which is responsible for identifying and notifying VIPs of ambient CMOs like cars, motorcycles, bicycles and pedestrians. Fig. 4.2 depicts the general schematic architecture of ARAware, comprising **four** major modules (some of them use existing methods, while others were specifically developed for this scheme): *Moving Object Tracker* (MOT), *2D-to-3D Projection Module (PM)*, *Collision and Risk Predictor* (CRP), and *Warning Emission Module* (WEM). The MOT detects and tracks *moving objects* in the video frames obtained from the WVM. The PM converts the *2D-pixel positions* of the moving objects in video frames into their corresponding *3D-metric positions* in the *Camera Coordinate System* (CCS, explained later in the subsection of 'The 2D-to-3D Projection Module (PM)' within Section 4.1.2). The CRP identifies CMOs that may collide with the VIP among objects tracked by the MOT and then estimates the *risk level* of each CMO. The WEM acoustically alerts the VIP about CMOs. The subsequent sections provide a detailed description of each module.

Moving Object Tracker (MOT)

The MOT aims to detect and track targeted classes of *moving objects* in the video frames captured by the WVM. This process is carried out in three steps: *i.e.*, *object detection*, *object filtering and tracking* and *motion detection*.

Object Detection. *Real-time* and *accurate* object detection is vital for VIPs to promptly respond to potential collisions. To achieve this, a *CNN-based object detector*, YOLOv8, is used. The YOLOv8 is the latest version (at the time of thesis writing) of the well-established YOLO object detection models. It is specifically chosen because, by incorporating an optimised CSP-

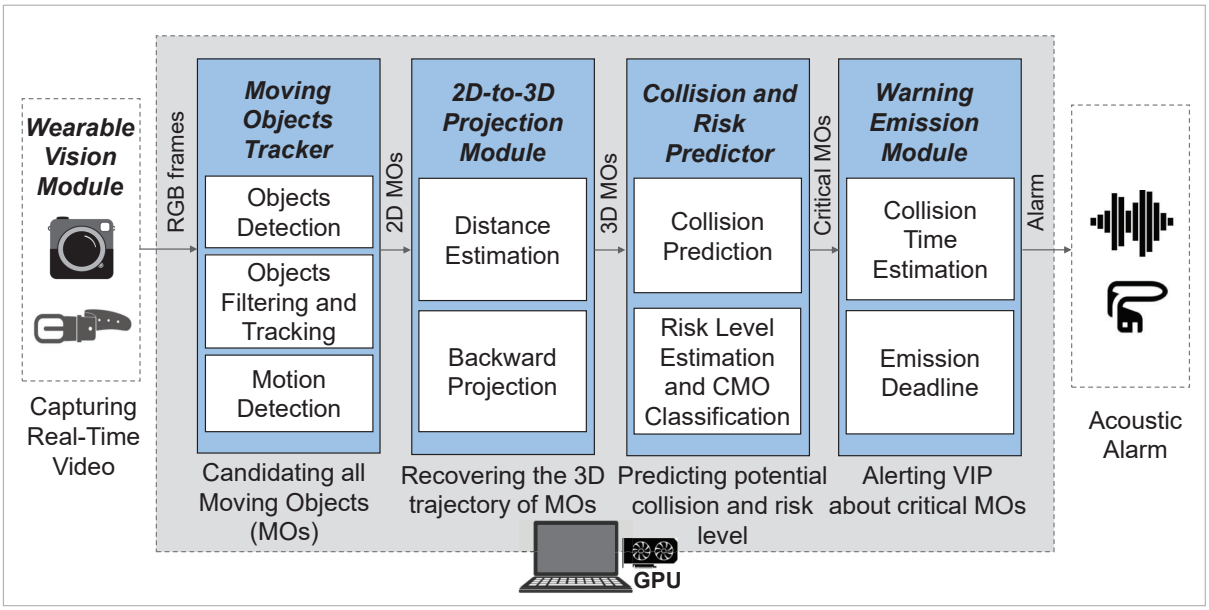


Figure 4.2: The General Schematic Architecture of ARAware.

Darknet53 backbone, an anchor-free detection, and a novel loss function, YOLOv8 achieves superior performance in detection accuracy and processing speed (end-to-end inference time) within the YOLO series [4, 222]; more details on the advancements of YOLOv8 are presented in Chapter 2. Due to its prominent and stable performance across various public datasets, such as COCO [92] and CrowdHuman [223], the YOLOv8 model pre-trained (with frozen weights) on the COCO dataset (a large-scale benchmark dataset for object detection) is employed as ARAware's object detector. The COCO dataset is a massive collection of $330K$ images containing everyday objects with 80 object categories. The object detector analyses each captured video frame (with a specific resolution) to localise and classify all objects present within the frame. It outputs a list of *bounding boxes* of detected objects. Each YOLOv8's bounding box is associated with an object *class label* and a *detection confidence* score, which indicates the likelihood that the box contains an object [222].

Object Filtering and Tracking. ARAware's pre-trained YOLOv8-based object detector detects 80 different classes of objects [222]. However, as this research focuses on moving objects that could threaten the VIP's safety in outdoor environments, a *class-based object filtering* process is conducted to extract *targeted objects*, including *cars*, *motorcycles*, *bicycles*, and *pedestrians*. These specific classes were chosen because cars and motorcycles can cause severe

accidents if not detected promptly due to their size and speed, while bicycles often share pedestrian pathways and can manoeuvre unpredictably, making their detection crucial for VIP safe navigation. Identifying other pedestrians helps VIPs navigate through crowds and maintain a safe distance from others. According to the VIP's requirements, any other class of objects, such as animals (dogs, cats, birds) that can appear suddenly and unpredictably, can be straightforwardly included without needing to retrain the detector. This is because, as mentioned above, the YOLOv8 detector used is pre-trained on the COCO dataset, which already includes a wide variety of object classes (80 classes), including many common animals. By incorporating such additional classes, ARAware could offer even more comprehensive assistance, ensuring VIPs can navigate safely and confidently in a wider range of scenarios.

The YOLOv8-based object detector produces independent bounding boxes for detected objects in each video frame without considering the correlation between the bounding boxes of the same object in different frames. Associating the detected objects across frames is crucial for estimating objects' movement and assessing their potential threat to the VIP. To achieve this, the *Deep Simple Online Real-Time* (Deep SORT) model, pre-trained on MOT16 [224] (with frozen weights), is used as ARAware's object tracker due to its capability to track multiple objects with fast processing speed and competitive accuracy [224], ensuring that the real-time requirement (processing frame rate $\geq 30fps$, see Chapter 1) is met. It receives bounding boxes of all targeted objects across all frames where object detection is conducted. It then assigns a unique ID to each object and sorts all bounding boxes of the object temporally to form a *trajectory*. This process is carried out in an *online* manner. To improve the overall processing speed of ARAware, in the trajectory of a specific object, ARAware only records sorted bounding boxes of the latest *two* detected frames, *i.e.*, f_p and f_c, the previous and current frames at time t_p and t_c, respectively. By recording only the latest object positions (bounding boxes), ARAware reduces computational load and memory usage, thus maintaining real-time performance while preserving sufficient up-to-date information for accurate VIP-Object collision prediction, explained later in the following subsections. Fig. 4.3 illustrates the online update process of the object trajectory.

Motion Detection. For computational simplicity, a straightforward motion detector is de-

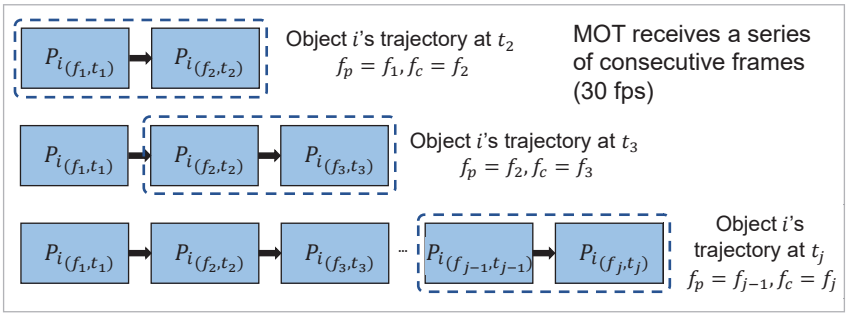


Figure 4.3: Moving Object Trajectory Update. Note: When the trajectory is updated at t_j, the estimated distance and estimated moving speed are also updated to calculate the latest expected collision time.

veloped to identify target objects that are moving, which can be used when the VIP (camera) is still or moving. This detector is also essential to distinguish camera motion from object motion, enabling accurate estimation of the object's speed, as explained later in the subsection of 'Collision and Risk Predictor (CRP)' within Section 4.1.2. Initially, the *oriented FAST and rotated BRIEF* (ORB) feature detector [225] is used to estimate the camera motion. This involves identifying the same *features* in two consecutive frames. The ORB is chosen due to its superior computation speed and cost compared to its counterparts, *i.e.*, SIFT and SURF. The ARAware's motion detector selects the best 25∼50 matches among all detected features. This selection helps to balance the trade-off between maximising feature utilisation and reducing the computational cost. Obtaining immobile features, such as those extracted from stationary structures like buildings, is essential to estimate camera motion accurately. Therefore, in the ARAware's motion detector, features that reside within the bounding boxes of the targeted detected moving objects are excluded. Then, for each feature M_i, its 2D-pixel positions $M_i^{(f_p,t_p)} = (x_m, y_m)_i^{(f_p,t_p)}$ at f_p and $M_i^{(f_c,t_c)} = (x_m, y_m)_i^{(f_c,t_c)}$ at f_c are compared. If most feature pairs (*i.e.*, $\geq 50\%$) have identical positions, the camera is identified as *static*. Otherwise, it is identified as *moving* (showing a noticeable change in the camera's position). This majority comparison is necessary because even after excluding the extracted features of targeted moving objects (as mentioned above), a part of features is likely to be extracted from non-targeted moving objects such as pets or immobile structures, *e.g.*, trees with moving leaves.

Then, for each targeted object O_i, its bounding box is defined as $(x_1, y_1, x_2, y_2)_i$, *i.e.*, the *minimum* and *maximum*, 2D-pixel coordinates of the rectangular border that fully encloses the

detected object in the video frame. By receiving $(x_1, y_1, x_2, y_2)_i^{(f_p,t_p)}$ and $(x_1, y_1, x_2, y_2)_i^{(f_c,t_c)}$ in object O_i's trajectory, the motion detector calculates the central position of O_i at f_p and f_c as $P_i^{(f_p,t_p)} = (x_O, y_O)_i^{(f_p,t_p)}$ and $P_i^{(f_c,t_c)} = (x_O, y_O)_i^{(f_c,t_c)}$, respectively. For *static camera*, if the Euclidean distance between the $P_i^{(f_p,t_p)}$ and $P_i^{(f_c,t_c)}$ is greater than ϵ,[1] O_i is identified as a *moving object*, and its trajectory is passed to CRP. For *moving camera*, the detector calculates the displacement between the $M_i^{(f_p,t_p)}$ and $M_i^{(f_c,t_c)}$ of the best matched M_i, at f_p and f_c, respectively, which is defined as $D_i^{M_p,M_c} = (x_m^{t_c} - x_m^{t_p}, y_m^{t_c} - y_m^{t_p})_i$. Then, if $P_i^{(f_c,t_c)} - D_i^{M_p,M_c} \neq P_i^{(f_p,t_p)} \pm \epsilon$, O_i is identified as a *moving object*, and it's trajectory is passed to CRP. Note that the $D_i^{M_p,M_c}$ will be used later in Eqs. (4.18) to (4.22) for camera motion separation.

The 2D-to-3D Projection Module (PM)

The 2D trajectories of all targeted moving objects tracked by MOT in the captured video frames are transmitted to PM. However, to accurately predict potential collisions between objects and VIPs, obtaining the real-world trajectories of these moving objects is necessary. PM accomplishes this by first estimating the real-world distance between the VIP and the moving objects. Then, it transforms the objects' 2D trajectories in the *Pixel Coordinate System* (PCS) into 3D trajectories in the *Camera Coordinate System* (CCS). A PCS is a 2D coordinate system used to represent objects' positions, in pixels, within a digital image or video frame, while a CCS is a real-world 3D coordinate where the camera acts as the origin (see Fig. 4.4) and measurement units in m [227].

Distance Estimation. Obtaining the distance from the camera to an object is fundamental to successfully recovering the object's 3D-metric position from its 2D-pixel position. Further, a precise and *long-range* distance estimation method contributes to more accurate and earlier collision predictions, which can help to guarantee more safety for VIPs. To fulfil this, ARAware utilises the pre-trained DisNet model [7] (trained based on YOLO's resulting bounding boxes [92]) over other counterparts' deep learning-based methods due to its simplicity, fast processing speed, and accuracy in estimating distances over long ranges using a monocular camera. Unlike

[1]The ϵ can be set as any small value to tolerate computational errors [226]. In the research's experiments, the ϵ is set to zero for stricter object motion detection as VIP's safety is the fundamental goal of this research.

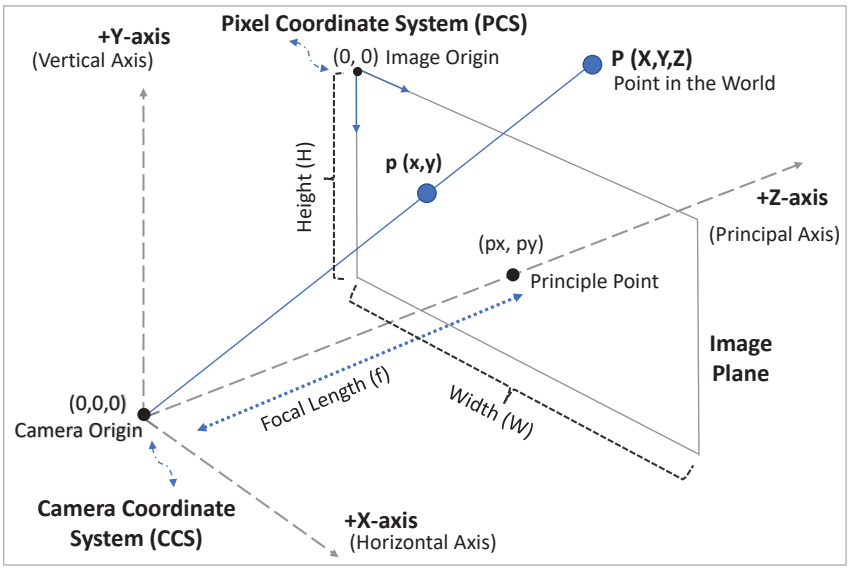


Figure 4.4: Demonstration of the Reference Camera Coordinate System. Note: The camera center is at the origin (0,0,0), X-axis and Y-axis are the horizontal and vertical axes that are parallel to the image plane axes, and Z-axis is the principal axis that is orthogonal to the image plane.

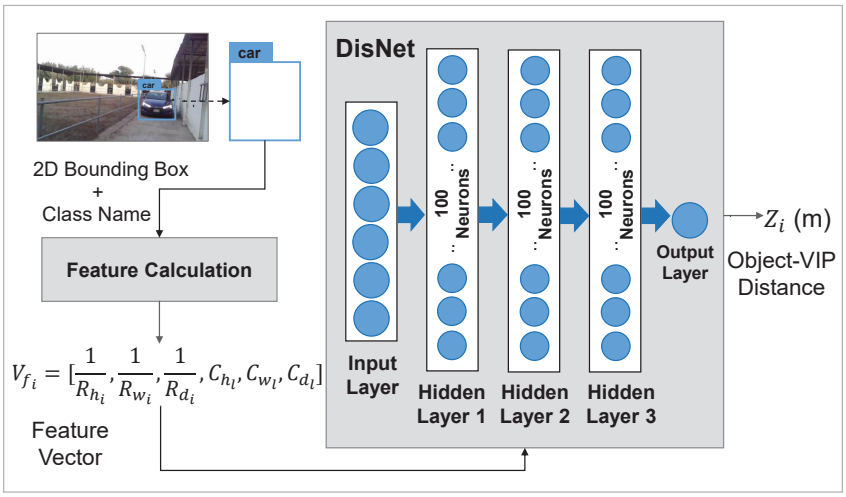


Figure 4.5: Distance Estimation Based on DisNet (Reproduced From [7]).

classical methods, DisNet's accuracy is unaffected by the object's distance from the camera [7]. It also does not require prior knowledge about the scene or the camera parameters.

DisNet is a multi-layer neural network specifically designed for autonomous obstacle detection over long distances in railway applications. As depicted in Fig. 4.5, DisNet comprises a 6-dimensional input layer representing an object's feature vector, three hidden layers, each containing 100 neurons, and a 1-dimensional output layer that provides the distance in m. For

the 2D bounding box of each detected object i, the input feature vector is calculated as:

$$V_{f_i} = [\frac{1}{R_{h_i}}, \frac{1}{R_{w_i}}, \frac{1}{R_{d_i}}, C_{h_l}, C_{w_l}, C_{d_l}], \tag{4.1}$$

where,

$$R_{h_i} = \frac{BB_{h_i}}{I_h}, R_{w_i} = \frac{BB_{w_i}}{I_w}, R_{d_i} = \frac{BB_{d_i}}{I_d}, \tag{4.2}$$

where R_{h_i}, R_{w_i} and R_{d_i} represent the ratios of the object i's bounding box dimensions (*i.e.*, height, width, and diagonal) to image dimensions in pixels, respectively. C_{h_l}, C_{w_l} and C_{d_l} are predefined values of average dimensions (*i.e.*, height, width, and depth) of an object of a particular class l in cm. The setups of the predefined dimensions of ARAware's targeted classes will be discussed later in Section 5.1.1 of Chapter 5. DisNet subsequently takes the feature vector V_{f_i} as input and produces the distance Z_i^t from the camera (i.e., the VIP) to the detected object at time t in m:

$$Z_i^t = Disnet(V_{f_i}). \tag{4.3}$$

Note that when a faraway object (*e.g.*, a motorcycle) has properties similar to another object (*e.g.*, a bicycle), the YOLOv8-based object detector may misclassify the object. This can affect DisNet's distance estimation (*i.e.*, estimation error becomes larger), as it strongly depends on the predefined object's class dimensions. However, this does not affect the ARAware's general performance because when an object is far away, its dimensions tend to be small in the image, and the differences in actual size between similar objects (bicycles and motorcycles) are typically slight ($15cm \sim 25cm$ based on the predefined object's class dimensions mentioned in Chapter 5, inspired by [7] and ARAware's empirical study), resulting in a negligible estimation error rate; also because any initial misclassification will be corrected as the object gets closer. Furthermore, when a correctly classified object (initially moving towards the VIP) changes its moving direction (moving left or right), the size of the bounding box surrounding the object increases temporarily (this increase decreases as the object exits the image frame). This can increase the error rate as the object appears closer than it actually is. Nevertheless, this also does not affect the ARAware's performance because if the object changes its direction, this implies no object-VIP collision, which is therefore disregarded.

Backward Transformation. General perspective projections, which involve converting 3D scenes into 2D images, do not preserve the real-world distance and angle of objects in a scene [228]. As a result, ARAware conducts a *backward transformation* to retrieve the 3D object's position in the CCS. This retrieval enables accurate estimation of the object's distance and moving angle, which is crucial for ARAware's collision prediction method (described in the subsequent section). According to the Pinhole Model [229] and given the distance Z_i^t in Eq. (4.3), the 2D object's position in the PCS is transformed into a 3D position in the CCS (see Fig. 4.4) as follows:

$$^{C}X_{O_i}^t = (x_{O_i}^t - p_x)\frac{Z_i^t}{f_x}, \tag{4.4}$$

$$^{C}Y_{O_i}^t = (y_{O_i}^t - p_y)\frac{Z_i^t}{f_y}, \tag{4.5}$$

$$^{C}Z_{O_i}^t = Z_i^t, \tag{4.6}$$

where $(x_{O_i}^t, y_{O_i}^t)$ is the 2D-pixel position of an object at time t, (p_x, p_y) is the camera's principal point in pixels, and f_x and f_y are the X-axis and Y-axis of the camera's focal length components in pixels. The principal point and focal length are two of the intrinsic parameters of a camera. These parameters can be estimated using the camera calibration tool in OpenCV 4.1,[2] where a set of known 3D points and their corresponding 2D projections are used for the estimation process. Here, PM transforms the 2D trajectories of all targeted moving objects in the PCS into corresponding 3D trajectories in the CCS, which are then relayed to CRP.

The Collision and Risk Predictor (CRP)

When the CRP receives 3D trajectories from the PM, it first predicts potential collisions by analysing the movements of objects. Then, it classifies critical objects based on their respective risk levels.

Collision Prediction. Based on vector calculus [230], a method for identifying objects likely to collide with the VIP, *i.e.*, Critical Moving Objects (CMOs), is developed. Note, to prevent the computational burden associated with complex projections in a real-world coordinate system,

[2] https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html

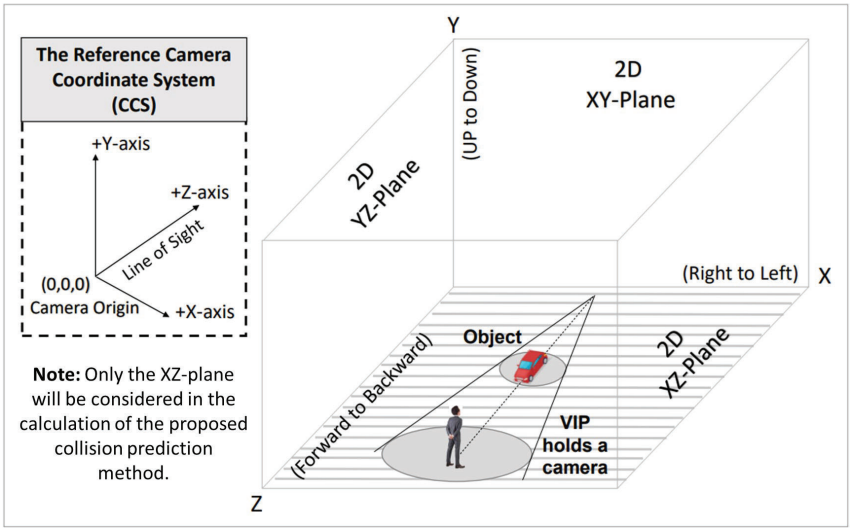


Figure 4.6: Demonstration of the 2D XZ-Plane Considered in Collision Prediction. Note: The 2D XZ-plane is the ground plane shared by the VIP and the detected object.

which has a fixed origin, the CCS is used as the reference coordinate system, see Fig. 4.4. This is because, in practice, the CCS's origin dynamically positioned at the current VIP's (camera's) position, which means the CCS's origin always moves with the VIP, facilitating ARAware calculations.

As demonstrated in Fig. 4.6 and Fig. 4.7, it is assumed that the VIP is located at point B^{t_p}, which is the centre of the VIP's circular *collision zone* C_B on the 2D XZ-plane, *i.e.*, the ground plane, at time t_p. This zone has a radius of r_B. There is also an object that occupies a circular area C_{O_i} with a centre at point $O_i^{t_p}$ and a radius of r_{O_i} at t_p. The object moves to point $O_i^{t_c}$ at t_c. To detect all potential threats, the VIP's collision zone should be larger enough than the object's zone. *i.e.*, $r_B \gg r_{O_i}$. For simplicity, it is assumed that the VIP (camera) is static, *i.e.*, $B^{t_c}==B^{t_p}$, and any object can move at a velocity of $\boldsymbol{v_{O_i}} = |\boldsymbol{v_{O_i}}|\angle\theta$, where $|\boldsymbol{v_{O_i}}|$ is the speed, and $\angle\theta$ is the moving angle, *i.e.*, the angle between the object moving direction and the Z-axis of the CCS.

CMO Identification. As illustrated in Fig. 4.7, the *critical zone* is defined with two boundaries, represented by $\overleftrightarrow{B_1O_1}$ and $\overleftrightarrow{B_2O_2}$, two common tangents of circles C_B and C_{O_i} intersected at point A. To determine whether detected objects are located within the *critical zone* (move toward the VIP), *three* angles are considered. Specifically, the angle $\angle\alpha_i$ between $\overline{AB^{t_p}}$ and

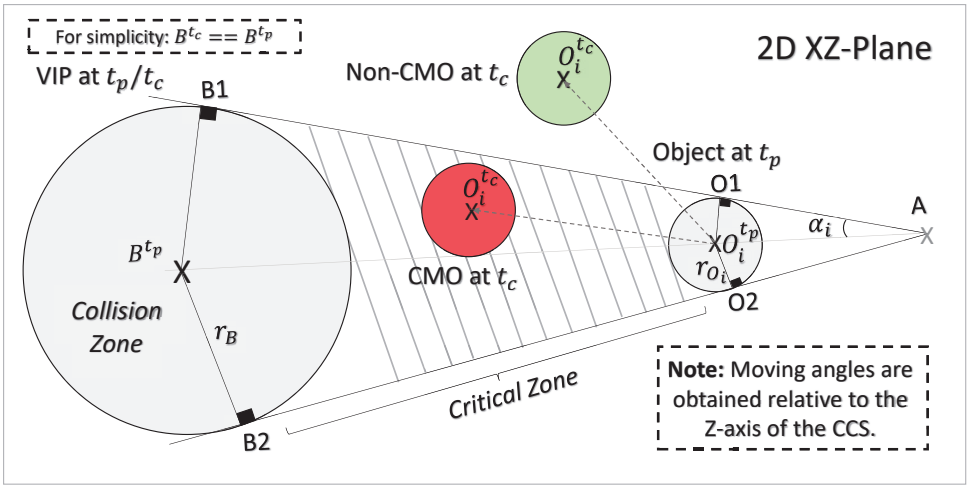


Figure 4.7: Critical Zone Estimation Based on Geometric Principles.

$\overline{AB_1}$ is first calculated. Obviously, since $\triangle AB^{t_p}B_1$ and $\triangle AO_i^{t_p}O_1$ are similar, there is:

$$\frac{|\overline{AB^{t_p}}|}{|\overline{AO_i^{t_p}}|} = \frac{r_B}{r_{O_i}}, \tag{4.7}$$

$$\frac{|\overline{AB^{t_p}}|}{|\overline{AO_i^{t_p}}|} = \frac{|\overline{(AO_i^{t_p} + O_i^{t_p}B^{t_p})}|}{|\overline{AO_i^{t_p}}|} = \frac{|\overline{AO_i^{t_p}}| + d_i}{|\overline{AO_i^{t_p}}|} = \frac{r_B}{r_{O_i}}, \tag{4.8}$$

thus,

$$|\overline{AO_i^{t_p}}| = \frac{r_{O_i}}{r_B - r_{O_i}} d_i, \tag{4.9}$$

where $d_i = |\overline{O_i^{t_p}B^{t_p}}|$ is the Euclidean distance between the VIP and the object at t_p, which is calculated using Eq. (4.4) and Eq. (4.6) as follows:

$$d_i = \sqrt{({}^C X_B^{t_p} - {}^C X_{O_i}^{t_p})^2 + ({}^C Z_B^{t_p} - {}^C Z_{O_i}^{t_p})^2}, \tag{4.10}$$

where ${}^C X_B^{t_p}$ and ${}^C Z_B^{t_p}$ are the X-axis and Z-axis coordinates of the VIP point B^{t_p}, and ${}^C X_{O_i}^{t_p}$ and ${}^C Z_{O_i}^{t_p}$ are the X-axis and Z-axis coordinates of the object point $O_i^{t_p}$ at t_p in the CCS. It is important to note that the Y-axis coordinates (*i.e.*, ${}^C Y_B^t = 0$ and ${}^C Y_{O_i}^t = 0$) are ignored because it is assumed that the camera's motion along the Y-axis is negligible since it is vertical to the ground plane. From the Pythagoras equation, there is:

$$|\overline{AO_1}| = \sqrt{|\overline{AO_i^{t_p}}|^2 - {r_{O_i}}^2}. \tag{4.11}$$

Inserting Eq. (4.9) into Eq. (4.11), there is:

$$|\overline{AO_1}| = r_{O_i}\sqrt{\frac{d_i^2}{(r_B - r_{O_i})^2} - 1}. \tag{4.12}$$

For α_i in $\triangle AO_i^{t_p}O_1$, there is:

$$tan(\alpha_i) = \frac{r_{O_i}}{|\overline{AO_1}|} = \frac{1}{\sqrt{\frac{d_i^2}{(r_B - r_{O_i})^2} - 1}}, \tag{4.13}$$

thus,

$$\alpha_i = tan^{-1}\left(\frac{1}{\sqrt{\frac{d_i^2}{(r_B - r_{O_i})^2} - 1}}\right). \tag{4.14}$$

Then, the angle $\angle\phi_i$ of the displacement vector from $O_i^{t_p}$ to B^{t_p} (defined as $O_i^{t_p}\vec{B}^{t_p} = |O_i^{t_p}\vec{B}^{t_p}|\angle\phi_i$) is calculated as:

$$\phi_i = tan^{-1}\left(\frac{{}^C X_B^{t_p} - {}^C X_{O_i}^{t_p}}{{}^C Z_B^{t_p} - {}^C Z_{O_i}^{t_p}}\right). \tag{4.15}$$

Finally, the angle $\angle\beta_i$ of the the displacement vector from $O_i^{t_p}$ to $O_i^{t_c}$ (defined as $O_i^{t_p}\vec{O}_i^{t_c} = |O_i^{t_p}\vec{O}_i^{t_c}|\angle\beta_i$) is calculated as:

$$\beta_i = tan^{-1}\left(\frac{{}^C X_{O_i}^{t_c} - {}^C X_{O_i}^{t_p}}{{}^C Z_{O_i}^{t_c} - {}^C Z_{O_i}^{t_p}}\right). \tag{4.16}$$

With the above angles α_i, β_i, and ϕ_i, if the condition $\phi_i - \alpha_i \leq \beta_i \leq \phi_i + \alpha_i$ is satisfied, the object is identified as a CMO as it is moving within the critical zone and poses a potential threat to the VIP. Otherwise, it is considered a non-critical object, *i.e.*, non-CMO. It is important to note that the identification of CMOs is based on the objects' moving angle (i.e., direction), not their relative location. For example, in Fig. 4.7, the moving object coloured in green will not be classified as a CMO as it is moving outside the critical zone (i.e., it does not move towards the VIP). Given objects' dynamic nature (varying speeds and course changes over time), analysing their movement direction provides a deeper understanding of their behaviour. This enables more effective prediction of potential collisions with VIPs compared to the static proximity measure obtained by simply measuring objects' relative position. Note, since the collision prediction

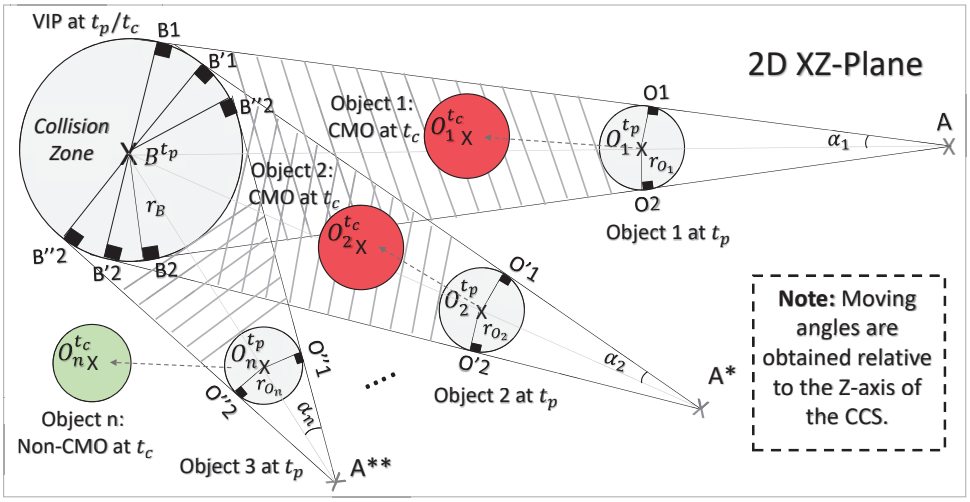


Figure 4.8: Critical Zone Estimation Based on Geometric Principles for Multiple objects.

depends on the object's moving direction only, ARAware's CMO identification method can be directly applied to scenarios where the VIP is moving without abrupt direction changes, as well as scenarios involving multiple objects (see Fig. 4.8).

Considerations for Radius Setting Up. Several factors are considered to set up the radius, r_{O_i}, and r_B. In particular, for r_{O_i} setting up, the circular object's zone should be large enough to cover the object's width in a particular class, and at the same time, it should not exceed the object's width too much to reduce false alarms. This is to balance the trade-off between VIP's safety and ARAware usability. For r_B setting up, to successfully ensure the detection of all possible CMOs threatening the VIP's safety, the VIP's collision zone should be considerably larger than the circular object's zone. Empirical studies based on extensive real-world experiments were conducted to determine proper radius values while addressing the considerations above. Specifically, in these experiments, the actual width of objects belonging to the targeted classes was measured, and the corresponding radius for a VIP circular zone that would ensure the VIP's safety was determined. Using these measurements, critical zones (cones) for each class were marked on the ground, and data on object movement both inside and outside these zones were collected and analysed. Based on this analysis, the corresponding radius values were determined. These values, customised based on the VIP's safety requirements, are presented in Section 5.1.1 of Chapter 5.

CMO Speed Estimation with Camera Motion Removal. For each CMO, its moving speed is estimated as follows (considering the removal of the camera motion effect):

$$v'_{O_i} = \sqrt{{v_{x_{O_i}}}^2 + {v_{y_{O_i}}}^2 + {v_{z_{O_i}}}^2}, \tag{4.17}$$

where,

$$v_{x_{O_i}} = \frac{({}^C X_{O_i}^{t_c} - D_{C_x}) - {}^C X_{O_i}^{t_p}}{t_c - t_p}, \tag{4.18}$$

$$v_{y_{O_i}} = \frac{({}^C Y_{O_i}^{t_c} - D_{C_y}) - {}^C Y_{O_i}^{t_p}}{t_c - t_p}, \tag{4.19}$$

$$v_{z_{O_i}} = \frac{{}^C Z_{O_i}^{t_c} - {}^C Z_{O_i}^{t_p}}{t_c - t_p} \pm C_z, \tag{4.20}$$

where,

$$D_{C_x} = (x_m^{t_c} - x_m^{t_p}) \frac{Z_i^{t_c}}{f_x}, \tag{4.21}$$

$$D_{C_y} = (y_m^{t_c} - y_m^{t_p}) \frac{Z_i^{t_c}}{f_y}. \tag{4.22}$$

Where, $v_{x_{O_i}}$, $v_{y_{O_i}}$ and $v_{z_{O_i}}$ represent the CMO's velocity components along the X-axis, Y-axis, and Z-axis, respectively. Both $({}^C X_{O_i}^{t_p}, {}^C Y_{O_i}^{t_p}, {}^C Z_{O_i}^{t_p})$ and $({}^C X_{O_i}^{t_c}, {}^C Y_{O_i}^{t_c}, {}^C Z_{O_i}^{t_c})$ are CMO's positions at t_p and t_c, respectively. D_{C_x} and D_{C_y} represent the camera displacement (in m) at the X-axis and Y-axis, respectively. C_z is the camera's velocity along the Z-axis in (m/s).

Removing camera movement is essential for accurately estimating the CMO's actual speed. Therefore, in Eqs. (4.18) and (4.19), D_{C_x} and D_{C_y} (the X-axis and Y-axis components of the camera displacement $D_i^{M_p,M_c}$, see Section 4.1.2, multiplied by the pixel-to-meter conversion ratio $Z_i^{t_c}/f$) are subtracted from ${}^C X_{O_i}^{t_c}$ and ${}^C Y_{O_i}^{t_c}$, respectively. This removes the influence of camera movement along the X-axis and Y-axis. However, ARAware's 2D feature-based motion detection (explained in Section 4.1.2) cannot estimate camera movement in the Z-axis (depth direction). To address this, it is assumed that the VIP walks at a slow velocity (*i.e.*, $\leq 3km/h$) along that axis, denoted by C_z in Eq. (4.20). The constant value of C_z can be added or subtracted ($\pm$) from the CMO's velocity based on the relative movement directions of the VIP and the CMO along the Z-axis. In general, along the Z-axis, both an object and the VIP

can exhibit two movement directions: forward (positive Z) or backward (negative Z), which results in four cases: (1) the object moving away from the VIP (+Z) and the VIP moving forward (+Z), (2) object moving away (+Z) and the VIP moving backward (-Z), (3) object approaching the VIP (-Z) and the VIP moving forward (+Z), and (4) object approaching (-Z) and the VIP moving backward (-Z). Consequently, when both the VIP and the object move in the same direction along the Z-axis, their velocities are subtracted. Conversely, if they move in opposite directions, their velocities are added. Since ARAware only estimates speed when objects heading toward the VIP, *i.e.,* CMOs , cases (1) and (2) are excluded. VIPs typically walk slower than sighted people, who generally move at an average speed of $5km/h$, and their walking speed decreases as the complexity or uncertainty of the surrounding environments increases [231]. Therefore, assuming a slow velocity, *i.e.,* $C_z \leq 3km/h$, for the VIP provides a reasonable estimate in most scenarios. The C_z can be adjusted based on the VIP's walking speed preferences. Note that to achieve accurate speed estimation with the removal of camera movement (discussed above), the camera should maintain a slow pace (e.g., $\leq 3km/h$ in ARAware experiments) and avoid strong shaking or sudden movements. This is because this method relies heavily on extracting features from a background.

Subsequently, by maintaining a record of v'_{O_i}, the CMO's average speed is computed using the following formula:

$$v'_{avg_{O_i}} = \frac{1}{n} \sum_{j=1}^{n} v'_{O_i j}, \tag{4.23}$$

where n represents the total number of speeds $v'_{O_i j}$ stored in the record. Considering that the calculation time of $v'_{avg_{O_i}}$ is negligible, $v'_{avg_{O_i}}$ is used rather than v'_{O_i} in predicting the collision time of the CMO (discussed later in Section 4.1.2) as it is less affected by outliers in the estimation of the CMO's speed. Indeed, depending on the VIP's requirements, either v'_{O_i} or $v'_{avg_{O_i}}$ can be used. For example, if a faster warning delivery is required, v'_{O_i} is recommended. However, if a more accurate collision time estimation is the priority, $v'_{avg_{O_i}}$ should be utilised; see Eq. (4.24).

Risk Level Estimation and CMO Classification. All identified CMOs are classified into several groups with different *risk levels* based on their potential danger to the VIP. Classifying

CMOs into different risk levels enables a systematic prioritisation and appropriate handling of each category. CMOs with higher risk levels receive greater attention and are processed with higher priority to prevent more serious damage to the VIP.

To develop a reasonable definition of risk levels, This research has conducted an empirical background study on injuries resulting from traffic collisions. The *UK Department for Transport* (DfT) classifies human injury severity in public road accidents into three levels: fatal, serious, and slight [232, 233]. Indeed, the severity of injuries caused by traffic collisions is influenced by various factors associated with the vehicles, individuals involved (*e.g.*, their age and health), and the surrounding environments. However, the fundamental factor significantly contributing to injury severity is the *speed* of the vehicles involved in the collision [234]. Numerous studies have investigated the relationship between the severity of injuries and the impact speed of vehicles, which is the absolute value of the relative speed between the vehicle and the impacted person at the moment of collision [233]. It is generally understood that car accidents involving impact speeds below $25km/h$ result in minor injuries, while those involving impact speeds above $45km/h$ result in fatal injuries [233, 234, 235]. Motorcycles can cause various fatal injuries [236]. However, there is a lack of numerical studies exploring the correlation between impact speed and injury severity for motorcycles. Given that motorcycles can reach high speeds similar to cars, this research assumes that the damages caused by motorcycles are generally close to those caused by cars. According to [237], when the impact speed exceeds $35km/h$, bicycles have the potential to cause secondary post-impact head injuries. Pedestrians typically walk at a low speed [238], and collisions involving pedestrians generally do not result in serious injuries.

Based on the understanding above, this research considers both the *type* and *moving speed* of CMOs as key criteria for classifying CMOs' risk levels. Specifically, four risk levels for all CMOs are defined: *severe* (e.g., fatal injuries), *serious* (e.g., fractures or injuries cause hospitalisation), *minor* (e.g., bruises), and *minimal* (e.g., no injuries or minor scrapes). Then, according to [237, 238, 239], the general average speed of cars, motorcycles, bicycles, and pedestrians are summarised, and a mapping between risk levels and CMOs moving at different speeds is set up, as shown in Figure 4.9. Note that this mapping can be straightforwardly modified to accommodate additional object classes per the VIP's needs without requiring additional effort,

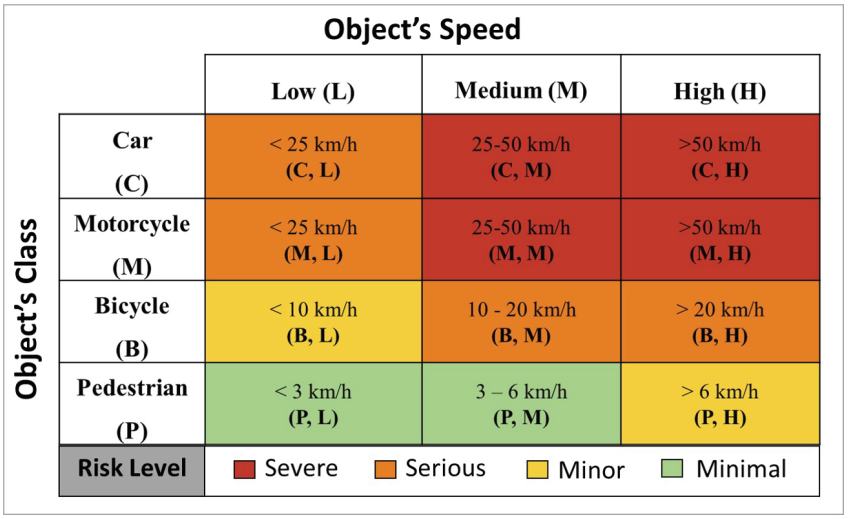


Figure 4.9: CMO Risk Level Classification Based on Potential Damages.

such as model training. This mapping assigns a risk level to each CMO, which serves as input for the WEM. With the VIP's safety in mind, it is important to highlight that the speed threshold in the mapping is intentionally set at half of the general average speed. This deliberate decision results in higher risk levels assigned to CMOs, ensuring a stricter warning approach for VIP safety. By risk classification, the aim is to enhance the efficiency of the ARAware scheme by providing early warnings for high-risk threats, *i.e.*, severe and serious risks, and reducing false alarms associated with low-risk threats, *i.e.*, minor and minimal risks. This prioritisation allows the VIP to react promptly and effectively.

The Warning Emission Module (WEM)

Considering the varying risk levels of CMOs, WEM aims to alert VIPs about potential threats in a timely, effective, and highly usable manner. To achieve this, WEM must satisfy the following three requirements:

- **In-Time Warning:** CMOs must be warned in time, *i.e.*, $t'_{cl_i} \leq t_{dl}$, see Equations (4.24) and (4.25), where sufficient time should be left for the VIP to react to the warning and take appropriate action.

- **Risk-Based Prioritisation:** Warning messages must be prioritised based on the CMOs' different risk levels, *i.e.*, scaled from 1 to 4, where 1 denotes the highest risk, and 4 denotes the lowest risk. This means a CMO with a higher risk level should receive a warning sooner than a CMO with a lower risk level.
- **False Alarms Reduction**: False alarms should be reduced as much as possible to ensure ARAware scheme usability. False alarms can make the VIP less likely to trust the scheme.

Based on the above requirements, a warning emission method is developed as follows.

Warning Emission Priority: The priority of warning emissions is determined based on the risk level of the CMOs. In cases where multiple warning messages for different CMOs arise simultaneously, messages of CMOs with higher risk levels will be sent earlier. Priorities to CMOs with different risk levels are assigned as follows: a CMO with 'severe' risk is received the '1st' priority, a CMO with 'serious' risk is assigned the '2nd' priority, a 'minor' risk CMO is given the '3rd' priority, and a 'minimal' risk CMO is received the '4th' priority. In cases where CMOs share the same priority, their warning messages are emitted based on the expected *collision time*, which means that the shortest time is prioritised. Neuroscience research [240, 241] and mobility assistance studies [242] for VIPs suggest that VIPs possess exceptional auditory processing capabilities, allowing them to perform sound-based tasks efficiently. Additionally, studies show that VIPs can make accurate and reliable decisions (like those of normally sighted pedestrians) when navigating dangerous environments, such as crossing streets [242]. These findings combined lead to the belief that VIPs have the potential to effectively perceive threat levels based on auditory warnings and cues and take appropriate actions to avoid them.

Trade-off between In-Time Warning Communication and False Alarm Reduction: On the one hand, sending warning messages as soon as the CMOs are identified is intuitive for ensuring timely warning emissions. Nevertheless, due to the property of ARAware's object detector and tracker, the accuracy of CMO identification tends to be lower when using early frames in which CMOs have just been identified, and objects are typically located at a considerable distance from the camera. Accordingly, the accuracy of collision and risk prediction will also decrease. Additionally, moving objects initially identified as critical (CMOs) may change

direction before actually approaching the VIP. Considering these factors, immediate emitting warnings may lead to a significant number of false alarms, which could hinder the usability of ARAware scheme. On the other hand, it is necessary to emit early warnings to ensure that the VIP has enough time to avoid predicted collisions. To balance the trade-off between timely warning and false alarm reduction, the *warning emission time* for each CMO is determined based on both the *estimated collision time* and the *warning emission deadline*. This approach differs from existing warning emission methods [243, 28], which only consider the object-VIP distance, making them impractical and unsafe solutions. The reason is that in distance-based methods, warning messages for objects with the same distance from the VIP are emitted at the same time, neglecting the crucial factor of varying speeds. Objects moving at different speeds may collide with the VIP at different time points, which poses significant threats to the VIP's safety. Further, with distance-based warning emission methods, the VIP may only be alerted to fast CMOs once it is too late.

In detail, once a CMO is identified, as shown in Fig. 4.7, its object-VIP distance estimation Z_i^t is obtained using Eqs. (4.3), and its average speed estimation ${v'}_{avg_{O_i}}$ is calculated using Eq. (4.23). Then, the CMO's collision time is estimated as:

$$t'_{cl_i} = \frac{Z_i^t}{{v'}_{avg_{O_i}} + v'_{VIP}}, \tag{4.24}$$

where $v'_{VIP} = C_z$, represents the VIP's approximate speed, mentioned in Section 4.1.2. Note that the estimated collision time for any identified CMO is continuously updated whenever its trajectory is updated.

Subsequently, different *warning emission deadlines* for CMOs are set up based on their respective risk levels. It should be emphasised that these deadlines are determined based on the empirical study of this research and the analysis of human behaviour in emergent road-traffic situations explored in [244]. However, they can be customised according to the safety requirements of VIPs. The warning emission deadline of CMOs categorised under a particular risk level refers to a specific time point before the estimated collision time, before which the warning

message must be emitted. Specifically, the warning emission deadline is defined as:

$$t_{dl} = t_{ra} + \delta, \tag{4.25}$$

where t_{ra} is an expected reaction time to avoid the collision, and δ represents a safety margin factor that is specific to the CMO's risk level.

Based on the findings in [244], approximately 70% of road-traffic accidents can be prevented if they are perceived $1.69 \sim 2.08s$ before the collision occurs. For the more vulnerable VIP, the time t_{ra} is set to $5s$ to allow sufficient time to perceive audio messages and to take an appropriate corresponding reaction. VIPs' safety is the top priority; therefore, to provide additional time for reacting to CMOs, different values of δ are assigned to CMOs with varying levels of risk based on the empirical study conducted by this research. These values are $6s$ for 'severe', $4s$ for 'serious', $2s$ for 'minor', and $0s$ for 'minimal'; again, they can be adjusted based on the VIP's safety requirements. In this case, the warning message for an identified CMO is emitted only when its estimated collision time reaches its warning emission deadline, *i.e.*, $t'_{cl_i} \leq t_{dl}$. It is important to note that if a CMO's initial estimated collision time falls within its warning emission deadline, its warning message is emitted immediately.

Early Detection of Small Size CMOs Moving at a Medium-to-High Speed: The ARAware's primary objective is to detect CMOs as soon as possible to provide early warnings, especially for CMOs moving at a moderate to high speed. This is crucial to ensure the VIP's physical safety. Hence, for small-size CMOs such as motorcycles and bicycles, the detection of the CMOs' riders is considered. This is because the rider can be detected (as a 'person') at earlier frames than the CMO itself, which can be attributed to their different sizes. Riders are typically taller, thus, more visible than their motorcycles or bicycles in the earlier video frames. For the differentiation between a cyclist, a motorcyclist, and a pedestrian, as they all belong to the same class, classification thresholds (predefined speed limits) are established based on CMOs' estimated average moving speeds $v'_{avg_{O_i}}$ (in km/h). These thresholds are determined based on two factors: (1) analysing the average speed limits of such objects on residential roads [236, 237] and the typical walking speed of pedestrians on sidewalks [231, 238], and (2)

experimental observations. During the experiments, the researcher observed that achieving a walking speed of $10km/h$ and cycling speeds exceeding $20km/h$ demonstrated challenging for the volunteers, requiring considerable effort to achieve such speeds. Therefore, the thresholds are defined as follows:

$$CMO(v'_{avg_{O_i}}) = \begin{cases} motorcyclist, & v'_{avg_{O_i}} \geq 20, \\ cyclist, & v'_{avg_{O_i}} \geq 10 \text{ and } v'_{avg_{O_i}} < 20, \\ pedestrian, & v'_{avg_{O_i}} < 10. \end{cases} \tag{4.26}$$

By this approach, motorcycles moving at speeds from $10km/h$ to $40km/h$ are detected $1.80s \sim 7.20s$ earlier, and bicycles moving at speeds from $8km/h$ to $25km/h$ are detected $3.89s \sim 12.16s$ earlier. It is essential to emphasise that this method is only used to extend the distance at which motorcycles and bicycles are detected (discussed later in Section 6.1.2 of Chapter 6).

However, this method can lead to a misclassification of the CMO class (similar to what is discussed in the previous subsection of DisNet-based distance estimation). For instance, if a motorcycle is travelling at $15km/h$, the ARAware system might classify it as a bicycle in the earlier frames (when only the rider is initially detected). Note that misclassifications here do not impact the performance of ARAware since the misclassified CMOs will be rectified in subsequent frames when their detection becomes more apparent. Moreover, even if a misclassification does occur when a warning must be emitted, ARAware can still effectively warn about the threat because it considers not only the CMO's class, but also factors such as the CMO's speed, VIP-CMO collision time, and the VIP's reaction time, which are more critical for ensuring the safety of VIPs compared to the CMO's class. As part of future work, the researcher plans to re-train YOLOv8 to distinguish between 'cyclists' and 'motorcyclists' instead of recognising them as 'persons'.

Warning Message Duplication Removal: By extending the detection distance of motorcycles and bicycles through the detection of their riders, as mentioned above, a situation may arise where two bounding boxes are produced, one for the rider and another for the CMO, both of which fall under the same risk level. This can result in duplicate alerts being emitted. To

avoid such duplication, WEM utilises the *Intersection Over Union* (IOU) metric to estimate the overlapping between the bounding boxes as:

$$IOU_{cmo} = \frac{BB^t_{i_1} \cap BB^t_{i_2}}{BB^t_{i_1} \cup BB^t_{i_2}}, \tag{4.27}$$

where BB^t_i is the bounding box of CMO i at time t. One alarm is triggered if $IOU_{cmo} \geq \eta$, where η is a predefined threshold indicating an overlapping (in ARAware experiments, $\eta = 0.25$).

Warning Message Design:

To assist VIPs in accurately and effectively perceiving potential threats, the warning message for a CMO includes crucial information such as the CMO's *type*, *risk level*, *incoming direction*, and *estimated collision time*. This information is conveyed to the VIP through bone-conduction headphones to avoid blocking the surrounding sounds for the VIP's safety. To concisely perceive a CMO's direction, the CMO's incoming direction is simplified as one of three options: 'front', 'left', and 'right', using ${}^C X^{t_c}_{O_i}$ in Eq. 4.4. Considering object's size, the three directions are identified as follows:

- 'Front': ${}^C X^{t_c}_{O_i} \in [-1m, 1m]$ if CMO is a car, and ${}^C X^{t_c}_{O_i} \in [-0.5m, 0.5m]$ otherwise.
- 'Right': ${}^C X^{t_c}_{O_i} > 1m$ if CMO is a car, and ${}^C X^{t_c}_{O_i} > 0.5m$ otherwise.
- 'Left': ${}^C X^{t_c}_{O_i} < -1m$ if CMO is a car, ${}^C X^{t_c}_{O_i} < -0.5m$ otherwise.

Using 'front', 'left', and 'right' is considered beneficial for VIPs in threat warnings as these simple terms offer quick comprehension and reduce cognitive load in stressful situations.

Furthermore, based on the VIP's requirements, the ARAware can be customised to provide the CMO's incoming direction in degrees. This customisation gives a more precise indication of the threat's direction, and it can be achieved by follow:

$$\Theta = tan^{-1}\left(\frac{{}^C X^{t_c}_B - {}^C X^{t_c}_{O_i}}{{}^C Z^{t_c}_B - {}^C Z^{t_c}_{O_i}}\right). \tag{4.28}$$

Degrees offer a more precise representation of an object's direction compared to the three categories ('front', 'left', 'right') for VIPs requiring a highly specific understanding of threats. This increased precision is crucial when multiple objects approach from various directions, facilitating precise evasive manoeuvres.

4.1.3 MinD: Local Path Planning Scheme

This section presents the description of the local path planning problem and the mathematical formulation of the MinD scheme. As mentioned earlier, ARAware aims to effectively identify and notify VIPs about potential threats (CMOs). However, relying solely on warnings about CMOs may not be sufficient in all scenarios. VIPs possess diverse cognitive, physical, and mobility capabilities, and solely relying on intuition for navigation and threat avoidance can be hazardous. Additionally, heavy reliance on alerts can induce VIP anxiety, potentially hindering clear thinking, decision-making, and accurate perception, ultimately increasing the risk of unsafe reactions during CMO avoidance. Therefore, this research introduces MinD, an advanced mode that goes beyond just notifying VIPs of threats. MinD actively plans the most suitable route for VIPs, factoring in their physical capabilities, to safely avoid CMOs.

Problem Description

As the aim is to guide VIPs in avoiding CMOs of the targeted classes, *i.e.,* cars, motorcycles, bicycles, and pedestrians, in unfamiliar dynamic environments, this problem is classified as *local path planning.* To tackle this problem effectively, for each CMO previously identified by ARAware, a further estimation of its future movement, *i.e.,* position and velocity, as well as VIP's corresponding trajectories, is needed. Generally, the assumptions made here are similar to those made previously in Section 4.1.2 regarding the circular shape and the movement of the CMO and the VIP. However, some key differences arise due to the different nature of this problem, which will be discussed below.

As depicted in Fig. 4.10, the VIP occupies a circular zone C'_B, centred at B^{t_c} with a radius of r'_B

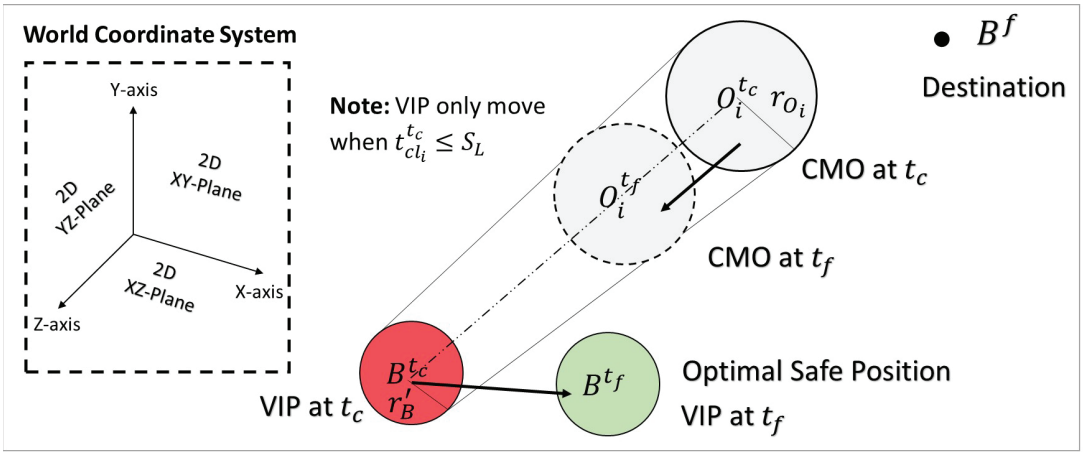


Figure 4.10: VIP Local Path Planning Problem.

at time t_c within the World Coordinate System (WCS).[3] Note that the WCS is employed here as it offers a global reference frame effectively encompassing the trajectories of both the VIP and the CMOs, thus facilitating accurate trajectory analysis and path planning. In this problem, the VIP moves forward towards a known final distention B^f at a velocity of $\boldsymbol{v_B} = |\boldsymbol{v_B}|\angle\theta'$, where $|\boldsymbol{v_B}|$ is the speed, and $\angle\theta'$ is the moving angle (the angle between the VIP moving direction and the Z-axis of the WCS). A CMO is located within a circle C_{O_i}, with a centre $O_i^{t_c}$ and a radius of r_{O_i} at time t_c, and moves to $O_i^{t_f}$ at velocity of $\boldsymbol{v_{O_i}} = |\boldsymbol{v_{O_i}}|\angle\theta$ at time t_f (next time in future frame). The VIP will be directed to move to B^{t_f} at time t_f only if two conditions are met: (1) a potential collision is identified (the object is moving towards the VIP, *i.e.,* classified as CMO using the method described in Section 4.1.2), (2) and the estimated collision time $t_{cl_i}^{t_c}$ between the CMO and the VIP at time t_c exceeds a predefined safety threshold S_L. If multiple CMOs are present, MinD prioritises avoiding the nearest threat. This is achieved by considering the CMO with the shortest estimated collision time, as outlined in Eq. (4.29). For the VIP's physical safety, r'_B should cover the entire width of the VIP, and r_{O_i} should cover the whole width of the CMO. Note that, in path planning, the safety radius r'_B should not be too large than the VIP's width. This is because finding a large space for the VIP to manoeuvre in a dynamic environment with many objects can be impractical and may limit the feasible paths for the VIP. Setting r'_B to the VIP's width ensures sufficient safe room for navigation while still accounting for the presence of CMOs and environmental constraints.

[3]It is a real-world 3D Cartesian coordinate system with a predefined origin.

Problem Formulation

Considering VIPs' limited physical abilities, such as walking at limited speed (generally $\leq 5km/h$ [231, 245]) and relying on auditory cues for environmental perception, path planning must be carefully designed to prioritise VIPs safety. It is crucial to early plan a safe path and allow sufficient time for VIPs to perceive acoustic guidance instructions and react safely, *i.e.,* follow the proposed path. This becomes especially important when high-risk CMOs, such as cars and motorcycles, are present. Further, preventing unnecessary transitions of VIPs with low-risk CMOs, such as bicycles and pedestrians, can improve the usability of the MinD scheme. Accordingly, MinD's path planning algorithm is activated based on a condition associated with the *minimum* estimated collision time between the VIP and the CMO at t_c as follows:

$$min(t_{cl_i}^{t_c}) \leq S_L, i = [1, .., n], \tag{4.29}$$

then,

$$t_{cl_i}^{t_c} = \frac{||O_i^{t_c} - B^{t_c}||_2}{|v_{O_i}^{t_c}| + |v_B^{t_c}|}. \tag{4.30}$$

where n represents the number of CMOs, and S_L is a pre-defined time threshold, in s, that is adjusted based on the class and speed of CMO i. The $B^{t_c} = (X, Y, Z)_B^{t_c}$ and $O_i^{t_c} = (X, Y, Z)_{O_i}^{t_c}$ are the *current* 3D positions of the VIP and CMO i at time t_c, respectively, in the WCS. The $v_B^{t_c} = (v_X, v_Y, v_Z)_B^{t_c}$ and $v_{O_i}^{t_c} = (v_X, v_Y, v_Z)_{O_i}^{t_c}$ are the *current* velocities of the VIP and CMO i at t_c, where v_X, v_Y, and v_Z are the velocity components of the X-axis, Y-axis, and Z-axis of the WCS, respectively, and they are defined as follows:

$$v_B^{t_c} = (\frac{X_B^{t_c} - X_B^{t_p}}{t_c - t_p}, \frac{Y_B^{t_c} - Y_B^{t_p}}{t_c - t_p}, \frac{Z_B^{t_c} - Z_B^{t_p}}{t_c - t_p}), \tag{4.31}$$

$$v_{O_i}^{t_c} = (\frac{X_{O_i}^{t_c} - X_{O_i}^{t_p}}{t_c - t_p}, \frac{Y_{O_i}^{t_c} - Y_{O_i}^{t_p}}{t_c - t_p}, \frac{Z_{O_i}^{t_c} - Z_{O_i}^{t_p}}{t_c - t_p}), \tag{4.32}$$

where $(X, Y, Z)_B^{t_p}$ and $(X, Y, Z)_{O_i}^{t_p}$ are the 3D coordinates of the *previous* positions (B^{t_p} and $O_i^{t_p}$) of the VIP and CMO i at t_p, respectively. Note that the sum of the absolute values of the VIP's and CMO's velocities is used in Eq.(4.30), assuming a worst-case scenario in which the

VIP and CMO are directly heading toward each other. This is to provide the VIP with the largest possible safety margin.

When the least value of $t_{cl_i}^{t_c}$ exceeds the threshold S_L, as shown in Eq. (4.29), the aim is to plan the shortest path that guarantees the VIP safely avoids various classes of CMOs moving at varying speeds while considering the aforementioned VIPs' limited capabilities. To achieve this, the VIP's next position, B^{t_f}, is optimised under six practical constraints that consider the VIP's abilities and the dynamic states of the CMOs to ensure a safe feasible path. These constraints are detailed below.

Constraint 1 (C_1): The *maximum travelling distance* (d_{trv}) that the VIP can travel while avoiding the CMO. This constraint considers the VIP's limited mobility and ensures they maintain proximity to the initial path, preventing disorientation or deviation from their intended destination. Mathematically, it can be expressed as follows:

$$d_{trv} \leq D_{max}, \tag{4.33}$$

$$||B^{t_f} - B^{t_c}||_2 \leq D_{max}, \tag{4.34}$$

where $B^{t_f} = (X, Y, Z)_B^{t_f}$ is the *optimal safe* 3D position of the VIP at t_f. The D_{max} represents a predefined maximum distance threshold in m.

Constraint 2 (C_2): The *maximum travelling time* (t_{trv}) the VIP needs to reach B^{t_f} safely according to their walking speed. This parameter can vary and can be tailored to the user based on walking measurements obtained, for example, by their mobile phone. Mathematically, it can be expressed as:

$$t_{trv} \leq T_{max}, \tag{4.35}$$

$$\frac{||B^{t_f} - B^{t_c}||_2}{|v_B^{t_f}|} \leq T_{max}, \tag{4.36}$$

where the $v_B^{t_f}$ represents the VIP's *next* velocity at t_f, and the T_{max} denotes the maximum expected time, in s, the VIP requires to travel from B^{t_c} to B^{t_f} successfully. The equivalent

expression of Eq.(4.36) is as follows:

$$||B^{t_f} - B^{t_c}||_2 \leq T_{max}|v_B^{t_f}|. \tag{4.37}$$

Constraint 3 (C_3): Ensuring VIPs' safety requires more than navigating them to an obstacle-free position. Due to the dynamic nature of CMOs, including their ability to adapt speed and change direction suddenly, predicting their *next* positions ($O_i^{t_f}$) and estimating the *time separation* between the positions of the VIP and all CMOs i at time t_f (in the future) is crucial. Besides, this time separation should be sufficiently large to provide an additional safety margin, ensuring ample time for the VIP to navigate to a new safe position even if a CMO unexpectedly changes direction towards them. Based on this, the *safe time separation* ($t_{sep_i}^{t_f}$) constraint is formulated as follows:

$$t_{sep_i}^{t_f} > s_l, \quad \forall i, i = [1, .., n], \tag{4.38}$$

$$\frac{||O_i^{t_f} - B^{t_f}||_2}{|v_{O_i}^{t_f}| + |v_B^{t_f}|} > s_l, \quad \forall i, i = [1, .., n], \tag{4.39}$$

$$O_i^{t_f} = O_i^{t_c} + v_{O_i}^{t_c}(t_f - t_c), \tag{4.40}$$

where $v_{O_i}^{t_f}$ is the CMO i's *next* velocity at t_f, and s_l is a predefined minimum time interval, measured in s, that must be maintained between B^{t_f} and $O_i^{t_f}$. The value of s_l is adapted according to the class and speed of the CMO i for a practical and safe solution. Specifically, with higher-risk CMOs such as cars and motorcycles, s_l should be large enough to guarantee sufficient safety for the VIP. The equivalent equation of Eq. (4.39) is as follows:

$$||O_i^{t_f} - B^{t_f}||_2 > s_l(|v_{O_i}^{t_f}| + |v_B^{t_f}|), \quad \forall i, i = [1, .., n] \tag{4.41}$$

To successfully avoid collision with the CMO, the VIP must move outside the *critical zone*. As it shown in Fig. 4.11, the *critical zone* is the shadowed potential threat area confined between the two tangents $\overleftrightarrow{O_1B_1}$ and $\overleftrightarrow{O_2B_2}$ of circles C_{O_i} and C'_B intersected at point A'. To achieve this, two *collision avoidance* constraints (C_4 and C_5) are defined below. Here, it is solely focused on the 2D XZ-plane of the WCS (see Fig. 4.10), which represents the ground plane shared between

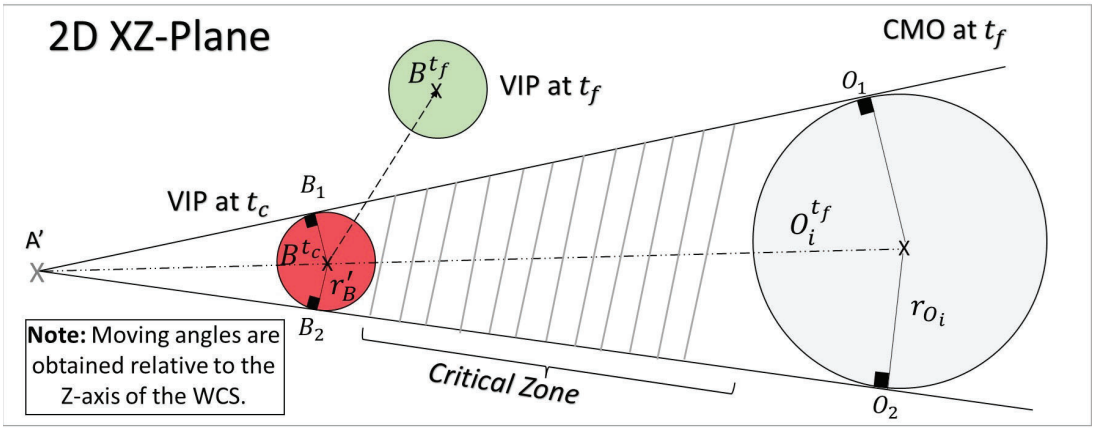


Figure 4.11: CMO Collision Avoidance.

the VIP and the CMO. This decision is based on the assumption that camera movement along the Y-axis is negligible and can be disregarded due to its perpendicular orientation to the ground plane.

Constraint 4 (C_4): To redirect the VIP away from the path of a CMO i, the *moving angle* constraint is imposed, which can be formulated as follows:

$$\beta' < (\phi'_i - \alpha'_i) - \delta_a, \quad \forall i, i = [1, .., n], \tag{4.42}$$

$$\alpha'_i = tan^{-1}\left(\frac{1}{\sqrt{\frac{{d'_i}^2}{(r_{O_i} - r'_B)^2} - 1}}\right), \tag{4.43}$$

$$\phi'_i = tan^{-1}\left(\frac{X_{O_i}^{t_f} - X_B^{t_c}}{Z_{O_i}^{t_f} - Z_B^{t_c}}\right), \tag{4.44}$$

$$\beta' = tan^{-1}\left(\frac{X_B^{t_f} - X_B^{t_c}}{Z_B^{t_f} - Z_B^{t_c}}\right), \tag{4.45}$$

where α'_i is the angle between $\overline{A'O_i^{t_f}}$ and $\overline{A'O_1}$, see Fig. 4.11 (note that the derivations of the above angles are similar to those explained in Section 4.1.2; however, the derivations are explicitly provided in Appendix .2), $d'_i = |\overline{B^{t_c}O_i^{t_f}}|$ is the Euclidean distance between B^{t_c} and $O_i^{t_f}$ and can be calculated as $d'_i = \sqrt{(X_{O_i}^{t_f} - X_B^{t_c})^2 + (Z_{O_i}^{t_f} - Z_B^{t_c})^2}$, ϕ'_i is the angle of the displacement vector from B^{t_c} to $O_i^{t_f}$ (relative to the Z-axis of the WCS), and β' is the angle of the displacement

vector from B^{t_c} to B^{t_f}. The δ_a is an *angle safety margin*; it is introduced to ensure that the VIP's moving direction is sufficiently away from the CMO's moving direction, i.e., δ_a should be large enough to prevent frequent transitions or overlaps between the VIP's trajectory and that of the CMO. This is particularly important when CMOs are detected early at a significant distance as the object appears smaller, resulting in a smaller α'_i value. The value of δ_a can be adjusted according to the VIP's safety requirements. The equivalent expression of Eq. (4.42) is:

$$(X_B^{t_f} - X_B^{t_c}) - (tan((\phi'_i - \alpha'_i) - \delta_a)(Z_B^{t_f} - Z_B^{t_c})) < 0, \quad \forall i, i = [1, .., n]. \tag{4.46}$$

Another form of the *moving angle* constraint is:

$$\beta' > (\phi'_i + \alpha'_i) + \delta_a, \quad \forall i, i = [1, .., n], \tag{4.47}$$

which is equivalent to:

$$(X_B^{t_f} - X_B^{t_c}) - (tan((\phi'_i + \alpha'_i) + \delta_a)(Z_B^{t_f} - Z_B^{t_c})) > 0, \quad \forall i, i = [1, .., n]. \tag{4.48}$$

Both Eq. (4.46) and Eq. (4.48) can be utilised interchangeably to guide the movement of the VIP either to the left or to the right, depending on the relative position of the CMO. For comprehensive VIP safety, considering the moving angles of all CMOs, not just the closest CMO, is necessary. This ensures that the MinD is aware of potential threats posed by multiple objects approaching from different angles, thus effectively planning a free-collision path. Besides, only considering the moving angle of the closest CMO can result in frequent re-directions and transitions of the VIP, which can significantly reduce the usability of the scheme (discussed later in Chapter 7). In other words, incorporating the moving angles of all CMOs into the path planning optimisation process mitigates the risk of oscillation issues, e.g., redirecting the VIP into the path of one CMO while attempting to avoid another; see Fig. 4.12.

Note that when using a conventional camera with a limited field of view, redirecting the VIP's path to avoid a potential threat (CMO) may unintentionally cause the CMO to disappear from the camera's view. To avoid this and ensure VIP safety, the use of a 360-degree camera could

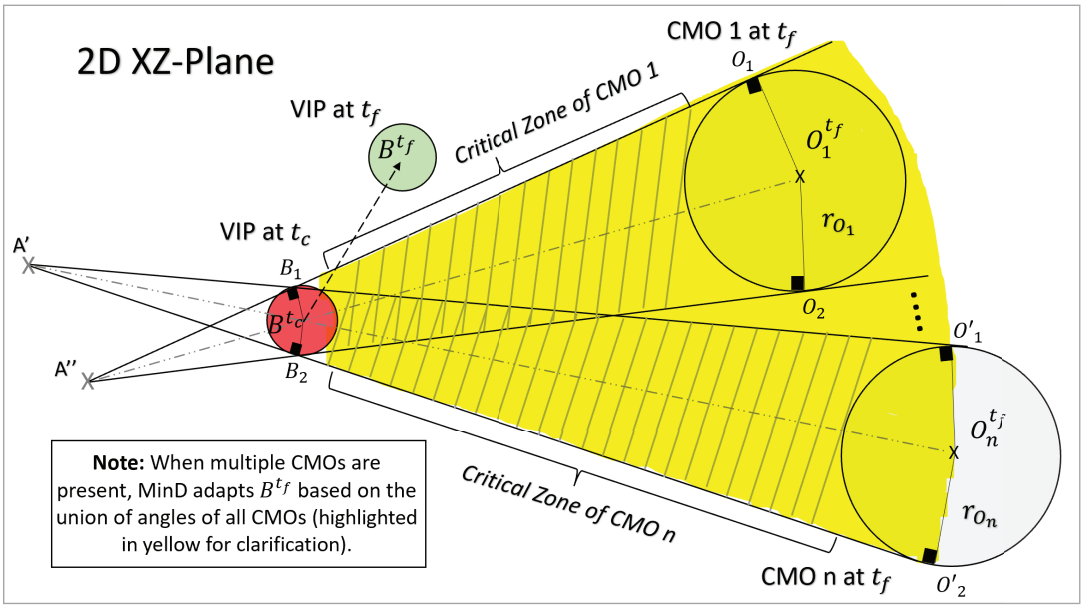


Figure 4.12: Collision Avoidance for Multiple CMOs. Note that the VIP is directed based on the combined angles of all CMOs.

be considered for the future integration of ARAware and MinD.

Constraint 5 (C_5): To guarantee that the VIP moves outside of the threat area, *i.e.*, the *critical zone* in Fig. 4.11, a constraint on the VIP's *moving distance* is imposed as follows:

$$||B^{t_f} - B^{t_c}||_2 > (r'_B + r_{O_i}) + \delta_d, \tag{4.49}$$

where δ_d is a *distance safety margin* that ensures that the VIP moves outside the critical zone with sufficient distance. It can be customised based on the VIP's safety requirements. This safety margin is used because optimal paths may not always be horizontal lines; they can also be diagonal lines. Further, objects often do not move in straight lines; their paths may include slight changes in thier moving direction. As a result, the scheme may detect multiple collisions between the VIP and the same object within a short period of time. Therefore, introducing δ_d helps prevent unnecessary frequent transitions of the VIP.

It should be noted that when multiple CMOs exist, this constraint only considers the closest CMO, which is the most dangerous. This is to mimic human avoidance behaviour, which is adaptable and prioritises the most immediate threat. Humans typically avoid multiple objects

gradually, making continuous adjustments to their trajectory rather than abrupt and large movements. This allows for smoother navigation and reduces the risk of collisions with other nearby obstacles. Therefore, this constraint focuses on progressively guiding the VIP away from the closest object, followed by the next closest, until all nearby CMOs are safely avoided. This gradual approach aligns with human behaviour and considers the VIP's physical limitations, preventing impractical large movements, particularly in dynamic environments with numerous objects and frequent changes.

Constraint 6 (C_6): To ensure that VIP continually advances toward the final destination B^f, the *forward moving* constraint is defined as follows:

$$||B^{t_f} - B^f||_2 \leq ||B^{t_c} - B^f||_2. \tag{4.50}$$

Based on all the above constraints definitions, a non-convex VIP-specific local path planning problem, the *non-convex MinD*, is formulated as follows:

$$\min_{B^{t_f}} f(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 \quad | \left\{ \min(t_{cl_i}^{t_c}) \leq S_L \quad \forall i, i = [1, .., n] \right\} \tag{4.51}$$

s. t.

$c_1(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 \leq D_{max}$ (maximum travelling distance)

$c_2(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 \leq T_{max}|v_B^{t_f}|$ (maximum travelling time)

$c_3(B^{t_f}) = ||O_i^{t_f} - B^{t_f}||_2 > s_l(|v_{O_i}^{t_f}| + |v_B^{t_f}|), \quad \forall i, i = [1, .., n]$ (safe time separation)

$c_4(B^{t_f}) = (X_B^{t_f} - X_B^{t_c}) - (tan((\phi_i' - \alpha_i') - \delta_a)(Z_B^{t_f} - Z_B^{t_c})) < 0, \quad \forall i, i = [1, .., n]$ (moving angle)

$c_5(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 > (r_B' + r_{O_i}) + \delta_d$ (moving distance)

$c_6(B^{t_f}) = ||B^{t_f} - B^f||_2 \leq ||B^{t_c} - B^f||_2$ (forward moving)

In the Eq. (4.51), the objective function $f(B^{t_f})$ and all the constraints are convex except C_3 and C_5. The both C_3 and C_5 are non-convex constraints because they violate the standard formulation of the convex optimisation problem, *i.e.,* $C(x) \leq 0$ [246]. For this reason, the MinD becomes a non-convex problem. When a problem lacks convexity, it cannot be solved using conventional methods.

Algorithm 1 The SCP of the convex MinD sub-problem

Given initial point $B_0^{t_f}$, iteration k, and tolerance $\epsilon > 0$
for $i = 0, 1, \ldots, k$ **do**
  Solve the convex sub-problem in Eq. (4.54)
  if Eq. (4.54) is feasible **then**
    $B_{k+1}^{t_f} \leftarrow B^{t_f}$
    if $||B_{k+1}^{t_f} - B_k^{t_f}||_2 < \epsilon$ **then**
      Stop, it is converged
    else
      $B_k^{t_f} \leftarrow B_{k+1}^{t_f}$
      Continue
    end if
  else
    Initialize a new point $B_k^{t_f}$
  end if
end for

Problem Convex Approximation

For solving the *non-convex MinD* problem in Eq. (4.51), the *Sequential Convex Programming* (SCP) method [246] is utilised. The SCP involves transforming the original non-convex problem into a sequence of convex sub-problems. This is achieved by replacing the non-convex constraints or objective function with convex approximations centred around an approximation point. These sub-problems are then solved iteratively using a convex algorithm, see Algorithm 1. The algorithm process starts with an initial guess and continues until convergence is reached (*i.e.*, finding an acceptable feasible solution), satisfying a predefined convergence criteria. Therefore, the non-convex constraints C_3 and C_5 are approximated and transformed to their respective convex counterparts by utilising the *first-order Taylor* series approximation. This means that the constraints are approximated with a linear function that is tangent to the constraint at a particular point. In the following, the equivalent convex constraints C_3' and C_5' of the C_3 and C_5, respectively:

$$C_3'(B^{t_f}, B_k^{t_f}) = ||O_i^{t_f} - B_k^{t_f}||_2 + [\frac{-(X_{O_i}^{t_f} - X_{B_k}^{t_f})}{||O_i^{t_f} - B_k^{t_f}||_2}, \frac{-(Y_{O_i}^{t_f} - Y_{B_k}^{t_f})}{||O_i^{t_f} - B_k^{t_f}||_2}, \frac{-(Z_{O_i}^{t_f} - Z_{B_k}^{t_f})}{||O_i^{t_f} - B_k^{t_f}||_2}](B^{t_f} - B_k^{t_f})$$
$$> s_l(|v_{O_i}^{t_f}| + |v_B^{t_f}|), \quad \forall i, i = [1, .., n], \quad \text{(safe time separation)} \tag{4.52}$$

$$C_5'(B^{t_f}, B_k^{t_f}) = ||B_k^{t_f} - B^{t_c}||_2 + [\frac{(X_{B_k}^{t_f} - X_B^{t_c})}{||B_k^{t_f} - B^{t_c}||_2}, \frac{(Y_{B_k}^{t_f} - Y_B^{t_c})}{||B_k^{t_f} - B^{t_c}||_2}, \frac{(Z_{B_k}^{t_f} - Z_B^{t_c})}{||B_k^{t_f} - B^{t_c}||_2}](B^{t_f} - B_k^{t_f})$$
$$> (r_B' + r_{O_i}) + \delta_d. \quad \text{(forward moving)} \tag{4.53}$$

Here, $B_k^{t_f}$ represents an approximation point used to approximate a solution centered around it. The approximation point helps in guiding the process of finding an appropriate solution within its immediate vicinity. Initially, it can be set to any random value that generates a feasible solution for the approximated sub-problem. Based on the above approximations, the Eq. (4.51) is re-formulated to the following convex sub-problem:

$$\min_{B^{t_f}} f(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 \quad | \left\{\min(t_{cl_i}^{t_c}) \leq S_L \quad \forall i, i = [1, .., n]\right\} \tag{4.54}$$

s. t.

$$c_1(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 \leq D_{max} \quad \text{(maximum travelling distance)}$$

$$c_2(B^{t_f}) = ||B^{t_f} - B^{t_c}||_2 \leq T_{max}|v_B^{t_f}| \quad \text{(maximum travelling time)}$$

$$C_3'(B^{t_f}, B_k^{t_f}) = ||O_i^{t_f} - B_k^{t_f}||_2 + [\frac{-(X_{O_i}^{t_f} - X_{B_k}^{t_f})}{||O_i^{t_f} - B_k^{t_f}||_2}, \frac{-(Y_{O_i}^{t_f} - Y_{B_k}^{t_f})}{||O_i^{t_f} - B_k^{t_f}||_2}, \frac{-(Z_{O_i}^{t_f} - Z_{B_k}^{t_f})}{||O_i^{t_f} - B_k^{t_f}||_2}](B^{t_f} - B_k^{t_f})$$

$$> s_l(|v_{O_i}^{t_f}| + |v_B^{t_f}|), \quad \forall i, i = [1, .., n] \quad \text{(safe time separation)}$$

$$c_4(B^{t_f}) = (X_B^{t_f} - X_B^{t_c}) - (tan((\phi_i' - \alpha_i') - \delta_a)(Z_B^{t_f} - Z_B^{t_c})) < 0, \quad \forall i, i = [1, .., n] \text{ (moving angle)}$$

$$C_5'(B^{t_f}, B_k^{t_f}) = ||B_k^{t_f} - B^{t_c}||_2 + [\frac{(X_{B_k}^{t_f} - X_B^{t_c})}{||B_k^{t_f} - B^{t_c}||_2}, \frac{(Y_{B_k}^{t_f} - Y_B^{t_c})}{||B_k^{t_f} - B^{t_c}||_2}, \frac{(Z_{B_k}^{t_f} - Z_B^{t_c})}{||B_k^{t_f} - B^{t_c}||_2}](B^{t_f} - B_k^{t_f})$$

$$> (r_B' + r_{O_i}) + \delta_d \quad \text{(moving distance)}$$

$$c_6(B^{t_f}) = ||B^{t_f} - B^f||_2 \leq ||B^{t_c} - B^f||_2 \quad \text{(forward moving)}$$

To solve the convex sub-problem outlined in Eq. (4.54), a random initial value $B_0^{t_f}$ is assigned, and a convergence criterion is established using a tolerance variable $\epsilon > 0$. This convergence criterion is used to halt the iteration k and and find the global optimal solution B^{t_f}. The complete algorithm for solving Eq. (4.54) is summarised in Algorithm 1.

Note that since MinD is a non-convex optimisation problem by nature, there is no guarantee that local minima will be avoided even when approximated using SCP. However, encountering

local minima at a certain time slot does not compromise MinD's overall effectiveness. This is because the VIP can still successfully navigate around obstacles, which is the primary goal of ensuring their safety, even in the presence of local minima.

In conclusion, this chapter has comprehensively outlined the design of the AR-AwareAvoid scheme, focusing on its two key components: ARAware and MinD. Furthermore, a detailed explanation of the VIP local path-planning problem and its mathematical foundation was provided. The next chapter will shift focus to the implementation of both ARAware and MinD, alongside the evaluation methodologies, metrics, datasets, and simulations meticulously chosen to validate the schemes' effectiveness.

Chapter 5

Implementation and Evaluation

Building upon the previously discussed schematic design of ARAware and MinD schemes, which together form the comprehensive scheme, AR-AwareAvoid, this chapter is dedicated to their implementation and evaluation. It explores the details of implementing both schemes and the evaluation methodologies, metrics, datasets, and simulations employed to validate their effectiveness.

5.1 ARAware Implementation

This section presents the prototype implementation of ARAware, real-world experiments, experimental datasets, and evaluation metrics.

5.1.1 ARAware's Prototype Implementation

For a comprehensive evaluation under real-world settings, a tangible prototype of ARAware was constructed using commercially available devices, as illustrated in Fig. 5.1. For WVM, an RGB camera, capturing video at a resolution of 640×480 and a frame rate of $30 fps$, was mounted on a stabiliser. This stabiliser is utilised to counteract body movements and ensure steady footage; it is securely attached to a comfortable chest belt mount strap, allowing the VIP's hands to remain



Figure 5.1: The ARAware's Real-World Prototype.

free. This setup allows the camera to be elevated to the VIP's eye level to provide a close, natural human field of view of the surroundings. The camera, measuring $50mm \times 28mm \times 0.9mm$ and weighing $5g$, has a fixed but unknown focal length. The camera is connected to a laptop, carried in a backpack by the VIP, via a USB cable for video transmission. The laptop has an Intel Core i9-9900K CPU, an NVIDIA GeForce RTX 2080 GPU, and $64GB$ RAM, hosting the following modules: MOT (Moving Object Tracker), PM (2D-to-3D Projection Module), CRP (Collision and Risk Predictor), and WEM (Warning Emission Module). While a laptop is used in ARAware experimentation, bringing this prototype to production would likely involve transitioning to smaller, lighter equipment. Bone-conduction headphones are used to convey warnings and ensure the VIPs remain aware of ambient sounds for their own safety. The WEM module communicates warnings to the headphones using Bluetooth technology.

All components of ARAware are implemented in `Python 3.7.7`. Specifically, the YOLOv8-based object detector was implemented using the Ultralytics Python library 8.0.1453.[1] Public implementations[2,3] based on TensorFlow-gpu 2.1.0 and Keras 2.3.1 were adapted and employed

[1] https://docs.ultralytics.com/modes/predict/#inference-sources
[2] https://github.com/theAIGuysCode/yolov3_deepsort
[3] https://github.com/guanjianyu/DisNet

for the Deep SORT-based object tracking and DisNet-based distance estimation components, respectively. Using camera calibration based on `OpenCV 4.8.1.78`,[4] the camera focal length in pixels was estimated as $f_x = 644.83px$ and $f_y = 644.07px$, and the camera principal point in pixels was obtained as $(p_x, p_y) = (322.97px, 238.56px)$.

Inspired by [7], the predefined dimensions of the classes targeted by this research (discussed earlier in Section 4.1.2 of Chapter 4) were set as follows:

- $Cars$: C_h= 160cm, C_w= 170cm, and C_d= 400cm,
- $Motorbikes$: C_h= 124cm, C_w= 65cm, and C_d= 190cm,
- $Bicycles$: C_h= 105cm, C_w= 40cm, and C_d= 175cm,
- $Persons$: C_h= 175cm, C_w= 55cm, and C_d= 30cm.

Empirical studies based on extensive real-world experiments (using the ARAware's prototype) were conducted to determine proper radius setups for r_{O_i} and r_B while fulfilling the consideration mentioned earlier in Section 4.1.2 of Chapter 4. The resulting radius setups are outlined as follows:

- $Cars$: $r_{O_i} = 0.9m$, $r_B = 3m$,
- $Motorbikes$: $r_{O_i} = 0.45m$, $r_B = 1.75m$,
- $Bicycles$: $r_{O_i} = 0.35m$, $r_B = 1.25m$,
- $Persons$: $r_{O_i} = 0.25m$, $r_B = 0.75m$.

5.1.2 Evaluation Methodology

To conduct a comprehensive evaluation, extensive real-world experiments[5] were first conducted using the ARAware's prototype. These experiments aimed to collect two evaluation datasets

[4] https://docs.opencv.org/master/dc/dbb/tutorial_py_calibration.html

[5] This study is approved by the Science Engineering and Technology Research Ethics Committee (SETREC) in the Imperial College London, see Appendix .3.

(discussed in the next section) by capturing videos of lightly choreographed (following predetermined scenarios) volunteers driving cars, motorcycles and bicycles and walking at different speeds. During these recordings, the researcher mimicked a VIP by holding the ARAware's prototype to capture the videos. Note that the researcher collected her own datasets to comprehensively evaluate ARAware performance as existing publicly available datasets (like COCO and KITTI) lack the necessary ground truth (GT) to effectively assess ARAware's functionalities, such as CMO identification and CMO risk estimation and classification.

Ethical Considerations For Collecting Experimental Dataset. In adherence to ethical standards, this research obtained approval from the Science Engineering and Technology Research Ethics Committee (SETREC) at Imperial College London, see Appendix .3. Before participation, all volunteers involved provided informed consent after being fully briefed on the purpose, procedures, and potential risks of the experiments. Participants were assured of their right to withdraw from the study at any time without consequence. To protect confidentiality, all personal information was anonymised, and data were stored securely on password-protected Imperial College servers accessible only to authorised individuals, i.e., the researcher and her supervisor. Additionally, participants were treated with the utmost respect, and efforts were made to ensure their comfort and safety during experiments. To minimise any potential harm or discomfort, rigorous fieldwork risk assessment was conducted, and appropriate measures were implemented to mitigate potential risks. For example, experiments were conducted on authorised private lands under controlled and supervised environments to ensure the safety of the participants and the public. Before commencing each experiment, the experimental environments and all necessary scenario settings were ensured to be safe for both participants and the researcher. Moreover, the researcher underwent training in *Fieldwork First Aid* with Marlin Training at Imperial College London to ensure preparedness in handling any unexpected situations or emergencies that may arise during experiments. Emergency response plans were also established to promptly address any unforeseen incidents. The experiments' results were reported in aggregate form to protect participant anonymity, with no identifiable information or images provided. Note that if including images is essential for clarity, participants' faces will be blurred to ensure privacy.

Following the data collection phase, the performance of ARAware's components was validated separately to obtain in-depth insight into their individual performance (discussed later in Chapter 6). Then, the overall performance of ARAware on real-time and accurate CMO identification and classification, as well as on warning emission, was assessed. For ARAware's CMO identification, a further evaluation was conducted under different k-frame skipping settings, *i.e.,* only one frame was fed to the CMO identification for every k frame (i.e., $k = 3$, 5, 15, 30 and 50), discussed later in Chapter 7. This is to assess how skipping more frames (higher k values) impacts the accuracy of CMO identification and thus can help in determining the acceptable level of computational reduction (to cope with limited computational resources) based on the required accuracy level for a specific scenario. When used in CMO identification, the accuracy and efficiency of YOLOv3 and YOLOv8 were also compared. Finally, the ARAware scheme was compared with the most comparable state-of-the-art approach, DEEP-SEE [28] (in Chapter 6).

5.1.3 Experimental Dataset

The experiments involved recruiting up to five volunteers to walk, cycle, and drive in outdoor spaces while the researcher held the ARAware prototype (determined the camera height as $1.52m$), and two experimental datasets were collected for the ARAware's performance evaluation as follows:

- **Video-Based Multiple Objects Tracking (VMOT):**

  The VMOT dataset comprises 96 short video clips (totaling 50.926K frames and 256.6 MB in size) with a combined duration of $28min$ captured in different private outdoor lands during both day and night (excluding overly bright or extremely low-light situations), see Fig. 5.2. Private lands were chosen to facilitate the convenient setup of experiments. Using private spaces allowed the researcher to easily establish precise configurations, such as identifying the radius of C_B and C_{O_i} (for object i), as well as the critical zone (see Fig. 4.7) on the ground using ropes to thoroughly and precisely evaluate the ARAware's CMO collision prediction method. This level of control and flexibility would have been

challenging, if not impossible, to achieve in public spaces due to limitations on controlling the presence of other objects and people. Another reason for choosing private lands is to ensure the researcher's, volunteers' and the public's safety. The video clips contain different numbers of objects (up to five) belonging to the targeted classes (car, motorcycle, bicycle and pedestrian) positioned at various distances (up to $20m$) from the VIP and moving at slow speeds ($< 10km/h$). For all VMOT clips, the ground truth (GT) annotations were manually identified using the MATLAB Ground Truth Labeler,[6] where objects moving within the critical zone (marked by yellow ropes in Fig. 5.2) were labelled as CMOs, while those moving outside the critical zone were labelled as non-CMOs. This labelling was performed at the frame level. The numbers of CMOs (*i.e.*, GT_c) and non-CMOs (*i.e.*, GT_{Nc}) under each class are detailed in Table 6.5 in Chapter 6, with a total of 539 GT_c and 217 GT_{Nc}.

- **Video-Based Collision and Risk Prediction (VCRP):** The VCRP dataset consists of 44 video clips (totaling 23.281K frames and 141.1 MB in size) with a combined duration of $15min$ and recorded in private outdoor lands, similar to that in VMOT (during both day and night, excluding situations with excessive brightness or insufficient light). Each clip focuses on a single object (*i.e.*, a CMO), such as a car, motorbike, bicycle, or pedestrian, which is initially located at different distances (up to $100m$) and moving uniformly at a fixed speed (up to $60km/h$) toward the camera. It is important to note that while ARAware can handle objects with varying speeds, the speed of the CMOs was made uniform throughout the video only for the purpose of evaluation, *i.e.*, to facilitate obtaining accurate ground truth speed measurements because precise speed measurement equipment was not available. For the object in each video clip, both the initial distance between the object and the VIP (*i.e.*, GT_D) and the object's moving speed (*i.e.*, GT_S) were manually determined, thus known (see Table 6.3 in Chapter 6). In GT_S labelling, cars and motorcycles are labelled using their own speedometers, while for bicycles and pedestrians, the time taken to travel a predefined distance was measured, and then the speed was calculated by dividing the distance by the time taken. Furthermore, each object

[6] https://uk.mathworks.com/help/driving/ground-truth-labeling.html



Figure 5.2: Illustration of Video Clip Collection and Experimental Environment Setting-up.

in the clips was tagged with a risk level label based on the risk classification scheme shown in Fig. 4.9. These risk level labels serve as the ground truth for evaluating ARAware's performance in CMO risk assessment, with categories including severe, serious, minor, and minimal.

5.1.4 Evaluation Metrics

To evaluate ARAware's performance, the following evaluation metrics are employed:

- **Mean Average Recall** (mAR): It is used to indicate the capability of ARAware to successfully identify all objects of different classes in terms of either objects' types (*i.e.*, car, motorcycle, bicycle, and pedestrian) or their associated risk levels (*i.e.*, severe, serious,

minor, and minimal). It can be defined as follows:

$$mAR = \frac{1}{M}\sum_{m=1}^{M}(\frac{1}{K}\sum_{k=1}^{K}\frac{TP_k^l}{TP_k^l + FN_k^l}), \tag{5.1}$$

where M represents the total number of video clips processed, K is the number of frames in each video clip m, TP_k^l denotes the number of objects belonging to class l correctly identified in frame k, and FN_k^l denotes the number of class l's objects missed in frame k.

- **Mean Average Precision** (mAP): It is used to indicate the capability of ARAware to correctly identify objects belonging to different classes. It can be defined as follows:

$$mAP = \frac{1}{M}\sum_{m=1}^{M}(\frac{1}{K}\sum_{k=1}^{K}\frac{TP_k^l}{TP_k^l + FP_k^l}), \tag{5.2}$$

where FP_k^l represents the number of non-class l's objects falsely identified as class l's objects in frame k. The IOU is used with a 0.5 threshold to classify the correct and false identifications *i.e.*, TP_k^l and FP_k^l.

- **Multiple Object Tracking Accuracy** (MOTA) [247]: It is used to indicate the capability of ARAware to correctly track objects in their trajectories. It can be expressed as follows:

$$MOTA = \frac{1}{M}\sum_{m=1}^{M}(1 - \frac{\sum_k(FP_k + FN_k + IDSW_k)}{\sum_k GT_k}), \tag{5.3}$$

where $IDSW_k$ is the number of object mismatches in a trajectory in frame k, and GT_k is the number of ground truth objects in frame k.

- **Multiple Object Tracking Precision** (MOTP) [247]: It is utilised to indicate the capability of ARAware to precisely localise objects in video frames, which can be formulated as follows:

$$MOTP = \frac{1}{M}\sum_{m=1}^{M}(\frac{\sum_{i,k} D_k^i}{\sum_k C_k}), \tag{5.4}$$

where D_k^i represents the measure of match between the ground truth bounding box and the tracked bounding box of object i in frame k, and C_k is the number of matched objects

in frame k. The IOU is used with a 0.5 threshold to classify the matched ground truth and tracked bounding box pairs.

- **Absolute Error** (E_A): It is used to measure the difference between the estimated and actual values of speed, distance and collision time of objects in each video clip:

$$E_A = X_{i,m}^{Est} - X_{i,m}^{Act}, \tag{5.5}$$

where $X_{i,m}^{Est}$ and $X_{i,m}^{Act}$ are the estimated and actual values (speed, distance, and collision time) of object i in video clip m, respectively. Note that the absolute value (||) in Eq. (5.5) is omitted to provide a more explicit representation. This omission is intentional and explicitly indicates the relationship between the estimated and the actual values. This means a negative value in the equation signifies that the estimated value is smaller than the actual value, while a positive value indicates the opposite.

- **End-to-End mean FPS** (EmFPS): It is used to indicatie the capability of ARAware to achieve real-time CMO identification and classification:

$$EmFPS = \frac{1}{M}\sum_{m=1}^{M} FPS = \frac{1}{M}\sum_{m=1}^{M} \frac{K_m}{t_m^e - t_m^s}, \tag{5.6}$$

where K_m denotes a clip m's total number of frames, and t_m^e and t_m^s are the end time and the start time of the clip m's end-to-end processing time, respectively.

5.2 MinD Implementation

MinD is also implemented in Python 3.7.7. (using the CVXPY 1.1.18 library for convex optimisation) to ensure consistency with the ARAware scheme. To evaluate MinD's performance in a highly practical setting under diverse scenarios while prioritising health and safety, an empirically parameterised simulator was developed, where its parameters were learned from the extensive real-world ARAware experimentation. Due to the inherent challenge of ensuring collision-free VIP navigation in dynamic real-world environments with high-risk moving objects

(e.g., cars, motorcycles), prior rigorous evaluation within a simulated environment is crucial. This helps to mitigate safety concerns, foster an in-depth understanding of MinD's performance, and facilitate refinement before real-world implementation. Before developing the simulator, the researcher began with a comprehensive analysis of ARAware's experimental scenarios and obtained outcomes. Specifically, the movements of objects involved in ARAware's real-world experiments were analysed, their collision times were calculated, and the emission deadlines of early alarms for the VIP to avoid CMOs effectively were studied. The parameters of setup ARAware were also carefully examined. This is to accurately parameterise the MinD's simulator and set practical parameters (*e.g.,* r'_B and T_{max} for VIP, and r_{O_i}, δ_d, S_L, s_l, for each CMO class) that would map a real-world implementation (*i.e.,* by leaving enough time for the VIP to react, especially with high-speed objects). Simulation allows safe 'what-if' experimentation to fully evaluate MinD effectively and in a near-to-realistic way.

Empirically Parameterised Simulator Setup: To this end, a 2D Python-based simulator was developed that allows for the creation of diverse random experimental scenarios. It employs basic geometric algorithms to simulate the CMOs' movements and the VIP's manoeuvres without compromising health and safety, with parameters derived from ARAware's real-world experiments. The simulator was run on Ubuntu 18.04.4 LTS, using an Intel Core i9-9900K CPU, an NVIDIA GeForce RTX 2080 GPU, and $64GB$ RAM. The simulation is conducted in a 2D environment; this is because the researcher assumes that the Y-axis coordinates of both the VIP and CMOs are zero. This assumption is justified by the fact that the Y-axis is perpendicular to the ground plane (a 2D XZ-plane), which serves as the shared plane of movement for both the VIP and CMOs (as illustrated in Fig. 4.10). To effectively evaluate the MinD algorithm, the Monte Carlo method [46] was utilised to generate various random simulations. Scenarios were generated for each class of CMO, where the CMO was set to move towards the VIP at eight different speeds: $11km/h \sim 108km/h$ for cars and motorcycles, $7km/h \sim 54km/h$ for bicycles, and $3.4km/h \sim 36km/h$ for pedestrians. A total of 100 samples were conducted for each speed, with random positions assigned to the CMO. This resulted in a total of 3,200 samples, with 800 samples for each CMO class. The simulated environments included car parks, driveways, pedestrian crossings, and walking footpaths, with dimensions of

Table 5.1: The Parameter Settings of MinD's Simulation Experiments for Each Class.

CMO	S_L(s)	s_l(s)	r_{O_i}(m)	δ_d(m)
Car	12	5	0.9	1.1
motorcycle	12	5	0.45	0.65
Bicycle	9	4	0.35	0.55
Pedestrian	7	3	0.27	0.47

Table 5.2: The Parameter Settings of MinD's Simulation Experiments for All Classes.

B^{t^*}(m)	B^f(m)	r_B(m)	D_{max}(m)	T_{max}(s)	δ_a	k	ϵ
(0,0)	(0,500)	0.25	10	5	30°	100	0.001

$20m \times 500m$. The initial position of the VIP was set at $B^{t^*} = (0, 0)$, and the final destination was set at $B^f = (0, 500)$. For simplicity, it is assumed that the CMOs move directly towards the VIP without changing their direction and speed ($|v_{O_i}^{t_f}| == |v_{O_i}^{t_c}|$). The VIP is expected to manoeuvre such threats by changing the moving direction. However, evaluating MinD with CMOs adapting their speed and direction is left for further work. It is also assumed that the VIP walks at a constant speed of $5km/h$ ($|v_B^{t_f}| == |v_B^{t_c}|$) forward towards the predefined known final destination B^f. Further work will evaluate MinD when the VIP adapts their speed as well.

For parameter setting, differentiation is made between CMOs classes in terms of risk level (*i.e.,* cars and motorcycles are high-risk CMOs, and bicycles and pedestrians are low-risk CMOs). Table 5.1 presents the parameter settings that are adjusted based on a CMO's risk level. To ensure the VIP's safety, larger threshold values (*i.e.,* S_L and s_l) have been set for cars and motorcycles compared to bicycles and pedestrians. This allows for early planning of a safe path for the VIP, considering their slow-moving speed and auditory perception, and leaving a greater time separation between the VIP and CMO at t_f. This time separation gives the VIP more time to react and navigate safely, especially when the CMO updates its direction toward the VIP after starting to avoid it. Conversely, for bicycles and pedestrians, which pose less risk, the threshold values have been reduced to avoid unnecessary transitions for the VIP. Table 5.2 presents the uniform parameters applied to all CMO classes.

5.2.1 Evaluation Methodology

To evaluate the performance of MinD in the absence of specific VIP local path planning algorithms, a comparison with two alternative methods was conducted using the evaluation metrics described in Section 5.2.2. The first method, serving as a baseline, is *Notification-Based Random Decision* (NBRD). In NBRD, it is assumed that the VIP makes random collision avoidance decisions solely based on alerts issued by a navigation assistance system that lacks path-planning capabilities. Here, the VIP can move forward within a limited range ($\leq 10m$) in any direction between 45° and 135° to manoeuvre around CMOs. The second method employed for the comparison is the state-of-the-art robot local path planning algorithm: improved *Artificial Potential Field* (APF) [45]. This improved APF addresses limitations of traditional APF, such as local minima and oscillation. The APF algorithm is widely utilised in robotics motion planning due to its simplicity, effectiveness, and real-time performance.

For comparison, the researcher implemented the improved version of the APF algorithm described in [45]. This version specifically addresses a common challenge in path planning, local minima, by integrating the Simulated Annealing algorithm. In the local minima, a robot gets stuck and fails to avoid an obstacle located along the same path toward its destination as the resultant force of obstacle repulsion and destination attraction becomes zero (cancel each other out). To ensure fair comparison, the researcher carefully tuned the parameters of the improved APF algorithm based on extensive trials while considering the effects of parameter settings studied in [248]. The parameters are set as follows: a repulsive coefficient $k_{rep} = 10^9$, a gravitational coefficient $k_{att} = 0.1$, a control annealing temperature $T = 10$ with an escape step size $E_s = 4$, and a repulsion influence distance $r_r = s_l(|v_{O_i}^{t_f}| + |v_B^{t_f}|)$. The value of r_r is adaptive based on the speed of the VIP, the speed of the CMO, and the value of s_l. Note that the value of s_l is set according to the CMO's risk level, as in the case of the MinD's parameter settings; see Section 5.2. It is important to note that k_{rep} is set at a very large value compared to k_{att} to prioritise the safety of the VIP; the main objective is to keep the VIP as far away from potential threats as possible. Additionally, to accelerate the speed of local minima escape processing, a larger value for E_s compared to [45] is chosen. In the experiments, the primary

concern is with the proposed path for avoiding the CMO rather than the entire path to reach the destination as planned by the improved APF algorithm.

5.2.2 Evaluation Metrics

For MinD's performance evaluation, the following metrics are defined:

- **Travelling Distance ($'d_{trv}$)**: This metric is used to measure the length of the path that the VIP needs to traverse to successfully avoid potential threats. It can be expressed as follows:

$$'d_{trv} = ||B^{t_f} - B^{t_c}||_2. \tag{5.7}$$

- **Safe Time Separation ($'t_{sep}$)**: This metric is used to indicate the time separation between the VIP and CMO in the optimal path, serving as an indicator of the safety of the optimal position B^{t_f}. It can be expressed as follows:

$$'t_{sep} = \frac{||O_i^{t_f} - B^{t_f}||_2}{|v_{O_i}^{t_f} + v_B^{t_f}|}. \tag{5.8}$$

- **Collision rate (r_{coll})**: This metric is used to indicate the effectiveness of MinD in planning collision-free paths. Here, a collision is considered if the $'t_{sep}$ is less than the threshold sl that is set based on the CMO's risk level, see Section 5.2. Then, the r_{coll} is calculated as follows:

$$r_{coll} = \frac{1}{'M} \sum_{'m=1}^{'M} (\frac{C}{'N}), \tag{5.9}$$

  where $'M$ is the total number of the samples that represent CMO moving speeds for a particular CMO class (*i.e.,* 8 speeds), $'N$ is the total number of random samples running under each CMO moving speed (*i.e.,* 100 samples), and C is the number of collisions.

- **Computation Time (C_t) and Convergence Iteration (C_{Itr})**: These metrics are used to evaluate MinD's efficiency in generating real-time path plans. The computation time measures the time taken by the algorithm to calculate a path from the current

position of the VIP to a safe position. A shorter computation time indicates faster path planning. The convergence iteration quantifies the number of iterations required for MinD to converge and generate a collision-free path. A lower number of iterations suggests quicker convergence and more efficient path planning.

In conclusion, this chapter has delved into the intricate details of implementing ARAware and MinD, the two proposed schemes. It has thoroughly explored the chosen methodologies, metrics, datasets, and simulations employed to evaluate their effectiveness comprehensively. The following chapter will present and discuss the results obtained from validating and evaluating ARAware and MinD. It will also conduct a comparative analysis, contrasting them against the relevant state-of-the-art approaches in assisting VIP mobility and robot local path planning.

Chapter 6

Results and Discussions

After discussing the implementation details, evaluation methods, and metrics outlined in the previous chapter, this chapter presents and discusses the results obtained from validating and evaluating the proposed schemes, ARAware and MinD. It commences by validating ARAware's core functionalities separately, e.g., moving object tracking (MOT), distance estimation, and speed estimation. Subsequently, it evaluates ARAware's end-to-end performance in CMO identification, CMO classification, and warning notification. Then, an in-depth comparison with DEEP-SEE, the closest state-of-the-art approach, is provided. Moreover, MinD's performance is evaluated by analysing travelling distance, time separation, collision rate, and efficiency. These results are compared against a baseline method, Notification-Based Random Decision (NBRD), and the state-of-the-art robot local path planning algorithm, Artificial Potential Field (APF).

6.1 ARAware Results and Discussions

6.1.1 MOT Validation

To validate the effectiveness of the MOT module, the VMOT dataset is fed to the MOT, and then the intermediate results of both the YOLOv8-based object detector and the Deep

Table 6.1: The Performance of YOLOv8-Based Object Detector on VMOT With Two Different Sizes of YOLOv8.

Model	Dataset	mAP_{50}	mAR
YOLOv8m	VMOT	97.61%	96.45%
YOLOv8m [95]	COCO [249]	50.20%	n/a
YOLOv8n	VMOT	93.68 %	90.15%
YOLOv8n [95]	COCO [249]	37.30 %	n/a
YOLOv8n [253]	KITTI [138]	89.30 %	n/a
YOLOv8n [223]	CrowdHuman [250]	82.00 %	n/a

Table 6.2: The Performance of Deep SORT-Based Object Tracker on VMOT.

Model	Dataset	MOTA	MOTP
Deep SORT (YOLOv8m)	VMOT	94.55%	80.88%
Deep SORT (YOLOV5l) [254]	UA-DETRAC [251]	84.40%	77.40%
Deep SORT (Faster R-CNN) [224]	MOT16 [252]	61.40%	79.10%

SORT-based object tracker are collected and analysed, illustrated in Tables 6.1 and 6.2, respectively. These results were obtained when the MOT detects and tracks all captured objects, including both CMOs and non-CMOs, in VMOT. Considering the performance of YOLOv8 and Deep SORT on other publicly available datasets such as COCO [249], KITTI [138], CrowdHuman [250], UA-DETRAC [251], and MOT16 [252], which are treated as acceptable in practice, it is obvious that the performance of the MOT surpasses this acceptable level on VMOT. These datasets were selected for comparison due to their wide acceptance in the research community and their comprehensive coverage of various object detection and tracking scenarios. Additionally, YOLOv8m (medium variant) exhibits a slight performance advantage over YOLOv8n (nano variant) on VMOT, with a difference of approximately 4% in mAP and 6% in mAR. This suggests that the larger model offers better detection capabilities. Therefore, for ARAware's evaluation, the YOLOv8m variant is chosen as the detector to leverage its potentially superior object detection accuracy.

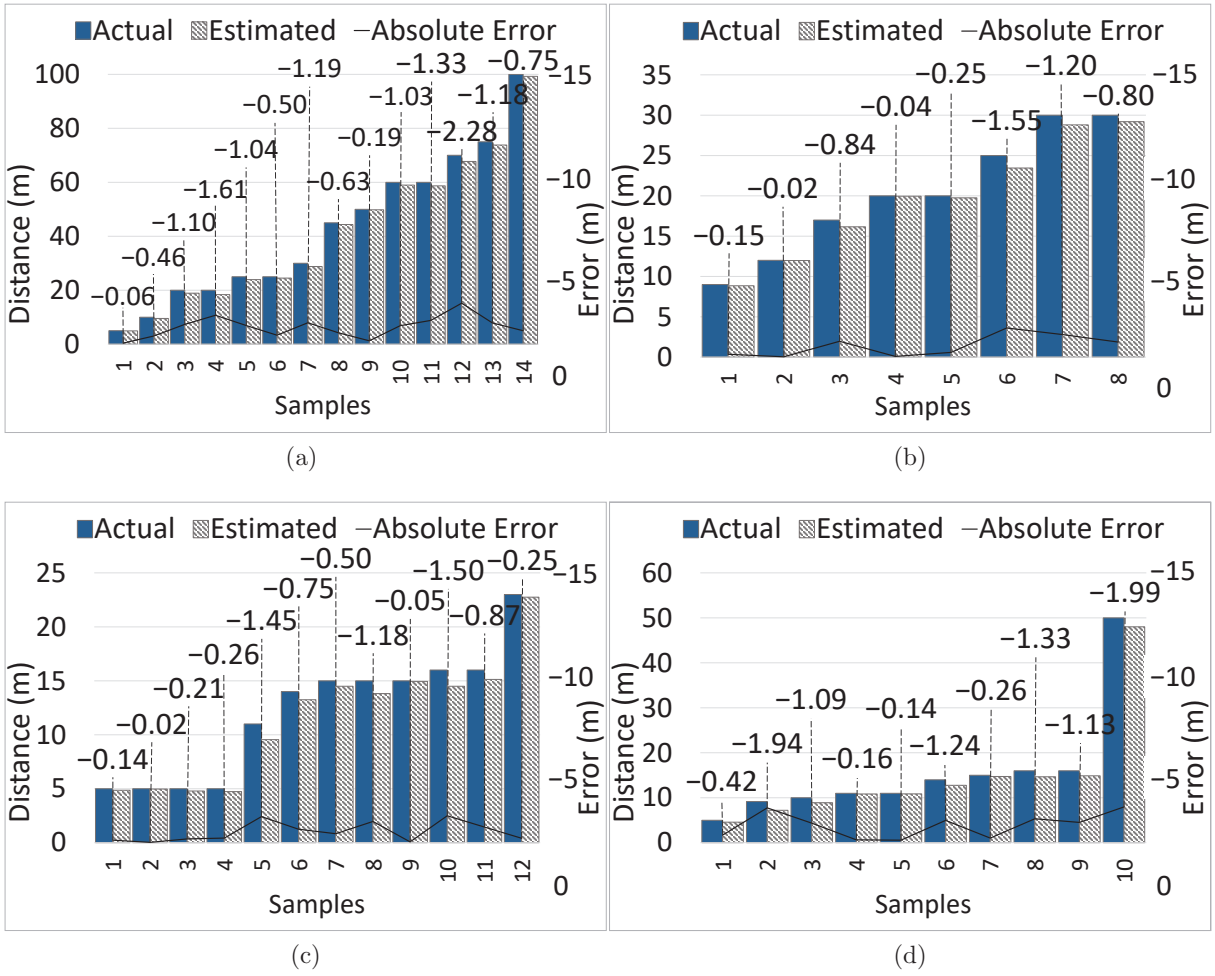


Figure 6.1: The Validation of Distance Estimation of All Four Classes in VCRP (44 Samples in Total). (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Samples, Y-axis (left): Distance, Y-axis (right): Error.

6.1.2 Validation of Distance and Speed Estimation

To validate the performance of ARAware's distance and speed estimation methods discussed in Section 4.1.2, the VCRP dataset is fed to ARAware prototype, and then the intermediate results of distance and speed estimation were collected and analysed, specifically the estimated object-VIP distance Z_i^t and the estimated object speed $v'_{avg_{O_i}}$.

Distance Estimation

Distance estimation results were collected for each class of CMOs separately. In particular, the initial actual object-VIP distance, the estimated distance, and the absolute error E_A of the estimated distance were recorded for each class. Additionally, the maximum distance that ARAware was able to detect for each CMO class was identified.

Fig. 6.1a presents the distance estimation results for 14 samples of the car class positioned at various distances ranging from $5m$ to $100m$. Car distance estimation exhibits a maximum observed E_A of -2.28m, with an average E_A of -0.95m. As mentioned in Section 5.1.4 of Chapter 5, a negative sign is solely used to indicate that the estimated distance falls below the actual value. For the motorcycle class, Fig. 6.1b illustrates the results of 8 samples at distances ranging from $9m$ to $30m$, achieving the maximum E_A of -1.55m and the average E_A of -0.61m. For the bicycle class, Fig. 6.1c shows the results of 12 samples at distances ranging from $5m$ to $23m$. The maximum and average E_A of bicycle distance estimation are -1.50m and -0.60m, respectively. Finally, Fig. 6.1d displays the results of the pedestrian class, involving ten samples with distances ranging from $5m$ to $50m$. The maximum and average E_A values of pedestrian distance estimation are -1.99m and -0.97m, respectively. Pedestrians' absolute error boundaries are higher than other CMOs. This is because pedestrians have articulated body parts like arms and legs that can move independently in various directions. The independent movement of such body parts causes changes in the sizes of bounding boxes surrounding pedestrians in images, affecting distance estimation accuracy.

Based on the results, it was observed that two factors influence the accuracy of distance estimation. The first factor is the precision of the predefined average dimensions (height, width, and depth) of a CMO of a particular class, discussed in Section 4.1.2 of Chapter 4. The second factor is the tightness or alignment of bounding boxes around the detected CMOs. As the object bounding box becomes looser (with more expanded boundaries around the object), the estimated distance tends to be shorter than the actual distance; consequently, the distance estimation error increases. The distance estimation results demonstrate that ARAware's pre-trained YOLOv8-based detector is reliable even without re-training it with the VCRP dataset.

The average E_A of the distance estimation results over all CMOs' classes is $-0.48m$, which is negligible compared to the overall average of the ground truth distances, *i.e.*, $22.67m$. This indicates that the predefined dimensions the researcher sets for different CMOs' classes are accurate enough.

Based on the VCRP dataset, the maximum detectable distances for each object class are as follows: $100m$ for cars, $30m$ for motorcycles, $23m$ for bicycles, and $50m$ for pedestrians, see Table 6.3 (E_R in the table refers to the relative error). Knowing maximum detectable distances helps to determine the earliest warning time achievable by the ARAware scheme for each class of CMOs and informs warning notification deadlines. In ARAware experiments, it was observed that different classes of objects have different maximum detectable distances. This is primarily attributed to their size differences. Cars, the widest among all classes, can be detected up to $100m$ away. Despite pedestrians having an average width of only $45 - 55cm$, their taller height (averaging $170 - 175cm$) allows them to be detected up to $50m$ away. In this research's experiments, motorcycles and bicycles have a smaller width than cars and a shorter height than pedestrians; therefore, they are detected at shorter distances.

Table 6.3: Performance of Distance and Speed Estimation on VCRP.

Samp.	CMO	C_{st}	GT_S(km/h)	$Est._S$(km/h)	SD_S(km/h)	GT_D(m)	E_A(m)	E_R(m)
1	Car	M	7	7.26	0.13	5	-0.06	-0.01
2		S	10	11.05	3.13	10	-0.46	-0.05
3		S	13	14.34	4.81	20	-1.10	-0.06
4		M	25	26.83	2.09	20	-1.61	-0.08
5		M	25	27.49	3.62	25	-1.04	-0.04
6		S	30	34.72	3.56	25	-0.50	-0.02
7		M	30	34.46	3.31	30	-1.19	-0.04
8		M	35	38.82	1.44	45	-0.63	-0.01
9		S	40	45.60	3.03	50	-0.19	-0.00
10		M	40	46.15	3.33	60	-1.03	-0.02
11		M	40	45.97	2.40	60	-1.33	-0.02
12		M	45	51.67	2.74	70	-2.28	-0.03
13		S	50	55.54	6.38	75	-1.18	-0.02
14		S	60	68.19	2.86	100	-0.75	-0.01
15	Motorcycle	M	10	11.09	0.32	9	-0.15	-0.02
16		S	15	16.57	6.49	12	-0.02	-0.00
17		M	20	22.28	2.11	17	-0.84	-0.05
18		S	25	27.76	5.63	20	-0.04	-0.00
19		M	25	27.91	3.57	20	-0.25	-0.01
20		M	30	35.22	2.48	25	-1.55	-0.06
21		S	35	40.64	3.57	30	-1.20	-0.04
22		S	40	46.13	4.59	30	-0.80	-0.03
23	Bicycle	S	8	8.34	3.50	5	-0.14	-0.03
24		M	8	8.59	0.82	5	-0.02	-0.00
25		S	10	10.94	1.66	5	-0.21	-0.04
26		S	11	11.91	2.34	5	-0.26	-0.05
27		M	11	11.74	1.74	11	-1.45	-0.13
28		M	12	13.98	0.65	14	-0.75	-0.05
29		S	12	13.60	0.85	15	-0.50	-0.03
30		S	16	17.02	4.09	15	-1.18	-0.08
31		M	16	17.37	2.04	15	-0.05	-0.00
32		M	17	18.30	4.22	16	-1.50	-0.09
33		M	18	19.61	2.61	16	-0.87	-0.05
34		S	25	27.40	3.59	23	-0.25	-0.01
35	Pedestrian	S	1.5	1.63	0.08	5	-0.42	-0.08
36		M	2	2.85	0.15	9.15	-1.94	-0.21
37		M	3	3.45	0.19	10	-1.09	-0.11
38		S	4	4.52	0.97	11	-0.16	-0.01
39		M	5	6.33	0.95	11	-0.14	-0.01
40		S	6	7.16	1.23	14	-1.24	-0.09
41		M	7	8.75	0.48	15	-0.26	-0.02
42		S	8	8.97	1.97	16	-1.33	-0.08
43		S	9	10.58	0.19	16	-1.13	-0.07
44		M	10	11.38	0.81	50	-1.99	-0.04

According to the findings from [244], pedestrians need a reaction time of $1.69 \sim 2.08s$ to avoid approximately 70% of perceived traffic-road collisions. Therefore, considering cars' average speed limit on urban roads (*i.e.*, $38 \sim 45km/h$ [239]), the researcher believes that the $100m$ detectable distance of cars is sufficient as it enables ARAware to emit an alarm $8 \sim 9.47s$ earlier than the potential collision. For motorcycles and bicycles, the detection of their riders (as persons, as mentioned in Section 4.1.2 of Chapter 4) at a distance of $50m$ helps to extend the time of alarm emission, where ARAware can warn $4 \sim 4.73s$ and $8.99 \sim 11.99s$ before potential collisions, for motorcycles and bicycles moving at average speeds of $38 \sim 45km/h$ and $15 \sim 20km/h$, respectively. It is clear that, compared to [244], ARAware leaves significantly more time for the VIP to react and avoid predicted collisions.

Speed Estimation

Speed estimation results were collected for all classes of CMOs under both static and moving camera (as well as the VIP) conditions. This involved collecting actual and estimated average speeds and calculating the absolute error E_A between them for each class of CMOs separately.

Static Camera: Fig. 6.2a displays the outcomes of six car samples moving at speeds from $10km/h$ to $60km/h$. At a $60km/h$ speed, the maximum E_A recorded is $8.19km/h$ and the average E_A across all car samples is $4.41km/h$. Fig. 6.2b shows the results of four samples from the motorcycle class, moving at speeds from $15km/h$ to $40km/h$. The maximum E_A observed is $6.13km/h$ when the motorcycle moves at $40km/h$, and the average E_A is $4.03km/h$. Fig. 6.2c illustrates the outcomes of six bicycle samples travelling from $8km/h$ to $25km/h$. The E_A varies between $0.34km/h$ and $2.40km/h$, with an average of $1.20km/h$. Finally, Fig. 6.2d demonstrates five pedestrian samples walking at speeds from $1.5km/h$ to $9km/h$, with E_A ranging from $0.13km/h$ to $1.58km/h$ and an average of $0.87km/h$.

Moving Camera: During the experiments, the VIP (and the camera) maintained a reasonable moving speed ($\leq 3km/h$). Fig. 6.3a displays the outcomes of eight car samples, with moving speeds ranging from $7km/h$ to $45km/h$. The maximum E_A observed is $6.67km/h$ when the car moves at $45km/h$, and the average E_A is $3.96km/h$. Fig. 6.3b shows the results of four

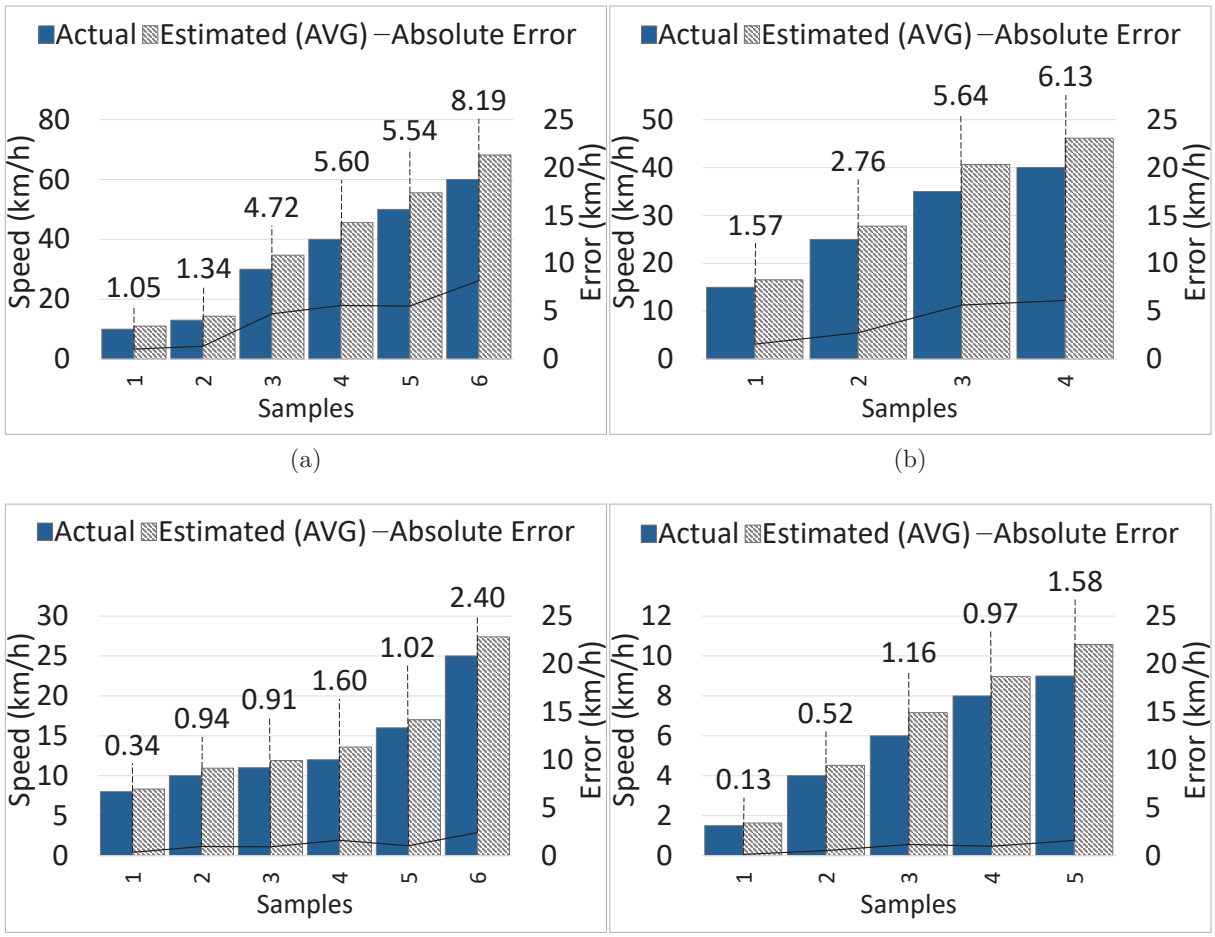


Figure 6.2: The Validation of Speed Estimation of All Four Classes in VCRP With Static Camera (21 out of 44 Samples, in Total). (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Samples, Y-axis (left): Speed, Y-axis (right): Error.

motorcycle samples moving from $10km/h$ to $30km/h$. The maximum E_A recorded is $5.22km/h$ when the motorcycle travels at $30km/h$; the average E_A is $2.88km/h$. For the bicycle class, Fig. 6.3c presents six samples travelling from $8km/h$ to $18km/h$, where E_A varies between $0.59km/h$ and $1.98km/h$, achieving an average of $1.27km/h$. For the pedestrian class, Fig. 6.3d illustrates the outcomes of five samples moving at speeds from $2km/h$ to $10km/h$, with E_A rates ranging from $0.45km/h$ to $1.75km/h$ and an average of $1.15km/h$. Table 6.3 summarises all the speed estimation results achieved using static and moving cameras (C_{st} denotes the camera status, and SD_S represents the standard deviation).

The analysis reveals a direct correlation between the increasing speed of CMOs and growing

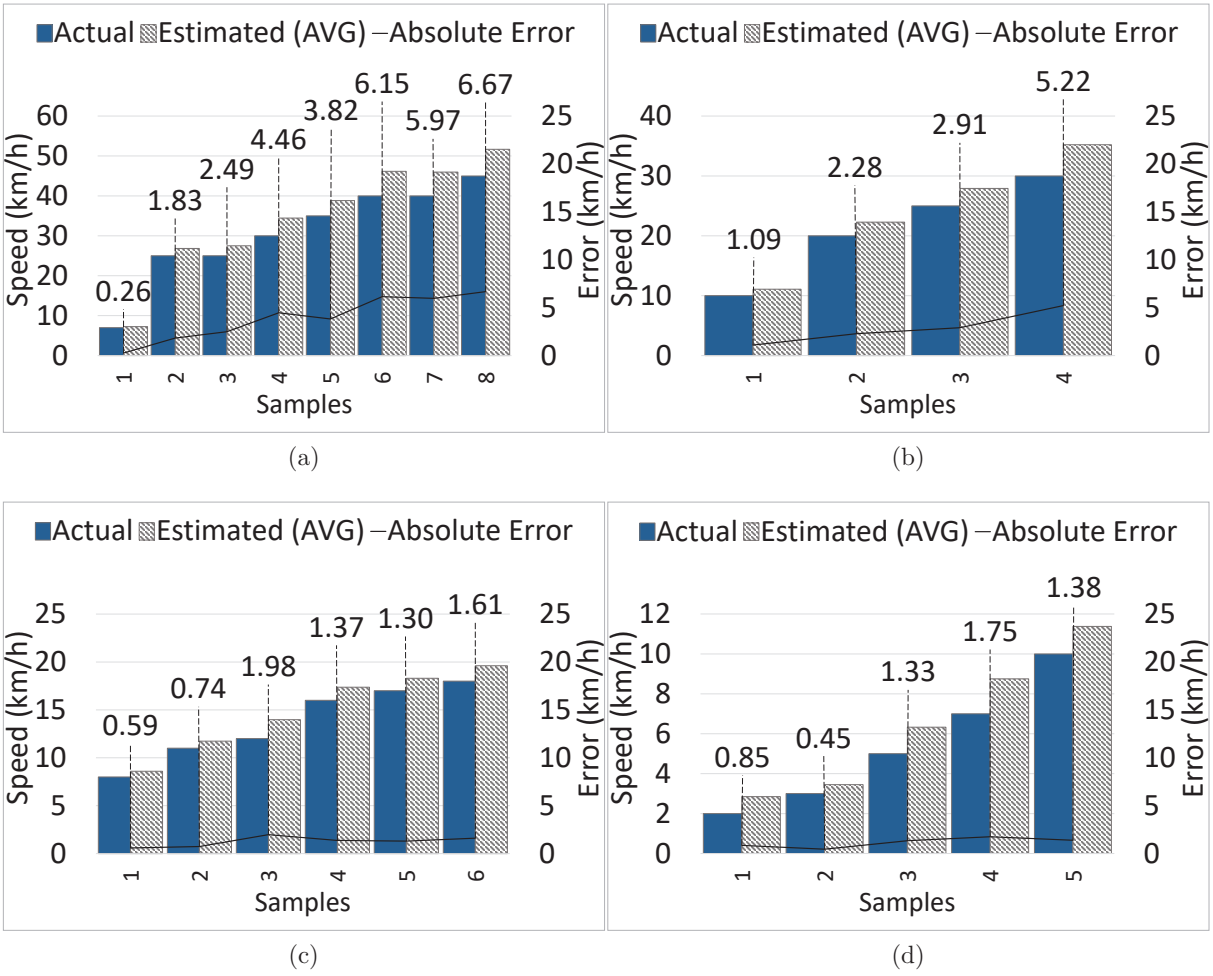


Figure 6.3: The Validation of Speed Estimation of All Four Classes in VCRP With Moving Camera (23 out of 44 Samples, in Total). (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Samples, Y-axis (left): Speed, Y-axis (right): Error.

speed estimation errors. With static and moving cameras, high-speed CMOs (cars and motorcycles) exhibit relatively higher errors than slower-moving objects under the same and other classes. Conversely, low-speed CMOs (bicycles and pedestrians) experience minimal estimation errors, and there is no noticeable difference in error rates among different CMO classes (such as bicycles and pedestrians) moving at low to moderate speeds (e.g., $< 25km/h$).

Three requirements are generally considered necessary for accurate speed estimation with ARAware. Firstly, distinct and good enough features (having enough detail to be easily identifiable and effectively matched) within the surrounding environment are required. Secondly, a secure and stable camera mounting is necessary to avoid camera shake or instability. Lastly, the

VIP's moving speed should be reasonably slow (*e.g.,* $\leq 3km/h$). This requirement stems from the reliance of the speed estimation method on extracting features from the background, which can be affected by camera motion. This requirement is deemed acceptable due to its alignment with previous research on VIP walking patterns [231] that indicate that VIPs typically walk at speeds below $5km/h$, with a noticeable reduction in speed when navigating complex or unfamiliar surroundings; therefore, this will not hinder their mobility. The average E_A of speed estimation results across all classes (for both static and moving camera cases) is $2.47km/h$, which is a relatively low error (12.90%) compared to the overall average of the ground truth speeds ($19.14km/h$). Considering that, in all experiments, the absolute error of speed estimation is positive, indicating that the CMOs' estimated speeds are always slightly higher than the actual speeds, the researcher accepts this low error as it leads to estimated collision times that are shorter than the actual collision times. This enables earlier notifications to be emitted, which gives the VIP more time to react.

6.1.3 Performance of ARAware

To validate the effectiveness of ARAware's design, its performance in real-time CMO identification and classification was evaluated. Specifically, the VMOT and VCRP datasets were fed to ARAware for end-to-end (E2E) CMO identification and classification, respectively. For overall efficiency, ARAware manages to achieve a 32 EmFPS, shown in Table 6.4, and process the incoming $30fps$ video stream in real-time with an average E2E processing time of $0.031s$, well within the real-time processing threshold of $\leq 0.033s$ ($33ms$) outlined in Chapter 1.

CMO Identification

Table 6.4 and Table 6.5 shows the overall and detailed results of CMO identification, respectively. As observed in both tables, when identifying all types of CMOs, ARAware achieves an overall mAP of 88.20% and an overall mAR of 97.26%. The analysis reveals that cars achieve the highest mAP score, followed by motorcycles, bicycles, and pedestrians, see Table 6.5. Pedestrians exhibit the lowest mAP due to their slower movement compared to other

Table 6.4: Overall Performance of ARAware.

Component	Over. mAP	Over. mAR	EmFPS
CMO identification CMO classification	88.20% 91.69%	97.26% 100%	32 fps



Figure 6.4: Illustration of CMO Identification on VMOT.

Table 6.5: Performance of ARAware's CMO Identification on VMOT.

CMO	GT_c	GT_{Nc}	TP	FP	FN	TN	mAP	mAR
Car	182	65	174	19	8	46	91.61%	95.62%
Motorcycle	112	31	108	19	4	12	88.96%	96.62%
Bicycle	77	35	75	17	2	18	86.44%	96.97%
Pedestrian	168	86	167	46	1	40	85.81%	99.83%
Total	539	217	524	101	15	116	88.20%	97.26%

classes of CMOs. Since the time difference between captured frames is negligible ($0.033s$), the positions of slow-moving objects, such as pedestrians, experience minimal changes or remain the same. This minimal change in position affects the precision of determining the object's moving direction (*i.e.*, angle β_i of vector $O_i^{t_p}\vec{O}_i^{t_c}$ in Fig. 4.7). Consequently, this leads to an increase in false positive cases (FP), ultimately resulting in a lower mAP score. Moreover, as the general moving speed decreases for different classes of CMOs (*i.e.*, from cars to pedestrians), the mAR increases due to a decrease in missed cases (false negatives, FN). This suggests



Figure 6.5: Illustration of CMO classification On VCRP.

that CMO speed is the primary factor influencing the mAR. However, it's important to note that CMO identification is evaluated in every frame, and missed cases were limited to a few frames (negligible time, just less than a tenth of a second) primarily due to the MOT module (i.e., Deep SORT tracker). This very short object's missing do not impact the ARAware's overall performance, as ARAware successfully identified all CMOs within all video clips of the VMOT dataset without entirely missing any CMO throughout an entire clip. Fig. 6.4 demonstrates ARAware's ability to identify CMOs across real-world scenarios with varying numbers of CMOs.

CMO Classification

Table 6.4 and Table 6.6 show the overall and detailed results of CMO classification, respectively. Based on the results, ARAware achieves an overall mAP of 91.69% and an overall mAR of 100% when classifying identified CMOs into different risk levels, as illustrated in Fig. 4.9. The results demonstrate the effectiveness of ARAware in classifying CMOs, especially for high-risk threats, where cars and motorcycles achieve an mAP of 99.32% and 96.01%, respectively. Although pedestrians and bicycles exhibit lower mAP scores compared to cars and motorbikes, their performance is more than acceptable as they are low-risk objects. Fig. 6.4 illustrates ARAware's capability to estimate the risk level of CMOs.

Table 6.6: Performance of ARAware's CMO classification On VCRP.

CMO	TP	FP	mAP	Over. mAP
Car	422	4	99.32%	91.69%
Motorcycle	105	7	96.01%	
Bicycle	159	47	88.77%	
Pedestrian	737	194	82.66%	

Table 6.7: Performance of ARAware's Warning Emission.

CMO	R_l	t_{cl} (s)	t'_{cl} (s)	$Pri._w$	$Ord._w$	$Emi._w$
Car	Serious	10	7.38	2	2	Imm.
Moto.$_R$	Serious	5.40	2.70	2	1	Imm.
Bic.$_R$	Minor	9.95	7.60	3	3	$t'_{cl} \leq 7s$
Ped.	Minimal	18.86	15.21	4	4	$t'_{cl} \leq 5s$

Warning Notification

Table 6.7 presents the performance of ARAware in emitting warnings for a video containing four CMOs with varying collision times and risk levels. The table includes details such as the CMOs' actual risk levels R_l, actual collision times t_{cl}, estimated collision times t'_{cl}, warning priorities assigned by ARAware $Pri._w$, the order in which the warning messages were sent $Ord._w$, and when the warning messages were actually sent by ARAware $Emi._w$. The table clearly demonstrates that the t'_{cl} for all classes of CMOs is consistently less than the t_{cl}, with an average E_A of $-2.83s$. This indicates that ARAware effectively protects the VIP's safety by issuing warnings ahead of time, as the CMO's estimated collision time is, on average, almost $3s$ earlier than the actual collision time. Moreover, ARAware successfully prioritises the communication of warnings based on CMO's risk level and collision time; it immediately warns about higher-risk CMOs, *i.e.,* car and motorcycle, while reducing false alarms by setting deadlines for lower-risk CMOs, *i.e.,* bicycle and pedestrian. For example, for pedestrian, ARAware sets a warning emission deadline of $5s$ and postpones the warning message by $10.21s$ (calculated as $15.21s - 5s$). Achieving a balance between reducing false alarms and delivering timely warnings ensures the safety of VIPs while enhancing the ARAware system's usability.

Table 6.8: Warning Emission Deadline Setups.

CMO	R_l	$Earl._{dt}$ (s)	t_{ra} (s)	δ (s)	t_{dl} (s)	$Emi._w$
Car	Severe	6-14.41	5	6	11	Imm. or $t'_{cl} \leq t_{dl}$
	Serious	14.99-51.55	5	4	9	$t'_{cl} \leq t_{dl}$
Moto.$_R$	Severe	4.50-7.20	5	6	11	Imm.
	Serious	7.50-17.99	5	4	9	Imm. or $t'_{cl} \leq t_{dl}$
Bic.$_R$	Serious	7.20-17.99	5	4	9	Imm. or $t'_{cl} \leq t_{dl}$
	Minor	20-22.52	5	2	7	$t'_{cl} \leq t_{dl}$
Ped.	Minor	17.99-25.77	5	2	7	$t'_{cl} \leq t_{dl}$
	Minimal	29.94-119.05	5	0	5	$t'_{cl} \leq t_{dl}$

The setup of the warning emission deadlines (explained in Section 4.1.2 of Chapter 4) is demonstrated in Table 6.8. These deadlines are determined based on the maximum detectable distance of each CMO and the actual moving speed considered in ARAware experiments, discussed in Section 6.1.2 of Chapter 6. The table presents CMOs alongside their respective risk levels R_l, the earliest time they can be identified before the predicted collision $Earl._{dt}$, the expected reaction time t_{ra}, the safety margin factor specified for the CMO risk level δ, the warning emission deadline t_{dl}, and the designated timing for ARAware to send warning messages $Emi._w$. It is clear that for severe risk levels, the desired t_{dl} can be achieved for cars moving at speeds between $25km/h$ and $34km/h$, while it cannot be achieved for motorcycles moving at speeds from $25km/h$ to $40km/h$. This is because motorcycles have a shorter detectable distance than cars by $50m$, as discussed in Section 6.1.2. However, ARAware resolves this issue by promptly emitting warning messages for motorcycles to ensure the VIP's safety. With ARAware, the VIP is provided with a considerably longer amount of time (*i.e.,* $\geq 4.50s$) than the generally accepted human reaction time range of $1.69 \sim 2.08s$, according to [244], for avoiding road-traffic accidents. Considering that ARAware can emit warnings up to almost $3s$ earlier than the actual collision time (as mentioned above, *i.e.,* $\geq (4.50s + 2.83s) = 7.33s$), this extended time allows the VIP sufficient time to react and successfully avoid collisions. For other risk levels, t_{dl} is achievable for all classes of CMOs. Based on the above discussion, ARAware's warning emission strategy efficiently handles CMOs with such varying risk levels.

Audible warning messages are transmitted to the VIP's bone-conduction headphones through

Bluetooth technology. These messages provide crucial information about the CMO's type, risk level, incoming direction, and estimated collision time, ensuring that the VIP clearly understands the impending threat.

6.1.4 ARAware versus DEEP-SEE

In this section, a comparison between the ARAware scheme and the state-of-the-art DEEP-SEE is provided. The comparison focuses on key shared components such as object tracking, collision identification, and object warning and risk classification. Additionally, an evaluation of the overall efficiency of both schemes is presented. Note that DEEP-SEE is similar to ARAware in offering a vision-based solution to assist VIPs' mobility in outdoor environments. While using different approaches, generally, both schemes identify collisions with obstacles like cars, motorcycles, bicycles, and pedestrians, classify their risks, and prioritise warning messages based on objects' risk level. However, unlike ARAware, DEEP-SEE lacks per-object distance estimation (instead, it estimates the distance to the first visible area on the floor), object speed estimation, and collision time prediction.

Due to the unavailability of open-source code for DEEP-SEE, the researcher carefully followed the methodologies outlined by the authors in their original work [28] to re-implement DEEP-SEE using Python 3.7.7. However, certain DEEP-SEE components, like GOTURN [255] and DeepCompare [256], rely on previously published research. Therefore, to speed up the implementation process and guarantee the accuracy of the obtained results, the researcher opted to utilise existing, well-established, open-source PyTorch implementations[1,2] for these specific components instead of rebuilding and retraining them from scratch.

Generally, both ARAware's and DEEP-SEE's trackers are built upon the well-known YOLO object detector, serving as a foundation for the tracking process. To ensure an even comparison, YOLOv8, the same version used in ARAware's Deep SORT tracker, is also employed to provide input data for DEEP-SEE's adapted GOTURN tracker, and both trackers are evaluated using

[1]https://github.com/amoudgl/pygoturn/tree/master
[2]https://github.com/szagoruyko/cvpr15deepcompare.

Table 6.9: Object Tracking Performance Evaluation: ARAware's Deep SORT-Based Tracker vs. DEEP-SEE's GOTURN-Based Tracker.

Model	MOTA	MOTP	Aver. Process. Time
Deep SORT (YOLOv8)	94.55%	80.88%	0.64ms
DEEP-SEE [28] (YOLOv8)	87.04%	68.75%	380ms

the VMOT dataset. Table 6.9 summarises the performance of both trackers in terms of MOTA, MOTP, and average processing time. The comparison reveals that ARAware surpasses DEEP-SEE in accurately and precisely tracking multiple objects, achieving a higher 94.55% MOTA and 80.88% MOTP. It's important to note that DEEP-SEE's tracker doesn't explicitly assign IDs to objects like ARAware's tracker does. Consequently, for the MOTA comparison, the researcher conservatively assumed zero ID switches (meaning no object identity changes). Even with this potentially optimistic assumption for DEEP-SEE, ARAware still achieves a significantly higher MOTA score. This highlights the superior robustness of ARAware in accurately tracking multiple objects. Additionally, ARAware demonstrates significantly faster tracking with an average processing time of only $0.64ms$, while DEEP-SEE requires $380ms$ (significantly larger than the real-time processing threshold of $\leq 33ms$, see Chapter 1), running on an NVIDIA GeForce RTX 2080 GPU. This substantial difference arises from DEEP-SEE's computationally intensive tracking, making it less suitable for real-time applications where fast processing speed is crucial, unlike ARAware.

Different from ARAware, DEEP-SEE cannot estimate the distance from the VIP to each object. Instead, it calculates the distance to the first visible area on the floor and defines a 'trapezium of interest' based on the camera's parameters, i.e., height and angle, where this trapezium serves as a safety zone for identifying potential collisions and obstacle proximity to the VIP in the real world. Using ARAware camera's parameters (height of $1.52m$ and angle of $74.9°$), the safety zone is estimated to extend approximately $5m$ in front of the camera. To identify potential collisions, DEEP-SEE classifies an object within the safety zone as 'urgent'; otherwise, it classifies it as 'normal'. To effectively evaluate ARAware and DEEP-SEE in identifying collisions, the researcher treated "urgent" objects as "CMO" and "normal" objects as "non-CMO", aligning with ARAware's classification for collision prediction (refer to Section 4.1.2 of

Table 6.10: CMO Identification Performance Evaluation: ARAware vs. DEEP-SEE.

Scheme	Over. mAP	Over. mAR	Aver. E2E Process. Time
ARAware	88.20%	97.26%	31ms
DEEP-SEE [28]	77.32%	54.64%	430ms

Chapter 4).

Table 6.10 shows a comparison between ARAware and DEEP-SEE in their ability to identify potential collisions on the VMOT dataset. ARAware outperforms DEEP-SEE with a higher mAP (by 10.88%), which indicates that ARAware is more accurate in identifying potential collisions across various scenarios and object types. Moreover, ARAware demonstrates a remarkable mAR of 97.26%, compared to DEEP-SEE's 54.64% mAR. This signifies that ARAware misses significantly fewer true CMOs; consequently, it offers better recall and ensures more reliable collision detection. This is because, unlike DEEP-SEE, ARAware goes beyond simply considering the objects' relative position to the VIP. It analyses a critical factor: objects' moving angles, providing a deeper understanding of their behaviour and thus enabling effective prediction of potential collision threats. Remarkably, ARAware achieves this with an impressive average E2E processing time of $31ms$, which is 93% faster than DEEP-SEE's $430ms$, both operating on the same hardware. Besides, DEEP-SEE's $5m$ safety zone is insufficient to identify potential collisions and ensure VIPs' safety, especially with high-speed moving objects.

Even with the restricted VIP safety zone, DEEP-SEE prioritises alerting the VIP about 'normal' cars and motorcycles located beyond the safety zone over low-risk 'urgent' objects (such as pedestrians) to improve VIP safety. Nevertheless, DEEP-SEE remains ineffective in preventing collisions due to its inability to estimate object speed, unlike ARAware. This limitation hinders DEEP-SEE's capability to adapt to changes in object velocities, potentially compromising VIP safety. Object speed is critical for providing timely warnings to the VIP regarding potential threats. To exemplify this, consider a scenario involving a car and a bicycle. The car is initially located $20m$ from the VIP and is travelling at a speed of $45km/h$. Conversely, the bicycle is located significantly closer, at a distance of only $5m$, while travelling at a slower speed of $3km/h$. DEEP-SEE classifies the bicycle as 'urgent' since it is inside the VIP's safety zone

while classifies the car as 'normal'. According to DEEP-SEE's warning prioritisation scheme, the VIP receives an alert solely for the object deemed the most critical threat, in this case, the bicycle. However, this poses a considerable risk to the VIP as the bicycle requires $6s$ to approach, whereas the car only takes $1.6s$ before a potential collision occurs. Conversely, ARAware promptly warns the VIP about the car. This is because ARAware classifies the car as a *'severe'* CMO due to its speed exceeding $25km/h$ and its estimated collision time is shorter than the collision time of the bicycle, which is classified as a *'minor'* CMO as its speed is less than $10km/h$. This shows that ARAware's method of classifying CMOs' risk and prioritising warning messages is more effective than DEEP-SEE, making it a safer solution for VIPs.

Furthermore, DEEP-SEE may generate frequent false alarms, potentially desensitising the VIP to genuine threats. To demonstrate this, a video was recorded for a car approaching the VIP at $5km/h$, initially positioned $30m$ away, with a potential collision in $21.58s$. At the $12m$ distance (likely collision in $8.63s$), the car safely moved away from the VIP's path. Based on DEEP-SEE's warning prioritisation, the car is alerted at first detection at $30m$ even though it remains outside the 5-meter safety zone. This approach can lead to excessive false alarms persisting for an extended duration. In this case, DEEP-SEE triggers false alarms lasting over $12.95s$ (calculated as $21.58s - 8.63s$) until the car exits the camera's view; this is because DEEP-SEE ignores the car's moving direction. Such long false alarms can distract VIPs from focusing on genuine threats requiring immediate attention. In contrast, with ARAware, the VIP receives alert $9s$ ahead (when deadline is met, see Section 4.1.2), instead of $21.58s$ in advance when the car is first identified. Besides, the alert lasts for only around $0.4s$, and it is promptly terminated once the car changes its direction of movement and is classified as 'non-CMO'(no longer considered a threat). This reveals the effectiveness of ARAware in reducing false alarms while emitting timely warnings, unlike DEEP-SEE.

In summary, based on the aforementioned findings, ARAware outperforms the state-of-the-art DEEP-SEE approach [28] by accurately tracking objects and effectively identifying potential collisions. Moreover, unlike DEEP-SEE, ARAware provides finer-grained estimation of an object's risk level, leading to more effective prioritisation of warning messages. By establishing risk-specific warning deadlines, ARAware also achieves a better balance between timely warn-

ings and false alarm reduction. All of these advancements are achieved with an impressive E2E processing speed that is 93% faster than that of DEEP-SEE.

6.2 MinD Results and Discussions

To validate the MinD scheme, results from diverse random scenarios generated by the empirically parameterised simulator, see Section 5.2, were collected and analysed for each CMO class. These scenarios, involving single and multiple CMOs moving at various speeds and located at random positions, were evaluated based on pre-defined metrics: travelling distance, time separation, collision rate, computation time and convergence iteration (all detailed in Section 5.2.2). The evaluation was conducted in comparison to both the baseline method, NBRD, and the state-of-the-art robot local path planning algorithm, APF, as discussed in Section 5.2.1.

6.2.1 Travelling Distance

Single CMO. Fig. 6.6 displays the results of the travelling distance required by the VIP to avoid each CMO class moving at eight different speeds, where the performance of MinD, NBRD, and the improved APF algorithm [45] are compared. Figure 6.6a presents the average results of 800 samples from the car class, covering speeds from $11km/h$ to $108km/h$, with 100 samples averaged for each speed. Similarly, Fig. 6.6b illustrates the results of the motorcycle class. The same number of motorcycle samples were used as car samples, moving at the same speeds. Additionally, Fig. 6.6c and Fig. 6.6d showcase the outcomes of the bicycle and pedestrian classes (800 samples for each) at speeds ranging from $7km/h$ to $54km/h$ and $3.6km/h$ to $36km/h$, respectively.

The analysis of single CMO scenarios reveals that the MinD scheme variably minimises the travelling distance based on the CMO class. Specifically, the average distances are found to be $2.91m$ for cars, $1.56m$ for motorcycles, $1.26m$ for bicycles, and $1.02m$ for pedestrians, and these distances remain constant regardless of the speed of the CMOs across all classes. For NBRD,

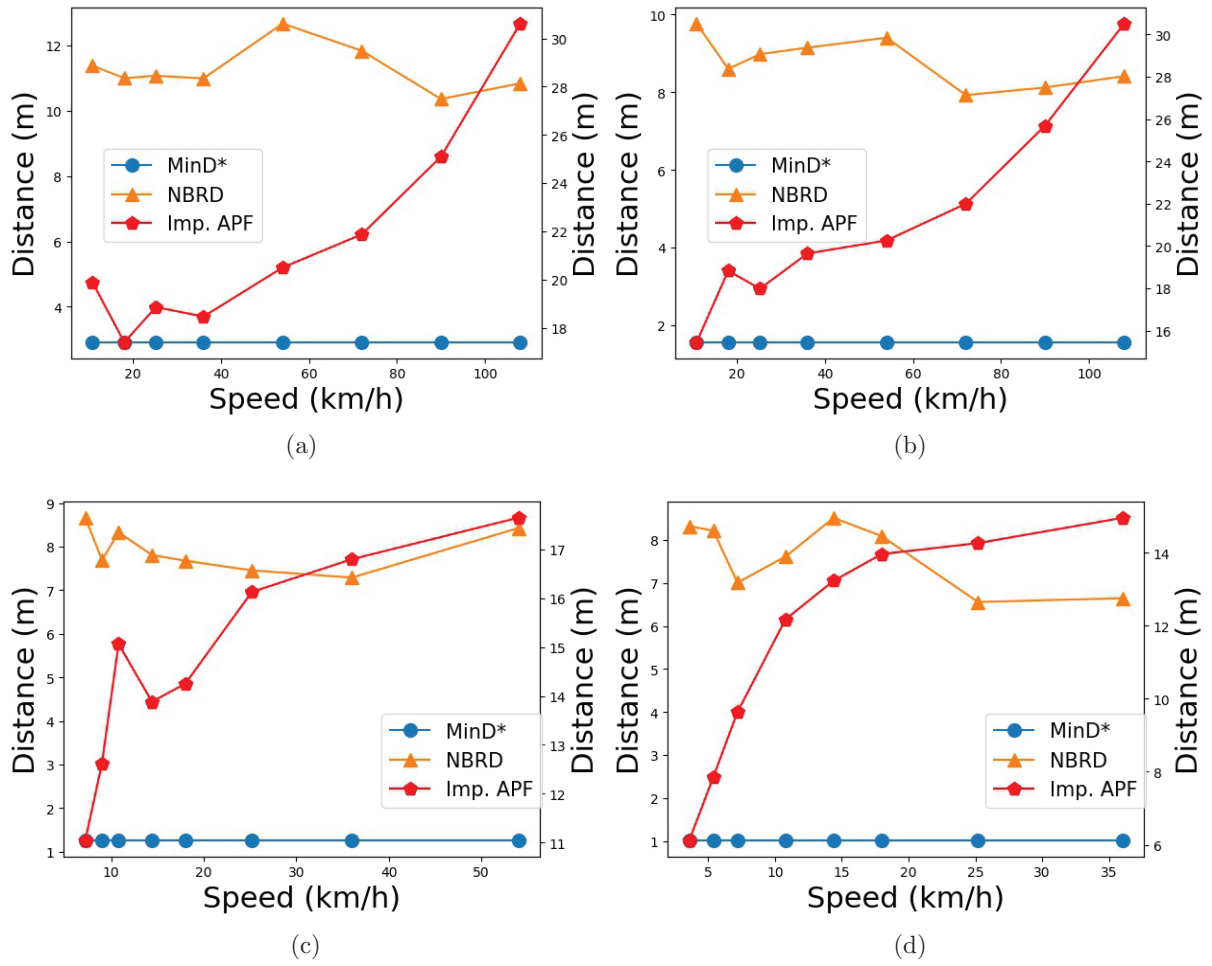


Figure 6.6: The Travelling Distance of Each CMO Class Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF).

the travelling distances are random and do not depend on the CMO class or speed, as these factors are not considered in the NBRD approach. In NBRD, the VIP continues to navigate and avoid the CMO as long as alerts are being sent, persisting until the danger is entirely avoided and no further alerts are received. This may result in random frequent transitions for the VIP, potentially increasing the overall travelling distance. Note, in all NBRD scenarios, the alerts are triggered when $\min(t^{t_c}_{cl_i}) \leq S_L$, see Eq. (4.29); this is to get comparable results with MinD, the same case with APF. The average travelling distances for NBRD are $11.27m$ for cars, $8.79m$ for motorcycles, $7.91m$ for bicycles, and $7.62m$ for pedestrians. These distances are greater than the average distances of the MinD scheme.

For the improved APF, the travelling distance method generally tends to increase as the CMO's speed increases across all CMO classes. However, there are a few exceptional cases where this linear relationship deviates. This increase occurs due to the reliance of the APF method on two artificial forces for path planning: the attraction force, which directs the VIP towards the destination, and the repulsion force, which pushes the VIP away from obstacles. The repulsion force is restricted by an obstacle's influence distance r_r, which represents the boundary for the strength of the repulsion force. This r_r distance increases as the CMO's speed increases (note, r_r is set as $r_r = s_l(|v_{O_i}^{t_f}| + |v_B^{t_f}|)$, see Section 5.2.1). Therefore, higher CMO speeds result in a larger r_r, which enables the repulsion force to act over a wider area and exert a stronger influence. This, in turn, increases the required travelling distance for the VIP to avoid the CMO. Further, the repulsion force is strongest at the centre of the obstacle and gradually weakens as the distance from the obstacle's centre increases towards the edges of the influence distance. Consequently, the APF algorithm lets the VIP navigate towards the destination (due to the attraction force), even in the presence of an obstacle in the same VIP's path, until the repulsion force becomes influential enough to prevent the potential collision with the obstacle. As a result, the paths generated by APF exhibit significantly longer average distances than MinD and NBRD. Specifically, the average distances are found to be $21.58m$ for cars, $21.29m$ for motorcycles, $14.68m$ for bicycles, and $11.53m$ for pedestrians. These long distances make APF impractical for VIPs as they have limited mobility constraints. Further, the deviation from the linear relationship in some cases of the APF line graph, as mentioned above, is because APF may encounter problems such as local minima and oscillation. These problems can cause the APF to get stuck in a suboptimal path or to repeatedly oscillate between two or more different paths. To overcome these problems, APF uses the Simulated Annealing algorithm, where the algorithm randomly chooses an escape point and then repeatedly tries new paths, sometimes accepting worse paths with a certain probability. This can lead to longer paths, which means that the VIP may travel further than expected.

To sum up, when considering the minimum travel distance required to avoid a single CMO, the MinD scheme achieves the lowest average distance across all CMO's classes, with a distance of $1.69m$, compared to NBRD and improved APF, which achieved $8.90m$ and $17.27m$ on average,

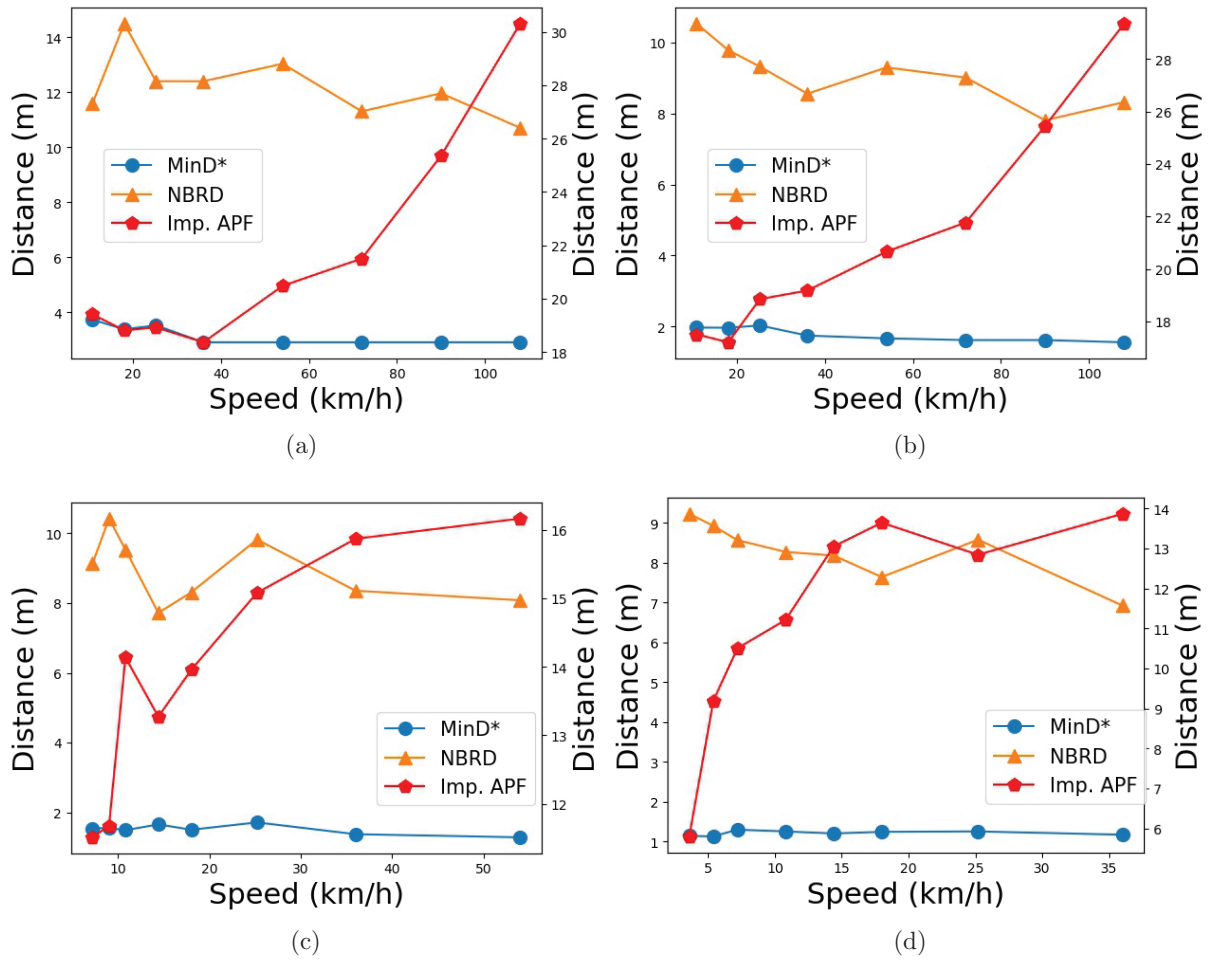


Figure 6.7: The Travelling Distance of Two CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF).

respectively. This implies that, with the MinD scheme, the VIP requires an average time of only $1.22s \sim 4.05s$ to evade a potential threat when moving at speeds $1.5km/h \sim 5km/h$. In contrast, with the NBRD method and the APF algorithm, VIP needs a longer time ($> 6.40s$ for NBRD and $> 12.43s$ for APF), potentially endangering the VIP's safety. This demonstrates the effectiveness of the MinD proposed scheme.

Multiple CMOs. Figures 6.7, 6.8, 6.9, and 6.10 show the average travelling distance results of MinD, NBRD, and the improved APF algorithm [45] for scenarios involving two, three, five, and seven CMOs of different classes, respectively. In each figure, the results of the four targeted classes are depicted in separate sub-figures, labelled with letters (a for cars, b for motorcycles, c for bicycles, and d for pedestrians). Similar to the single CMO scenarios, these results are

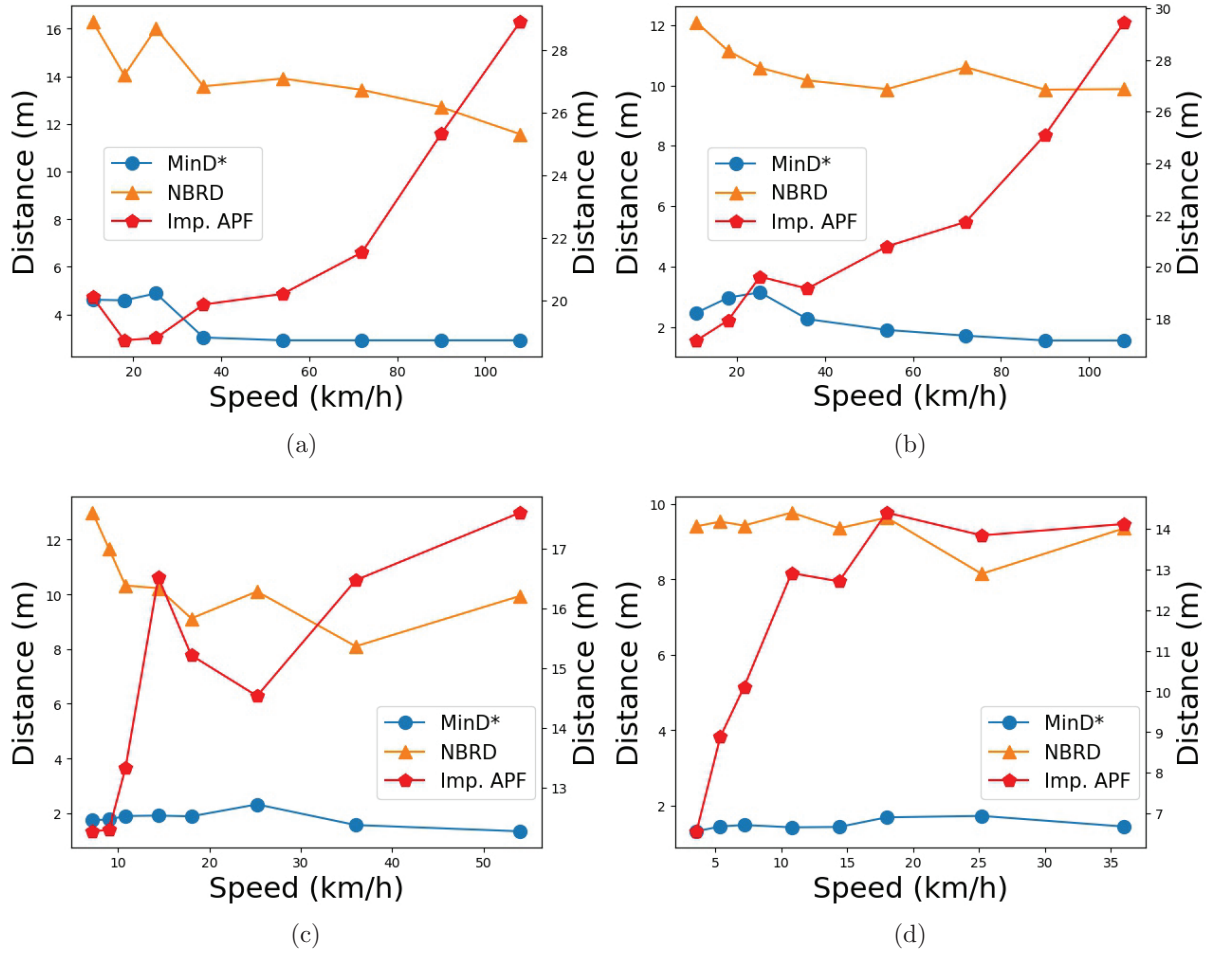


Figure 6.8: The Travelling Distance of Three CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF).

based on 800 samples and encompass eight different moving speeds for each class (ranging from $11km/h$ to $108km/h$ for cars and motorcycles, from $7km/h$ to $54km/h$ for bicycles and from $3.6km/h$ to $36km/h$ for pedestrians), with an average of 100 samples for each speed. The results show that, compared to the other algorithms, MinD consistently achieves significantly shorter travel distances for the VIP to avoid varying numbers of CMOs (*i.e.,* from 2 to 7). However, all MinD graphs exhibit fluctuations in travel distance, which increase in intensity as the number of objects increases. These fluctuations are an expected occurrence due to the dynamic nature of the scenarios (VIP and objects are constantly moving). Specifically, as the VIP and objects continue to move towards their respective destinations, despite the VIP initially avoiding the object's path, there are instances where the VIP may inadvertently align with the object's path

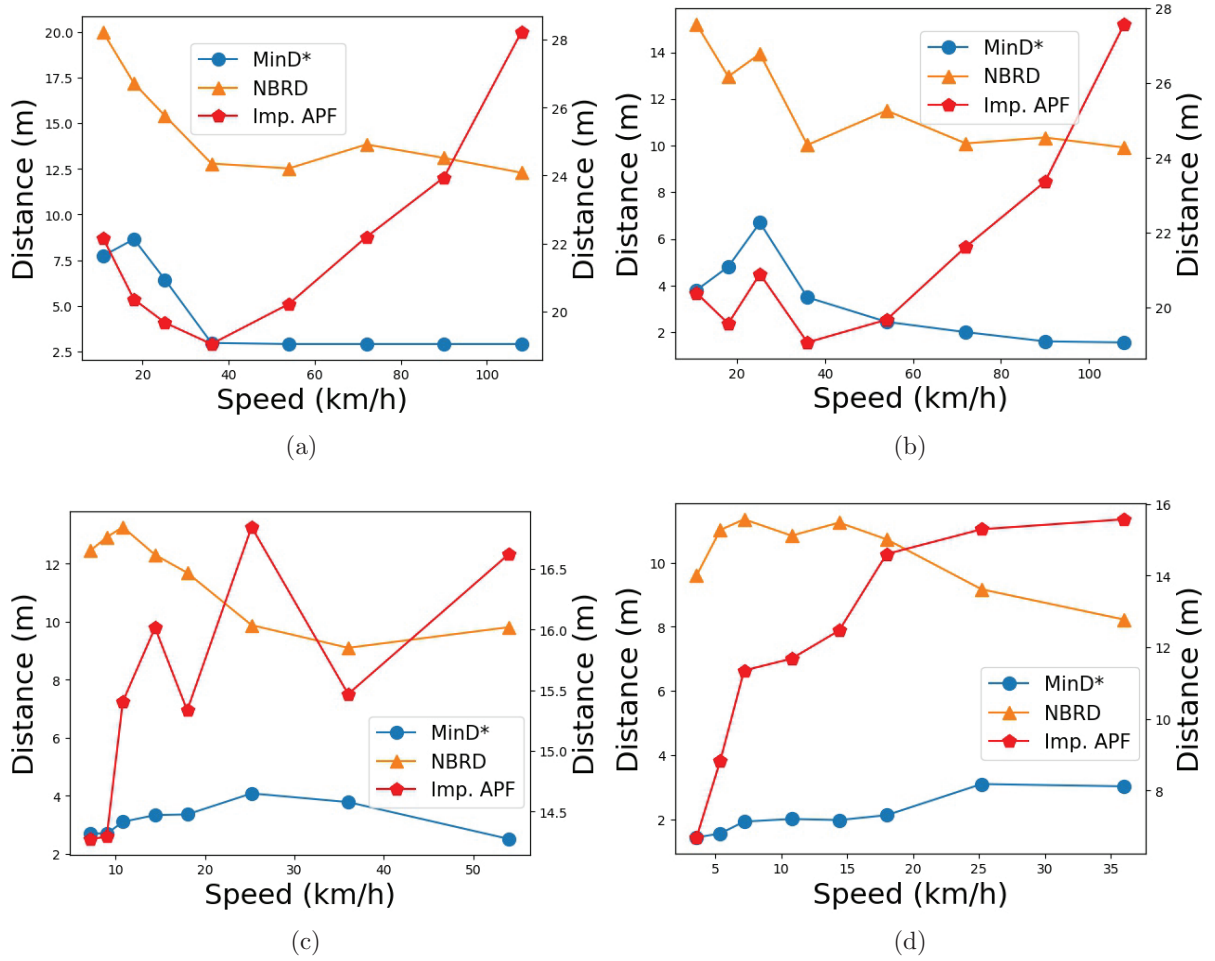


Figure 6.9: The Travelling Distance of Five CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF).

again after a while, necessitating a transition to a new path. This process may be repeated multiple times according to the number of CMOs in the scene, resulting in an overall increase in the total distance travelled. The average distances achieved by MinD vary depending on the type and the number of CMOs. Table 6.11 presents the average distances MinD achieves for each CMO class under scenarios where 2, 3, 5, and 7 CMOs are present.

For NBRD, the results observed in the scenarios involving multiple CMOs are generally similar to those observed in the case of a single CMO, which is characterised by randomness. This implies that the distance travelled by the VIP can vary significantly from one scenario to another, even when using the same parameters. However, the average distances achieved by NBRD for

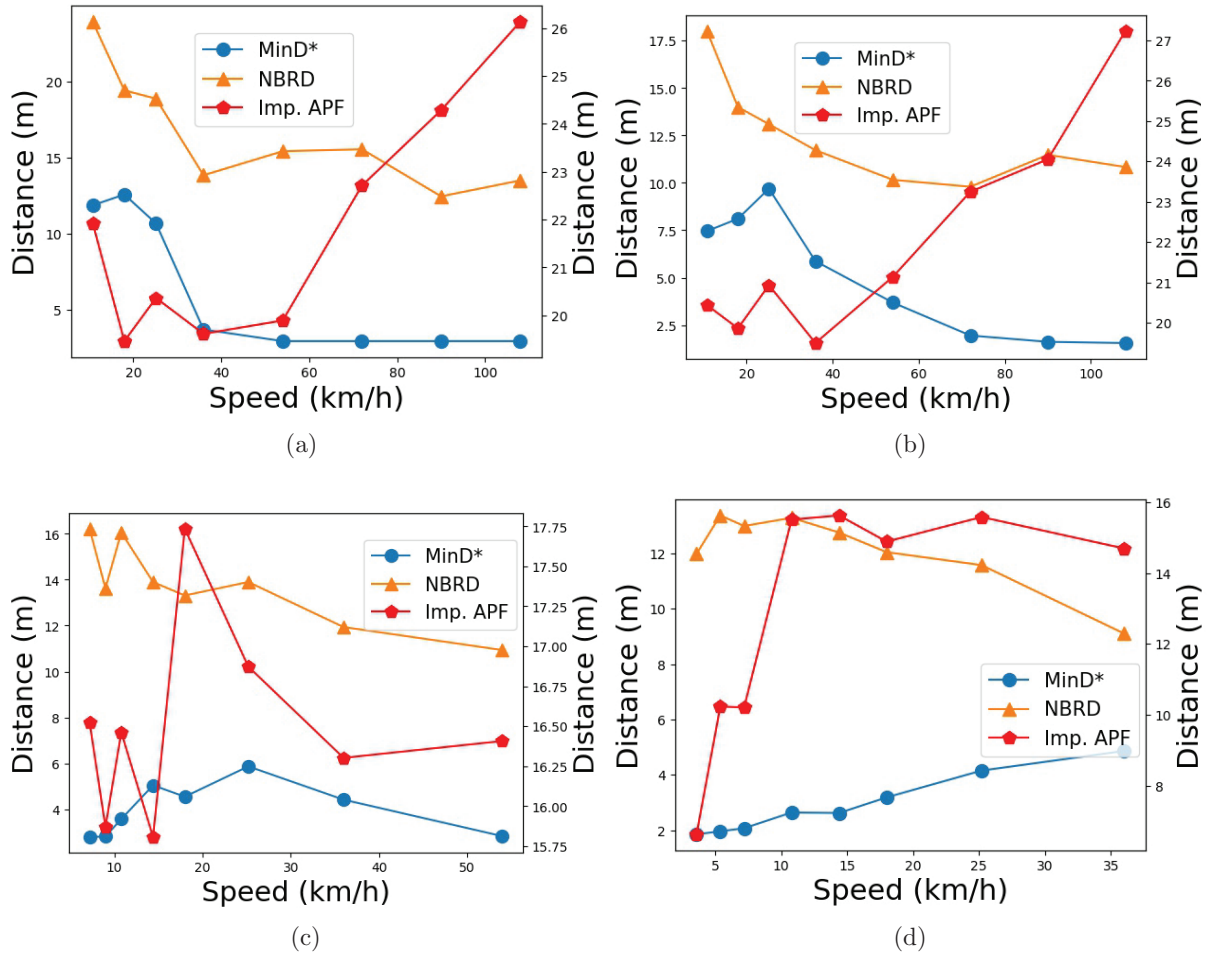


Figure 6.10: The Travelling Distance of Seven CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance (MinD* and NBRD), Y-axis (right): Distance (Imp. APF).

each CMO class are presented in Table 6.11. For APF, similar to the single CMO scenarios, the travelling distance generally increases as the object's speed increases (the reason is mentioned above). However, there are instances where this relationship is violated. In fact, the violation becomes more intense as the number of CMOs increases. Particularly, when the number of CMOs ≥ 5, the pattern significantly deviates, and a linear increase is no longer preserved. This is because when the number of CMOs increases, the scope of overall repulsion influence grows accordingly. This occurs due to the overlapping and combined repulsion potential fields of individual CMOs, creating a larger overall repulsion area. This is especially true when the CMOs are close together. As a result, when APF gets trapped in suboptimal solutions, it requires larger escape steps (selected randomly) to overcome the overall increased repulsion in-

Table 6.11: Average Travelling Distances Achieved for Each CMO Class.

Class	Num. of CMOs	MinD	NBRD	APF
Car	2	$3.15m$	$12.24m$	$21.65m$
	3	$3.59m$	$13.94m$	$21.69m$
	5	$4.68m$	$14.64m$	$21.98m$
	7	$6.29m$	$16.61m$	$21.79m$
Motorcycle	2	$1.78m$	$9.07m$	$21.25m$
	3	$2.20m$	$10.53m$	$21.37m$
	5	$3.30m$	$11.75m$	$21.52m$
	7	$4.99m$	$12.38m$	$22.05m$
Bicycle	2	$1.51m$	$8.92m$	$13.96m$
	3	$1.80m$	$10.31m$	$14.78m$
	5	$3.19m$	$11.43m$	$15.53m$
	7	$3.99m$	$13.73m$	$16.50m$
Pedestrian	2	$1.21m$	$8.29m$	$11.26m$
	3	$1.49m$	$9.33m$	$11.69m$
	5	$2.14m$	$10.27m$	$12.06m$
	7	$2.92m$	$12.15m$	$12.92m$

fluence, which results in longer paths in such situations. Table 6.11 shows the average travelling distances APF achieves for each CMO class.

To summarise, the MinD algorithm outperforms NBRD and APF in achieving the minimum average travelling distance when multiple CMOs are present. Across all classes, the average distances achieved with MinD, measuring at $1.91m$, $2.27m$, $3.33m$, and $4.55m$ for 2, 3, 5, and 7 CMOs, respectively, which are significantly lower than the distances achieved by NBRD and APF, which are $9.63m$ and $17.03m$ for 2 CMOs, $11.03m$ and $17.38m$ for 3 CMOs, $12.02m$ and $17.77m$ for 5 CMOs, $13.72m$ and $18.31m$ for 7 CMOs, respectively. Across all CMOs (from 2 to 7) of all classes, MinD achieves a significantly lower overall average travel distance of only $3.02m$, which is 74.66% and 83.03% lower than those achieved by NBRD and APF, respectively. This implies that when the VIP is moving at speeds ranging from $1.5km/h$ to $5km/h$, the MinD scheme enables the VIP to avoid $2 \sim 7$ potential threats in an average time of only $2.17s \sim 7.19s$. In contrast, the NBRD method and APF algorithms require longer times ($8.34s \sim 27.62s$ for NBRD and $12.68s \sim 41.96s$ for APF). This highlights the effectiveness of the MinD scheme in providing shorter paths for the VIP.

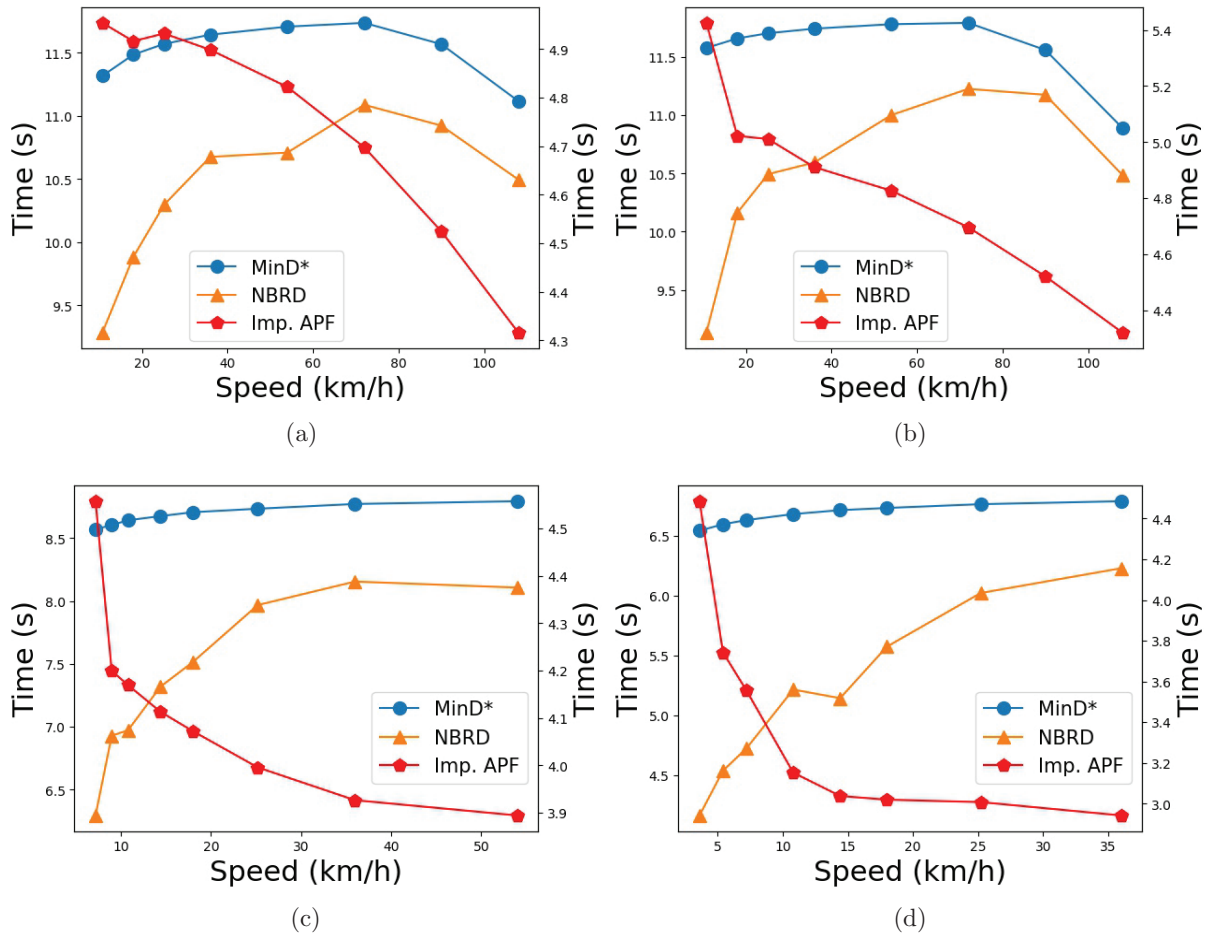


Figure 6.11: The Time Separation of Each CMO Class Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF).

6.2.2 Time Separation

Single CMO. Fig.6.11 shows the time separation results of MinD, NBRD, and APF [45] for each CMO class. Specifically, the results for the cars and motorcycles are depicted in Fig. 6.11a and Fig. 6.11b, respectively. Fig. 6.11c displays the results for the bicycles, while Fig. 6.11d presents the results for the pedestrians. The results clearly show that, with MinD, the time separation consistently increases as the speed increases across all bicycle and pedestrian scenarios. Conversely, for both car and motorcycle classes, the time separation exhibits an initial gradual increase as the speed increases from $11km/h$ to $72km/h$, followed by a decrease when the speed exceeds or is equal to $90km/h$. This decrease is logical due to the significantly high speeds of cars and motorcycles compared to the other classes. As MinD considers the

VIP's mobility constraint, these higher speeds reduce the amount of time that MinD can leave between the object and the VIP at B^{t_f} (optimal safe position) as a safe distance. Despite the reduced time separation between the VIP and faster-moving objects ($\geq 90km/h$), the time separation remains sufficiently large ($> 11s$) for VIP safety. On average, MinD achieves a time separation of $11.52s$ for cars, $11.59s$ for motorcycles, $8.69s$ for bicycles and $6.68s$ for pedestrians.

In NBRD, a similar pattern to MinD is observed (the time separation increases as the speed increases to a certain limit and then decreases). However, slight fluctuations are present, which can be attributed to the NBRD's inherent randomness, where the VIP takes a random collision avoidance based on triggered alarms, mentioned previously. On average, NBRD achieves $10.42s$ for cars, $10.53s$ for motorcycles, $7.41s$ for bicycles, and $5.20s$ for pedestrians. For improved APF, the time separation generally decreases as the speed increases for all CMO classes, achieving, on average, $4.76s$ for cars, $4.84s$ for motorcycles, $4.12s$ for bicycles and $3.37s$ for pedestrians. The decrease in time separation is because, in APF, the repulsion force generated by obstacles is inversely proportional to the distance. This means the closer the CMO is to the VIP, the stronger the repulsion force will be. Therefore, when the CMO moves slowly, it is detected closer to the VIP (considered a lower-risk object), leading to an increased repulsion force. Conversely, when the CMO moves quickly, it is detected further away from the VIP (considered a higher-risk object), resulting in a decreased repulsion force. This behaviour of APF contradicts the objective of providing more time separation as the speed of CMO increases. Therefore, based on these observations, APF is unsuitable choice for ensuring the VIP's safety during local path planning, even with a single CMO.

To sum up, in single CMO scenarios, MinD significantly outperforms NBRD and APF in achieving larger time separation across all CMO classes. On average, it achieves a $9.62s$ (safety buffer for VIPs to react to sudden changes), which is 14.63% and 125.21% larger than NBRD and APF, respectively.

Multiple CMOs. Figures 6.12, 6.13, 6.14, and 6.15 demonstrates the average time separation results of MinD, NBRD, and the improved APF algorithm [45] in scenarios involving two, three, five, and seven CMOs, respectively. Each sub-figure of these figures represents a specific

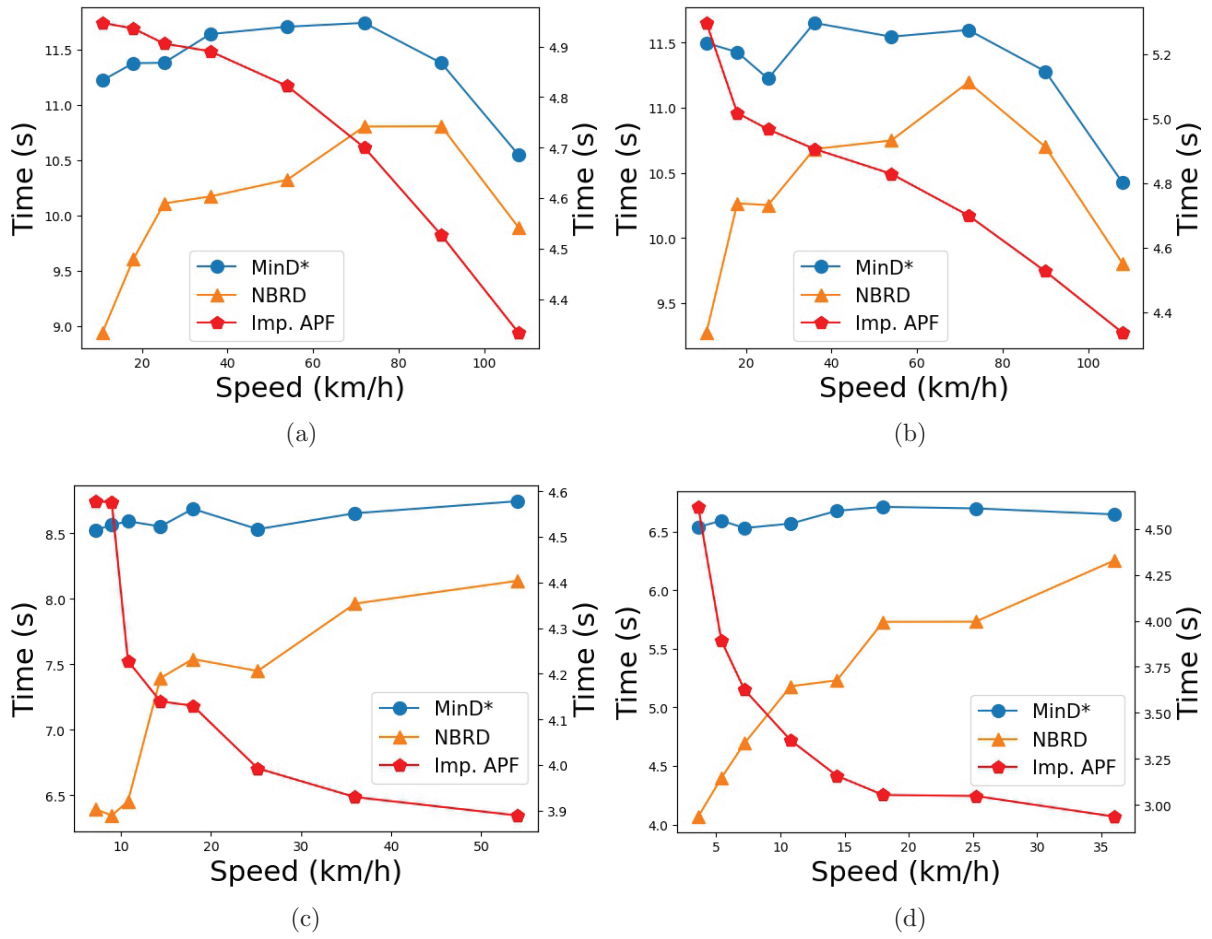


Figure 6.12: The Time Separation of Two CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF).

class: a for cars, b for motorcycles, c for bicycles, and d for pedestrians. The results show that MinD generally maintains a larger time separation between the VIP and varying numbers of CMOs (from 2 to 7) compared to the other algorithms, with fluctuations that increase in intensity as the number of CMOs increases. These fluctuations arise due to the dynamic nature of the experimental environments, similar to what was discussed in Subsection 6.2.1. In dynamic environments, as objects continue to move, the VIP may find themselves moving toward them again after a while. Consequently, the MinD re-plans a new path; this path may have a smaller time separation than the previous path as the objects and the VIP become closer. Even in such cases, the MinD algorithm successfully achieved a substantial time separation that can effectively handle situations where the objects suddenly change their direction towards the

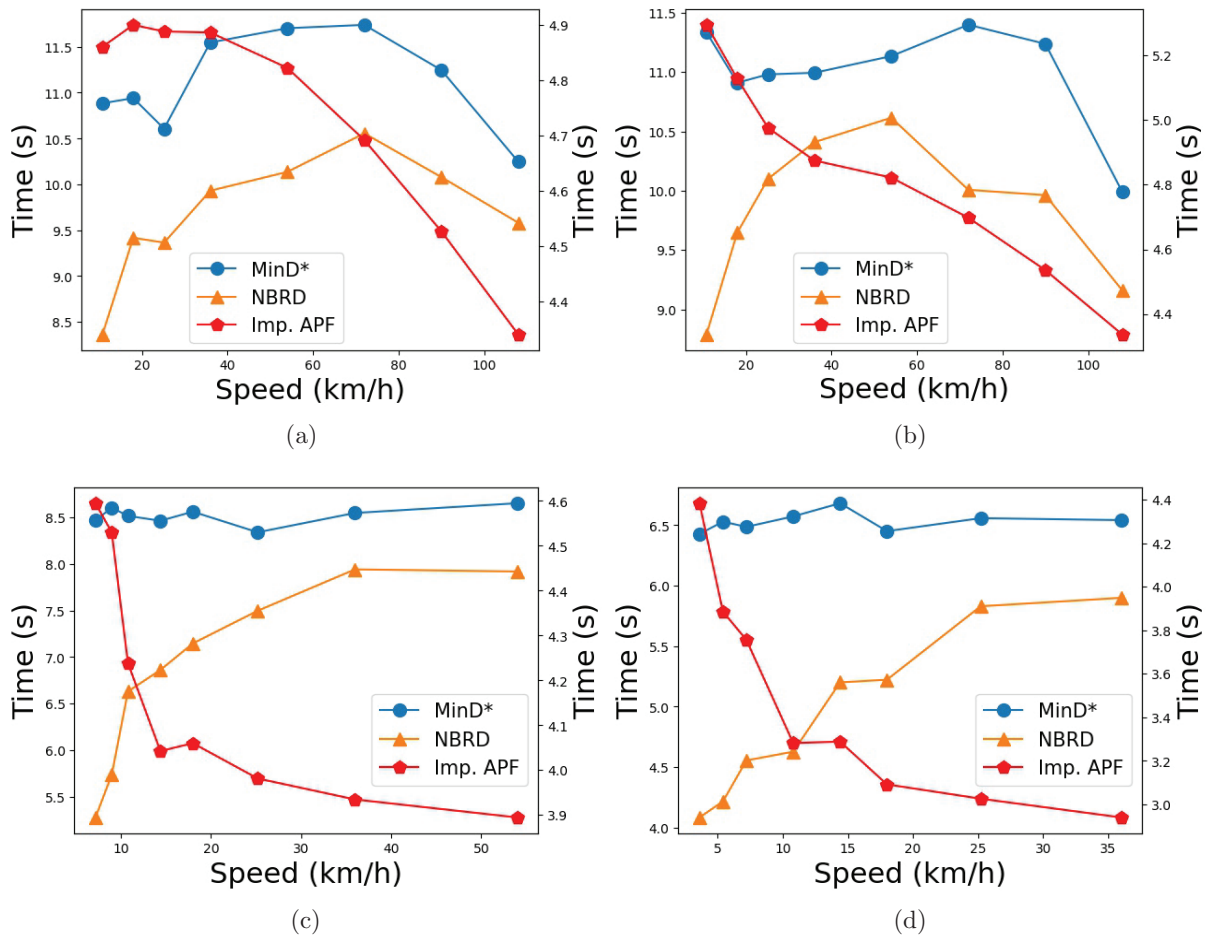


Figure 6.13: The Time Separation of Three CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF).

VIP. Table 6.12 summarises the average time separations MinD achieves for each class when scenarios include 2, 3, 5 and 7 CMOs, considering the time separation between the VIP and all CMOs in a scenario.

NBRD and APF also exhibit fluctuations due to their inherent randomness strategies (NBRD's random avoidance and APF's random escape, as discussed in Subsection 6.2.1) and the dynamic nature of the environment. However, in contrast to MinD and NBRD, with the APF, the time separation decreases as the speed of the CMO increases for all CMOs (as in the case of a single CMO). The average separation times achieved by NBRD and APF for each class are presented in Table 6.12.

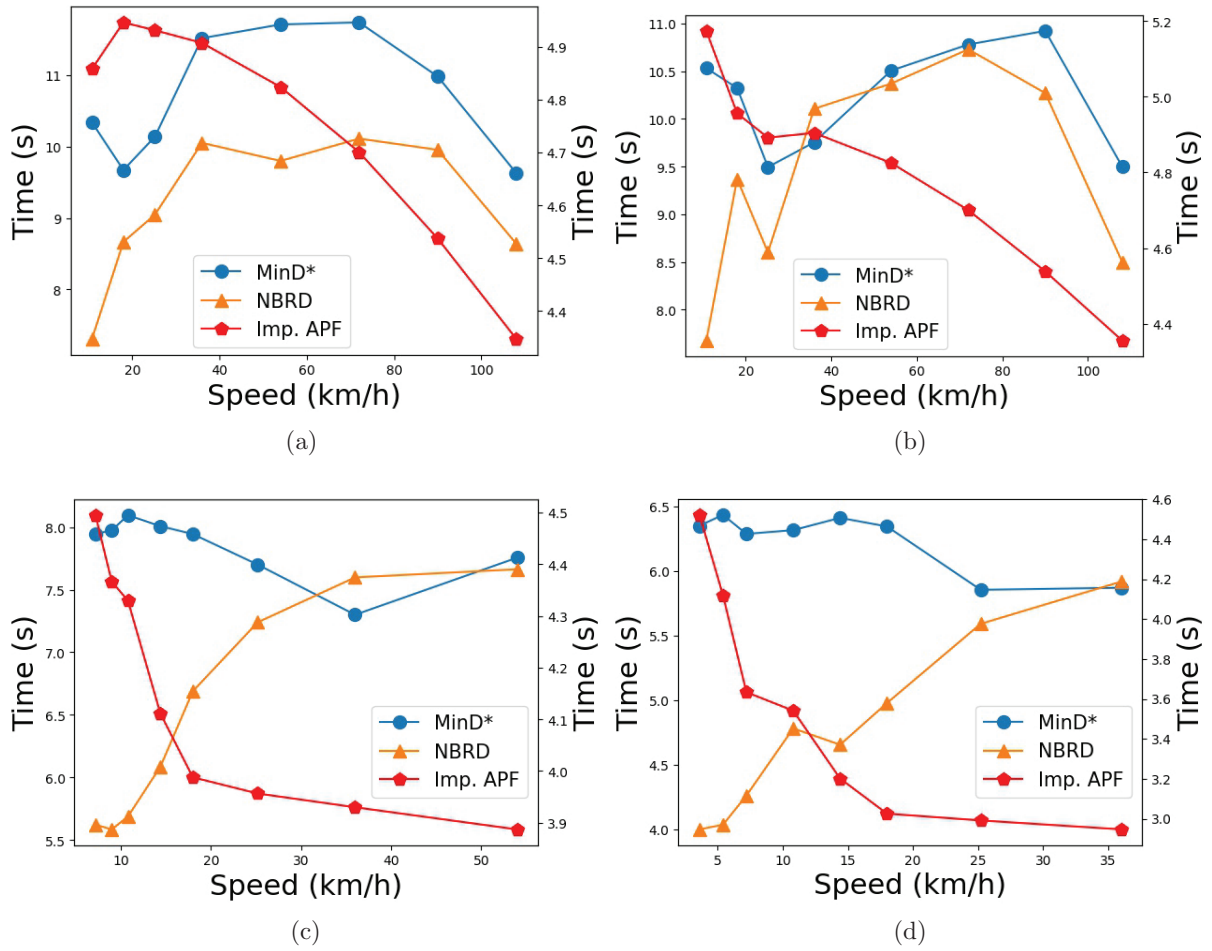


Figure 6.14: The Time Separation of Five CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF).

To sum up, MinD maintains a larger safety gap between the VIP and CMOs compared to NBRD and APF in scenarios with multiple CMOs. Across all CMOs ($2 \sim 7$) of all classes, MinD achieves an overall average time separation of $8.98s$, which is 15.98% and 107.74% larger than NBRD and APF, respectively. This extra time allows the VIP to react effectively when previously avoided CMOs change route, reducing the potential risks of collisions.

6.2.3 Collision Rate

Table 6.13 presents a comparison of collision rates between MinD, NBRD, and improved APF [45] for scenarios with single and multiple CMOs. The results show that MinD successfully

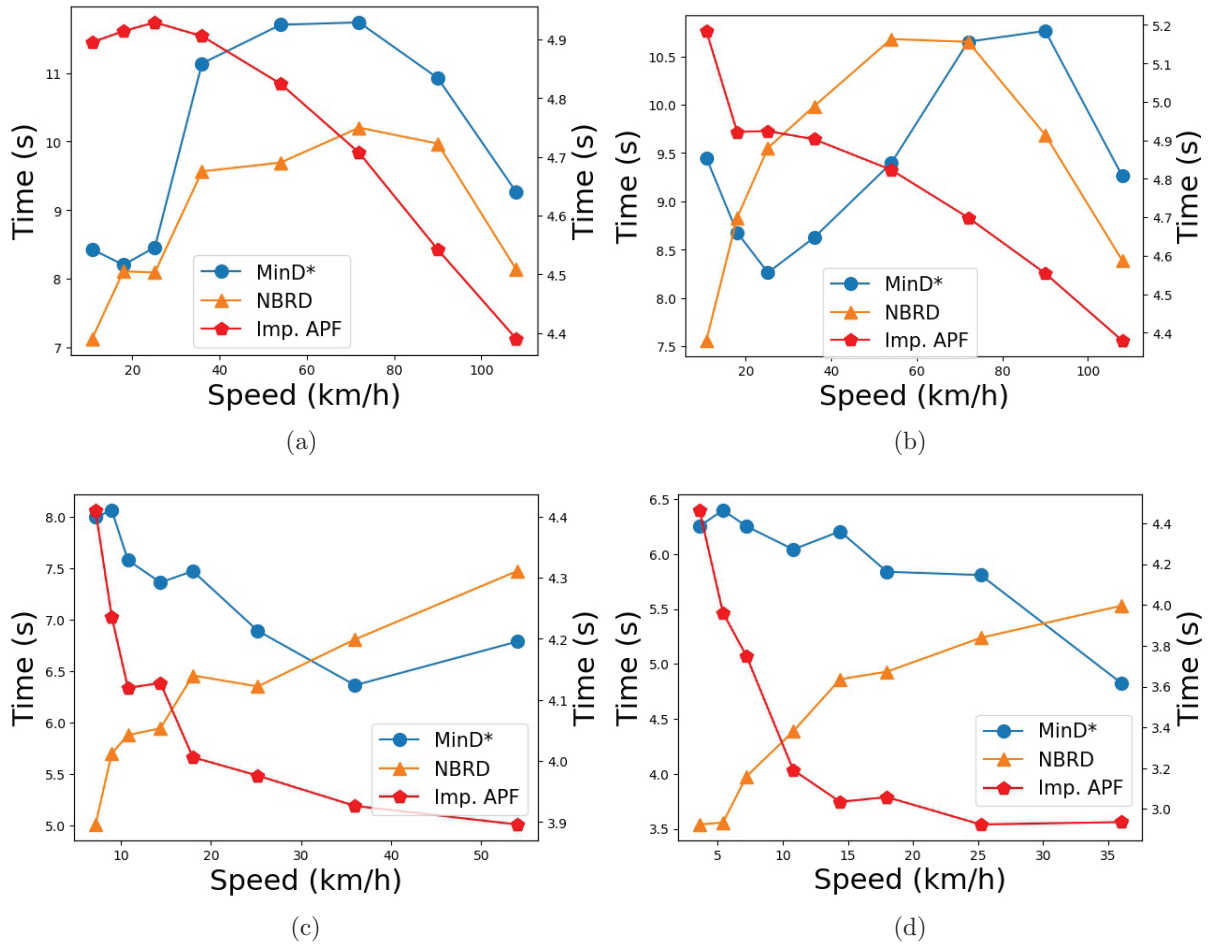


Figure 6.15: The Time Separation of Seven CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Time (MinD* and NBRD), Y-axis (right): Time (Imp. APF).

achieves a 0% average collision rate in all scenarios involving 1, 2, 3, 5, and 7 CMOs across all CMO classes, while NBRD and improved APF exhibit average collision rates of 5.75% and 86.94% for 1 CMO, 7.77% and 83.84% for 2 CMOs, 9.79% and 86.10% for 3 CMOs, 13.91% and 88.07% for 5 CMOs, and 18.04% and 89.80% for 7 CMOs, respectively. The overall average collision rates across all CMOs (from 1 to 7) of all classes are 0% for MinD, 11.05% for NBRD, and 86.95% for APF. This demonstrates that MinD's adaptive optimisation of the VIP's position, considering the VIP's limited capabilities, CMOs' dynamic moving patterns and estimated collision times, enables effective collision avoidance of threats of different classes moving at various speeds, making it a reliable and robust solution for VIPs. However, future work will investigate more complex scenarios involving sudden adjustments in the speed and

Table 6.12: Average Time Separation Achieved for Each CMO Class.

Class	Num. of CMOs	MinD	NBRD	APF
Car	2	11.37s	10.09s	4.76s
	3	11.13s	9.71s	4.74s
	5	10.75s	9.30s	4.76s
	7	10.26s	9.02s	4.77s
Motorcycle	2	11.33s	10.37s	4.83s
	3	11.00s	9.85s	4.85s
	5	10.32s	9.54s	4.82s
	7	9.63s	9.52s	4.82s
Bicycle	2	8.61s	7.21s	4.21s
	3	8.52s	6.90s	4.19s
	5	7.88s	6.58s	4.19s
	7	7.43s	6.30s	4.16s
Pedestrian	2	6.62s	5.16s	3.50s
	3	6.53s	4.97s	3.49s
	5	6.26s	4.83s	3.57s
	7	6.06s	4.56s	3.53s

direction of objects.

6.2.4 Computation Time and Convergence Iteration

Table 6.14 compares MinD and improved APF [45] in terms of the average computation time and convergence iteration over all CMOs (up to 7) of all classes. Note that NBRD is excluded from the table as it does not represent actual local path planning. Although improved APF exhibits a notably shorter overall computation time of just $0.04ms$, MinD successfully achieves real-time performance by only requiring $1.20ms$ ($\leq 33ms$, the real-time processing threshold outlined in Chapter 1), running on an NVIDIA GeForce RTX 2080 GPU. Moreover, AFP requires 58 overall iterations to converge, while MinD requires only 4 overall iterations; this is due to the differences in the underlying avoidance strategies of both algorithms. In particular, improved APF uses a random escape mechanism, where the algorithm repeatedly tries to escape from local minima in the potential field. This random escape process can sometimes take more iterations to converge to a suitable solution, especially in complex and dynamic environments. Conversely, MinD employs optimisation-based adaptive approach; it simultaneously considers the VIP's limited

Table 6.13: Collision Rate Among MinD Scheme and Other Methods.

CMO	Class	MinD	NBRD	Imp. APF [45]
1	Car	0%	2.75%	99.75%
	Motorcycle		3.38%	94.88%
	Bicycle		5.63%	84.38%
	Pedestrian		11.25%	68.75%
Average		0%	5.75%	86.94%
2	Car	0%	5.38%	99.75%
	Motorcycle		5.63%	95.25%
	Bicycle		8.63%	80.63%
	Pedestrian		11.44%	59.72%
Average		0%	7.77%	83.84%
3	Car	0%	6.13%	99.75%
	Motorcycle		7.38%	95.75%
	Bicycle		10.5%	82.88%
	Pedestrian		15.13%	66.00%
Average		0%	9.79%	86.10%
5	Car	0%	8.63%	99.5%
	Motorcycle		12.5%	98.25%
	Bic.		14.75%	86.63%
	Ped.		19.75%	67.88%
Average		0%	13.91%	88.07%
7	Car	0%	14.29%	99.75%
	Motorcycle		14.20%	98.25%
	Bicycle		19.00%	89.43%
	Pedestrian		24.67%	71.76%
Average		0%	18.04%	89.80%

Table 6.14: The Average Computation Time and Convergence Iteration.

Method	Computation Time	Convergence Iteration
MinD	$1.20ms$	4
Imp. APF [45]	$0.04ms$	58

mobility, the dynamic moving patterns of CMOs and estimated collision times and performs a look-ahead safety prediction of an optimal position. This accelerates convergence to a collision-free path that effectively accommodates the VIP in challenging environments. All these are done within a negligible processing time ($1.20ms$). This demonstrates the effectiveness of MinD in achieving real-time collision-free path planning.

In summary, while APF's computational simplicity leads to faster computation times, it may

necessitate more iterations to escape from unfavourable situations and reach a solution, and this solution may not be safe enough for VIPs. In contrast, MinD's optimisation-based approach contributes to faster convergence (with negligible processing time) while prioritising VIP safety by effectively considering all aforementioned constraints during local path planning.

In conclusion, this chapter has comprehensively evaluated ARAware and MinD schemes. It meticulously validated ARAware's core functionalities and overall performance in CMO identification, classification, and warning notification. This in-depth analysis was followed by a detailed comparison with DEEP-SEE, the closest state-of-the-art approach. MinD's effectiveness was evaluated by analysing factors like travelling distance, time separation, collision rate, and efficiency. These results were compared against a baseline method NBRD and the robot local path planning algorithm APF. The real-world experimental results showed the effectiveness of ARAware in accurately identifying CMOs (97.26% mAR and 88.20% mAP) and precisely classifying their risk levels (100% mAR and 91.69% mAP) while maintaining real-time processing with an average E2E processing speed of $32fps$ ($31ms$). It also showed that ARAware surpasses DEEP-SEE in CMO identification accuracy by 42.62% higher mAR and 10.88% higher mAP, with 93% faster end-to-end processing speed. Simulation-based results showed that MinD surpasses APF and NBRD in minimising travelling distance (84.47% and 75.94% shorter than APF and NBRD, respectively), maintaining a larger safety buffer between the VIP and CMOs ($\geq 4.80s$ and $\geq 1.24s$ compared to APF and NBRD, respectively), ensuring a 0% collision rate, and achieving a negligible average processing time of $1.20ms$. The next chapter will explore ARAware's and MinD's performance under diverse settings and explore their limitations. This deeper examination will provide valuable insights for future research and refinement of these promising schemes.

Chapter 7

Alternative Approaches and Limitations

Following the evaluation of ARAware and MinD presented in the previous chapter, this chapter delves deeper into their performance by examining them under different settings and exploring their limitations. Specifically, it compares the performance of YOLOv3 and YOLOv8 object detectors and evaluates their impact on ARAware's overall effectiveness. It also examines the impact of different frame-skipping settings on ARAware's end-to-end (E2E) CMO identification. Additionally, MinD's effectiveness in planning collision-free paths is examined under two different configurations of CMO moving angles. The limitations of both schemes' algorithms, design, and evaluation are also discussed. This is to provide a deeper understanding of both schemes' capabilities and areas for improvement.

7.1 Performance Comparisons: YOLOv8 vs YOLOv3

This section compares YOLOv8, the newest iteration (at the time of thesis writing) of the You Only Look Once (YOLO) family, with the well-established older version YOLOv3. As the detector plays a crucial role in the proposed scheme's core functionality, this comparison aims to identify the detector that effectively meets the scheme requirements, i.e., accuracy and real-

time performance. This comparison focuses on object detection and its influence on ARAware's other core functionalities: object tracking and E2E CMO identification. To clarify, YOLOv3 was initially selected as the object detector for the ARAware scheme. This choice was based on two key considerations. Firstly, based on existing research (e.g., [91, 92]), YOLOv3 offered a compelling balance between accurate object detection and real-time processing speed, which can align well with the ARAware scheme requirements. Secondly, it was widely adopted in computer vision applications, particularly during the period when the researcher commenced her studies in early 2019. At that time, YOLOv3 stood out as the latest and most outstanding iteration within the YOLO family. However, as the researcher progressed in her research and received valuable feedback from IMWUT reviewers, discussions were initiated regarding the effectiveness of YOLOv3 in achieving real-time without relying on constructing a frame-skipped method (inspired by [257]) for performance improvement. These conversations led to a reevaluation of the ARAware scheme, prompting updates and modifications to ensure that ARAware aligns more effectively with the objectives and requirements of this thesis. Consequently, the comparison between YOLOv3 and YOLOv8 is provided in Section 7.1.1. Additionally, although frame skipping was initially designed to enhance YOLOv3's performance (as mentioned above), the researcher's curiosity extended to examine its effects on YOLOv8-based CMO identification. This investigation explored various metrics, including mAR, mAP, EmFPS, and average processing time, discussed later in Section 7.2.

7.1.1 Object Detection

YOLOv8 represents a significant advancement over YOLOv3 due to various key enhancements [4]. For example, YOLOv8 employs CSPDarknet53 as its backbone architecture, a more efficient and optimised network than YOLOv3's Darknet-53. YOLOv8 also introduces the C2f module (a new convolution layer), which significantly boosts detection accuracy by integrating contextual information with high-level features. Unlike YOLOv3's reliance on predefined anchor boxes, YOLOv8 adopts anchor-free detection, which reduces candidate bounding boxes and streamlines the Non-Maximum Suppression (NMS) process (a complex post-processing phase),

Table 7.1: Object Detection Performance on VMOT: YOLOv8 vs. YOLOv3.

Model	mAP_{50}	mAR	Aver. Process. Time
YOLOv8m	97.61%	96.45%	11.85ms
YOLOv8m (50% conf.)	98.77%	87.30%	11.82ms
YOLOv8n	93.68 %	90.15%	5.98ms
YOLOv3	98.83 %	90.06%	52.70ms

resulting in faster overall object detection [4]. YOLOv8 offers different variants, ranging from nano to extra-large, to accommodate diverse applications and computational resources [95]. These advantages in YOLOv8 encouraged the researcher to evaluate its performance against YOLOv3 to ensure achieving faster and more accurate object detection.

Table 7.1 compares the object detection performance between YOLOv8 variants, i.e., YOLOv8n (nano) and YOLOv8m (medium), and YOLOv3 on the VMOT dataset regarding mAP, mAR and the average processing time. YOLOv8m achieves a mAP of 97.61% and a mAR of 96.45%, with an average processing time of 11.85ms. Increasing the confidence threshold to 50% slightly boosts the mAP to 98.77% but at the cost of a decrease in mAR to 87.30%. In contrast, the YOLOv8n, a faster variant, exhibits a lower mAP of 93.68% yet maintains a good mAR of 90.15% while significantly reducing the average processing time to 5.98ms. Compared with YOLOv8m, YOLOv3 achieves a competitive mAP of 98.83% but with a relatively lower mAR of 90.06% and a considerably longer average processing time of 52.70ms. The findings underline the trade-offs between detection accuracy and processing efficiency among the evaluated models. However, in ARAware, YOLOv8m is preferred as it aligns better with the scheme's accuracy and real-time performance requirements. Its high mAP and mAR and relatively low processing time make it well-suited for the scheme's core functionality. Most importantly, YOLOv8m's higher mAR (more than 6%) than other models contributes to reducing missing cases in object detection, which is a critical factor for VIP safety, especially with high-risk objects. Note that YOLOv8n's lower processing requirements make it a potential candidate for deploying ARAware on mobile devices (more exploration will be conducted in the future).

Table 7.2: Object Tracking Performance on VMOT: YOLOv8 vs. YOLOv3.

Model	MOTA	MOTP
Deep SORT (YOLOv8m)	94.55 %	80.88%
Deep SORT (YOLOv3)	87.81 %	78.08%

7.1.2 Object Tracking

Since ARAware's Deep Sort-based tracker relies on object detection to initiate tracking, the choice between YOLOv8 and YOLOv3 significantly impacts the overall tracking accuracy. Table 7.2 demonstrates this correlation. It compares the performance of Deep SORT when paired with both YOLOv3 and YOLOv8 on the VMOT dataset, using MOTA and MOTP as metrics. Utilising YOLOv8m with Deep SORT demonstrates remarkable object tracking performance, achieving a MOTA of 94.55% and a MOTP of 80.88%, surpassing YOLOv3 by 6.74% and 2.80%, respectively. This suggests that YOLOv8 provides better object detection for tracking, leading to more accurate overall tracking performance. In particular, the higher MOTA achieved with YOLOv8m indicates fewer false positives, missed detections, and identity switches than YOLOv3. Additionally, the increased MOTP further reinforces the precision in object localisation during tracking with YOLOv8m.

7.1.3 E2E CMO Identification

Given that ARAware's CMO collision prediction (see Chapter 4) depends on the bounding boxes produced by the Deep Sort tracker, which in turn relies on the object detector (as mentioned above), the selection between YOLOv8 and YOLOv3 significantly impacts the ARAware's E2E CMO identification. Table 7.3 summarises the performance comparison of ARAware's CMO identification when using both YOLOv8 and YOLOv3 object detection models (on the VMOT dataset) regarding mAP, mAR, and EmFPS. For the CMO identification task, while both models achieve very high overall mAR above 95%, YOLOv8 maintains a slightly higher value (97.26% vs. 95.91%) and boasts a 5.9% improvement in overall mAP (88.20% vs. 82.30%), with an impressive EmFPS of $32fps$ (nearly doubles YOLOv3's $17fps$). This indicates that in-

tegrating YOLOv8 with ARAware significantly improves its E2E CMO identification compared to YOLOv3. In particular, the higher mAR obtained with YOLOv8 suggests that it contributes to reducing missed CMOs more than YOLOv3, which is crucial for improving VIP safety. For more clarification, in real-world ARAware experiments, the researcher observed that, in night scenarios, YOLOv8 successfully detected cars with bright headlights, while YOLOv3 sometimes failed. This successful detection allows ARAware to analyse object movements and thus successfully identify CMOs. YOLOv8's ability to handle such lighting conditions significantly contributes to the increased mAR of the ARAware's E2E CMO identification compared to YOLOv3. Additionally, the higher mAP indicates that YOLOv8 enables ARAware to identify CMOs more precisely than YOLOv3. This translates to fewer false positives (mistaking non-CMOs for CMOs), ultimately enhancing the overall reliability of ARAware. It is worth noting that when further evaluating YOLOv8 and YOLOv3 using the average IoU (Intersection over Union) metric, YOLOv8 achieved a score of 79.82%, demonstrating better object localisation accuracy (tighter bounding boxes around objects) compared to YOLOv3, which scores 76.96%. This improved localisation accuracy directly contributes to more accurate collision prediction and, thus, more precise CMO identification. Moreover, using YOLOv8 significantly boosts ARAware's efficiency. Its higher EmFPS ($32fps$) successfully ensures ARAware meets the real-time processing requirement (at least $30fps$, as outlined in Chapter 1), unlike YOLOv3.

All of these advantages that YOLOv8 provides to ARAware's E2E CMO identification can be attributed to the advancements and unique architectural features of YOLOv8, as discussed in Sections 7.1.1 and 2.2.1. These advancements, particularly those aimed at enhancing feature extraction and detection accuracy (such as the C2f module) and improving inference speed (like anchor-free detection), set YOLOv8 apart from YOLOv3. This makes YOLOv8 the superior choice for ARAware's CMO identification, offering both significantly faster processing and superior accuracy compared to YOLOv3. Many research efforts are currently underway to enhance YOLOv8's performance. For example, the study by [258] optimises the YOLOv8s (small) variant to improve the accuracy of small object detection for UAV Aerial Photography Scenarios. While it achieves an accuracy improvement of 7.7% mAP, this comes at the expense of an approximately 61% reduction in inference time. Such improvements can be useful only for

Table 7.3: Overall Performance of ARAware's CMO Identification: YOLOv8 vs. YOLOv3.

Component	Over. mAP	Over. mAR	EmFPS
CMO identification (YOLOv8)	88.20%	97.26%	32 fps
CMO identification (YOLOv3)	82.30%	95.91%	17 fps

applications that prioritise detection accuracy over speed. However, a deeper investigation into the potential of existing YOLOv8 improvements to further enhance ARAware performance is left for future work.

7.2 Impact of Frame-Skipping on ARAware

Many hardware, like mobile phones or embedded systems, have limited processing power and battery life. Processing video frames requires resources, and running a complex algorithm on every single frame can quickly drain the battery or slow down the device. By skipping frames, video processing can become less demanding on the hardware, which significantly reduces the overall workload on the hardware's processor and memory, thus improving the overall processing performance [257]. While ARAware achieves real-time performance even when processing every video frame (achieves 32 EmFPS running on GPU), the researchers investigated its performance under different frame-skipping settings. This analysis is to provide deeper insights into ARAware's performance, thus paving the way for improvements that enable the ARAware to run on resource-constrained hardware. It's important to note that skipping frames also means less information is analysed, which can potentially impact the accuracy of the ARAware's CMO identification. Finding a good balance between processing speed (skipping frames reduces processing load) and accuracy (enough frames are processed for effective CMO identification) can facilitate the deployment of ARAware on devices where resources are scarce (future work).

To examine the impact of frame skipping, the researcher evaluated the performance of ARAware's E2E CMO identification using the VMOT dataset under *six* different k-frame skipping settings: 1-frame, 3-frame, 5-frame, 15-frame, 30-frame and 50-frame. Under the k-frame setting, only one frame is fed to the ARAware for every k-input frame. As detailed in Table 7.4, the corre-

Table 7.4: Results of ARAware's CMO Identification Under Different Frame-Skipping Settings.

Setting	Inter-Frame Delay	Over. mAP	Over. mAR	EmFPS	Total Process. Time
1-frame	$0.03s$	88.20%	97.26%	$32fps$	$26.31min$
3-frame	$0.10s$	97.89%	92.76%	$94fps$	$8.94min$
5-frame	$0.17s$	99.72%	81.71%	$158fps$	$5.32min$
15-frame	$0.50s$	100.00%	44.02%	$636fps$	$1.33min$
30-frame	$1.00s$	100.00%	19.14%	$1089fps$	$46s$
50-frame	$1.67s$	100.00%	1.25%	$1594fps$	$31s$

sponding inter-frame delays (the time between processed frames) for these six frame-skipping settings are $0.03s$, $0.10s$, $0.17s$, $0.5s$, $1s$, and $1.67s$, respectively. Table 7.4 also shows that when more frames are skipped, ARAware achieves higher EmFPS and overall mAP while its overall mAR decreases. This is because, for EmFPS, as the number of skipped frames (inter-frame delay) increases, ARAware performs fewer complex computations per second, which translates to a decrease in total processing time, see Eq. (5.6). Consequently, EmFPS increases, indicating faster processing speed compared to settings with lower frame skipping (or shorter inter-frame delay). This aligns with the findings in [257]. Note that, in the 3-frame to 50-frame skipping settings, the EmFPS metric, as defined in Eq. (5.6), explicitly considers skipped frames to reflect how skipping frames reduce processing time for VMOT videos. This is particularly evident in the observed total processing times of all VMOT videos: $26.31min$ for the 1-frame setting, $8.94min$ for the 3-frame setting, $5.32min$ for the 5-frame setting, $1.33min$ for the 15-frame setting, $46s$ for the 30-frame setting, and $31s$ for the 50-frame setting, as shown in Table 7.4.

For mAP, the reason for its increase is that ARAware identifies CMOs based on the difference between their positions in two frames, *i.e.* f_p and f_c, as shown in Fig. 4.7, and since the time difference between each frame is negligible (*e.g.,* $0.03s$ for 1-frame setting), the objects' position changes very slightly or may even remain unchanged, particularly for slow-moving objects. Consequently, when more frames are skipped (i.e., more delay between processed frames), more obvious changes in the object positions are collected (i.e., the difference between such positions increases). As a result, the movement direction of objects is determined more precisely, which leads to an increase in overall mAP. However, when more frames are skipped, detecting the changes in objects' direction becomes more difficult, especially for fast-moving objects or objects

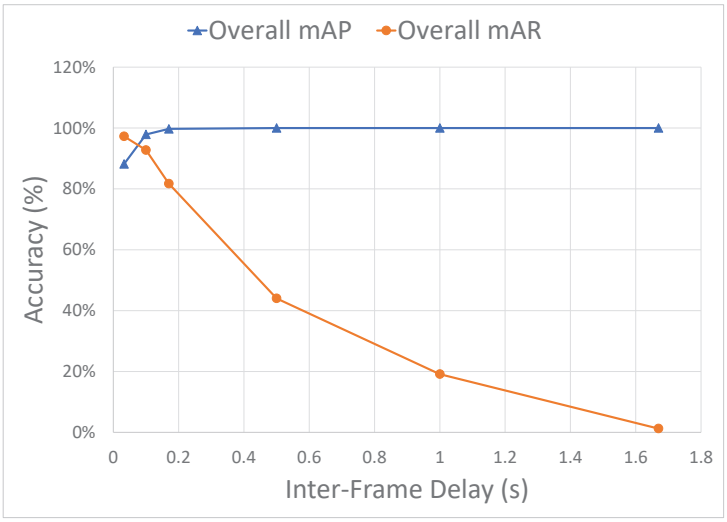


Figure 7.1: The performance of ARAware's CMO Identification With Different Inter-Frame Delay.

that make sudden directional changes. This can lead to a decrease in overall mAR. Fig. 7.1 illustrates the impact of inter-frame delays, resulting from the six different frame-skipping settings, on the overall performance of the ARAware's CMO Identification. Specifically, the mAP increases slightly with the initial increase in inter-frame delay ($0.03s \sim 0.17s$) and then remains at 100% for inter-frame delays of $0.50s$ and above, while the mAR decreases significantly as the inter-frame delay increases, reaching a low of 1.25%.

The performance of CMO identification for each class under different frame-skipping settings is presented in Table 7.5. The results show that, across the 1-frame, 3-frame and 5-frame settings, pedestrians consistently exhibit the lowest mAP and the highest mAR compared to other object classes. This is primarily due to their slower speed than other classes, which results in minimal positional changes between frames. As mentioned above, these minimal changes can make it challenging for ARAware to precisely determine the pedestrians' moving direction in some cases, unlike other classes. For cars, the mAP is generally higher than other classes, except for the 3-frame setting, where it is close to motorcycles, and the 5-frame setting, where it is similar to bicycles. For the 15-frame, 30-frame, and 50-frame settings, the mAP for cars remains at 100%, consistent with other classes. However, in all settings, cars consistently have the lowest mAR among all classes due to their higher speeds, which increase the likelihood of missed cases. Similar observations were found for motorcycles and bicycles.

Table 7.5: Performance of ARAware's CMO Identification on VMOT Under Different Frame-Skipping Settings for Each Class of CMOs.

CMO	GT_c	GT_{Nc}	TP	FP	FN	TN	mAP	mAR
1-Frame Setting (0.03s Inter-Frame Delay)								
Car	182	65	174	19	8	46	91.61%	95.62%
Motorcycle	112	31	108	19	4	12	88.96%	96.62%
Bicycle	77	35	75	17	2	18	86.44%	96.97%
Pedestrian	168	86	167	46	1	40	85.81%	99.83%
Total	539	217	524	101	15	116	88.20%	97.26%
3-Frame Setting (0.10s Inter-Frame Delay)								
Car	182	65	164	1	18	64	99.20%	90.13%
Motorcycle	112	31	103	2	9	29	99.29%	91.14%
Bicycle	77	35	72	2	5	33	98.11%	93.94%
Pedestrian	168	86	156	13	12	73	94.95%	95.83%
Total	539	217	496	18	44	198	97.89%	92.76%
5-Frame Setting (0.17s Inter-Frame Delay)								
Car	182	65	143	0	39	65	100.00%	79.18%
Motorcycle	112	31	90	1	22	30	99.70%	79.24%
Bicycle	77	35	63	0	14	35	100.00%	83.64%
Pedestrian	168	86	137	1	31	85	99.17%	84.80%
Total	539	217	433	2	106	206	99.72%	81.71%
15-Frame Setting (0.50s Inter-Frame Delay)								
Car	182	65	79	0	103	65	100.00%	39.34%
Motorcycle	112	31	59	0	53	31	100.00%	40.47%
Bicycle	77	35	36	0	41	35	100.00%	43.14%
Pedestrian	168	86	94	0	74	86	100.00%	53.13%
Total	539	217	268	0	271	217	100.00%	44.02%
30-Frame Setting (1.00s Inter-Frame Delay)								
Car	182	65	31	0	151	65	100.00%	14.38%
Motorcycle	112	31	30	0	82	31	100.00%	17.42%
Bicycle	77	35	13	0	64	35	100.00%	19.13%
Pedestrian	168	86	62	0	106	86	100.00%	25.63%
Total	539	217	136	0	403	217	100.00%	19.14 %
50-Frame Setting (1.67s Inter-Frame Delay)								
Car	182	65	1	0	181	65	100.00%	0.40%
Motorcycle	112	31	1	0	111	31	100.00%	0.79%
Bicycle	77	35	2	0	75	35	100.00%	1.14%
Pedestrian	168	86	8	0	160	86	100.00%	2.67%
Total	539	217	12	0	527	217	100.00%	1.25%

The above observations indicate that the frame-skipping method improves the ARAware's overall processing speed and CMO identification precision, *i.e.,* by skipping 3 frames and above, but increases the chance of missing CMOs. The researcher believes that skipping frames from 3

to 5 (with an inter-frame delay of $0.17s$ or less) is acceptable for two reasons: (1) the accuracy metrics (both mAP and mAR) remain relatively high and balanced and (2) the missing CMOs are temporary for a few frames (as discussed in Section 6.1.3), i.e., just over a tenth of second. However, more aggressive skipping (15, 30, and 50 frames), corresponding to delays $\geq 0.50s$, significantly compromises CMO identification recall (below 50%), making it unacceptable for practical applications.

Nevertheless, this does not affect the ARAware's usability as it offers the flexibility to be configured as a 1, 3, or 5 frame setting based on the surrounding environment, VIP requirements and hardware processing resources. For example, for VIPs using a device with limited battery life to navigate lower-risk environments like parks or sidewalks, ARAware can be configured to the 5-frame setting to reduce the overall workload on the device's processor (potentially leading to significant improvements in battery life) and achieve high CMO identification precision. For high-risk environments (e.g., road crossings), the ARAware can be set to the 1-frame setting to guarantee fewer missed CMOs. The 3-frame setting can be used to trade-off between overall processing speed (overall workload on the hardware's resources), precision, and recall. In this thesis, the ARAware is configured to the 1-frame setting to maximise VIPs safety. At the moment, this is performed manually. However, in future work, the researcher will explore various adaptive parameter setting schemes that will automate this process and fine-tune ARAware according to its specific environment. This meta-level training will add practical enhancements to the solution, allowing for improved performance and adaptability. Furthermore, it is important to note that increasing the number of skipped frames (up to 5) will not cause a significant delay in the detection of objects or the decision to emit a warning message. This is because the inter-frame delay remains negligible, i.e., $0.17s$ or less.

7.3 Impact of Moving Angle Constraint on MinD

To ensure VIPs' safety when navigating environments with multiple CMOs, the researcher investigates the performance of the MinD scheme under two different settings regarding CMOs'

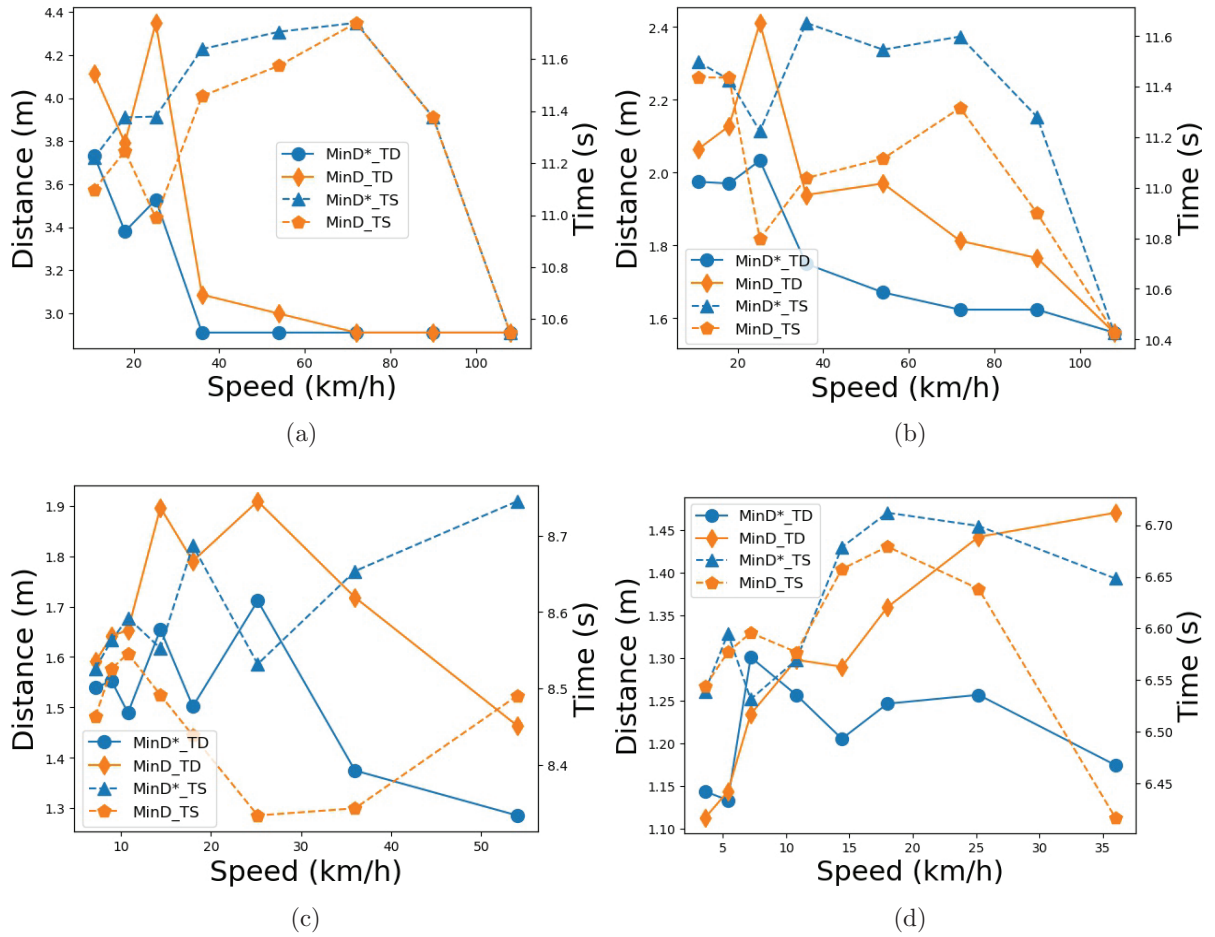


Figure 7.2: The Travelling Distance and Time Separation of Two CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time.

moving angle: *nearest neighbour focus* setting and *comprehensive awareness* setting. When planning a safe path for the VIP, the *nearest neighbour focus* setting considers the moving angle of only the closest CMO, while the *comprehensive awareness* setting considers the moving angles of all CMOs (as specified in constraint 4 of the MinD formulation in Section 4.1.3). This investigation is to identify the setting that best promotes effective free-collision path planning with larger safety buffers and minimal travel distances.

Figures 7.2, 7.3, 7.4, and 7.5 compare the performance of the two aforementioned path planning settings in terms of travelling distance and time separation between the VIP and CMOs at t_f for scenarios involving two, three, five and seven CMOs of different classes, respectively. Each sub-figure (a - d) of these figures shows the results of a particular class: a for cars, b

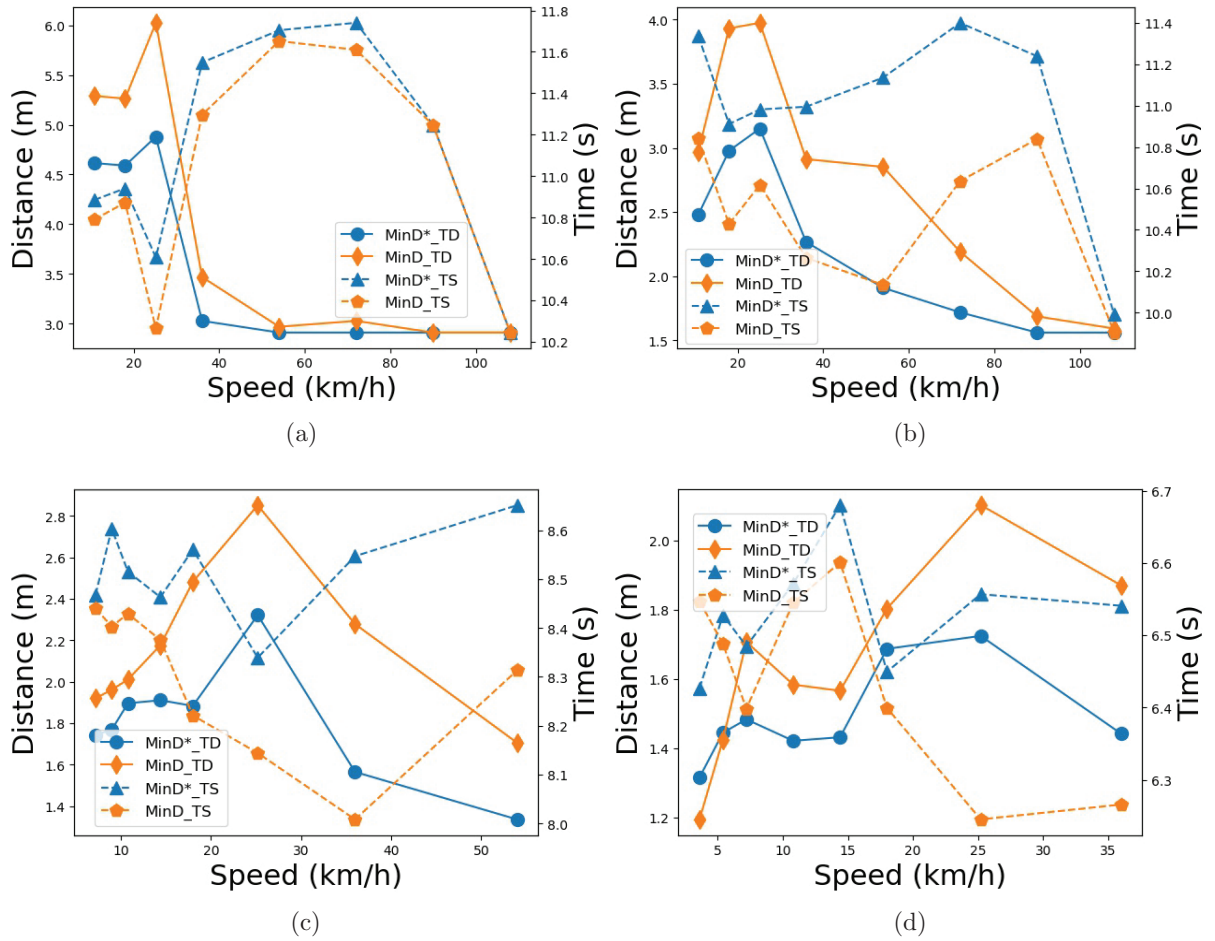


Figure 7.3: The Travelling Distance and Time Separation of Three CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time.

for motorcycles, c for bicycles, and d for pedestrians. In these figures, MinD (orange lines) represents the path planning results using the *nearest neighbour focus* setting, while MinD* (blue lines) represents the path planning results using the *comprehensive awareness* setting. The results reveal significant differences in travelling distance (TD in the figures) and time separation (TS) between the two path planning settings. Notably, in the *nearest neighbour* setting, the VIP travels considerably further to avoid obstacles (from 2 to 7 CMOs), and the time separation (safety buffer) between the VIP and the obstacles is smaller compared to the *comprehensive awareness* setting. As shown in the figures, this increase in travel distance and decrease in time separation varies depending on the number of CMOs and CMOs distributions in the scenes. On average, compared to MinD*, MinD achieves a travel distance that is 9.42%, 16.30%, 17.42%,

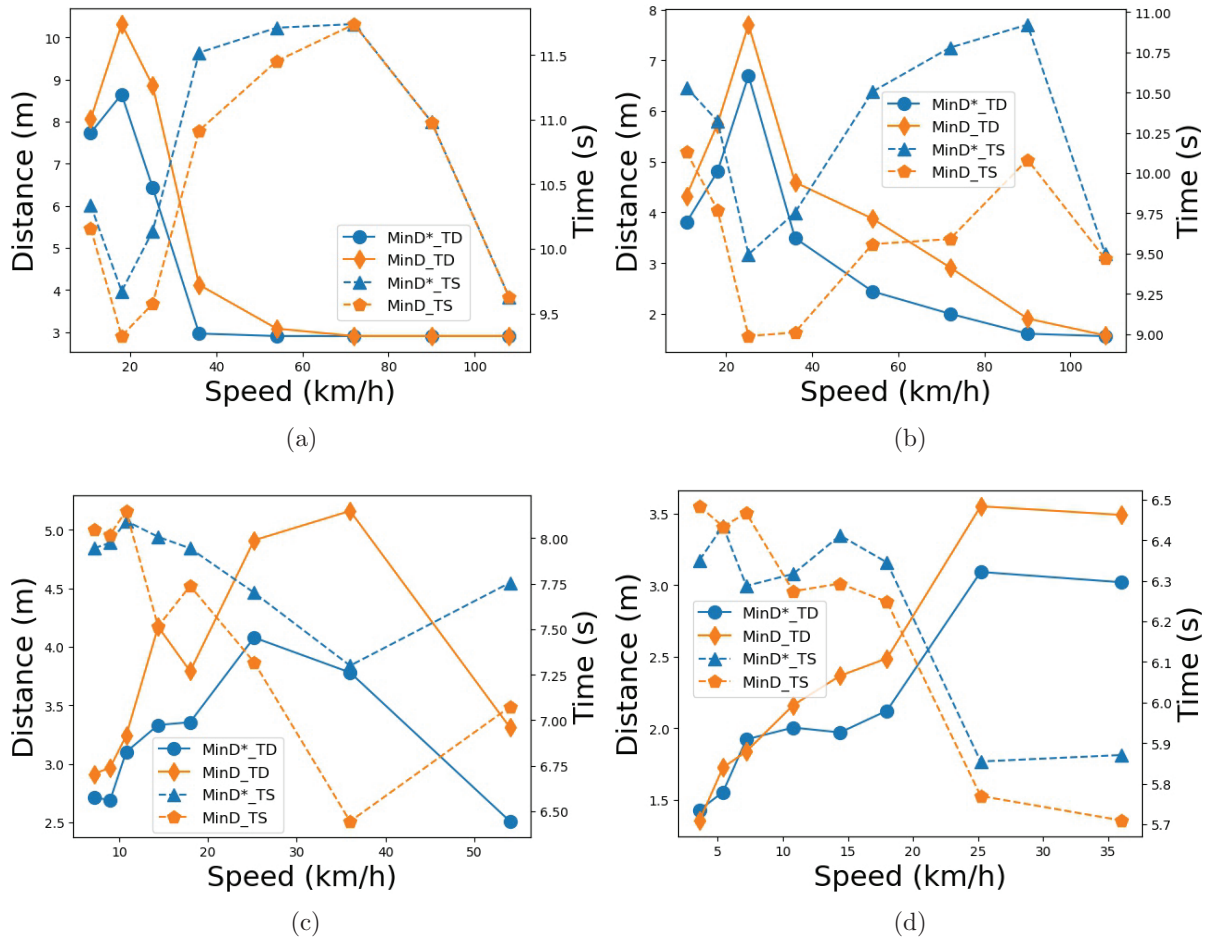


Figure 7.4: The Travelling Distance and Time Separation of Five CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time.

and 12.75% longer and a time separation that is 1.55%, 2.43%, 2.95%, and 3.00% shorter for 2, 3, 5, and 7 CMOs, respectively. Overall, with MinD, the VIP needs to travel 13.97% further to avoid obstacles ($2 \sim 7$ CMOs), with a 2.48% decrease in time separation between the VIP and CMOs compared to MinD*. The reason for this lies in MinD's limited perspective, i.e., as the *nearest neighbour* setting considers the moving angle of only the closest CMO, it may guide the VIP towards paths with hidden risks from other CMOs. This leads to adjusting the VIP path frequently to avoid CMOs, thus increasing travel distance, which can impact the algorithm's usability. It also results in smaller safety zones between the VIP and such CMOs, potentially increasing the chance of infeasible solutions and compromising VIP physical safety. Conversely, MinD* takes a more comprehensive and cautious approach. Considering the moving angles of

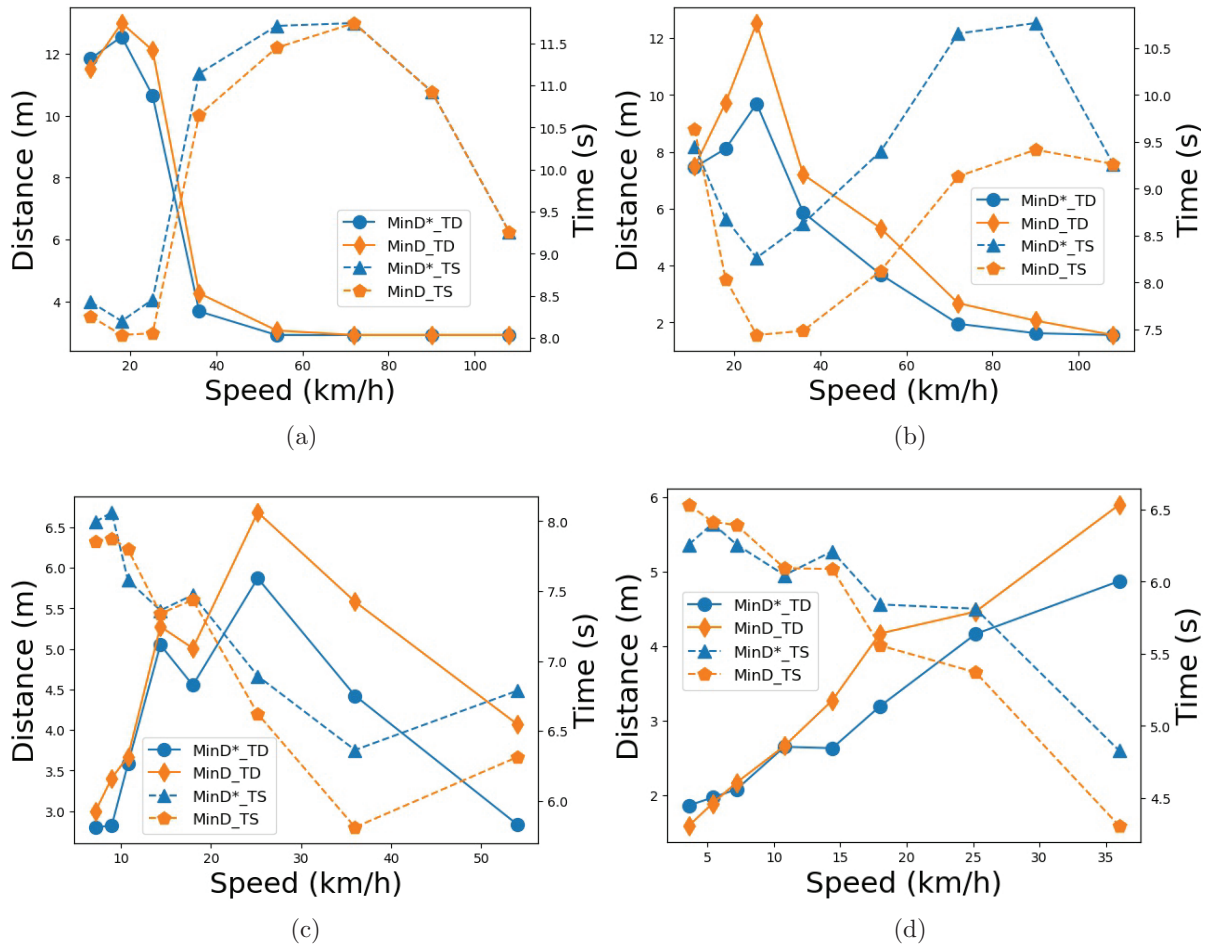


Figure 7.5: The Travelling Distance and Time Separation of Seven CMOs Under Different Moving Speeds. (a) Cars, (b) Motorcycles, (c) Bicycles, and (d) Pedestrians. X-axis: Speed, Y-axis (left): Distance, Y-axis (right): Time.

all surrounding CMOs can identify safer routes for the VIP, especially in dynamic environments. This advantage makes the *comprehensive awareness* setting a more favourable path-planning setting than the *nearest neighbour* setting when prioritising VIP safety. It offers a greater time safety buffer between the VIP and CMOs and minimises the VIP avoidance distance.

7.4 Limitations

The approaches provided in ARAware and MinD bring substantial improvements to assistance technologies for VIPs. However, there are still paths for future improvement. The following section discusses the limitations of both schemes.

7.4.1 Limitation of ARAware

Further optimisation can be carried out to improve the ARAware in terms of motion detection and prototype form-factors and human-machine interface.

The motion detection method of ARAware combines trajectory classification-based and object tracking-based techniques [259]. Specifically, it relies on the Deep SORT tracker for object localisation and uses the ORB extractor to extract features from a scene background to differentiate between movements induced by moving objects and those caused by camera motion, see Section 4.1.2. Unlike background modelling-based approaches [259], ARAware's method offers reduced computational complexity as it does not require background modelling and homography transformations. This helps to achieve faster performance. As discussed at the end of Section 6.1.2, although the ARAware scheme compensates for camera movement, the effectiveness of ARAware's motion detector can deteriorate under certain conditions. These conditions include instances where the camera is moved faster than typical VIP walking speeds (or is involved, *e.g.*, abrupt movements, strong camera shaking, etc.) or when the scene lacks distinctive features. Nevertheless, there are existing studies that specifically address such movements [260] and exhibit improved feature discrimination in challenging scenes [261]. Exploring these works could be investigated to improve ARAware's motion detector; this will be the future work of this thesis.

The ARAware's real-world prototype used a high-performance laptop placed in a backpack during the conducted experiments, and this setup was not without limitations. For example, given the mobile nature of the solution, the proposed scheme has room to use more energy-efficient algorithms. Therefore, for future work, approaches such as lightweight neural networks [262] tailored for mobile devices and a light version of YOLOv8 (e.g., YOLOv8n) [95] can be explored. Additionally, leveraging specialised hardware like the OpenNCC NCB,[1] designed for Deep Learning applications, holds promising potential to greatly enhance the usability of ARAware.

Furthermore, the ARAware scheme offers an effective warning strategy that emits timely alarms

[1] https://www.crowdsupply.com/eyecloud/openncc-ncb

and delivers descriptive acoustic warning messages to VIPs about potential hazards in their surroundings. For the prototype of ARAware, bone-conduction headphones were opted to allow ambient sounds to remain audible since VIPs often rely on sound for navigation. However, this initial prototype of ARAware inherently needs $1 \sim 2s$ to generate and deliver warning messages, primarily due to the text-to-speech conversion process for warning messages. Future work will focus on developing a rapid and effective alert mechanism proportional to the surrounding threats' severity while guaranteeing concise threat perceptions. This will significantly improve the performance of ARAware's prototype warning emission. Exploring the possibility of encoding instructions into signals or focusing on what makes sense to the VIP requires additional studies focused on human-computer interaction (HCI). To this end, the researcher will investigate and evaluate various warning methods for VIPs [263, 243] for a better trade-off between delivering alarms quickly and ensuring that VIPs can effectively perceive them.

7.4.2 Limitations of MinD

The MinD demonstrates superior performance in collision-free local path planning compared to the state-of-the-art APF and the baseline NBRD. This is achieved by minimising CMO avoidance distance, ensuring significant time separation (safety buffer) between VIPs and CMOs, and maintaining a 0% collision rate, all while achieving real-time performance. However, there is still room for further improvement to ensure even more comprehensive safety for VIPs. In the current version of MinD, the path planning process primarily focuses on avoiding CMOs (objects with a potential collision with the VIP), neglecting non-CMOs (objects with no potential collision). Although MinD plans safe paths that are free from CMOs, there remains the potential for unsafe situations to arise. These situations involve instances where a planned path, while clear of CMOs, could still lead to a potential collision with other objects (non-CMOs) in the environment. To avoid such situations, future work will focus on refining MinD to account for non-CMO objects. This could involve introducing additional constraints or reformulating existing ones. For example, the safe time separation constraint (C_3, see Section 4.1.3) could be redefined to incorporate non-CMOs and ensure a safe buffer between the VIP and such

objects during navigation. Through such adjustment, MinD can be more reliable in planning free-collision paths that maximise VIP safety by considering all surrounding objects, including both CMOs and non-CMOs.

Evaluating MinD within a practically parameterised simulator, where its parameters were learned from real-world ARAware experimentation, provides valuable insights through a safe' what-if' experimentation and in a near-to-realistic way. However, further evaluation in complex real-world environments is necessary for comprehensive validation. Therefore, future work will focus on real-world validation by integrating MinD with the ARAware's prototype, and this combined prototype will then be deployed for VIP navigation tasks in diverse environments with varying crowd densities and lighting conditions. Evaluating MinD's performance under these real-world complexities will provide valuable insights into its adaptability and operational efficiency. These insights will be crucial for refining MinD's robustness and generalisability for real-world VIP navigation scenarios.

Moreover, before the MinD's real-world testing, future work will further evaluate MinD using the simulator. This evaluation will focus on more complex scenarios with CMOs exhibiting sudden trajectory changes and variations in velocities and showing manoeuvre behaviour to VIPs. This additional testing aims to ensure that MinD's algorithm maintains enough safe separation distances from CMOs even when they exhibit erratic behaviour. This will reduce the risk of collisions in real-world situations where objects' movement might be unpredictable. This further evaluation also aims to recognise situations where generating a safe and feasible path for the VIP might be impossible due to unpredicted objects' behaviour. Identifying these scenarios will allow future work to improve MinD's ability to handle such infeasible scenarios, potentially through the adaptation of more constraints and/or objectives or the incorporation of different planning methods.

In conclusion, this chapter has delved deeper into exploring the performance of ARAware and MinD under different settings to gain valuable insights into their effectiveness, and it has critically examined their limitations. Specifically, it compared the performance of YOLOv3 and YOLOv8 object detectors, assessed their impact on ARAware's overall effectiveness, and exam-

ined the effects of different frame-skipping settings on CMO identification. It explored MinD's effectiveness in planning collision-free paths under two different configurations. It then discussed the limitations inherent in the algorithms, design choices, and evaluation methodologies of ARAware and MinD, providing a foundation for future research and development efforts. The next chapter will offer a comprehensive conclusion to this thesis, summarising key findings and outlining future works to improve the scheme proposed by this thesis.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

While existing vision-based VIP mobility assistant solutions are promising, they cannot effectively prevent collisions with different types of dangerous objects travelling at low-to-high speeds (i.e., Critical Moving Objects, CMOs). Therefore, this research presented AR-AwareAvoid (or Ambient Risk Awareness and Avoidance), a novel AI-powered assistance scheme designed to help VIPs safely navigate challenging outdoor environments involving various classes of CMOs, i.e., cars, motorcycles, bicycles, and pedestrians, moving at varying speeds. AR-AwareAvoid tackles the limitations of current solutions through two core components: the CMO identification (ARAware) scheme and the local path planning (MinD) scheme. ARAware effectively and efficiently identifies CMOs and provides longer VIP-reaction times for CMO avoidance by simultaneously addressing the combined issues of real-time CMO identification, fine-grained risk level-based CMO classification, and prioritised CMO warning. Meanwhile, MinD plans the shortest path required to avoid CMOs while considering CMOs' motion constraints and VIPs' limited physical abilities (walking speed and auditory perception) and provides look-ahead safety prediction on the planned route. The key contributions of this research lie in developing and evaluating these two novel core schemes within AR-AwareAvoid as follows.

CMO Identification. For CMO identification, the researcher proposed ARAware (or Ambient

Risk Awareness), a real-time E2E camera-based CMO avoidance scheme, from video capture to message delivery. In constructing the ARAware pipeline, the design and optimisation of all scheme modules were explicitly explored, considering the properties and perceptions of the VIP in practice. Specifically, the researcher developed a Deep Learning-based CMO identifier capable of analysing incoming video in real time without losing critical information. Furthermore, a novel object-VIP collision prediction method based on objects' moving directions was developed for effective collision detection. Through extensive empirical background study, a CMO risk estimation and classification method was also proposed, along with a risk-aware prioritised CMO warning strategy that achieves a subtle trade-off between VIP safety and scheme usability. A 2D to 3D coordinate projection method was also constructed to enhance the accuracy of the ARAware collision and risk predictor. ARAware was designed with the future in mind. Its modular architecture allows for integrating different object detectors, trackers, and distance estimators. This flexibility allows for prioritising higher accuracy or faster processing speeds based on VIPs' specific needs, ensuring ARAware remains adaptable to individual requirements. Extensive prototype-based experiments conducted in different real-world scenarios demonstrated the effectiveness of ARAware in achieving accurate CMO identification (with an overall mAR of 97.26% and overall mAP of 88.20%) and precise CMO risk level classification (with an overall mAR of 100% and overall mAP of 91.69%) and illustrated the trade-offs between in-time warnings and the reduction of false alarms. The experiments also showed that ARAware successfully achieves real-time performance, processing $30fps$ input video at a rate of $32fps$ ($31ms$, running on an NVIDIA GeForce RTX 2080 GPU), making it a practical solution for real-world scenarios. Compared to the closest state-of-the-art approach, DEEP-SEE [28], ARAware achieved significantly higher CMO identification accuracy by 42.62% in mAR and 10.88% in mAP, with 93% faster end-to-end processing speed.

Path Planning. The researcher also developed MinD, a novel collision-free local path planning scheme to assist VIPs' mobility in outdoor environments. MinD was specifically designed to be consistent with VIPs' limited capabilities; it plans the shortest path required by the VIP to avoid CMOs, where the VIP's walking speed and auditory perception and different CMO types and movement patterns are comprehensively considered. This is to ensure a sufficient, safe

time for VIPs to react and effectively avoid collisions. It utilises adaptive thresholds based on CMO risk levels; this is to enable timely path planning for high-risk CMOs (like cars) to ensure safe collision avoidance while avoiding unnecessary VIP redirection for low-risk CMOs (like pedestrians) to improve scheme usability. It also incorporates look-ahead safety prediction for optimal path selection to ensure more safety for VIPs. Another contribution is that to address the non-convex nature of the VIP-local path planning problem, the researcher reformulated the non-convex problem into solvable convex local path planning by leveraging Sequential Convex Programming (SCP). This is to find a near-optimal solution while reducing the computational complexity of the original non-convex problem. Evaluation using an empirically parameterised simulator demonstrated that MinD outperforms the well-known APF path planning and the baseline NBRD in minimising the travelling avoidance distance (requiring an average avoidance distance of only $2.75m$ for all CMO classes (up to 7) moving at various speeds (up to $108km/h$), which is significantly shorter than APF by 84.47% and NBRD by 75.94%), allowing a large time separation between the VIP and CMO ($11.00s$ for cars, $10.78s$ for motorcycles, $8.23s$ for bicycles and $6.43s$ for pedestrians, which is, on average, $\geq 4.80s$ and $\geq 1.24s$ compared to APF and NRD, respectively), and ensuring a 0% collision rate while maintaining a negligible average processing time of only $1.2ms$, running on an NVIDIA GeForce RTX 2080 GPU.

Further Contributions. A key decision for ARAware's effectiveness involved a deep comparison of YOLOv8 and YOLOv3, aiming to identify the object detection model that best fulfils the scheme's accuracy and real-time performance requirements. This comparison evaluated both models' effectiveness in object detection, object tracking, and E2E CMO identification, considering both accuracy metrics and processing speed. The evaluation revealed that, in object detection, YOLOv8 outperforms YOLOv3 in achieving higher mAR (more than 6%, indicating fewer missed detections) with a significantly lower average processing time of 11.85ms (77.51% faster than YOLOv3). When paired with the ARAware's Deep SORT-based tracker, YOLOv8 provides more accurate object detection for tracking than YOLOv3 (surpassing YOLOv3 by 6.74% MOTA and 2.80% MOTP), leading to more precise overall tracking. For the CMO identification task, YOLOv8 boasts a 5.9% improvement in overall mAP (88.20% vs. 82.30%), with an impressive EmFPS of 32fps (nearly doubles YOLOv3's 17fps). These findings emphasised

YOLOv8 as the preferred choice for ARAware, as it perfectly aligns with the system's accuracy and real-time performance requirements.

A further contribution is that recognising the potential limitations of computational resources on some devices, the researcher investigated the effectiveness of YOLOv8-based CMO identification with the frame-skipping approach. This evaluation explored different k-frame skipping settings (e.g., k = 1, 3, 5, 15, 30, and 50) to identify a sweet spot between reduced processing load on the hardware and maintaining acceptable accuracy. This approach paves the way for deploying ARAware on resource-constrained devices in the future to improve its usability. Finally, to ensure VIP safety when navigating environments with multiple CMOs, the MinD path planning scheme was also investigated under two distinct settings: *neighbour focus* and *comprehensive awareness*. By evaluating these settings, the research identified the configuration that best promotes safer and more efficient mobility for VIPs: *comprehensive awareness* setting.

Overall, this research has made significant contributions towards enhancing the safety and independence of VIPs navigating outdoor environments. ARAware assists VIPs with real-time CMO identification, CMO risk estimation, and prioritised warnings, enabling timely responses to threats. MinD, the collision-free local path planning, further reinforces VIP safety by generating optimal paths that minimise avoidance distances and ensure large, safe buffers. These advancements pave the way for a fully practical and comprehensive VIP mobility assistance system, ultimately promoting greater independent mobility for VIPs.

Requirements Fulfillment. At the end of Chapter 3, several requirements are listed to overcome the limitations of current studies proposing assistive mobility solutions for VIPs. Table 8.1 demonstrates that this thesis has successfully fulfilled all specified requirements, with the exception of one that will be addressed in future work.

Table 8.1: VIP Assistive Mobility Solution Requirements

n	Requirement	Achieved	Details
1	Detection and recognition of common obstacles encountered in everyday outdoor environments during day and night.	Yes	ARAware can detect common outdoor objects (up to five objects), i.e., up to two cars, one motorcycle, up to three bicycles, and up to five pedestrians as verified through experimentation; it also can be customised to detect more object classes based on VIP requirements. ARAware was evaluated using datasets collected during day and night, ensuring robust performance across different lighting scenarios. However, future work requires further testing under challenging lighting conditions, such as extremely bright sunlight and very darkness, to ensure ARAware 's reliability in all possible real-world scenarios. Future work will also involve evaluating ARAware with more than five objects.

Continued on next page

Table 8.1 – continued from previous page

n	Requirement	Achieved	Details
2	Recognising and tracking moving objects.	Yes	ARAware can track and recognise moving objects to predict their potential collision with VIPs by analysing their movement angle relative to the VIP's position (see Table 6.4). It can also estimate their moving speed (see Figures 6.2 and 6.3 and Table 6.3).
3	Prediction of potential collisions with dynamic obstacles.	Yes	ARAware can effectively predict potential collisions with CMOs by analysing their moving angles; see Tables 6.4 and 6.5.
4	Accurate estimation of VIP-object distance.	Yes	ARAware can accurately estimate the proximity of the objects relative to the VIP; see Fig. 6.1 and Table 6.3.
5	Extended collision identification area beyond $10m$.	Yes	ARAware identifies threats up to $100m$; see Table 6.3.
6	Extended distance estimation for early warning.	Yes	At a range of $100m$, ARAware can warn about a car travelling at $60km/h$ $6s$ before a potential collision.

Continued on next page

Table 8.1 – continued from previous page

n	Requirement	Achieved	Details
7	Extended collision prediction window beyond $1 \sim 2s$.	Yes	ARAware offers a dynamic collision prediction window that considers the estimated speeds of the CMO and the VIP, along with the relative distance between them and the risk level of the CMO. This window typically guarantees a minimum threshold of around $5s$ for all CMO types, including cars, motorcycles, bicycles, and pedestrians, moving up to $60km/h$; see Table 6.8.
8	Identifying threats moving at varying speeds and prioritising their warnings according to their risk level and potential collision time.	Yes	ARAware identifies threats moving from $1.5km/h$ to $60km/h$ and prioritises warning emissions based on the estimated CMO's risk level (considering the CMO's speed and type) and predicted collision time between the VIP and the CMO.
9	Detection of faster-moving objects ($\geq 22km/h$).	Yes	ARAware detects CMOs moving up to $60km/h$; see Table 6.3.
10	Providing detailed information about detected threat.	Yes	ARAware provides the VIP with warning messages that include crucial information about the CMO's type, risk level, incoming direction, and estimated collision time.

Continued on next page

Table 8.1 – continued from previous page

n	Requirement	Achieved	Details
11	Reduction of false alarms.	Yes	ARAware successfully achieves a balance between providing timely warnings and reducing false alarms; see Table 6.7 and discussions in Section 6.1.3.
12	Real-time performance ($\geq 30fps$, processing time for each frame should be $\leq 33ms$).	Yes	Both ARAware and MinD achieve real-time performance; ARAware achieves an average processing time of $31ms$ ($32fps$), see Tables 6.4 and 6.10, and MinD requires an average processing time of $1.20ms$, see Table 6.14.
13	Resilience to infrared radiation interference (sunlight).	No	Since the proposed scheme in this thesis relies on camera-based technology, the camera may experience interference from intense sunlight. Future efforts will address this issue to ensure robust performance under varying lighting conditions.
14	Resilience to background noise and environmental acoustic interference.	Yes	Background noise and acoustics do not impact a camera's ability to capture clear images or videos.
15	Reduced scanning burden through expanded field of view.	Yes	The camera offers a large enough field of view, ensuring all obstacles within the VIPs' vicinity are captured.

Continued on next page

Table 8.1 – continued from previous page

n	Requirement	Achieved	Details
16	Optimal collision-free local path planning.	Yes	MinD calculates the shortest collision-free local path required by the VIP to avoid different types of CMOs (cars, motorcycles, bicycles, and pedestrians) moving at varying speeds (from $3.6km/h$ to $108km/h$); see Figures 6.6 to 6.10 and Table 6.11.
17	Dynamic collision-free path adaptation.	Yes	MinD dynamically calculates collision-free local paths to adapt to real-time environmental changes.
18	Generating paths that are unblocked with other threats in the near future.	Yes	MinD conducts a look-ahead safety prediction trajectory analysis of the optimal path to ensure more safety for VIPs and avoid unnecessary transitions leading to longer travel distances.
19	Incorporation of object movement patterns in path planning.	Yes	MinD considers CMOs' and VIP movement patterns when generating collision-free navigation paths.
20	Consideration of VIP walking speeds in path planning	Yes	MinD is specially crafted to accommodate the constrained capabilities of VIPs, including limited walking speed and audio perception.
21	Efficient path planning (processing time for each frame should be $\leq 33ms$.	Yes	MinD achieves fast convergence and a very low average computation time; see Table 6.14.

Continued on next page

Table 8.1 – continued from previous page

n	Requirement	Achieved	Details
22	Avoiding reliance on artificial force-based path planning.	Yes	MinD does not rely on the concept of artificial force.
23	Reduced dependency on extensive data collection and training.	Yes	MinD operates without extensive data collection and training for path planning.

8.2 Future Work

While ARAware and MinD demonstrate significant potential, there is always room for improvement. Building upon the limitations identified in these schemes, discussed in Chapter 7, this section identifies several areas for future exploration that remained due to time constraints in conducting the current research.

Generally, this thesis successfully implemented a real-world prototype of ARAware and conducted extensive experiments in outdoor environments. These environments included spacious private outdoor lands, providing the necessary control and flexibility for precise setup and evaluation while ensuring volunteers' and public safety. The experiments in such environments involved up to five volunteers who walked, cycled, and drove at different speeds (up to $60km/h$) and located at various distances (up to $100m$). These volunteers followed predefined scenarios, such as moving inside or outside a critical zone (see Figs. 4.7 and 6.4) and moving toward the camera at specific speeds. The scenarios included a combination of objects from different classes: up to two cars, one motorcycle, up to three bicycles, and up to five pedestrians. For safety consideration, these objects were distributed relatively sparsely (not densely packed together). Experiments were conducted during daytime and nighttime, excluding periods with excessively bright sunlight or extreme darkness.

As a result, ARAware can currently be considered a promising stand-alone system for identifying and alerting to potential threats (CMOs) faced by VIPs in outdoor environments. However, delivering this system to VIPs and fostering their trust in its utility necessitates further endeavours. Primarily, ARAware should be subjected to rigorous evaluation by VIPs during their everyday outdoor activities. This is to ensure ARAware's effectiveness in protecting VIPs from threats in diverse and complex scenarios and perfectly aligns with their needs and usability preferences. Central to this endeavour is securing ethical approvals to conduct experiments and adhering rigorously to safety and privacy protocols. These measures are imperative to uphold the safety and privacy of both VIPs and the public, reinforcing the integrity of the ARAware system. The researcher will prioritise addressing these considerations to enhance the accessibility and reliability of ARAware for the VIP community.

While simulation with empirical parameters provided valuable insights into the MinD scheme's performance, implementing a real-world prototype of MinD is essential. This is to facilitate the evaluation of MinD's performance in planning collision-free local paths under a wide range of realistic dynamic conditions, ensuring that it meets the strict requirements of VIP safety. The MinD scheme has demonstrated promising results in simulated environments, but its performance in real-world settings may differ due to several factors. Specifically, real-world data is more variable and unpredictable than synthetic or simulation data, including differences in objects' behaviour, environmental conditions, and unexpected obstacles. For example, in the real world, objects such as pedestrians and vehicles often move in less straight and more erratic paths compared to simulation scenarios. They can suddenly accelerate, decelerate, or change direction, making path planning more challenging. Additionally, objects can interact and influence each other's movements, leading to unpredictable trajectories and complex scenarios. This complexity makes it difficult for MinD to accurately predict and react to these movements. Additionally, objects may suddenly appear or jump into the VIP's path, such as pedestrians abruptly altering their course or pets darting out from behind others, making path planning even more challenging. Real-world scenarios also introduce variability in lighting conditions, weather effects, and environmental factors like shadows and reflections, which can hinder MinD's ability to accurately predict object movements. As a result, MinD may face

challenges in maintaining the same level of accuracy and effectiveness. To ensure VIPs' safety, MinD must be robust enough to handle such real-world challenges effectively. Transitioning MinD from simulation to reality and conducting rigorous real-world evaluations are essential next steps to assess its practical utility and reliability before handing it to VIPs.

Since MinD was designed to complement the functionality of ARAware, where ARAware identifies and warns about threats and MinD plans optimal routes to avoid them, the researcher's future step is to integrate these components into a unified system, i.e., AR-AwareAvoid. This integrated system aims to boost VIPs' safety and independence by combining ARAware and MinD's capabilities. Achieving this integration demands further development and refinement. The researcher will ensure seamless communication and interoperability between ARAware and MinD, facilitating efficient information exchange and effective coordination. Extensive testing and validation will verify AR-AwareAvoid's reliability and performance across diverse real-world scenarios. Additionally, VIP feedback will be gathered and incorporated into the system's design and functionality, ensuring AR-AwareAvoid's usability and accessibility.

In the following, the future improvements for each scheme are outlined in more detail. For ARAware, the researcher plans to conduct the following improvements:

- As mentioned in Chapter 7, current ARAware's motion detection method can deteriorate with rapid camera movements or scenes lacking distinctive features. To address this, future work will explore existing research on feature discrimination in challenging scenes (e.g., [260, 261]) and will also investigate the background modelling-based approaches (e.g., [259]) and their suitability for achieving end-to-end real-time performance within ARAware. It's important to acknowledge that while some existing methods might achieve real-time performance individually, integration with other modules can introduce new processing demands, impacting the overall efficiency.

- Future work will expand the current CMO risk-level classification criteria beyond the CMO's class and moving speed. It will incorporate additional factors associated with objects (e.g., weight, shape, etc.) and VIPs (e.g., age, health, etc.). Considering these

additional factors, ARAware's CMO risk-level estimation can provide a more personalised and accurate assessment of potential threats, ultimately enhancing VIP safety.

- While ARAware currently classifies riders (cyclists and motorcyclists) based on speed thresholds to extend the distance at which small-size CMOs, i.e., motorcycles and bicycles, are detected (discussed in Chapter 4), future work will explore re-training YOLOv8 to directly distinguish between cyclists, motorcyclists, and pedestrians and reduce reliance on estimated speed for classification and to investigate the impact of the re-trained YOLOv8 on the overall ARAware's CMO identification accuracy.

- To push ARAware's performance to new heights, future work will involve a comprehensive evaluation of many alternative approaches for object detection (e.g., SSD [90] and other YOLO variants like YOLOv6 [264], YOLOv7 [265], etc.), object tracking (e.g., Deep OC-SORT [266], SiamMask [267], etc.), and long-range distance estimation (e.g., improved version of Disnet [268], LiDAR-based method [269], etc.).

- To enhance ARAware's usability, future work will focus on developing a mobile-friendly prototype. This will involve utilising lightweight neural network architectures designed specifically for mobile devices (e.g., [262]) or employing a lightweight version of YOLOv8, YOLOv8n [95], and leveraging specialised hardware like the Intel NCS2[1] designed for Deep Learning applications, which potentially improves processing efficiency and power consumption, making the mobile prototype more practical.

- Future work will explore human-computer interaction (HCI) studies and evaluate various warning methods for VIPs (e.g., [263, 243]) to reduce the $1 \sim 2s$ delay in delivering warnings caused by text-to-speech conversion (as discussed in Chapter 7). This is to develop a rapid and effective alert mechanism proportional to threat severity, ensuring clear and concise communication of potential dangers to VIPs.

- While ARAware has shown promising results, several factors not covered in this research could significantly impact its performance. These include the quality and specifications of the camera lens, head movements, VIP height, and gait pattern (including variations

[1] https://software.intel.com/content/www/us/en/develop/hardware/neural-compute-stick.html

in posture and limb movement). Variations in lens quality can affect object detection and tracking accuracy, while unintended head movements and gain can introduce noise into the captured video data. Additionally, the VIP's height can influence the perspective at which the camera captures the environment, affecting the perceived distance and size of objects, thus impacting the overall accuracy of ARAware. Future work will explore calibrating ARAware for different lenses, developing algorithms to compensate for head movements and gain variations, and employing adaptive height calibration techniques for different VIPs' heights. Addressing these factors is crucial for refining ARAware and providing more personalised and reliable assistance for VIPs.

- Future work will prioritise investigating ARAware's performance in more intricate real-world scenarios, particularly within urban environments characterised by high dynamic nature. To achieve this, comprehensive ethical guidelines and procedures will be established for conducting trials with VIPs in real-world settings, ensuring the safety and well-being of participants. Then, extensive data will be collected from various real-world settings, such as parks, sidewalks, and streets, to capture a wide range of conditions and scenarios that VIPs might encounter. This includes different daylight hours (including sunrise and sunset) and nighttime with various lighting conditions (e.g., streetlights and bright cars' headlights), weather conditions (e.g., rain, snow, and fog), levels of pedestrian and vehicular traffic, and a wide range of object densities, speeds, and unpredictable movement patterns (e.g., jaywalking pedestrians). These experiment trials will be conducted with a large and diverse group of VIPs to assess ARAware's effectiveness and usability across different VIP needs and preferences. Future work will also include continuously collecting feedback from VIPs and making iterative improvements to ARAware based on real-world usage experiences. This is to ensure that ARAware evolves effectively and meets the VIPs' needs. Additionally, edge computing and distributed processing techniques [270] will be explored to efficiently handle ARAware's scalability and manage larger numbers of objects while maintaining real-time performance.

To further enhance MinD's capabilities, future work will focus on the following improvements:

- Future work will prioritise conducting further simulation-based evaluations focusing on more challenging scenarios in which CMOs exhibit sudden trajectory changes, velocity variations, and VIP manoeuvring. This will rigorously assess MinD's ability to plan collision-free paths and maintain safe separation distances between the VIP and CMOs in such complex scenarios, allowing for further refinement of its effectiveness and thus ensuring safe path planning in real-world deployments.

- Currently, MinD focuses on avoiding CMOs. Therefore, future work will refine MinD to consider non-CMO objects as well to ensure more safety for VIPs. This could involve introducing additional constraints or reformulating existing ones to ensure safe buffers between the VIP and all surrounding objects during navigation.

- Following further rigorous simulation-based evaluations, MinD will undergo testing in real-world scenarios. This will involve integrating MinD with the existing ARAware prototype to create a combined AR-AwareAvoid prototype. Extensive real-world experiments will also be conducted under varying complexities, including diverse crowd densities and lighting conditions. These experiments aim to gather valuable in-depth insights and ensure the practicality and effectiveness of MinD, as well as the fully integrated scheme, AR-AwareAvoid, in real-world environments.

- The fully integrated scheme, AR-AwareAvoid, will likely require further improvements to effectively assist VIP navigation in real-world environments. This will potentially involve developing more components or enhancing existing functionalities. For example, developing a real-time mechanism that ensures the VIP adheres to the proposed path at the intended pace and triggers notifications to guide the VIP back on the path in case of deviations or slowing down walking speed, and also considering more types of obstacles, including static obstacles (e.g., obstructions) for a more safe independent navigation experience.

By achieving these future work improvements, the research presented in this thesis can be further extended to provide an even more robust, practical, user-friendly, and ultimately safe VIP mobility assistance solution.

Bibliography

[1] J. E. Dowling and J. L. Dowling, *Vision: How It Works and What Can Go Wrong*. The MIT Press, 03 2016.

[2] R. Szeliski, *Computer vision: algorithms and applications*. Springer Nature, 2022.

[3] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. Awwal, and V. K. Asari, "A state-of-the-art survey on deep learning theory and architectures," *electronics*, vol. 8, no. 3, p. 292, 2019.

[4] M. Sohan, T. Sai Ram, R. Reddy, and C. Venkata, "A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*. Springer, 2024, pp. 529–545.

[5] L. Leal-Taixé, "Multiple object tracking with context awareness," *arXiv preprint arXiv:1411.7935*, 2014.

[6] H.-y. Zhang, W.-m. Lin, and A.-x. Chen, "Path planning for the mobile robot: A review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.

[7] M. A. Haseeb, J. Guan, D. Ristic-Durrant, and A. Gräser, "Disnet: a novel method for distance estimation from monocular camera," in *Proceedings of the 10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS*, 2018, pp. 1–6.

[8] R. R. Bourne, S. R. Flaxman, T. Braithwaite, M. V. Cicinelli, A. Das, J. B. Jonas, J. Keeffe, J. H. Kempen, J. Leasher, H. Limburg *et al.*, "Magnitude, temporal trends, and

projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis," *The Lancet Global Health*, vol. 5, no. 9, pp. e888–e897, 2017.

[9] T. Ahmed, P. Shaffer, K. Connelly, D. Crandall, and A. Kapadia, "Addressing physical safety, security, and privacy for people with visual impairments," in *The Twelfth Symposium on Usable Privacy and Security*, Jun. 2016, pp. 341–354.

[10] A. Short, R. Grzebieta, and N. Arndt, "Estimating bicyclist into pedestrian collision speed," *International Journal of Crashworthiness*, vol. 12, no. 2, pp. 127–135, Aug. 2007.

[11] D. C. Richards, "Relationship between speed and risk of fatal injury: pedestrians and car occupants," Department for transport, London, United Kingdom, Tech. Rep., Sep. 2010.

[12] N. S. Mala, S. S. Thushara, and S. Subbiah, "Navigation gadget for visually impaired based on iot," in *2017 2nd International Conference on Computing and Communications Technologies (ICCCT'17)*, Feb. 2017, pp. 334–338.

[13] F. E.-Z. El-Taher, A. Taha, J. Courtney, and S. Mckeever, "A systematic review of urban navigation systems for visually impaired people," *Sensors*, vol. 21, no. 9, pp. 1–35, 2021.

[14] R. Manduchi and S. Kurniawan, "Mobility-related accidents experienced by people with visual impairment," *Insight: Research and Practice in Visual Impairment and Blindness*, vol. 4, no. 2, pp. 44–54, Feb. 2011.

[15] S. Bhatlawande, M. Mahadevappa, J. Mukherjee, M. Biswas, D. Das, and S. Gupta, "Design, development, and clinical evaluation of the electronic mobility cane for vision rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 6, pp. 1148–1159, Nov. 2014.

[16] R. Jafri, R. L. Campos, S. A. Ali, and H. R. Arabnia, "Visual and infrared sensor data-based obstacle detection for the visually impaired using the google project tango tablet development kit and the unity engine," *IEEE Access*, vol. 6, pp. 443–454, Oct. 2017.

[17] A. Ali and M. Abou Ali, "Blind navigation system for visually impaired using windowing-based mean on microsoft kinect camera," in *2017 Fourth International Conference on Advances in Biomedical Engineering (ICABME)*. IEEE, 2017, pp. 1–4.

[18] R. Jafri, R. L. Campos, S. A. Ali, and H. R. Arabnia, "Visual and infrared sensor data-based obstacle detection for the visually impaired using the google project tango tablet development kit and the unity engine," *IEEE Access*, vol. 6, pp. 443–454, 2018.

[19] L. Ran, S. Helal, and S. Moore, "Drishti: An integrated indoor/outdoor blind navigation system and service," in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the*. IEEE, 2004, pp. 23–30.

[20] O. Elkhalili, O. M. Schrey, P. Mengel, M. Petermann, W. Brockherde, and B. J. Hosticka, "A 4/spl times/64 pixel cmos image sensor for 3-d measurement applications," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1208–1212, Jul. 2004.

[21] S. Venkateswar and N. Mehendale, "Intelligent belt for the blind," *International Journal of Scientific and Engineering Research*, vol. 3, no. 11, pp. 1–3, Nov. 2012.

[22] J. Borenstein, "The navbelt - a computerized multi-sensor travel aid for active guidance of the blind," in *CSUN's Fifth Annual Conference on Technology and Persons with Disabilities*, Mar. 1990, pp. 107–116.

[23] N. G. Bourbakis and D. Kavraki, "An intelligent assistant for navigation of visually impaired people," in *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*. IEEE, 2001, pp. 230–235.

[24] R. Abobeah, M. E. Hussein, M. M. Abdelwahab, and A. A. Shoukry, "Wearable rgb camera-based navigation system for the visually impaired," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018))*, 2018, pp. 555–562.

[25] S. L. Joseph, J. Xiao, X. Zhang, B. Chawda, K. Narang, N. Rajput, S. Mehta, and L. V. Subramaniam, "Being aware of the world: Toward using social media to support the

blind with navigation," *IEEE transactions on human-machine systems*, vol. 45, no. 3, pp. 399–405, 2015.

[26] B. Lin, C. Lee, and P. Chiang, "Simple smartphone-based guiding system for visually impaired people," *Sensors*, vol. 17, no. 6, pp. 1–22, Jun. 2017.

[27] S. Kayukawa, K. Higuchi, J. Guerreiro, S. Morishima, Y. Sato, K. Kitani, and C. Asakawa, "Bbeep: A sonic collision avoidance system for blind travellers and nearby pedestrians," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[28] R. Tapu, B. Mocanu, and T. Zaharia, "Deep-see: Joint object detection, tracking and recognition with application to visually impaired navigational assistance," *Sensors*, vol. 17, no. 11, pp. 1–24, Oct. 2017.

[29] W. Ou, J. Zhang, K. Peng, K. Yang, G. Jaworek, K. Müller, and R. Stiefelhagen, "Indoor navigation assistance for visually impaired people via dynamic slam and panoptic segmentation with an rgb-d sensor," *arXiv preprint arXiv:2204.01154*, 2022.

[30] S. Suman, S. Mishra, K. S. Sahoo, A. Nayyar *et al.*, "Vision navigator: a smart and intelligent obstacle recognition model for visually impaired users," *Mobile Information Systems*, vol. 2022, 2022.

[31] R. C. Joshi, S. Yadav, M. K. Dutta, and C. M. Travieso-Gonzalez, "Efficient multi-object detection and smart navigation using artificial intelligence for visually impaired people," *Entropy*, vol. 22, no. 9, p. 941, 2020.

[32] T. Q. Khoi, N. A. Quang, and N. K. Hieu, "Object detection for drones on raspberry pi potentials and challenges," *IOP Conference Series: Materials Science and Engineering*, vol. 1109, no. 1, pp. 1–13, May 2021.

[33] J. Lee and K.-i. Hwang, "Yolo with adaptive frame control for real-time object detection applications," *Multimedia Tools and Applications*, vol. 81, no. 25, pp. 36 375–36 396, 2022.

[34] Z. Chen, X. Liu, M. Kojima, Q. Huang, and T. Arai, "A wearable navigation device for visually impaired people based on the real-time semantic visual slam system," *Sensors*, vol. 21, no. 4, p. 1536, 2021.

[35] T. B. Shaik and R. Mal, "Algorithm to assist visually impaired person for object detection in real time," in *International Conference on Emerging Electronics and Automation*. Springer, 2022, pp. 147–157.

[36] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[37] A. B. Najjar, A. R. Al-Issa, and M. Hosny, "Dynamic indoor path planning for the visually impaired," *Journal of King Saud University-Computer and Information Sciences*, pp. 1–11, 2022.

[38] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Virtual-blind-road following-based wearable navigation device for blind people," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 1, pp. 136–143, 2018.

[39] A. Yamashita, K. Sato, and K. Matsubayashi, "Walking navigation system for visually impaired people based on high-accuracy positioning using qzss and rfid and obstacle avoidance using hololens," *International Journal of Innovative Computing, Information and Control*, vol. 16, no. 4, pp. 1459–1467, 2020.

[40] T. S. T. Yusof, S. F. Toha, and H. M. Yusof, "Path planning for visually impaired people in an unfamiliar environment using particle swarm optimization," *Procedia Computer Science*, vol. 76, pp. 80–86, 2015.

[41] D. Nandini and K. Seeja, "A novel path planning algorithm for visually impaired people," *Journal of King Saud University-Computer and Information Sciences*, vol. 31, no. 3, pp. 385–391, 2019.

[42] P. T. Mahida, S. Shahrestani, and H. Cheung, "Dynapath: Dynamic learning based indoor navigation for vip in iot based environments," in *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*. IEEE, 2018, pp. 8–13.

[43] K. Cai, C. Wang, J. Cheng, C. W. de Silva, and M. Q. Meng, "Mobile robot path planning in dynamic environments: A survey," *CoRR*, vol. abs/2006.14195, 2020.

[44] W. Nawaz, K. U. Khan, and K. Bashir, "A review on path selection and navigation approaches towards an assisted mobility of visually impaired people," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 8, pp. 3270–3294, 2020.

[45] N. He, Y. Su, X. Fan, Z. Liu, B. Wang *et al.*, "Dynamic path planning of mobile robot based on artificial potential field," in *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*. IEEE, 2020, pp. 259–264.

[46] S. Raychaudhuri, "Introduction to monte carlo simulation," in *2008 Winter Simulation Conference*, 2008, pp. 91–100.

[47] S. R. Flaxman, R. R. Bourne, S. Resnikoff, P. Ackland, T. Braithwaite, M. V. Cicinelli, A. Das, J. B. Jonas, J. Keeffe, J. H. Kempen *et al.*, "Global causes of blindness and distance vision impairment 1990–2020: a systematic review and meta-analysis," *The Lancet Global Health*, vol. 5, no. 12, pp. e1221–e1234, 2017.

[48] A. S. on Developmental, B. Pediatrics, R. G. Voigt, M. M. Macias, S. M. Myers, and C. D. Tapia, *AAP Developmental and Behavioral Pediatrics*. American Academy of Pediatrics. [Online]. Available: https://doi.org/10.1542/9781610021357

[49] F. Rowe, R. Wormald, R. Cable, M. Acton, K. Bonstein, M. Bowen, C. Bronze, C. Bunce, D. Conroy, K. Cowan, K. Evans, M. Fenton, H. Giles, I. Gordon, L. Halfhide, R. Harper, A. Lightstone, M. Votruba, H. Waterman, and A. Zekite, "The sight loss and vision priority setting partnership (slv-psp): overview and results of the research prioritisation survey process," vol. 4, no. 7, 2014.

[50] N. T. Lepcha, C. K. Chettri, K. Getshen, B. B. Rai, S. Bindiganavale Ramaswamy, S. Saibaba, P. K. Nirmalan, E. H. Demarchis, G. Tabin, M. Morley *et al.*, "Rapid assessment of avoidable blindness in bhutan," *Ophthalmic epidemiology*, vol. 20, no. 4, pp. 212–219, 2013.

[51] A. Khan and S. Khusro, "An insight into smartphone-based assistive solutions for visually impaired and blind people: issues, challenges and opportunities," *Universal Access in the Information Society*, vol. 20, pp. 265–298, 2021.

[52] I. J. L. Paul, S. Sasirekha, S. Mohanavalli, C. Jayashree, P. M. Priya, and K. Monika, "Smart eye for visually impaired-an aid to help the blind people," in *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*. IEEE, 2019, pp. 1–5.

[53] T. I. Saskia, "Loneliness among adult with visual impairment," *Sriwijaya Journal of Ophthalmology*, vol. 4, no. 1, pp. 75–78, 2021.

[54] M. C. McDonnall and Z. Sui, "Employment and unemployment rates of people who are blind or visually impaired: Estimates from multiple sources," *Journal of Visual Impairment & Blindness*, vol. 113, no. 6, pp. 481–492, 2019.

[55] S. R. Nyman, M. A. Gosney, and C. R. Victor, "Psychosocial impact of visual impairment in working-age adults," *British Journal of Ophthalmology*, vol. 94, no. 11, pp. 1427–1431, 2010.

[56] F. E.-Z. El-Taher, L. Miralles-Pechuán, J. Courtney, K. Millar, C. Smith, and S. Mckeever, "A survey on outdoor navigation applications for people with visual impairments," *IEEE Access*, vol. 11, pp. 14 647–14 666, 2023.

[57] R. D. Jacobson, "Cognitive mapping without sight: Four preliminary studies of spatial learning," *Journal of Environmental Psychology*, vol. 18, no. 3, pp. 289–305, 1998.

[58] K. Papadopoulos, E. Koustriava, and M. Barouti, "Cognitive maps of individuals with blindness for familiar and unfamiliar spaces: Construction through audio-tactile maps and walked experience," *Computers in Human Behavior*, vol. 75, pp. 376–384, 2017.

[59] K. Papadopoulos, M. Barouti, and E. Koustriava, "Differences in spatial knowledge of individuals with blindness when using audiotactile maps, using tactile maps, and walking," *Exceptional Children*, vol. 84, no. 3, pp. 330–343, 2018.

[60] R. Kitchin and R. Jacobson, "Techniques to collect and analyze the cognitive map knowledge of persons with visual impairment or blindness: Issues of validity," *Journal of Visual Impairment & Blindness*, vol. 91, no. 4, pp. 360–376, 1997.

[61] J. Juurmaa, "Transposition in mental spatial manipulation: A theoretical analysis." *American Foundation for the Blind, Research Bulletin*, 1973.

[62] R. Passini and G. Proulx, "Wayfinding without vision: An experiment with congenitally totally blind people," *Environment and behavior*, vol. 20, no. 2, pp. 227–252, 1988.

[63] D. Picard and R. Pry, "Does knowledge of spatial configuration in adults with visual impairments improve with tactile exposure to a small-scale model of their urban environment?" *Journal of Visual Impairment & Blindness*, vol. 103, no. 4, pp. 199–209, 2009.

[64] S. Millar, "Models of sensory deprivation: The nature/nurture dichotomy and spatial representation in the blind," *International Journal of Behavioral Development*, vol. 11, no. 1, pp. 69–87, 1988.

[65] K. S. Papadopoulos, "A school programme contributes to the environmental knowledge of blind people," *British Journal of Visual Impairment*, vol. 22, no. 3, pp. 101–104, 2004.

[66] A. Koutsoklenis and K. Papadopoulos, "Auditory cues used for wayfinding in urban environments by individuals with visual impairments," *Journal of Visual Impairment & Blindness*, vol. 105, no. 10, pp. 703–714, 2011.

[67] ——, "Olfactory cues used for wayfinding in urban environments by individuals with visual impairments," *Journal of Visual Impairment & Blindness*, vol. 105, no. 10, pp. 692–702, 2011.

[68] ——, "Haptic cues used for outdoor wayfinding by individuals with visual impairments," *Journal of Visual Impairment & Blindness*, vol. 108, no. 1, pp. 43–53, 2014.

[69] K. Papadopoulos, K. Papadimitriou, and A. Koutsoklenis, "The role of auditory cues in the spatial knowledge of blind individuals." *International journal of special education*, vol. 27, no. 2, pp. 169–180, 2012.

[70] R. F. Wang, "Spatial representations and spatial updating," in *Psychology of Learning and Motivation*. Elsevier, 2003, vol. 42, pp. 109–156.

[71] M. Nardini, N. Burgess, K. Breckenridge, and J. Atkinson, "Differential developmental trajectories for egocentric, environmental and intrinsic frames of reference in spatial memory," *Cognition*, vol. 101, no. 1, pp. 153–172, 2006.

[72] K. Morsley, C. Spencer, and K. Baybutt, "Is there any relationship between a child's body image and spatial skills?" *British Journal of Visual Impairment*, vol. 9, no. 2, pp. 41–43, 1991.

[73] D. H. Warren, *Blindness and children: An individual differences approach*. Cambridge University Press, 1994.

[74] Z. J. Muhsin, R. Qahwaji, F. Ghanchi, and M. Al-Taee, "Review of substitutive assistive tools and technologies for people with visual impairments: recent advancements and prospects," *Journal on Multimodal User Interfaces*, vol. 18, no. 1, pp. 135–156, 2024.

[75] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[76] C. S. Group. (2023, April) Impact of artificial intelligence and machine learning on software development. Accessed on April 8, 2023. [Online]. Available: https://www.cdnsol.com/blog/impact-of-artificial-intelligence-and-machine-learning-on-software-development/

[77] IBM. (2023, April) What is a neural network? Accessed on May 11, 2023. [Online]. Available: https://www.ibm.com/topics/neural-networks

[78] O. Sporns, *Networks of the Brain*. MIT press, 2016.

[79] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[80] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

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

[82] Q. Wu, C. Shen, P. Wang, A. Dick, and A. Van Den Hengel, "Image captioning and visual question answering based on attributes and external knowledge," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1367–1381, 2017.

[83] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*. Springer, 2014, pp. 297–312.

[84] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[85] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," in *Journal of Physics: Conference Series*, vol. 1544, no. 1. IOP Publishing, 2020, p. 012033.

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

[87] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[88] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[89] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "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.

[90] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[91] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[92] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, pp. 1–6, Apr. 2018.

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

[94] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, K. Michael, J. Fang, Z. Yifu, C. Wong, D. Montes *et al.*, "ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation," *Zenodo*, 2022.

[95] G. Jocher, A. Chaurasia, and J. Qiu. (2023) Yolo by ultralytics. [Online]. Available: https://github.com/ultralytics/ultralytics

[96] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[97] W. Liu, I. Hasan, and S. Liao, "Center and scale prediction: Anchor-free approach for pedestrian and face detection," *arXiv preprint arXiv:1904.02948*, 2019.

[98] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, pp. 13–es, 2006.

[99] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2019, pp. 1328–1338.

[100] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1913–1921.

[101] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV 12*. Springer, 2012, pp. 215–230.

[102] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004.

[103] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[104] L. Fan, Z. Wang, B. Cail, C. Tao, Z. Zhang, Y. Wang, S. Li, F. Huang, S. Fu, and F. Zhang, "A survey on multiple object tracking algorithm," in *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2016, pp. 1855–1862.

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

[106] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng, "Strongsort: Make deepsort great again," *IEEE Transactions on Multimedia*, 2023.

[107] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[108] C. Kim, F. Li, and J. M. Rehg, "Multi-object tracking with neural gating using bilinear lstm," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 200–215.

[109] C. Kim, L. Fuxin, M. Alotaibi, and J. M. Rehg, "Discriminative appearance modeling with multi-track pooling for real-time multi-object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9553–9562.

[110] S. Guo, S. Wang, Z. Yang, L. Wang, H. Zhang, P. Guo, Y. Gao, and J. Guo, "A review of deep learning-based visual multi-object tracking algorithms for autonomous driving," *Applied Sciences*, vol. 12, no. 21, p. 10741, 2022.

[111] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

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

[113] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, "Towards real-time monocular depth estimation for robotics: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 940–16 961, 2022.

[114] A. Zaarane, I. Slimani, W. Al Okaishi, I. Atouf, and A. Hamdoun, "Distance measurement system for autonomous vehicles using stereo camera," *Array*, vol. 5, p. 100016, 2020.

[115] A. Prusak, O. Melnychuk, H. Roth, I. Schiller, and R. Koch, "Pose estimation and map building with a time-of-flight-camera for robot navigation," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3-4, pp. 355–364, 2008.

[116] F. Utaminingrum, T. A. Kurniawan, M. A. Fauzi, R. Maulana, D. Syauqy, R. C. Wihandika, Y. A. Sari, and P. P. Adikara, "A laser-vision based obstacle detection and distance estimation for smart wheelchair navigation," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2016, pp. 123–127.

[117] S. J. Kim and B. K. Kim, "Dynamic ultrasonic hybrid localization system for indoor mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4562–4573, 2012.

[118] P. Coronel, S. Furrer, W. Schott, and B. Weiss, "Indoor location tracking using inertial navigation sensors and radio beacons," in *The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland, March 26-28, 2008. Proceedings*. Springer, 2008, pp. 325–340.

[119] G. Benet, F. Blanes, J. E. Simó, and P. Pérez, "Using infrared sensors for distance measurement in mobile robots," *Robotics and autonomous systems*, vol. 40, no. 4, pp. 255–266, 2002.

[120] D. Forouher, M. G. Besselmann, and E. Maehle, "Sensor fusion of depth camera and ultrasound data for obstacle detection and robot navigation," in *2016 14th international conference on control, automation, robotics and vision (ICARCV)*. IEEE, 2016, pp. 1–6.

[121] N. Aswini and S. Uma, "Obstacle avoidance and distance measurement for unmanned aerial vehicles using monocular vision," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 5, p. 3504, 2019.

[122] A. Zaarane, I. Slimani, W. Al Okaishi, I. Atouf, and A. Hamdoun, "Distance measurement system for autonomous vehicles using stereo camera," *Array*, vol. 5, p. 100016, 2020.

[123] W. Hassen and H. Amiri, "Block matching algorithms for motion estimation," in *2013 7th IEEE International Conference on e-Learning in Industrial Electronics (ICELIE)*, 2013, pp. 136–139.

[124] A. Kadambi, A. Bhandari, and R. Raskar, "3d depth cameras in vision: Benefits and limitations of the hardware: With an emphasis on the first-and second-generation kinect models," *Computer vision and machine learning with RGB-D sensors*, pp. 3–26, 2014.

[125] L. Wang and R. Yang, "Global stereo matching leveraged by sparse ground control points," in *CVPR 2011*. IEEE, 2011, pp. 3033–3040.

[126] M. P. Muresan, M. Negru, and S. Nedevschi, "Improving local stereo algorithms using binary shifted windows, fusion and smoothness constraint," in *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2015, pp. 179–185.

[127] R. Spangenberg, T. Langner, S. Adfeldt, and R. Rojas, "Large scale semi-global matching on the cpu," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 195–201.

[128] A. Wedel, U. Franke, J. Klappstein, T. Brox, and D. Cremers, "Realtime depth estimation and obstacle detection from monocular video," in *Joint Pattern Recognition Symposium*. Springer, 2006, pp. 475–484.

[129] H. Javidnia and P. Corcoran, "Accurate depth map estimation from small motions," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2453–2461.

[130] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2144–2158, 2014.

[131] S. H. Raza, O. Javed, A. Das, H. Sawhney, H. Cheng, and I. Essa, "Depth extraction from videos using geometric context and occlusion boundaries," *arXiv preprint arXiv:1510.07317*, 2015.

[132] X. Chen, X. Chen, and Z.-J. Zha, "Structure-aware residual pyramid network for monocular depth estimation," *arXiv preprint arXiv:1907.06023*, 2019.

[133] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2019, pp. 1043–1051.

[134] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.

[135] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[136] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, 2022.

[137] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 5848–5854.

[138] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[139] A. Atapour-Abarghouei and T. P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2800–2810.

[140] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 824–840, 2008.

[141] F. E.-Z. El-Taher, A. Taha, J. Courtney, and S. Mckeever, "A systematic review of urban navigation systems for visually impaired people," *Sensors*, vol. 21, no. 9, p. 3103, 2021.

[142] X. Wang, G. Zhang, J. Zhao, H. Rong, F. Ipate, and R. Lefticaru, "A modified membrane-inspired algorithm based on particle swarm optimization for mobile robot path planning," 2015.

[143] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, 2022, pp. 287–290.

[144] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[145] P. Martin and A. Del Pobil, "Application of artificial neural networks to the robot path planning problem," *WIT Transactions on Information and Communication Technologies*, vol. 6, 1970.

[146] H. John, "Holland. adaptation in natural and artificial systems," 1975.

[147] M. Dorigo and T. Stutzle, "Ant colony optimization. mit press, cambridge, ma," 2004.

[148] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. ieee, 1995, pp. 1942–1948.

[149] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Simulated annealing," *Journal of Chemical Physics*, vol. 21, no. 161-162, pp. 1087–1092, 1953.

[150] P. Li, X. Huang, and M. Wang, "A novel hybrid method for mobile robot path planning in unknown dynamic environment based on hybrid dsm model grid map," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 23, no. 1, pp. 5–22, 2011.

[151] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[152] C. Qixin, H. Yanwen, and Z. Jingliang, "An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3331–3336.

[153] L. Zhou and W. Li, "Adaptive artificial potential field approach for obstacle avoidance path planning," in *2014 Seventh International Symposium on Computational Intelligence and Design*, vol. 2. IEEE, 2014, pp. 429–432.

[154] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[155] A. M. Whitbrook, U. Aickelin, and J. M. Garibaldi, "Idiotypic immune networks in mobile-robot control," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 6, pp. 1581–1598, 2007.

[156] M. Marefat and J. Britanik, "Case-based process planning using an object-oriented model representation," *Robotics and Computer-Integrated Manufacturing*, vol. 13, no. 3, pp. 229–251, 1997.

[157] X. Zhong, X. Peng, and M. Miao, "Planning of robot paths through environment modelling and adaptive window," *J. Huazhong Univ. Sci. Technol*, vol. 38, pp. 107–111, 2010.

[158] A. Antoniou and W.-S. Lu, *Practical optimization*. Springer, 2007.

[159] P. Pan, "A fast simplex algorithm for linear programming," *Journal of Computational Mathematics*, pp. 837–847, 2010.

[160] R. Anand, D. Aggarwal, and V. Kumar, "A comparative analysis of optimization solvers," *Journal of Statistics and Management Systems*, vol. 20, no. 4, pp. 623–635, 2017.

[161] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[162] E. W. Cheney and A. A. Goldstein, "Newton's method for convex programming and tchebycheff approximation," *Numerische Mathematik*, vol. 1, no. 1, pp. 253–268, 1959.

[163] E. D. Anderson, J. Gondzio, C. Mészáros, and X. Xu, "Implementation of interior-point methods for large scale linear programs," *Interior Point Methods of Mathematical Programming*, pp. 189–252, 1996.

[164] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," *Advances in neural information processing systems*, vol. 27, 2014.

[165] J. Iqbal, M. Ullah, S. G. Khan, B. Khelifa, and S. Ćuković, "Nonlinear control systems-a brief overview of historical and recent advances," *Nonlinear Engineering*, vol. 6, no. 4, pp. 301–312, 2017.

[166] M. G. Plessen, P. F. Lima, J. Mårtensson, A. Bemporad, and B. Wahlberg, "Trajectory planning under vehicle dimension constraints using sequential linear programming," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[167] F. Palacios-Gomez, L. Lasdon, and M. Engquist, "Nonlinear optimization by successive linear programming," *Management science*, vol. 28, no. 10, pp. 1106–1120, 1982.

[168] P. Lu, "Entry guidance and trajectory control for reusable launch vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 143–149, 1997.

[169] R. Palumbo, G. Morani, and M. Cicala, "Reentry trajectory optimization for mission analysis," *Journal of Spacecraft and Rockets*, vol. 54, no. 1, pp. 331–336, 2017.

[170] J. Schierman and J. Hull, "In-flight entry trajectory optimization for reusable launch vehicles," in *AIAA guidance, navigation, and control conference and exhibit*, 2005, p. 6434.

[171] Z. Wang and Y. Lu, "Improved sequential convex programming algorithms for entry trajectory optimization," *Journal of Spacecraft and Rockets*, vol. 57, no. 6, pp. 1373–1386, 2020.

[172] C. Liu, C. Zhang, H. Jiang, and F. Xiong, "Online trajectory optimization based on successive convex optimization," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 2577–2582.

[173] M. Szmuk and B. Acikmese, "Successive convexification for 6-dof mars rocket powered landing with free-final-time," in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 0617.

[174] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1725–1740, 2014.

[175] S.-B. Cho and H.-L. Choi, "A convex programming-based approach to trajectory optimization for survivability enhancement of homing missiles," *International Journal of Aeronautical and Space Sciences*, vol. 23, no. 5, pp. 992–1008, 2022.

[176] R. Velázquez, "Wearable assistive devices for the blind," *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment: Issues and Characterization*, pp. 331–349, 2010.

[177] M. M. Islam, M. S. Sadi, K. Z. Zamli, and M. M. Ahmed, "Developing walking assistants for visually impaired people: A review," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 2814–2828, 2019.

[178] T. Ifukube, T. Sasaki, and C. Peng, "A blind mobility aid modeled after echolocation of bats," *IEEE Transactions on biomedical engineering*, vol. 38, no. 5, pp. 461–465, 1991.

[179] I. Ulrich and J. Borenstein, "The guidecane-applying mobile robot technologies to assist the visually impaired," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 2, pp. 131–136, 2001.

[180] M. H. A. Wahab, A. A. Talib, H. A. Kadir, A. Johari, A. Noraziah, R. M. Sidek, and A. A. Mutalib, "Smart cane: Assistive cane for visually-impaired people," *arXiv preprint arXiv:1110.5156*, 2011.

[181] W. Penrod, M. D. Corbett, and B. Blasch, "Practice report: A master trainer class for professionals in teaching the ultracane electronic travel device," *Journal of Visual Impairment & Blindness*, vol. 99, no. 11, pp. 711–715, 2005.

[182] B. Hoyle and D. Waters, "Mobility at: The batcane (ultracane)," *Assistive technology for visually impaired and blind people*, pp. 209–229, 2008.

[183] L. H. Villamizar, M. Gualdron, F. González, J. Aceros, and C. V. Rizzo-Sierra, "A necklace sonar with adjustable scope range for assisting the visually impaired," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2013, pp. 1450–1453.

[184] W.-J. Chang, L.-B. Chen, M.-C. Chen, J.-P. Su, C.-Y. Sie, and C.-H. Yang, "Design and implementation of an intelligent assistive system for visually impaired people for aerial obstacle avoidance and fall detection," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 199–10 210, 2020.

[185] R. Farcy, R. Leroux, A. Jucha, R. Damaschini, C. Grégoire, and A. Zogaghi, "Electronic travel aids and electronic orientation aids for blind people: Technical, rehabilitation and everyday life points of view," in *Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Technology for Inclusion*, vol. 12. Citeseer, 2006, p. 12.

[186] A. Dernayka, M.-A. Amorim, R. Leroux, L. Bogaert, and R. Farcy, "Tom pouce iii, an electronic white cane for blind people: Ability to detect obstacles and mobility performances," *Sensors*, vol. 21, no. 20, p. 6854, 2021.

[187] P. Kwiatkowski, T. Jaeschke, D. Starke, L. Piotrowsky, H. Deis, and N. Pohl, "A concept study for a radar-based navigation device with sector scan antenna for visually impaired people," in *2017 First IEEE MTT-S International Microwave Bio Conference (IMBIOC)*. IEEE, 2017, pp. 1–4.

[188] V. Kulyukin, C. Gharpure, and J. Nicholson, "Robocart: Toward robot-assisted navigation of grocery stores by the visually impaired," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2845–2850.

[189] D. López-de Ipiña, T. Lorido, and U. López, "Indoor navigation and product recognition for blind people assisted shopping," in *Ambient Assisted Living: Third International Workshop, IWAAL 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings 3*. Springer, 2011, pp. 33–40.

[190] A. Meliones and D. Sampson, "Blind museumtourer: A system for self-guided tours in museums and blind indoor navigation," *Technologies*, vol. 6, no. 1, p. 4, 2018.

[191] K. Yelamarthi, D. Haas, D. Nielsen, and S. Mothersell, "Rfid and gps integrated navigation system for the visually impaired," in *2010 53rd IEEE International Midwest Symposium on Circuits and Systems*. IEEE, 2010, pp. 1149–1152.

[192] R. Tapu, B. Mocanu, and T. Zaharia, "Wearable assistive devices for visually impaired: A state of the art survey," *Pattern Recognition Letters*, vol. 137, pp. 37–52, 2020.

[193] P. Chanana, R. Paul, M. Balakrishnan, and P. Rao, "Assistive technology solutions for aiding travel of pedestrians with visual impairment," *Journal of rehabilitation and assistive technologies engineering*, vol. 4, p. 2055668317725993, 2017.

[194] J. Villanueva and R. Farcy, "Optical device indicating a safe free path to blind people," *IEEE transactions on instrumentation and measurement*, vol. 61, no. 1, pp. 170–177, 2012.

[195] M. H. A. Wahab, A. A. Talib, H. A. Kadir, A. Johari, A. Noraziah, R. M. Sidek, and A. A. Mutalib, "Smart cane: Assistive cane for visually-impaired people," *arXiv preprint arXiv:1110.5156*, 2011.

[196] A. Burlacu, S. Bostaca, I. Hector, P. Herghelegiu, G. Ivanica, A. Moldoveanul, and S. Caraiman, "Obstacle detection in stereo sequences using multiple representations of the disparity map," in *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2016, pp. 854–859.

[197] M. Kang, S. Chae, J. Sun, J. Yoo, and S. Ko, "A novel obstacle detection method based on deformable grid for the visually impaired," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 3, pp. 376–383, Aug. 2015.

[198] M. Kang, S. Chae, J. Sun, S. Lee, and S. Ko, "An enhanced obstacle avoidance method for the visually impaired using deformable grid," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 2, pp. 169–177, May 2017.

[199] A. Aladrén, G. López-Nicolás, L. Puig, and J. J. Guerrero, "Navigation assistance for the visually impaired using rgb-d sensor with range expansion," *IEEE Systems Journal*, vol. 10, no. 3, pp. 922–932, Sept. 2016.

[200] S. Lin, K. Wang, K. Yang, and R. Cheng, "Krnet: A kinetic real-time convolutional neural network for navigational assistance," in *International Conference on Computers Helping People with Special Needs*, Jun. 2018, pp. 55–62.

[201] N. Parikh, I. Shah, and S. Vahora, "Android smartphone based visual object recognition for visually impaired using deep learning," in *2018 International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2018, pp. 0420–0425.

[202] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia, "A smartphone-based obstacle detection and classification system for assisting visually impaired people," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Jun. 2013.

[203] A. N. Zereen and S. Corraya, "Detecting real time object along with the moving direction for visually impaired people," in *2016 2nd International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)*, Mar. 2016, pp. 1–4.

[204] S. Vaidya, N. Shah, N. Shah, and R. Shankarmani, "Real-time object detection for visually challenged people," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, May 2020, pp. 311–316.

[205] S. Shadi, S. Hadi, M. A. Nazari, and W. Hardt, "Outdoor navigation for visually impaired based on deep learning," in *Proc. CEUR Workshop Proc*, vol. 2514, 2019, pp. 97–406.

[206] S. Kumar, D. N. Mishra, S. M. Ganie, R. Bharathikannan, and K. Vijayakanthan, "Artificial intelligence solutions for the visually impaired: A review," *Handbook of Research on AI and Knowledge Engineering for Real-Time Business Intelligence*, pp. 198–207, 2023.

[207] L. Rana, A. U. Rehman, S. Javaid, and T. M. Ali, "A novel model-driven approach for visual impaired people assistance optic ally," in *2022 Third International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT)*. IEEE, 2022, pp. 1–8.

[208] A. Saxena, J. Schulte, A. Y. Ng *et al.*, "Depth estimation using monocular and stereo cues." in *IJCAI*, vol. 7, 2007, pp. 2197–2203.

[209] S. Duman, A. Elewi, and Z. Yetgin, "Design and implementation of an embedded real-time system for guiding visually impaired individuals," in *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*. IEEE, 2019, pp. 1–5.

[210] T. Turker, O. K. Sahingoz, and G. Yilmaz, "2d path planning for uavs in radar threatening environment using simulated annealing algorithm," in *2015 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2015, pp. 56–61.

[211] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[212] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1928–1935.

[213] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 5917–5922.

[214] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3475–3482.

[215] S. J. Guy, J. Chhugani, C. Kim, N. Satish, M. Lin, D. Manocha, and P. Dubey, "Clearpath: Highly parallel collision avoidance for multi-agent simulation," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Association for Computing Machinery, 2009, pp. 177–187.

[216] C. Qixin, H. Yanwen, and Z. Jingliang, "An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 3331–3336.

[217] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, "The hybrid path planning algorithm based on improved a* and artificial potential field for unmanned surface vehicle formations," *Ocean Engineering*, vol. 223, p. 108709, 2021.

[218] I. Nishitani, T. Matsumura, M. Ozawa, A. Yorozu, and M. Takahashi, "Human-centered x–y–t space path planning for mobile robot in dynamic environments," *Robotics and Autonomous Systems*, vol. 66, pp. 18–26, 2015.

[219] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[220] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[221] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[222] G. Jocher, A. Chaurasia, and J. Qiu. (2023) Yolo by ultralytics. [Online]. Available: https://github.com/ultralytics/ultralytics

[223] X. Xiao and X. Feng, "Multi-object pedestrian tracking using improved yolov8 and ocsort," *Sensors*, vol. 23, no. 20, p. 8439, 2023.

[224] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, pp. 1–5, Mar. 2017.

[225] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571.

[226] A. Muhammad and V. Zalizniak, *Practical Scientific Computing*. Sawston, United Kingdom: Woodhead Publishing, Feb. 2011.

[227] C. Wöhler, *3D computer vision: efficient methods and applications*. Springer Science & Business Media, 2012.

[228] E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, New Jersey 07458, United States: Prentice Hall, 1998.

[229] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. University Printing House, Cambridge CB2 8BS, United Kingdom: Cambridge University Press, Dec. 2004.

[230] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998.

[231] D. D. Clark-Carter, A. D. Heyes, and C. I. Howarth, "The efficiency and walking speed of visually impaired people," *Ergonomics*, pp. 779–789, Jun. 1985.

[232] D. Robineau, P. Baden, A. Dhani, M. Dark, A. Bhagat, and H. Mann, "Reported road casualties great britain: 2017," Department for Transport, London, United Kingdom, Tech. Rep., Sep. 2018.

[233] D. C. Richards, "Relationship between speed and risk of fatal injury: pedestrians and car occupants," Department for transport, London, United Kingdom, Tech. Rep., Sep. 2010.

[234] F. Walz, M. Hoefliger, and W. Fehlmann, "Speed limit reduction from 60 to 50 km/h and pedestrian injuries," Institute of Forensic Medicine University of Zurich, Winterthurerstrasse, Zürich, Switzerland, Tech. Rep., 1983.

[235] P. F. Niederer, M. Schlumpf, F. Mesqui, and P.-A. Hartmann, "The reliability of anthropometric test devices, cadavers, and mathematical models as pedestrian surrogates," *SAE Transactions*, pp. 775–784, 1983.

[236] M. C. Rebollo-Soria, C. Arregui-Dalmases, D. Sánchez-Molina, J. Velázquez-Ameijide, and I. Galtés, "Injury pattern in lethal motorbikes-pedestrian collisions, in the area of barcelona, spain," *Journal of Forensic and Legal Medicine*, vol. 43, pp. 80–84, Oct. 2016.

[237] A. Short, R. Grzebieta, and N. Arndt, "Estimating bicyclist into pedestrian collision speed," *International Journal of Crashworthiness*, vol. 12, no. 2, pp. 127–135, Aug. 2007.

[238] S. Chandra and A. K. Bharti, "Speed distribution curves for pedestrians during walking and crossing," *Procedia - Social and Behavioral Sciences*, vol. 104, pp. 660–667, Dec. 2013.

[239] C. Freer, "Vehicle speed compliance statistics, great britain: 2018," Department of Transport, London, United Kingdom, Tech. Rep., Jun. 2019.

[240] Z. Cattaneo and T. Vecchi, *Blind vision: the neuroscience of visual impairment*. MIT press, 2011.

[241] P. Voss, F. Gougoux, R. J. Zatorre, M. Lassonde, and F. Lepore, "Differential occipital responses in early-and late-blind individuals during a sound-source discrimination task," *Neuroimage*, vol. 40, no. 2, pp. 746–758, 2008.

[242] S. E. Hassan, "Are normally sighted, visually impaired, and blind pedestrians accurate and reliable at making street crossing decisions?" *Investigative ophthalmology & visual science*, vol. 53, no. 6, pp. 2593–2600, 2012.

[243] S. Kayukawa, K. Higuchi, J. Guerreiro, S. Morishima, Y. Sato, K. Kitani, and C. Asakawa, "Bbeep: A sonic collision avoidance system for blind travellers and nearby pedestrians," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[244] B. Nie, Q. Li, S. Gan, B. Xing, Y. Huang, and S. E. Li, "Safety envelope of pedestrians upon motor vehicle conflicts identified via active avoidance behaviour," *Scientific Reports*, vol. 11, no. 1, pp. 1–10, Feb. 2021.

[245] E. S. da Silva, G. Fischer, R. G. da Rosa, P. Schons, L. B. T. Teixeira, W. Hoogkamer, and L. A. Peyré-Tartaruga, "Gait and functionality of individuals with visual impairment who participate in sports," *Gait & Posture*, vol. 62, pp. 355–358, 2018.

[246] A. Antoniou and W.-S. Lu, *Practical optimization*, 1st ed. New York, NY: Springer, 2007.

[247] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, Apr. 2008.

[248] F. Janabi-Sharifi and D. Vinke, "Integration of the artificial potential field approach with simulated annealing for robot path planning," in *Proceedings of 8th IEEE International Symposium on Intelligent Control*, 1993, pp. 536–541.

[249] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.

[250] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.

[251] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, 2020.

[252] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, pp. 1–12, May 2016.

[253] Y. Sun, Y. Zhang, H. Wang, J. Guo, J. Zheng, and H. Ning, "Ses-yolov8n: automatic driving object detection algorithm based on improved yolov8," *Signal, Image and Video Processing*, vol. 18, no. 5, pp. 3983–3992, 2024.

[254] Y. Zhao, C. Yan, and Q. Wang, "Cpu tracking algorithm for lightweight vehicles based on deepsort," in *2022 18th International Conference on Computational Intelligence and Security (CIS)*. IEEE, 2022, pp. 102–106.

[255] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam,*

The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, 2016, pp. 749–765.

[256] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4353–4361.

[257] I. Song, J. Chung, T. Kim, and Y. Bengio, "Dynamic frame skipping for fast speech recognition in recurrent neural network based acoustic models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4984–4988.

[258] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "Uav-yolov8: a small-object-detection model based on improved yolov8 for uav aerial photography scenarios," *Sensors*, vol. 23, no. 16, p. 7190, 2023.

[259] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: A survey," *Computer Science Review*, vol. 28, pp. 157–177, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013716301794

[260] H. Zhang, J. Zhang, Q. Wu, X. Qian, T. Zhou, and H. Fu, "Extended kernel correlation filter for abrupt motion tracking," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 9, p. 4438 – 4460, 2017.

[261] J. Kuen, K. M. Lim, and C. P. Lee, "Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle," *Pattern Recognition*, vol. 48, no. 10, pp. 2964–2982, 2015.

[262] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, pp. 1–9, Apr. 2017.

[263] S. Meers and K. Ward, "A substitute vision system for providing 3d perception and gps navigation via electro-tactile stimulation," in *International Conference on Sensing Technology*, Dec. 2005.

[264] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[265] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "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.

[266] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, "Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 3025–3029.

[267] W. Hu, Q. Wang, L. Zhang, L. Bertinetto, and P. H. Torr, "Siammask: A framework for fast online object tracking and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3072–3089, 2023.

[268] M. Franke, V. Gopinath, C. Reddy, D. Ristić-Durrant, and K. Michels, "Bounding box dataset augmentation for long-range object distance estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1669–1677.

[269] G. A. Kumar, J. H. Lee, J. Hwang, J. Park, S. H. Youn, and S. Kwon, "Lidar and camera fusion approach for object distance estimation in self-driving vehicles," *Symmetry*, vol. 12, no. 2, p. 324, 2020.

[270] M. Tao, X. Li, R. Xie, and K. Ding, "Pedestrian identification and tracking within adaptive collaboration edge computing," in *2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2023, pp. 1124–1129.

.1 Notations

Table 1: Notations Used Throughout the Thesis.

Symbol	Description
f_p, f_c	Previous and current video frames, respectively.
t	Time (in general).
t_p, t_c, t_f	The current, past, and future times, respectively.
M_i	Feature extracted from a captured video frame.
$M_i^{(f_p,t_p)}$, $M_i^{(f_c,t_c)}$	Previous and current 2D-pixel position of feature M_i at f_p and f_c frames, respectively.
$D_i^{M_p,M_c}$	The displacement between the $M_i^{(f_p,t_p)}$ and $M_i^{(f_c,t_c)}$ of the best matched M_i in pixels.
O_i	Target object that is detected and tracked at a captured video frame.
BB_i	The bounding box of an object O_i.
$(x_1, y_1, x_2, y_2)_i$	The minimum and maximum 2D-pixel coordinates of the bounding box of an object O_i.
$P_i^{(f_p,t_p)}$, $P_i^{(f_c,t_c)}$	Previous and current 2D-pixel position (central position of the bounding box) of an object O_i at f_p and f_c frames, respectively, and they are defined as $(x_O, y_O)_i^{(f_p,t_p)}$ and $(x_O, y_O)_i^{(f_c,t_c)}$.
ϵ	Small, positive value for computational error tolerance.
BB_{h_i}, BB_{w_i}, BB_{d_i}	The height, width, and diagonal of an object's bounding box (BB_i) in pixels, respectively.
I_h, I_w, I_d	The height, width, and diagonal of a captured image in pixels, respectively.
R_{h_i}, R_{w_i}, R_{d_i}	The ratios of the object's bounding box dimensions (BB_{h_i}, BB_{w_i}, and BB_{d_i}), to the captured image dimensions (I_h, I_w, and I_d) in pixels, respectively.

Continued on next page

Table 1 – continued from previous page

Symbol	Description
C_{h_l}, C_{w_l}, C_{d_l}	Predefined values of average dimensions, *i.e.,* height, width, and depth, of an object of a particular class l in cm.
V_{f_i}	The feature vector of an object O_i.
Z_i^t	The estimated distance from the camera (the VIP) to an detected object O_i at time t in m.
$(x_{O_i}^t, y_{O_i}^t)$	The 2D-pixel position of an object in the Pixel Coordinate System (PCS) at time t.
$({}^C X_{O_i}^t, {}^C Y_{O_i}^t, {}^C Z_{O_i}^t)$	The 3D-metric position of an object in the Camera Coordinate System (CCS) at time t; note, t can be denoted as t_p or t_c to refer to previous or current positions, respectively.
$(X_{O_i}^t, Y_{O_i}^t, Z_{O_i}^t)$	The 3D position of an object in the World Coordinate System (WCS) at time t. Note, t can be denoted as t_p, t_c, t_f to refer to previous, current, or future positions, respectively.
$({}^C X_B^t, {}^C Y_B^t, {}^C Z_B^t)$	The 3D-metric position of the VIP in the CCS at time t, which is the camera origin (0,0,0).
(X_B^t, Y_B^t, Z_B^t)	The 3D position of the VIP in the WCS at time t.
(p_x, p_y)	The principal point of the camera in pixels.
(f_x, f_y)	The focal length of the camera in pixels.
C_B, C'_B	The circular collision zones of the VIP in the Camera Coordinate System (CCS) and the World Coordinate System (WCS), respectively.
r_B, r'_B	The radius of the VIP's C_B and C'_B, respectively.
C_{O_i}	The object's circular collision zone in the CCS and WCS.
r_{O_i}	The radius of the object's C_{O_i}.
B^{t_p}, B^{t_c}, B^{t_f}	The 3D previous, current and *optimal* next positions of the VIP, respectively.
B_f	The final known destination of the VIP.

Continued on next page

Table 1 – continued from previous page

Symbol	Description
$O_i^{t_p}$, $O_i^{t_c}$, $O_i^{t_f}$	The 3D previous, current and next positions of an object, respectively.
v_B	The velocity of the VIP.
$v_B^{t_c}$, $v_B^{t_f}$	The current and predicted next velocities of the VIP.
v_{O_i}	The velocity of an object.
$v_{O_i}^{t_c}$, $v_{O_i}^{t_f}$	The current and predicted next velocities of an object.
$\overleftrightarrow{B_1O_1}$, $\overleftrightarrow{B_2O_2}$	The common tangents of circles C_B and C_{O_i} in the CCS.
A	The intersected point of $\overleftrightarrow{B_1O_1}$ and $\overleftrightarrow{B_2O_2}$ in the CCS.
$\overline{AB^{t_p}}$, $\overline{AB_1}$	The sides of the triangle $\triangle AB^{t_p}B_1$ connecting point A to points B^{t_p} and B_1, respectively, see Fig. 4.7.
$\overline{AO_i^{t_p}}$	The side of the triangle $\triangle AO_i^{t_p}O_1$ opposite angle $\angle$A, connecting point A to point $O_i^{t_p}$, see Fig. 4.7.
α_i	The angle between $\overline{AB^{t_p}}$ and $\overline{AB_1}$ of $\triangle AB^{t_p}B_1$ in the CCS.
d_i / $\lvert\overline{O_i^{t_p}B^{t_p}}\rvert$	The Euclidean distance between the VIP, B^{t_p}, and the object, $O_i^{t_p}$, at t_p in the CCS.
ϕ_i	The angle of the displacement vector from $O_i^{t_p}$ to B^{t_p} ($O_i^{t_p}\vec{B}^{t_p}$) in the CCS.
β_i	The angle of the the displacement vector from $O_i^{t_p}$ to $O_i^{t_c}$ ($O_i^{t_p}\vec{O}_i^{t_c}$) in the CCS.
v'_{O_i}	The CMO's estimated speed in the CCS.
D_{C_x}, D_{C_y}	The camera displacement (in m) at the X-axis and Y-axis, respectively.
C_z/v'_{VIP}	The assumed camera's velocity in the Z-axis (in m/s).
$v'_{avg_{O_i}}$	The CMO's estimated average speed in the CCS.
t'_{cl_i} / $t_{cl_i}^{t_c}$	The estimated collision time between the VIP and CMO at t_c.
t_{dl}	The warning emission deadline of a CMO.
t_{ra}	An expected VIP's reaction time to avoid a collision.
δ	A deadline safety margin factor.

Continued on next page

Table 1 – continued from previous page

Symbol	Description
IOU_{cmo}	A metric to estimate the overlap between CMOs' bounding boxes.
η	A predefined threshold to indicate the overlap between CMOs' bounding boxes.
Θ_i	CMO's incoming direction.
S_L	A predefined time threshold in s to activate *MinD's* path planning algorithm.
d_{trv}	The maximum VIP travelling distance to avoid CMO.
t_{trv}	The maximum travelling time the VIP needs to arrive B^{t_f} safely.
D_{max}, T_{max}	Predetermined thresholds for the maximum allowable travel distance (in m) and time (in s) for the VIP, respectively.
$t^{t_f}_{sep_i}$	The safe time separation between the VIP and CMO at t_f.
s_l	A predefined minimum time interval (in s) that must be maintained between the VIP and CMO at t_f.
$\overleftrightarrow{O_1B_1}$, $\overleftrightarrow{O_2B_2}$	The common tangents of circles C_{O_i} and C'_B in the WCS.
A'	The intersected point of $\overleftrightarrow{O_1B_1}$ and $\overleftrightarrow{O_2B_2}$ in the WCS.
$\overline{A'O_i^{t_f}}$, $\overline{A'O_1}$	The sides of the triangle $\triangle A'O_i^{t_f}O_1$ connecting point A' to points $O_i^{t_f}$ and O_1, respectively, see Fig. 4.11.
α'_i	The angle between $\overline{A'O_i^{t_f}}$ and $\overline{A'O_1}$ in the WCS.
ϕ'_i	The angle of the displacement vector from B^{t_c} to $O_i^{t_f}$ in the WCS.
β'	The angle of the displacement vector from B^{t_c} to B^{t_f} in the WCS.
$'d_i/\ \lvert\overline{B^{t_c}O_i^{t_f}}\rvert$	The Euclidean distance between B^{t_c} and $O_i^{t_f}$ in the WCS.
δ_a	An angle safety margin.
δ_d	A distance safety margin.
$B_k^{t_f}$	An approximation point.
k	Iteration index.

.2 Derivation of the α_i' angle

As shown in Fig. 4.11, the angle $\angle\alpha_i'$ between $\overline{A'O_i^{t_f}}$ and $\overline{A'O_1}$ is calculated. Obviously, since $\triangle A'O_i^{t_f}O_1$ and $\triangle A'B^{t_c}B_1$ are similar, there is:

$$\frac{|\overline{A'O_i^{t_f}}|}{|\overline{A'B^{t_c}}|} = \frac{r_{O_i}}{r_B'}, \tag{1}$$

$$\frac{|\overline{A'O_i^{t_f}}|}{|\overline{A'B^{t_c}}|} = \frac{\overline{|(A'B^{t_c} + B^{t_c}O_i^{t_f})|}}{|\overline{A'B^{t_c}}|} = \frac{|\overline{A'B^{t_c}}| + d_i'}{|\overline{A'B^{t_c}}|} = \frac{r_{O_i}}{r_B'}, \tag{2}$$

thus,

$$|\overline{A'B^{t_c}}| = \frac{r_B'}{r_{O_i} - r_B'}d_i', \tag{3}$$

From the Pythagoras equation, there is:

$$|\overline{A'B_1}| = \sqrt{|\overline{A'B^{t_c}}|^2 - {r_B'}^2}. \tag{4}$$

Inserting Eq. (3) into Eq. (4), there is:

$$|\overline{A'B_1}| = r_B'\sqrt{\frac{{d_i'}^2}{(r_{O_i} - r_B')^2} - 1}. \tag{5}$$

For α in $\triangle AB^tB_1$, there is:

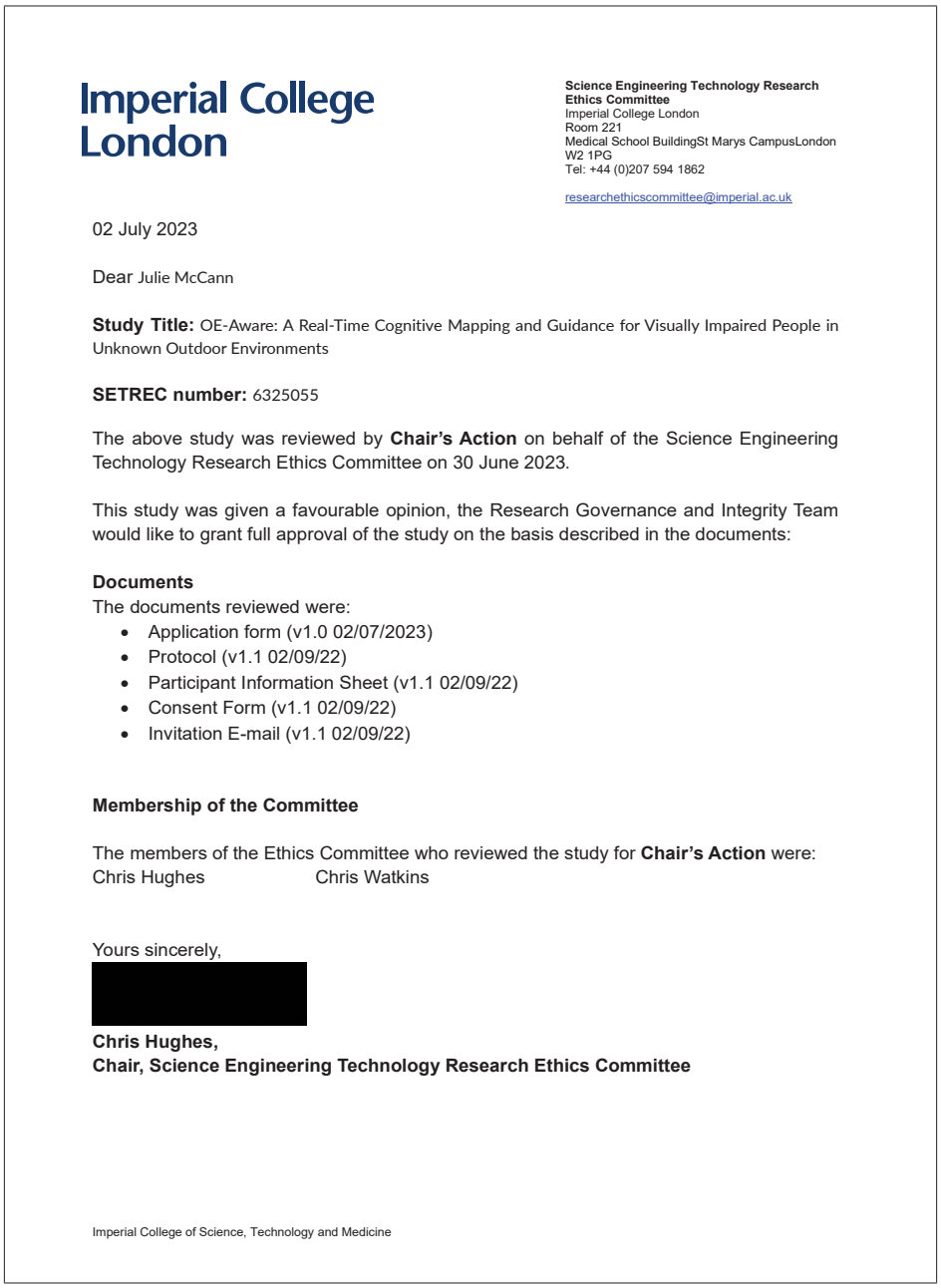
$$tan(\alpha_i') = \frac{{}'r_B}{|\overline{A'B_1}|} = \frac{1}{\sqrt{\frac{{d_i'}^2}{(r_{O_i} - r_B')^2} - 1}}, \tag{6}$$

thus,

$$\alpha_i' = tan^{-1}\left(\frac{1}{\sqrt{\frac{{d_i'}^2}{(r_{O_i} - r_B')^2} - 1}}\right). \tag{7}$$

.3 Ethical Approval

This research adhered to the ethical guidelines set forth by the Research Governance and Integrity Team (RGIT) at Imperial College London and received ethical approval from the Science Engineering and Technology Research Ethics Committee (SETREC) with reference number 6325055. Fig. 1 shows the ethical approval form obtained by the SETREC.



Imperial College London

Science Engineering Technology Research Ethics Committee
Imperial College London
Room 221
Medical School BuildingSt Marys CampusLondon
W2 1PG
Tel: +44 (0)207 594 1862

researchethicscommittee@imperial.ac.uk

02 July 2023

Dear Julie McCann

Study Title: OE-Aware: A Real-Time Cognitive Mapping and Guidance for Visually Impaired People in Unknown Outdoor Environments

SETREC number: 6325055

The above study was reviewed by **Chair's Action** on behalf of the Science Engineering Technology Research Ethics Committee on 30 June 2023.

This study was given a favourable opinion, the Research Governance and Integrity Team would like to grant full approval of the study on the basis described in the documents:

Documents
The documents reviewed were:

- Application form (v1.0 02/07/2023)
- Protocol (v1.1 02/09/22)
- Participant Information Sheet (v1.1 02/09/22)
- Consent Form (v1.1 02/09/22)
- Invitation E-mail (v1.1 02/09/22)

Membership of the Committee

The members of the Ethics Committee who reviewed the study for **Chair's Action** were:
Chris Hughes Chris Watkins

Yours sincerely,

[redacted]

Chris Hughes,
Chair, Science Engineering Technology Research Ethics Committee

Imperial College of Science, Technology and Medicine

Figure 1: Ethical Approval Form.