Autonomous Robot Navigation through a Crowded and Dynamic Environment

Using A Novel form of Path Planning to Demonstrate *Consideration* towards Pedestrians and other Robots

Ross Walker

Department of Automatic Control and Systems Engineering

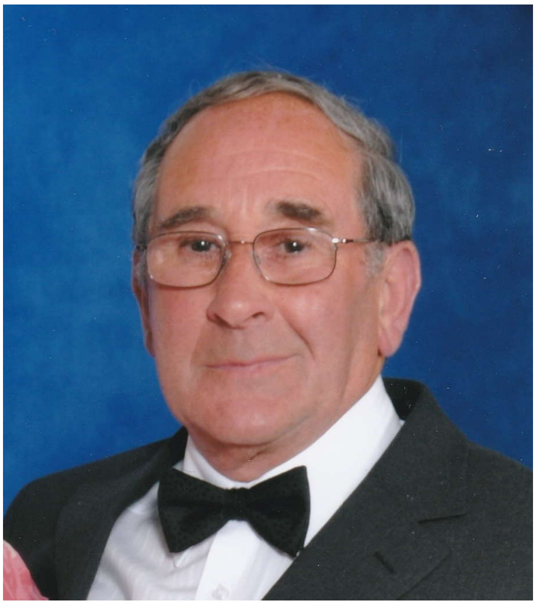
University of Sheffield

This thesis is submitted for the degree of Doctor of Philosophy

Doctor of Philosophy

August 2017

I would like to dedicate this thesis to the memory of my late grandfather.



Norman Cleasby
(1934 - 2014)

Acknowledgements

I would like to make the following acknowledgements to all the people who have helped me through this journey. Firstly I would like to thank my supervisor, Professor Tony Dodd, for allowing me to propose my own thesis subject, which gave me free reign to get creative. I would like to thank my parents for all the financial support they have provided me with, for my entire life. And lastly, I would like to thank my fiancé, the soon to be Elizabeth Helen Walker, for helping me keep my sanity (just).

Abstract

This thesis presents a novel path planning algorithm for robotic crowd navigation through a pedestrian environment. The robot is designed to negotiate its way through the crowd using *considerate* movements. Unlike many other path planning algorithms, which assume cooperation with other pedestrians, this algorithm is completely independent and requires only observation.

A *considerate navigation strategy* has been developed in this thesis, which utilises *consideration* as an directs an autonomous mobile robot. Using simple methods of predicting pedestrian movements, as well as simple relative distance and trajectory measurements between the robot and pedestrians, the robot can navigate through a crowd without causing disruption to pedestrian trajectories.

Dynamic pedestrian positions are predicted using *uncertainty* ellipses. A novel Voronoi diagram-visibility graph hybrid roadmap is implemented so that the path planner can exploit any available gaps in between pedestrians, and plan considerate paths. The aim of the *considerate path planner* is to have the robot behave in specific ways when moving through the crowd. By predicting pedestrian trajectories, the robot can avoid interfering with them. Following preferences to move behind pedestrians, when cutting across their trajectories; to move in the same direction of the crowd when possible; and to slow down in crowded areas, will prevent any interference to individual pedestrians, as well as preventing an increase in congestion to the crowd as a whole.

The effectiveness of the *considerate navigation strategy* is evaluated using simulated pedestrians, multiple mobile robots loaded with the path planning algorithm, as well as a real-life pedestrian dataset. The algorithm will highlight its ability to move with the aforementioned *consideration* towards each individual dynamic agent.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The Three Laws:
1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

— Issac Asimov*, Handbook of Robotics, 56th Edition, 2058 A.D.*

ROBOTS have become ubiquitous in modern society, ranging from stationary industrial CNC machines with rigidly defined operational motions to highly dexterous humanoid robots that can operate autonomously and mimic many human movements [25–27]. The term robot was coined in the 1920 play *Rosumovi Univerzální Roboti* (Rossum's Universal Robots) by Karel Čapek, and since the first electronic computer to use integrated circuitry was invented (c.1950) the technology has increased exponentially through the self-fulfilling prophecy of Moore's Law [28]. In recent years there have been vast increases in computer processing power, including the advent of affordable super-computers [29, 30]. This has allowed machines to become increasingly capable of performing more and more complex computational tasks in a shorter time-frame.

This improvement is an underlying feature that allows robots to operate with increased precision, speed, and efficiency. Robots do not suffer from fatigue and can operate consistently operating in environments that are often unsuitable for humans [31]. However, aside from using robots to outperform humans at certain tasks, robots are being developed in an assistive capacity, whereby humans and robots can interact with one another. The fast-paced and evolving field of robot autonomy is resulting in robots becoming less segregated from humans through human-robot interaction (HRI), and where less and less direct user control is required for the robot to operate. This leads to the motivation behind this thesis, whereby if humans are to interact with more intelligent robots, then the robots must be able to safely move around the same environment as a human.

Much of the HRI arena is dominated by *socially assistive robots* [32], used as support for disabled people, autistic children, and therapeutic needs, etc. However, outside the broad concept of Isaac Asimov's three laws of robotics, which covers a robot's behaviour towards all humans, current realistic HRI involves a command flow between robot and human in order to complete a particular task. Traditionally the level of HRI is defined by the amount of input a robot requires in order to complete a pre-determined task [33], using a robot that is already known to the user. Whilst for more modern systems that have a higher level of versatility, the purpose of the robot may be unknown to an interacting user.

Rather counter-intuitively the traditional rating system of HRI performance results in a higher score if the human provides less input. However, for assistive robots additional human input would be required, allowing the robot to assist more effectively. The input commands can take any form, including: remote control; verbal; gesture recognition; etc; as long as an instruction is directed towards and able to be processed by the robot. An overview is given to highlight these different approaches toward defining HRI performance:

a_1 Traditional: ***Increasing*** robot autonomy with ***Decreasing*** human input

- $a_{1.1}$ **Human Control:** The robot is directly controlled by explicit commands from an experienced user, with an automatic pre-programmed response.
- $a_{1.2}$ **Casual Commands:** The robot's purpose is known to the user, and the commands given allow the robot to respond in a selection of pre-determined ways it calculates to be the most suitable.
- $a_{1.3}$ **Robot Control:** Robot is able to complete a pre-determined task set by the user from start to finish. All the user is required to do is set-up the equipment appropriately.

a_2 Modern: ***Increasing*** robot autonomy with ***Increasing*** human input

- $a_{2.1}$ **Direct Learning:** Each human input allows the robot to learn through every experience. The robot may provide suggestions that make the task more efficient, based on acquired knowledge through previous experience of the task's execution.
- $a_{2.2}$ **Implicit Response:** The robot's purpose may not be known by the user, however the robot can assess the situation independently and respond without a users direct input, as the requirement is understood without being stated.
- $a_{2.3}$ **Independent Cognition:** Theoretically, the machine can realise a task should be undertaken without any prompt, or even prior knowledge, predicting possibilities from a related knowledge base. The roles of the human and robot may be reversed, with the robot seeking to ask assistance from the human to complete a task or even offering advice.

Increasing the level of human input does not mean more human control, rather the human input should result in a collaborative effort between human and robot. Although human input can be explicit and HRI can produce cooperative behaviour through direct interaction, it can also be implicit and behave considerately through passive observation. This latter notion of a robot's *considerate* behaviour is the seed to the motivation of this thesis, by using passive input from observing pedestrians move.

1.1 Motivation

As observed through the evolving HRI field, robot interfaces are becoming more human-like. Already humanoid robots are designed to walk like humans [26, 27], talk like humans [34], even used to assist humans as they walk [35], and a robot should be able to interact with a human as easily as another human would. Robots that are achieving more collaborative interactions is developing the field social robots, including how they move around with us. Designing a human-aware robot navigation that can be implicitly understood by humans will increase the diversity of HRI implementations.

HRI with autonomous mobile robots (AMRs) began in the previous century, gaining notability with museum tour guides (e.g. [36, 37]). These implementations relied heavily on patience towards pedestrians, pausing the AMR's movement until any pedestrians had moved out of the way. The AMRs navigated differently from huamns and did not implement a co-operative interaction strategy with the pedestrians in order to navigate, which is often how pedestrians themselves are observed to work together [38, 39]. Even in recent implementations, which intend to adapt more human-like behaviours (e.g [40–42]), the robots move at a much slower pace than the pedestrians they navigate around. Pedestrians manoeuvre around the robots, allowing the robots to move relatively unhindered along a straight path. Even human-human navigation is still a highly researched area (e.g [43–45]), providing evidence that our knowledge of how people successfully navigate together is insufficient. To design a robot capable of successfully navigating among pedestrians, when even the navigation system of the pedestrians are unknown, presents a very difficult real-world challenge.

1.1.1 Improving Human-Robot Interaction

Creating navigation systems capable of negotiating though crowded pedestrianised environments can be beneficial to advancing all forms of autonomous robotics. It can progress HRI by reducing the gap between the environments that humans and robots can safely co-inhabit, as well as increasing navigational autonomy overall. One of the main reasons for mobile robots still being segregated from humans is due to their inability to *manoeuvre* in similar ways. Even very slight adjustments in movement can affect the equilibrium required for multiple dynamic agents to successfully navigate together, such as with flocking algorithms [46], due to collective behaviour and swarm intelligence [47]. For example, cyclists are segregated from pedestrians due to their larger size, faster speed, and reduced ability to manoeuvre. Despite the fact they are both moving humans the behaviour of the human cyclist would not be able to navigate amongst human pedestrians because of these subtle differences. Robots are segregated even more due to additional factors, such as: slower movement speeds, no adaptability (a cyclist can always dismount and become a pedestrian), and dissimilar navigation protocols.

For multiple dynamic agents to interact within the same space successfully, each must be able to anticipate one another's movements. If a pedestrian does not move within a crowd with the same behaviour as the others, they will become a problem for those they interact with. Their responses will be unexpected, which would increase confusion, as well as increasing collision potential and congestion [39]. Some current robots may be likeable to humans in regards to physical appearance, however they have not been integrated into co-habitation environments as they are unable to navigate within them effectively (e.g. [25]). Robots are unable to perfectly communicate with humans, due to a lack of advanced intelligence and adaptive cognitive ability, associated with human reasoning. This means that each robot can only operate within specific parameters, designed for specific environments.

For robots to manoeuvre through crowds of pedestrians they do not require the ability to process information like a human[1]. However, they should at least appear to adopt human-like qualities, so they are capable of easily integrating with minimal disruption. The research presented in this thesis does not propose a complete robot platform which is indistinguishable from a human, rather it provides a novel framework for a robot's navigation system. The focus is on the area of pedestrian crowd navigation, using a novel path planner to predict paths in order to move considerately around pedestrians and avoid potential collisions before they have a chance of developing. This considerate navigation strategy (CNS) relies on an *implicit response*, Item $a_{2.2}$, to a minimal amount of passively observed pedestrian input. Rather than requiring explicit command inputs from a user, the CNS will move an AMR so that it causes minimal disruption to any surrounding pedestrians. As well as provoking implicit cooperation between the AMR and other pedestrians, by manoeuvring in ways similar to other pedestrians. A specific localisation system is not considered within this thesis, as the focus is on planning considerate paths using a novel considerate path planner (CPP). The AMR's ability to correctly determine its current position and goal location is a considerably large research area

[1] The *Human Brain Project* demonstrated the complexity of the human brain's processing power, with their simulation taking 20 minutes to emulate 1 second of brain activity [48]

of its own, such as SLAM (simultaneous localisation and mapping), and the CPP is only designed to plan considerate paths between AMR and goal from whatever data is provided on their positions.

1.1.2 Real-World Applications

Applications of HRI with AMRs have already proved effective in assistance for rehabilitation [49–51]. The development of this thesis' CPP will improve these current applications. Especially for visually or physically impaired individuals, if implementing it into autonomous mobility devices that can assist how they move. The CNS can also be beneficial for emergency response services that require fast access through a crowd, in order to attend a medical emergency. As the CNS navigates with implicit consideration it will not rely on pedestrian cooperation, and so could efficiently navigate through a crowd faster and without hindrance.

Autonomous Mobility Devices

For individuals suffering from blindness, or visual impairment, navigating through a crowd can be difficult as it is not possible to assess how other pedestrians are moving in front of them. Although audio and touch are still available, audible details cannot define the locations or orientations of individuals whilst touch requires a direct contact that should ideally be avoided. Current non-robotic methods of navigation for the blind involve assistance from guide dogs, which are only capable of manoeuvring around obstacles if the owner already knows the route[2]. Robotic devices, such as the *UltraCane* [52], only detect objects in the immediate vicinity and require the operator to decide on what action to take based on vibrations they receive through the handle. The problem with these systems is that they only assess immediate obstacles. The motivation for this thesis is to allow visually impaired pedestrians to be lead through much more complex dynamic environments. The CNS would not require any of the other crowd members to make allowances for it, and the user will be directed through the crowd without hindrance.

A further possibility includes the integration of the CNS into autonomous wheelchairs, to help benefit those with severe movement difficulties. For individuals that cannot successfully move their wheelchairs around without assistance, the CNS will allow the wheelchair to plot considerate paths that it can then move along by itself.

Emergency Response

When attending to someone who requires first aid often time is of the essence, especially for those who have suffered a heart attack or stroke. If such an incident occurs in a crowded area, being able to get to the scene as efficiently and quickly

[2] http://www.guidedogs.org.uk/

as possible can add valuable seconds required to saving someone's life. As the CPP uses predictive models to anticipate crowd movements, the planned path could be more efficient than if performed by a human. This is not only beneficial in getting to someone as quickly as possible, but also moving them out and away from the crowd (e.g. to an ambulance), in order to get them the attention they require as soon as possible.

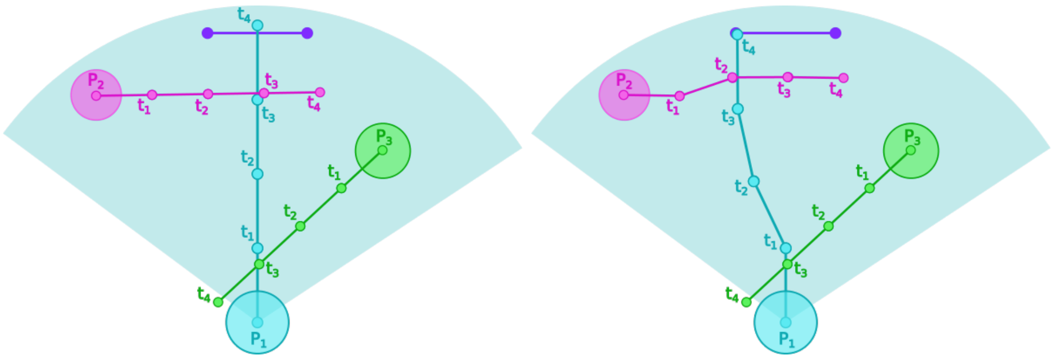
1.2 Overview of Related Research

Current research into AMRs focus on specific applications, such as [53]: moving over challenging terrain, completing tasks dangerous for humans, and exploration. The success of mobile robot autonomy within the field of navigation is dependent upon the environment the robot navigates. Successful cooperation in a multi-robot environment, such as swarm robotics and the Kiva robot system [54], have achieved excellent navigational results. However, these occur in human-free zones, with both systems failing if a pedestrian enters the environment. This limitation is assumed to be because the dynamic behaviour of pedestrians is very difficult to predict, when focusing on the interactions of individuals, as people are not just limited to moving efficiently, but by a myriad of social, personal, and emotional issues [55, 56]. To integrate a navigational robot into a pedestrian environment, both crowd behaviour and path planning techniques must be investigated.

1.2.1 Pedestrian Modelling

Pedestrian crowd dynamics and modelling has been studied for many years, with force-based models dating back to the 1950s [57]. Although these ansatz models predicted a generalised movement of a crowd they failed at predicting the exact movement of individual pedestrians. Modern fluid dynamics offer an excellent method to analyse the flow of large groups on a macroscopic level [58, 59], but again not the individual.

At a microscopic level pedestrians can be observed to interact using cooperation between individuals [60], using vision to assess collision potential [2]. They move by predicting the locations of other pedestrians a few steps ahead of themselves, whilst aiming to move as directly as possible towards their goal [61]. However, unpredictable variables may prevent any model from producing a perfect movement prediction of each pedestrian [62]. Only estimations can be made to within an acceptable boundary. In order for microscopic pedestrian models to adopt more behavioural predictions, goal orientation is assumed *a priori*. This has allowed order to appear from chaos, producing uni/bi-directional traffic and lane formation [63, 64], resulting in more easily predicted trajectories.



(a)

(b)

Fig. 1.1 Generic visualisation of how a pedestrian may select their trajectory based upon the potential collisions with others, as suggested in [1, 2]. P_1 is a pedestrian walking towards a door (purple line) with a 90° FoV, which contains two more pedestrians P_2 and P_3. (a) If no path planning is implemented P_1 and P_2 will collide three time-steps into the future. Although P_2 moves directly in front of P_1, the collision point of the two trajectories do not coincide. Therefore, the straight line path of P_3 can be maintained. (b) By implementing a simple form of considerate path planning, both P_1 and P_2 begin to adjust their trajectories at $t = 1$. At $t = 2$, P_2 has moved out of the direct path of P_1 and so resumes their original direction. P_1 considerately continues to move behind P_2 until $t = 3$. Both P_1 and P_2 have momentarily deviated form their ideal paths in order to avoid a collision.

Goal orientation is a key assumption that creates more elegant outputs, however for a free-flowing environment (e.g. an exhibition space) this long-term prediction horizon must be cropped. Pedestrians are generally observed to deviate as little as possible from a straight path to their destination, deviating only to avoid collisions [1, 2], Fig. 1.1. Simple short-term linear trajectories predictions should therefore suffice for predictive path planning in the near future [17], as unknown destinations need not be assumed.

1.2.2 Human-aware Mobile Robot Navigation

Autonomous mobile robots that engage with HRI primarily began with the museum tour guide RHINO [36] in 1998. Since then many AMR incarnations have attempted to address the problem concerned with appropriate navigation in a pedestrian environment, with one of the most recent being Obelix [40] in 2014. Unfortunately both of these examples, including others (e.g [41, 42]), have produced a robot that moves much slower then any of the crowd members. This fails to address the problems of interdependent navigation, which occurs between pedestrians [60]. If the robot's move slower than a human they are navigating along a path that is irrelevantly occupied by pedestrians, as the pedestrians navigate around the robot.

Robot path planners that are capable of navigating with *real* consideration is currently an unexplored area in the path planning literature. Social path planners *politely* stop-and-wait if a pedestrian crosses their path (e.g [7, 65, 66]), which may be considerate to the individual but not to other crowd members moving near the robot. Collaborative path planners are also very optimistic in assuming a pedestrian will engage in a *joint* form of collision avoidance (e.g [67–69]), which may lead the robot into densely populated areas it can move though only if the pedestrians decide to allow it. The CNS would allow consideration to be shown simultaneously to multiple pedestrians within a crowded environment, via simple implicit reaction to predicted pedestrian motion.

1.3 Proposal of A Considerate Navigation Strategy

This thesis proposes a CNS, designed specifically for a robot to move within an environment of free-flowing pedestrians. No explicit input or interaction is required from any pedestrian, only implicit input obtained from observation. A suitable path planner, and the aims and objectives of the CNS as a whole is presented.

1.3.1 The Considerate Path Planner

The definition of "considerate" is subjective and dependent upon interpretation:

> **considerate** /kənˈsɪd(ə)rət/
> *adjective*
> 1. careful not to inconvenience or harm others.
> (Oxford English Dictionary)

For the CNS to plan paths that respect this definition, the novel CPP has been designed. Traditional path planners choose the path of shortest distance to the robot's destination. This is excellent within static environments, however in a dynamic environment the shortest distance does not guarantee the quickest time, let alone the safest one.

Using a model to anticipate microscopic pedestrian movement will allow the CPP to plan paths that avoid any potential collisions before they can occur. Paths can be planned that not only do not interfere with pedestrian trajectories at the current moment, but also cause minimal distruption in the future. Using linear extrapolation over a short prediction time-horizon, the future positions of pedestrians can be predicted. Whilst updating a model of the environment over this prediction time-horizon will allow the CPP to exploit anticipated gaps in between individual pedestrians, and allow considerate paths to be predicted through the crowd as a whole.

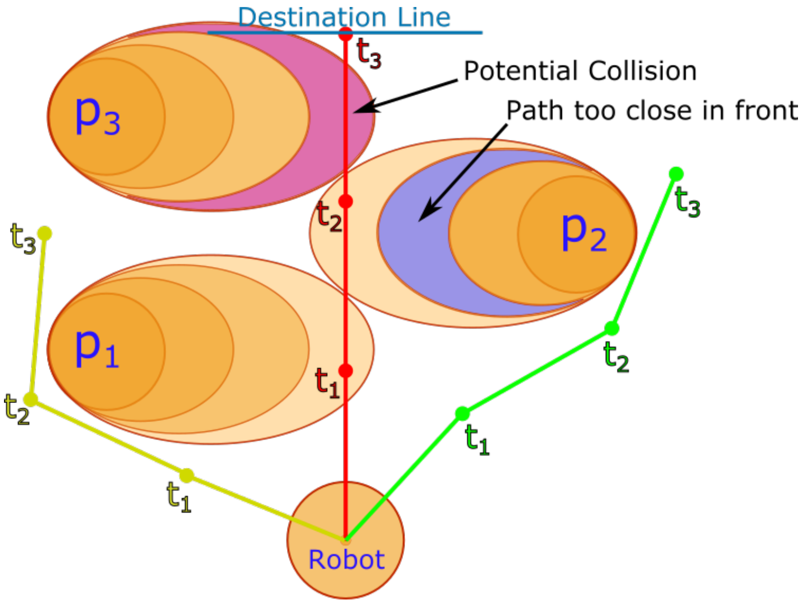


Fig. 1.2 Visualisation of how paths showing consideration (green/yellow) are advantageous when compared to direct paths (red). Demonstrated over three time-steps, t_1 to t_3, the considerate paths will prevent collisions and avoid congestion. p_1 to p_3 are pedestrians moving with a CVM, represented with a novel uncertainty ellipse that elongates at each time-step. The blue ellipse area of p_2 indicates the red direct path is too close, cutting in front of the pedestrian at $t = 2$. Purple ellipse area of p_3 indicates a potential collision with the red direct path at $t = 3$. The green most considerate path chooses to move behind p_2, which sacrifices distance and time for a collision free path. The yellow alternative considerate path shows an example of an alternative path that moves behind p_1 and p_3 instead. This is a result of the CPPs exploration of the model environment, allowing a number of paths to predicted before the best one is chosen.

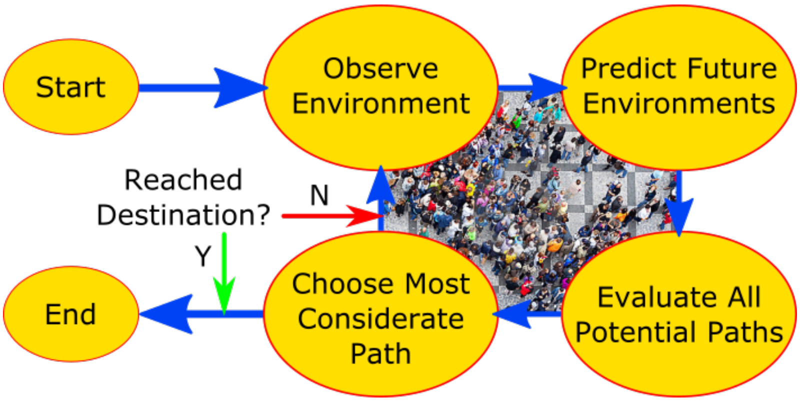

Start
Observe Environment
Predict Future Environments
Reached Destination?
N
Y
End
Choose Most Considerate Path
Evaluate All Potential Paths

Fig. 1.3 Overview of the CNS's autonomy cycle. Each stage will consist of a number of dynamical systems, from which each produce an output appropriate for the input of the next stage. This repetition allows the robot to act autonomously without requiring any explicit human operation. The robot can then manoeuvre through the pedestrian crowd, fulfilling all objective requirements until it arrives at its destination.

1.3.2 Aims and Objectives

The main aim for this thesis is to develop a *considerate* navigation algorithm that can be implemented into any robotic system capable of moving within a pedestrian environment. It will show consideration towards all agents within the environment, as well as minimise the disruption to the overall flow of the crowd, by not increasing congestion and reducing collision potential.

From the aim of the CNS the following targetable objectives can be defined for the CNS and the CPP to achieve:

b_1 Objectives for the CNS to fulfil when it is directing an AMR through a crowd

  $b_{1.1}$ Move with consideration towards other pedestrians

  $b_{1.2}$ Arrive successfully at a prescribed destination

  $b_{1.3}$ Avoid collisions with anything

  $b_{1.4}$ Minimize any additional congestion the robot may cause to the crowd as a whole

b_2 Objectives for the CPP to fulfil as it plans paths, which will allow the primary objective, Item $b_{1.1}$, to be achieved

  $b_{2.1}$ Plan paths that minimise potential collisions

  $b_{2.2}$ Plan paths that move in the same pedestrian direction

  $b_{2.3}$ Plan paths that avoid crossing another pedestrians path

The CPP objectives can be used to quantify the performance of the CNS, by also ascribing them to the movement the AMR makes in the real-world.

1.3.2.1 Quantifying Performance

To effectively evaluate the CNS's performance, a number of quantitative performance metrics must be established to quantify the success of the system. Although the CPP aims to plan considerate paths, the CNS will not result in the AMR behaving considerately if those planned paths are not executed well. For autonomous navigation and obstacle avoidance a number of typical performance criteria exist, however each vary dependent upon the specific goals of the system. Either the use of a specific set of items to evaluate [70], or through physical tests [71]. The metrics must be comparable with the specific objectives of the algorithm, which when regarding the novelty of the CPP must orientate around *consideration.* The performance of the CNS will be evaluated using a newly defined set of metrics for consideration, defined in Section 6.2.3, as an AMR navigates around dynamic agents from both a microscopic pedestrian simulator, Chapter 6, and a real-life pedestrian dataset, Chapter 7. Using traditional metrics to evaluate journey efficiency, and novel metrics that measure the objectives of the CPP.

1.4 Impact and Novel Contributions

In regards to human-robot navigation the CNS has real-world impact potential by aiding in the development of autonomous mobility devices, which has been a motivation for this thesis, Section 1.1.2. The CPP is a novel path-planner that generates *considerate* paths, preventing a robot from interfering with the navigation of other dynamic agents. This novel form of path planning is very useful as it will allow a robot to be released into a high density dynamic environment and navigate with minimal disruption toward any of the other agents.

When assessing a robot's level of autonomy, traditional taxonomies consider the level of human interaction as a negative effect on a robot's autonomy, i.e. greater human input results in a lower autonomy level. Developing autonomous robotics, especially for HRI scenarios, actually requires input from humans in order to achieve the highest level of autonomy. A new taxonomy is proposed to account for HRI and autonomous mobile robots in Section 2.4.3, Table 2.5, which traditional definitions of autonomy fail to successfully address.

The Voronoi diagram of circles [72] has been adapted in order to allow the CPP to approximate the Voronoi diagram of fully-intersecting and arbitrarily placed ellipses in Chapter 4, as an exact method for constructing one has not yet been developed. The Voronoi diagram is achieved with an extremely high level of accuracy and efficiency, including a time complexity nearly equal to the original Voronoi diagram of circles algorithm.

Specific novel contributions are:

c_1 Section 2.4: A new taxonomy for autonomous mobile robot navigation in a pedestrian environment

- A new taxonomy for human-mobile robot interaction, Table 2.5. True autonomy must counter-intuitively require input from humans, either by explicit or implicit means, which traditional methods do not account for.

c_2 Section 3.2: The uncertainty ellipse, to embrace the uncertainty of pedestrian movements

- Embracing the uncertainty, based on predictions from velocity. An uncertainty profile, Fig. 3.6, used to create a split ellipse, Fig. 3.5, based on pedestrian walking variations [73].

c_3 Section 3.3: The Considerate Path Planner

- The function that allows *considerate* paths to be plotted. Simple measurements, involving proximity and relative trajectory between robot and pedestrian, can be applied to a search algorithm in order to plan considerate paths as part of the CNS.

c_4 Section 4.1: Approximating the Voronoi Diagram of Fully-Intersecting Ellipses

- Adapting the algorithm for the Voronoi diagram of circles [72] to create a Voronoi diagram of pseudo-ellipses, formed from four tangentially aligned circles.

c_5 Section 4.4: A Voronoi diagram-visibility graph hybrid roadmap.

- A Voronoi diagram inspired roadmap, spliced with a modified visibility graph. The roadmap can move along the safest path of the VD, but can at any moment shift to a modified visibility graph that has a naturally larger object clearance safety-margin.

c_6 Section 6.2: Quantitative performance metrics for evaluating how *considerately* the CNS performs:

- A novel quantitative performance metric that allows the consideration performed by the AMR to be quantified, as the AMR moves within a crowd.

The research developed in this thesis has contributed to [74]:

- R.Walker and T. Dodd, “A novel path planning approach for robotic navigation using consideration within crowds,” in Towards Autonomous Robotic Systems (C. Dixon and K. Tuyls, eds.), vol. 9287 of Lecture Notes in Computer Science, pp. 270–282, Springer International Publishing, 2015.

1.5 Thesis Outline

The thesis is structured to show the CNS progress from its initial conception, through to the development stages, before it is finally implemented and evaluated using simulated pedestrians and a real-life pedestrian dataset.

Chapter 2 To begin with a comprehensive literature review is undertaken. This is designed to highlight the current research being undertaken in the field of human-aware mobile robot path planning. Current AMR path planning methods and collision avoidance strategies are reviewed, as well as human-human path planning and collision avoidance. Polite human-robot collision avoidance strategies are currently being developed by researchers, however these are not designed to perform in a crowded environment. Those that are do no not exhibit human-like behaviour, which is the aim of the CNS.

Chapter 3 This chapter begins with a problem statement that defines the current challenges faced, following the review of the literature. An appropriate pedestrian model is developed. The movement model is kept simple, whilst the uncertainty of the model is *embraced* in order to cover more areas of where they may move. These uncertainties are constrained to ellipses, modifiable due to: initial walking speeds; movement likelihood; and spatial views of the pedestrians. The AMR model is also defined, as well as the environmental mapping procedures. This includes the development of the VD-VG roadmap, the search algorithm, and the *considerate weights* that allow the most considerate paths to be planned.

Chapter 4 A novel approximation of the Voronoi diagram (VD) of ellipses implemented. Using a novel adaptation of the VD or circles, ellipses are approximated using tangentially aligned circles to create pseudo-circles. From the VD of pseudo-circles, the VD of ellipses can be approximated by converging the vertex points of the VD of pseudo-circles onto the vertex points for the VD of ellipses. The algorithm is scalable as it executes with a logarithmic time complexity.

Chapter 5 The CNS is now implemented and all the relative stages of the algorithm are detailed. This includes how the AMR obtains environmental data; how the search algorithm is dynamically modified in order to search considerately; as well how the search effort is reduced by preventing homotopic path repetition. Also, steps are included that encourage paths diversity by exploring the surrounding environment. Finally, the execution of the most desirable path is covered, along with the AMR movement model used.

Chapters 6 and 7 The CNS is evaluated to ensure that it fulfils all behavioural objectives (Section 1.3.2) and behaves considerately. Each movement of the AMR is quantified and evaluated using the proposed QPMs for considerate movement (Section 1.3.2.1).

> **Chapter 6** is designed to discover the optimal system variables, by testing various parameters and scenarios using an environment populated with simu-

lated pedestrians, and the optimal prediction time-horizon required to produce the most successful paths is determined. From review of the literature, pedestrians are observed to move with consideration to one another. Therefore, the experimental set-up used in these empirical studies is used first. Consideration and implicit cooperation are observed between the AMR and an individual dynamic agent. Experiments are then devised to evaluate how considerately the AMR moves within a crowd, which are compared to both a traditional A* search algorithm as well as a modified *considerate* A* search algorithm.

Chapter 7 then evaluates the CNS using a real-life pedestrian dataset, in order to see how *considerately* the AMR responds to the movement of humans. Experiments are set-up so that the AMR can move with the overall crowd flow, as well as perpendicular to pedestrian traffic. The paths the AMR moves along are shown to be considerate as they do not interfere with pedestrian trajectories (moving behind them rather than crossing in front) and do not attribute towards increased congestion (moving in the same direction as neighbouring pedestrians).

Chapter 8 The thesis concludes with an evaluation and summary of all the work presented in the thesis, along with an appropriate critique how the objectives were achieved. The potential for future work is also include, which would be to implement the CNS on a real robot platform.

Chapter 2

Literature Review

"Everything should be made as simple as possible, but not simpler."

— Albert Einstein

THIS chapter will evaluate the present literature in the field of human-aware mobile robot social path planning. An overview of autonomous mobile robots (AMRs) is given, pertaining to how AMR's are deemed autonomous and perceived as intelligent, Section 2.1. The requirements of an AMR to then navigate within a dynamic pedestrian environment is provided, which will initially ascertain the AMR's navigational and behavioural requirements. A review of current path navigation strategies is then undertaken, Section 2.2, including what forms of environmental mapping can be implemented, along with path planning collision avoidance techniques.

As the AMR is required to navigate within a crowd of pedestrians human path planning techniques are also investigated, Section 2.3. By understanding the mechanics of crowd phenomena a path planning algorithm can be devised that imitates human motion. This is achievable by respecting social proxemics and modelling direct human-human interaction. Various pedestrian models are evaluated, along with empirical collision avoidance experiments between individual pedestrians.

Human-aware mobile robot path planning focuses on replicating human motion patterns. Therefore, the chapter concludes, Section 2.4, by evaluating current algorithms against the aforementioned pedestrian path planning techniques and observed interactions. Cooperation between human and AMR is a consistent assumption made by most socially-aware path planners, with humans accepting the robot's presence as another human. The motion of the AMR must be "legible" to a human so that they can anticipate the AMR's movement, and vice-versa, so that cooperation can be achieved. A novel taxonomy is also proposed that contests the traditional view of robot autonomy stated at the start of the chapter. As an AMR receives greater levels of input from a pedestrian its autonomy will actually increase, rather than decrease.

2.1 Autonomous Mobile Robots

> "The problems of autonomous robotics include things such as: making sure the robot is able to function correctly and not run into obstacles autonomously." [75] (Dr. George A. Bekey)

This quote represents the difficulty associated with developing a successful AMR. An AMR must be able to successfully differentiate between obstacles, and then be capable of manoeuvring around them. This thesis focuses on developing an AMR capable of independent navigation through a pedestrian crowd. To achieve this the AMR must be able to perceive and interact with its surrounding environment, and the objects within it, appropriately.

A classically defined robot is a machine that can carry out complex actions automatically [76]. Depending on whether a human operator or the robot itself is in control of these actions determines if the robot is not, or is, autonomous, respectively. A robot could be considered as simply either autonomous or not autonomous [77]. However, this generalization is rather simplistic and autonomy is an elastic concept that can change its definition based on the subject, as well as a subjective evaluation. Robot autonomy is defined as being capable of carrying out all of its actions without the input of a human operator [75, 78, 79]. Traditionally a robot's autonomy directly concerns the level of influence a human operator has over the robot's behaviour. Taxonomies that describe robot autonomy have levels that range anywhere between 4 and 12 [80], however 10, Table 2.1, is the most frequently used in order to describe the separate levels of robot autonomy [81–83][1].

When robots are tasked to operate within uncertain environments, such as with other dynamic agents, as the human control reduces the robot's intelligence must increase. This can sub-divide autonomous robots into three types [20]:

a_1 **Programmable Automatic Robots:** Programmed to perform a set of repetitive tasks in known environments. e.g. Industrial robot.

a_2 **Adaptive Robots:** Capable of using sensors to detect surrounding obstacles and adjust its output accordingly. e.g. Kiva robots [54] used for automated material handling in warehouse organisation, controlled through a central computer.

a_3 **Intelligent Robots:** Capable of analysing a task and then choosing the most appropriate course of action. e.g. Robots moving in uncertain and dynamic environments.

[1] range from 1978 to 2013

#	Level of Automation	Description
1	Manual Control	The human monitors, generates options, selects options (makes decisions), and physically carries out options.
2	Action Support	The automation assists the human with execution of selected action. The human does perform some control actions.
3	Batch Processing	The human generates and selects options; then they are turned over to automation to be carried out (e.g., cruise control in automobiles).
4	Shared Control	Both the human and the automation generate possible decision options. The human has control of selecting which options to implement; however, carrying out the options is a shared task.
5	Decision Support	The automation generates decision options that the human can select. Once an option is selected, the automation implements it.
6	Blended Decision Making	The automation generates an option, selects it, and executes it if the human consents. The human may approve of the option selected by the automation, select another, or generate another option.
7	Rigid System	The automation provides a set of options and the human has to select one of them. Once selected, the automation carries out the function.
8	Automated Decision Making	The automation selects and carries out an option. The human can have input in the alternatives generated by the automation.
9	Supervisory Control	The automation generates options, selects, and carries out a desired option. The human monitors the system and intervenes if needed (in which case the level of automation becomes Decision Support).
10	Full Automation	The system carries out all actions.

Table 2.1 Example levels of robot autonomy as defined by Endsley & Kaber (1999), reproduced from [19]. The scale begins with a fully teleoperated robot with no autonomy, over which a human operator has complete control. As the level of autonomy increases the amount of human involvement gradually diminishes, until the robot is able to operate without assistance.

Dynamic Stages	**Sensors**	**Perception**	**Modelling**	**Planning**	**Task Execution**	**Motor Control**	**Actuators**
AMR	Object Data	Object Recognition	Environment Predictions	Path Planning	Robot Movement	Path Adjustment	Robot Position
Pre-programmed	1	-	-	-	2	-	3
Adaptive	1	2	-	-	3	4	5
Intelligent	1	2	3	4	5	6	7

Table 2.2 Demonstration of how additional sub-systems are required if a robot is to become more intelligent [20]. The various dynamic stages [21] are directly compared to the sub-systems required for an AMR. The numbers represent the order of the sub-system execution.

The definitions between autonomy and intelligence are often blurred, and intelligence does not equal autonomy [84]. Autonomy [85] is considered as a system's ability to determine its actions for itself, be it exclusively through an internal state or through a sensor monitoring some physical condition of the environment. Whilst intelligence [86] is considered as a system's ability to act appropriately in uncertain environments, where the action increases the probability of success. For an AMR to be considered fully autonomous it requires an environmental input, and then through a series of internal dynamic operations it selects a movement via its own decision process [84]. This can be can be divided into 3 main capabilities [87]

b_1 **Perception:** The ability to recognise the surrounding environment

b_2 **Action:** The ability to respond to a sensation

b_3 **Cognition:** The ability to reason

The final point however progresses onto cognitive robotics whereby a response to stimuli is insufficient, requiring interpretation of intention as well. To create a robot robust enough to cope with *life* just like a human, the neural network of the human brain could be utilised (e.g. [88]). The processing of simple animal brains has already been used to control AMRs (e.g. [89]), however this area of robotics is beyond the scope of this thesis.

Once the required level of autonomy is sufficiently defined for a robot's purpose, it must be implemented through a network of multiple systems integrated together.

For example, an AMR will have several elements that perform separate functions, each processing an output suitable for the next stage's input, such as [21]:

Sensors → perception | modelling | planning | task execution | motor control → **Actuators**

However, even these have additional sub-systems, thus highlighting the level of complexity required to produce a *perfect* humanoid robot, with a huge number of interactive dynamic systems.

An AMR navigating through a crowd of people should require no human-operator and be classified as *intelligent*, Table 2.2. This is only an emergent intelligence, a result of the various interactions between the separate autonomous sub-systems. Harmonising how the data is used at each step is vital to ensure they are all processing data appropriately, allowing the subsequent stage to function. For an AMR with at least *adaptive* autonomy it's movements are not deterministic, as it chooses its own motion patterns. How it behaves whilst in operation and how successfully it reaches its sub-goals and final goal is the important factor [90]. Therefore, to appropriately evaluate the success of a AMR's autonomy requires specific performance metric evaluation. Although evaluation criteria for specific scenarios has been previously created (e.g. search and rescue [91]), there is no defined test criteria involving pedestrian navigation. The assumption that no collisions must occur can be made for obvious reasons, but how the robot should move through the crowd is completely dependent upon subjective design parameters. In response to this, an updated taxonomy is proposed in Section 2.4.3, Table 2.5, following a review of how current AMR's navigate with pedestrians.

2.1.1 The Autonomy Cycle

There are many scenarios and/or environments an AMR can be designed for. Each involves different mission objectives, sensory inputs, enviromental factors, etc, which lead to a very high level of customization. Conceptually, autonomy is a division between environment and machine with a sensing and actuation process that allows interaction between the two, Fig. 2.1a. In order to facilitate this interaction, the navigation problem for an AMR can be divided into four sub-problems [4], Fig. 2.1b. A task specific autonomy cycle can then be developed specifically for the goal of an individual AMR, Fig. 2.1c.

Each stage of the generalised AMR autonomy cycle, Fig. 2.1b, is a highly diverse and extensive research area. This thesis will focus primarily on the path planning element, with the other elements of the autonomy cycle being simplified so that the AMR can still be tested as a complete robotic system. As each step of the autonomy cycle requires appropriate outputs from the previous in order to achieve its goal it is important to consider each stage.

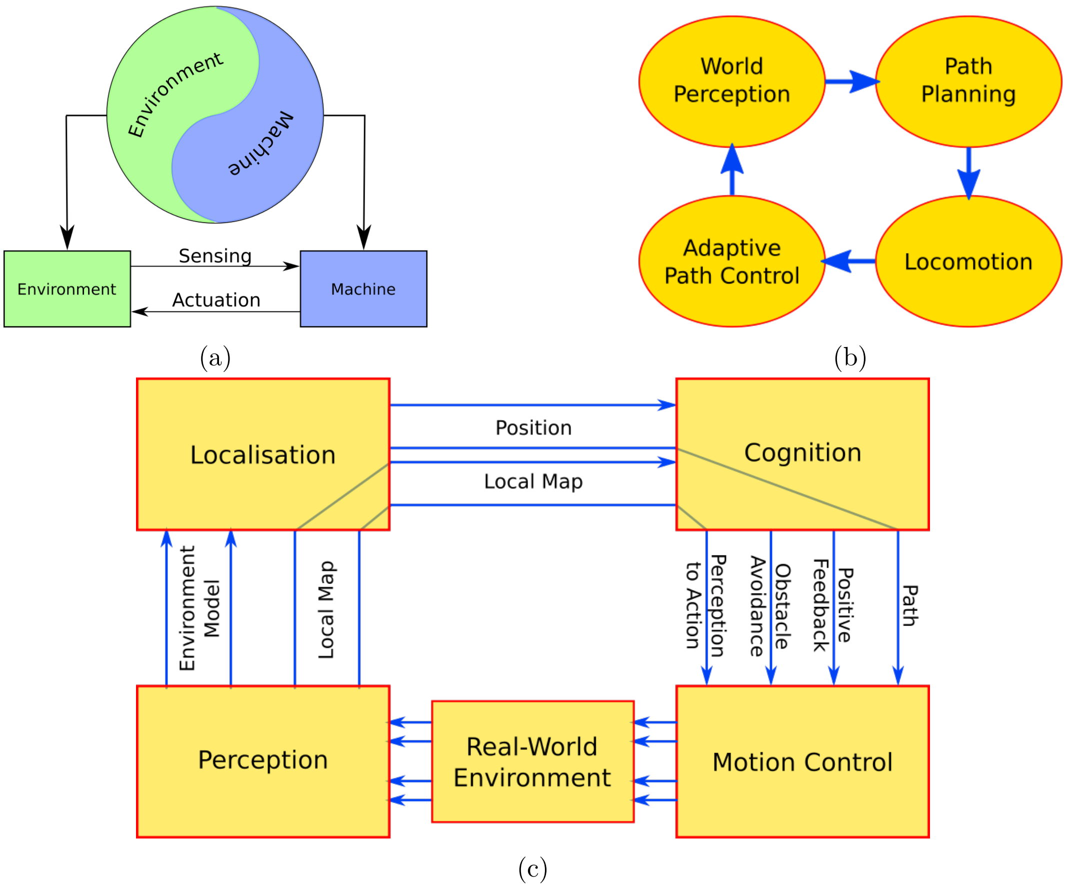


Fig. 2.1 Defining an AMR's autonomy cycle. (a) Concept of dividing the environment and the machine at an arbitrary line. Once the boundary is drawn it is assumed that the machine interacts with the environment using sensing and actuation [3]. (b) Initial definition of the navigation problem. The four basic elements that must be considered to develop an AMR [4]. (c) Detailed example of the interaction between the four basic elements of navigation [5].

World Perception

To predict appropriate paths, the AMR must be able to extract information from the environment. This can include dynamics agents, along with their respective position, dimensions, and kinematics, as well as static obstacles such as walls [92]. A variety of sensors can be used to obtain information about the surroundings, and each can provide a different form of output for the information. Common exteroceptive sensors include [93]: optical, aural, and tactile

Accurate spatial extraction and identification of pedestrians within the AMR's immediate vicinity is required. To accurately assess the environment, detect obstacles and their movement, and identify individual objects, a sensor based on the light spectrum is ideal. Laser range-finders provide excellent low noise obstacle detection, whilst camera based stereo vision provides depth perception, as well as the ability for object recognition due to the high spectrum of visible wavelengths. Multisensor fusion [94], such as combining a camera with a laser, can provide increased accuracy for localisation problems [95].

Path Planning

The type of path an AMR can plan not only depends on the path planning objectives, but also on what information the sensors have acquired from the environment and how it has been processed. Two forms of path planning exist, which include: local reactive object avoidance, and global preference-based behavioural path planning [92]. These are either mapless (reactive) or mapped (predictive) techniques, respectivly.

For AMR navigation both local and global methods are required [96]. However, global techniques should be used during this path planning stage and local techniques should be used in the *adaptive path control* stage. Global path planning is a preference-based method aimed at satisfying user preferences when designating a path to move along. It allows a complete path from start to goal to be created and have the AMR move in a specific manner. Therefore combining local path planning into the path control stage makes it excellent at correcting movement along a pre-determined global path.

Locomotion

AMRs can be designed to move within almost any environment. The path planning algorithm creates a path suitable for the environment that must consider how the robot can move. To solve the locomotion problem the mechanism and kinematics, dynamics and control theory of the robot must be considered [5]. Common locomotion systems include [97]: wheels, tracks, and legs for land. Propellers, and jets for air. Propellers, rotors, and paddles for sea.

It is assumed that the AMR for this thesis will be confined to an urban environment, the surface of which will be essentially flat, enabling extremely easy and versatile manoeuvrability. Therefore, during the path planning stage no additional strategy will be required in order to cope with any unknown terrain. Essentially any land based locomotion system can be implemented, as long as the platform's dynamics can allow the AMR to execute the planned path with a mobility similar to humans.

Adaptive Path Control

Successful global path planning is only the preliminary stage of an AMR, as the robot must then correctly move along that path. It is essential to ensure that the AMR moves along the path without causing any collisions, by adjusting for any errors in the model of the environment. As well as ensuring correct localisation, so that the AMR really does move along the planned path. Common path control techniques include [98, 99]: proximity fields and tactile sensors, for local collision avoidance. GPS, triangulation, landmark references, and dead reckoning, for localisation.

2.1.2 Dynamic Environments

For an AMR to navigate successfully its autonomy cycle, Fig. 2.1, must be appropriately devised depending on the environment it will be subjected to and its path planning objectives. A major difference between a dynamic and a static environment is the increased levels of uncertainty, relating to: identifying objects correctly; obtaining accurate agent positions; self-localisation; and predicting how the environment may change.

Within a dynamic environment an AMR must successfully interact with any other dynamic agents. Any form of interaction requires at least an *adaptive* level of autonomy [100, 101], Table 2.2, using sensors to detect and respond to other agents. Unlike pre-programmed robots that repeat a set pattern, interactive robots may change their behaviour based on the direct influence of elements around it. The challenge is how to programme the robot to interact *correctly*. Systems that incorporate multiple similar robots are highly effective and efficiently run, as the 'intentions' of the robots are known. It is the addition of humans that is a major hurdle towards successfully achieving adaptable and cooperative human-robot interactions [41, 102].

2.1.2.1 Mobile Robot Only Interaction

An AMR-only interactive system is typically one of swarm intelligence, involving multiple similar robots of simple design [103]. Using these simple robots, programmed with simple rules, organization[2] can occur and result in more difficult tasks being completed. Each robot is individually programmed with the same "rules", achieving their combined goal through simple communication methods, resulting in a decentralized system that has a high redundancy [104].

Alternative systems would typically be controlled by a centralized distribution computer network. The Kiva Robot system [54] is such an example, operating with high efficiency in the organisation of multiple robots simultaneously. All of this is achieved without any human input or supervision, however the robots must be confined to a *Human Exclusion Zone*. The system operates so effectively as all the AMRs have their paths planned for them by a centralised computer network. Therefore, the AMRs will never move along paths that conflict with paths given to other AMRs at any given time, and so collisions are avoided.

2.1.2.2 Human and Mobile Robot Interaction

The introduction of independent and less predictable dynamic agents, such as humans, will be far more challenging as an AMR must anticipate their movement and behaviour and react to them accordingly. The article: "Human-aware robot navigation: A survey" [105] was published in *Robotics and Autonomous Systems*, Volume 61, Issue 12 in December 2013, and provides a comprehensive collection of literature on human-aware robot navigation. When considering papers that explicitly deal with human-aware robot navigation the area begins to gather interest in 2001, and continues to grow. This coincides with the independent research undertaken for this thesis, as navigation within pedestrian environments began with RHINO the museum tour guide [36] in 1997, discussed later in Section 2.4.1. A total of 106 papers are reviewed, allowing for an insight to be formed into the current research outcomes of various aspects of human-aware navigation systems. The categories, challenges, and technologies the survey highlights are significant for human-aware navigation are listed below, along with the number of publications associated with each attribute.

c_1 Categories focused on:

- $c_{1.1}$ Comfort, **44**
- $c_{1.2}$ Naturalness, **38**
- $c_{1.3}$ Sociability, **16**
- $c_{1.4}$ Other, **19**

c_2 Specific challenges for robots navigating among humans:

- $c_{2.1}$ Follow a person, **8**
- $c_{2.2}$ Solve blocked passage and dense crowd, **4**

[2] can take many hours to organize correctly

- $c_{2.3}$ Guide person, **3**
- $c_{2.4}$ Move in formation, **3**

c_3 Technologies focused on:

- $c_{3.1}$ Pose selection, **14**
- $c_{3.2}$ Global planning, **33**
- $c_{3.3}$ Behaviour selection, **14**
- $c_{3.4}$ Local planning, **27**

The survey concludes that from the reviewed literature, the major human-aware capabilities that a robot should exhibit during navigation are:

d_1 Respect personal zones

d_2 Respect affordance spaces

d_3 Avoid culturally scorned upon behaviours

d_4 Avoid erratic motions or noises that cause distraction

d_5 Reduce velocity when approaching a person

d_6 Approach a person from the front

This is a list that concurs with independent research conducted for this thesis, specifically discussed in *Behavioural Collision Avoidance* (Section 2.2.3.1), *Path Planning Methodology and Interaction* (Section 2.3.1), and *Human-Mobile Robot Interaction* (Section 2.4).

A major element in dealing with these points is predicting what any pedestrian may do. When predicting pedestrians' future positions, two approaches exist: 1) Prediction based on reasoning; and 2) Prediction based on learning. For reasoning-based techniques, predictions are based on assumptions of how pedestrians behave in general. For learning-based techniques, predictions are based on observations of how pedestrians behave, especially in particular circumstances and within particular environments. Examples of prediction techniques for moving humans include:

e_1 Linear [106]

e_2 Likely state transitions in a grid [107]

e_3 Using potential field with walls as repulsors and a perpendicular (dashed) line in front of the person [108]

e_4 Growing uncertainty [109]

e_5 Using random samples and stochastic weights [110]

Fig. 2.1b	Table 2.2	Task
World Perception	Sensors	Spatial data collection of environment
	Perception	Individual pedestrian identification
Path Planning	Modeling	Spatio-temporal predictions of individual pedestrian movements
	Planning	Global planning over prediction time-horizon
Locomotion & Adaptive Path Control	Task Execution	Select most ‘human-aware’ path
	Motor Control	Translate path prediction into real-world movement
	Actuators	Move AMR along path

Table 2.3 A collation of the standard autonomy cycle for an AMR, Fig. 2.1b, and the dynamic stages required for an AMR, Table 2.2. A statement of what task must be performed is also included.

e_6 Social Force model, person attracted to goal, to walking partner, but also repulsed by partner when too close [111]

e_7 Using library of observed paths [112]

e_8 Using closest frontal annotated waypoints on map [113]

It can be assumed that to accurately navigate a pedestrian environment an appropriate pedestrian model is required. Accurately predicting human behaviour is a means to predict a suitable path that will conform to all the navigation requirements in List *d*.

Table 2.3 provides a comparison between the basic autonomy cycle, Fig. 2.1b, and the dynamic stages of an AMR, Table 2.2. As both the modelling and planning form part of the dynamic stages the AMR can be classified as "intelligent". The autonomy cycle for the AMR would therefore require a modelling and path cost assessment stage when path planning, the data flow of which is represented in Fig. 2.2. Various path planning strategies could be implemented, and so an evaluation of various forms is undertaken in the next section.

2.2 Strategies for Navigation

To create an effective AMR navigation strategy both global path planning and local collision avoidance must be used to implement the system as a whole [96]. The autonomy cycle, Fig. 2.1b, makes it clear that global and local navigation techniques are required for the path planning and adaptive path control sections, respectively. Although *local* and *global* methods are both path planning strategies, local methods are generally implemented for the assessment of immediate collision avoidance,

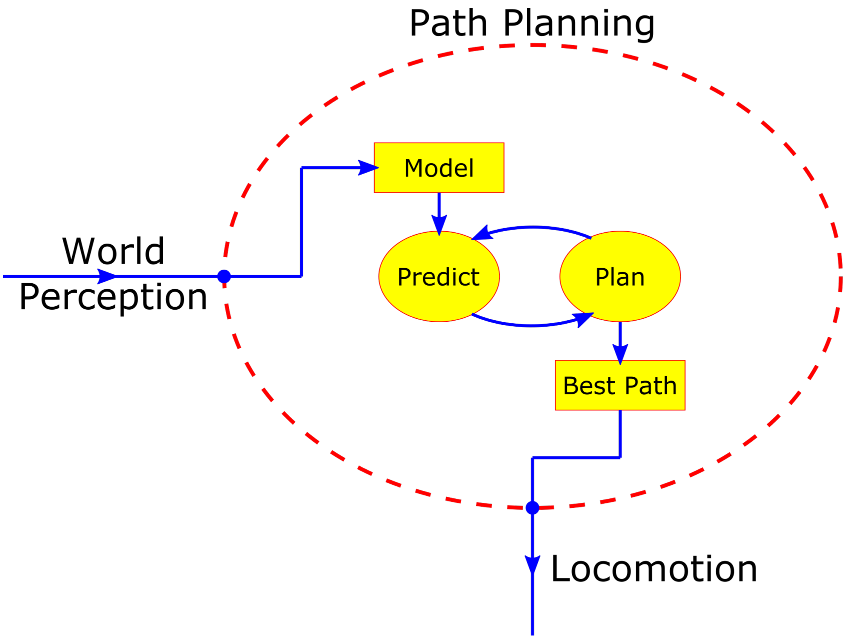
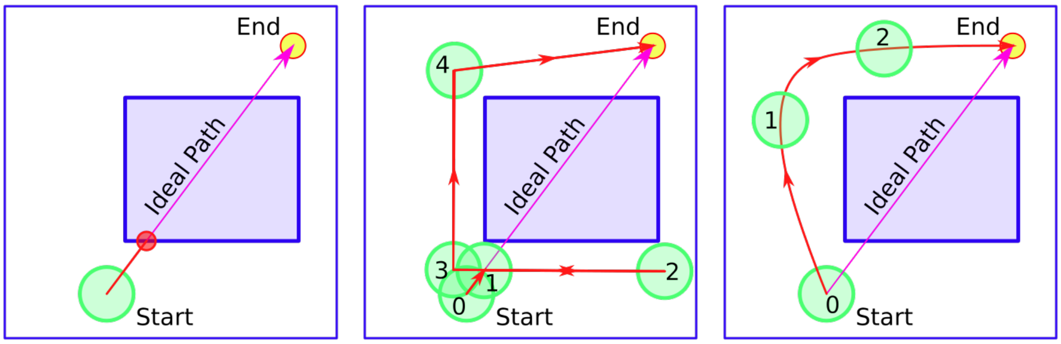


Fig. 2.2 Example of an additional predictive element added to the path planning stage of an AMR's autonomy cycle, Fig. 2.1b. This replicates the "tactical" level of a pedestrian's path planning choices, Fig. 2.9 discussed in Section 2.3.1.2.

whilst global methods plot a path from start to finish [114], "as local planning approaches are generally not able to solve complex interaction situations in a consistent way due to the limited look-ahead". [7] However, local path planning has benefits in: processing speed, navigation within unknown environments, and ease of implementation; whilst global path planning has benefits in: finding the optimal path, allowing long-term behaviour to be implemented into the path, and anticipating potential collisions. The effects of path planning for a simple object avoidance scenario is demonstrated in Fig. 5.16.

2.2.1 Environmental Maps

The first stage of a path planning strategy is to map the AMR's environment. A 2D plan view traditionally represents whatever can be detected by the AMRs sensors, with any objects typically either be static, such as walls, or dynamic, such as pedestrians. The objects can be mapped onto occupancy grids, Section 2.2.1.1, where the plan view is divided into cells, and if an object fills a cell it is occupied, otherwise it is left empty. Alternatively the objects can be represented as geometric shapes and a roadmap, Section 2.2.1.2, can be formed that represents how available space in between obstacles can be connected, helping enhance certain path planning attributes [115].



(a) (b) (c)

Fig. 2.3 The three forms of collision avoidance that are involved with path planning. The robot is represented by a green circle, whilst the number is relevant to its progression along the path (larger number ⇒ further along in time). (a) No collision avoidance: The robot would immediately crash into the object ahead (red circle). (b) Local collision avoidance: The robot would respond to the object as it is about to collide. The robot moves right as that is the direction with the shortest distance around the object, to the end point. As the gap between the edge of the environment and the object is too small to fit through, the robot turns back and moves the other way around. (c) Global collision avoidance: A path is planned before the robot moves. The path extends from the robot's start to the end point, and is free of collisions. The robot can now move to the end point without having to reassess another path, and can do so with the greatest level of efficiency.

2.2.1.1 Occupancy Grids

The intention of this method is to: "account for all of the free space" [6]. Free space is partitioned into a collection of cells, and a connectivity graph is constructed by connecting cells that share common boundaries [116].

Fixed Cell Decomposition This cell decomposition method [116] (Fig. 2.4a) is excellent for systems with restricted processing speed, as unlike roadmaps such as the VD or VG, or the vertical cell decomposition method, not everything must be processed. Regardless of the environmental complexity a level of cellular decomposition can be found. However, some complex shapes may not be described appropriately as a grid of fixed resolution is applied over the plane, which segments the environment into smaller sections that may not incorporate high levels of detail.

Quadtree Cell Decomposition This cell decomposition method [117] (Fig. 2.4b) provides an expansion to the approximate method, where consecutively smaller resolutions are achieved. This rectifies the fixed resolution problem of fixed cell decomposition, as a grid is applied over the plane. This is then subdivided into smaller cells over areas where more detailed information can be extracted until either all details have been appropriately considered or the time constraints of a deterministic systems have expired. This is very effective for deterministic systems that can utilise the increase in resolution through parallel processing.

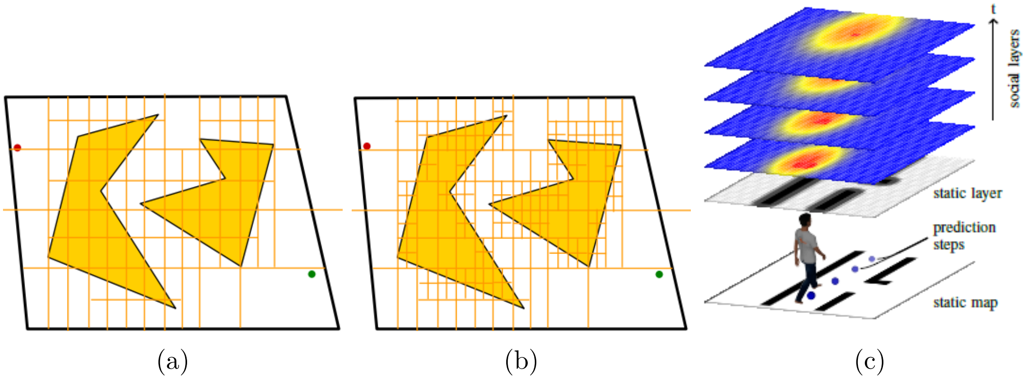
Costmaps In a standard occupancy grid the normalized state of a cell (0/1) represents the presence of some objects' position, and prevents a path being planned that enters an occupied cell (1). This limits a path planning algorithm by assigning all free cells as equally valid (0), regardless of their position or orientation to the object. For a human-aware path planner this can result in a path that does not respect the requirements of socially acceptable behaviour, stated in List *d*. Rather than assigning discrete values for unoccupied/occupied cells, their values could instead range anywhere between these two values based on the cost, x, of planning a path through them, where $\{x \mid x \in \mathbb{R},\ 0 \leq x \leq 1\}$. The cost attributed to each cell could be based on anything, such as: proxemics surrounding pedestrians; dangerous areas; traffic flow direction; etc, allowing a path to be planned that responds more appropriately to different scenarios.

Traditionally a costmap stores the costs in a singular grid of values, which would combine all the various costs together in a "monolithic" costmap. However, by storing all compiled cost data in a single data structure the origins of the information, and what each value represents, would be lost. An alternative is a layered costmap [118] that maintains an ordered list of layers that each represent a certain attribute, e.g. proxemics, obstacles, caution zones. Each individual layer can track data related to a specific functionality, maintaining the context of the information and making updates much easier. Layers of the costmap can also represent the dynamics of a moving object, such as dynamic social costs of pedestrians [7]. Each layer represents the social navigation constraints for one prediction time-step, Fig. 2.4c, allowing a path to be planned that considers where the pedestrians will also be.

An initial static map can be created, in the form of a traditional occupancy grid, and then any number of layers can be added to influence the behaviour of the AMR when path planning. However, too many layers may lead to too much overall cost, and a path planning search may not be able to find a route. This may lead to the freezing robot problem [69] where a safe path cannot be found and the AMR does not move, as the more layers produced the higher the overall costs.

2.2.1.2 Roadmaps

Using roadmaps is an excellent method to explore the environment in a host of various ways. The intention is to: "reduce the N-dimensional configuration space to a set of 1-dimensional paths to search" [6]. A number of nodes are positioned within the environment, which forms a connectivity graph. These nodes in the graph can then be connected in whatever way is desired, with the various node connections



(a) (b) (c)

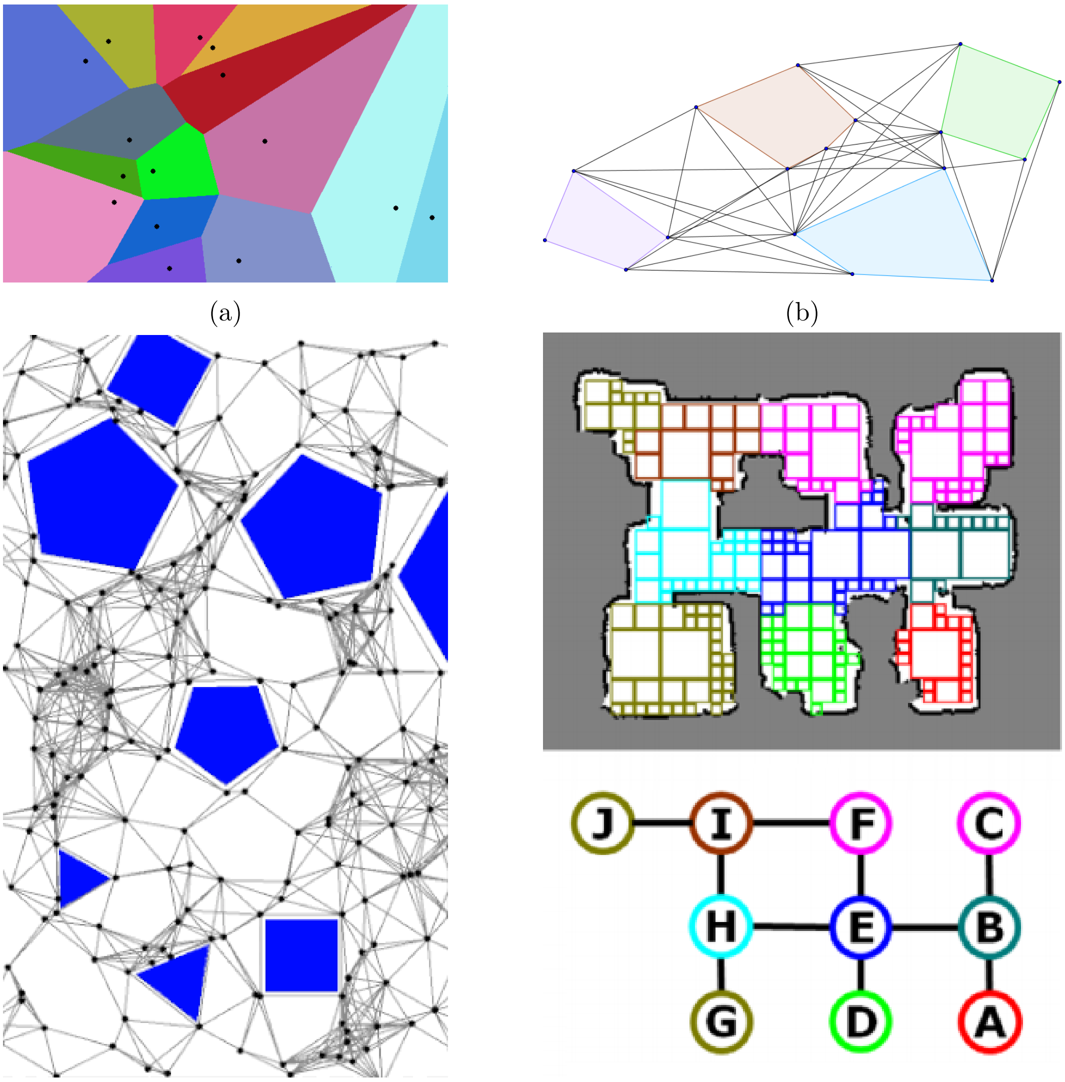
Fig. 2.4 Examples of occupancy grids. (a) Fixed (Reproduced from [6]). (b) Quadtree (Reproduced from [6]). (c) Layered Costmap that uses social proxemics (Reproduced from [7])

making the resultant roadmap highly customizable [119]. The traditional and most used roadmaps, Fig. 2.5, include:

Voronoi diagram This roadmap (Fig. 2.5a) [120] is formed from a set of points that are equidistant between neighbouring objects. The VD therefore produces the *safest* path, which is always simultaneously the farthest from all neighbouring objects. Although this can mean that for sparsely populated environments large detours may occur, for more densely populated environments the VD provides the best chance at searching for a collision free path.

Visibility graph This roadmap (Fig. 2.5b) [121] is suited for polygonal obstacles, nodes represent points on the polygons, and edges form between two visibly connected nodes. The roadmap created allows the shortest path to be calculated, without requiring any unnecessary detours. However, no consideration is made towards a safety margin surrounding the objects, which can make potential collisions more likely as paths pass very close to neighbouring objects.

Probabilistic roadmap This roadmap (Fig. 2.5c) [122] does not allow uniform exploration of the environment, and consists of connections between randomly created points that populate the configuration space. Therefore it is highly likely to avoid small details of the environment, e.g. being unable to describe narrow passages. It is also likely to repeatedly sample large open areas where no additional description of the environment is required. A roadmap is created with no specific structure, and cannot divide the environment to describe certain desirable traits, such as the equidistant or shortest connections between objects, like the VD or VG, respectively.



(a)

(b)

(c)

(d)

Fig. 2.5 Examples of roadmaps (Section 2.2.1.2). (a) Voronoi diagram . (b) Visibility graph [8]. (c) Probabilistic roadmap [9]. (d) Topological map [10].

Topological map This roadmap (Fig. 2.5d) [123] is more conceptual and offers a more abstract method of mapping, linking points based on distance and orientation through a connection of recognizable places. A local metric map is applied so when the robot is at a distinctive place local sensory information and orientation is assessed. When the robot moves, the position is described in relation to where it is coming from, the distance travelled, lateral distance information, and current orientation. This form of navigation could be deemed similar to human navigation [124], using landmarks to manoeuvre between places. However the topological map only ensures correct localisation for one map, without providing distinct characteristics of obstacle clearance, as with the others.

2.2.2 Path Planning

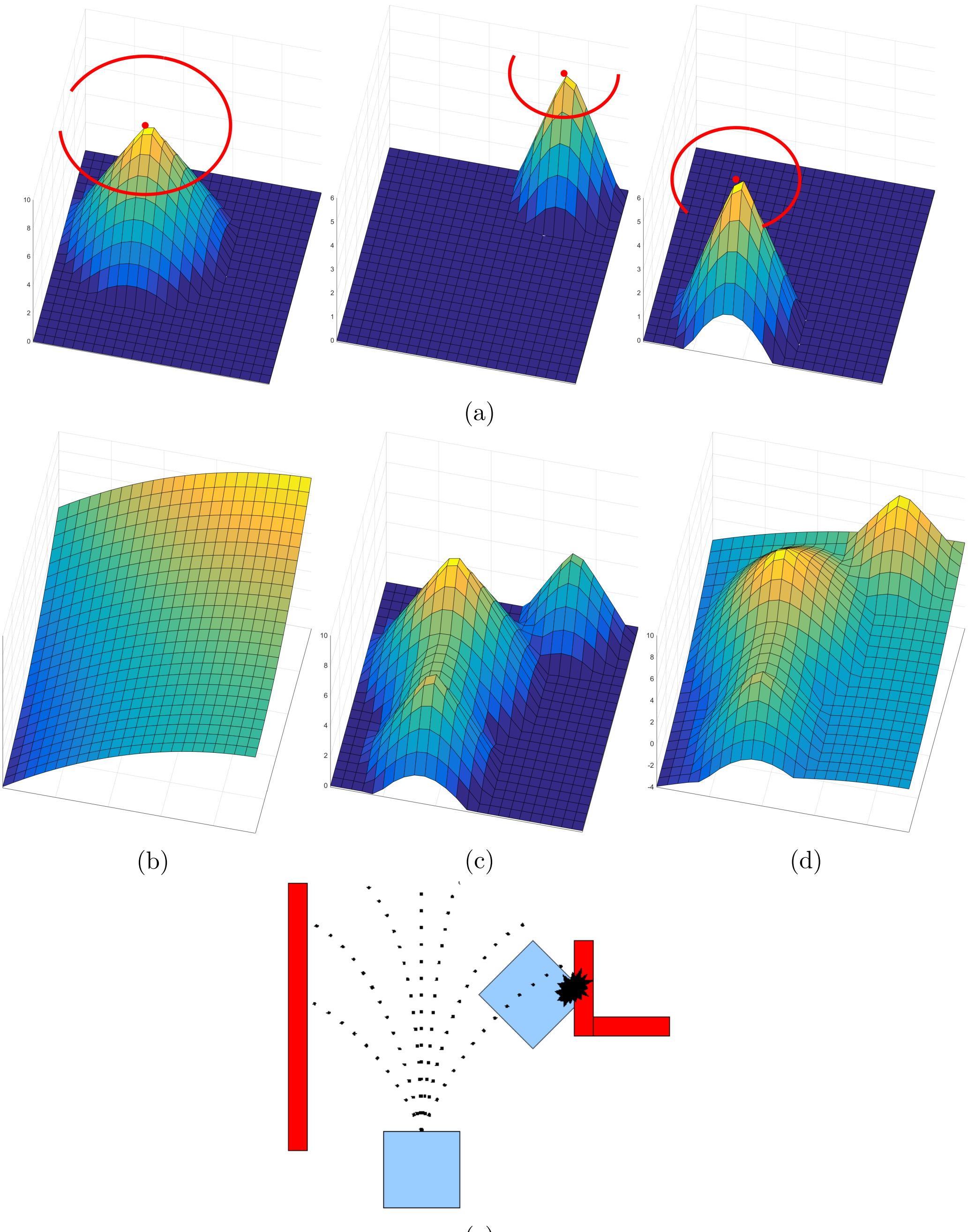
When calculating paths only two spatial dimensions are used, formed from a cross-sectional plan view of a three dimensional environment. Local techniques are quick reactive approaches that respond only to the robot's immediate environmental obstacles, without plotting a path all the way to the robot's goal. Global techniques employ search algorithms that attempt to find a path through the entire environment all the way to the robot's goal, which can allow more sophisticated paths to be plotted via proactive obstacle avoidance. These methods are now discussed regarding their effectiveness towards path planning in dynamic and uncertain environments.

2.2.2.1 Locally Reactive Movement

Local path planning techniques are designed to move an AMR directly towards its goal, avoiding collisions as they occur en route by reacting to the immediate presence of any obstacles. Collision avoidance was originally considered a high-level planning problem (1986), when artificial potential fields [125] were developed, allowing real-time robot operations in a complex environment. Alternatives to potential fields include: virtual force fields [126], vector field histograms [127], and the dynamic window approach [128], Fig. 2.6e, which all generate movement using velocity potentials of the AMR.

Artificial Potential Fields The method is very simple to implement, requiring only two opposing elements: 1) An *attractive* potential is implemented at the goal, drawing the robot towards it; and 2) *Repulsive* potentials are situated around the obstacles to avoid, repelling the robot away from them, Fig. 2.6b. The gradient of the combined potentials creates a path that the robot can move along. However, due to these interacting fields local minima traps can occur [129], which create an area of no potential that the AMR can become stuck in.

As this method is *reactive*, only plotting and moving along a path based on the immediate environment, the processing speed of local collision avoidance is very fast, allowing real-time implementation. Depending on the literature they have been considered as either global [130] or local [131]. However, artificial field methods traditionally rely on reactive means of collision avoidance, rather than mapping the entire environment [132]. For whatever purpose the fields are used for the same characteristics still emerge, but on slightly different scales [133]. Due to this the reactive nature of local path planning is limited for autonomous navigation as: "a reactive agent running on a deterministic program would be considered autonomous, but since such an agent has no intentions, and is incapable of introspection, autonomy seems to be a concept of limited usefulness in the context of strictly reactive agents." [85].



(a)

(b) (c) (d)

(e)

Fig. 2.6 Examples of local reactive path planning techniques (Section 2.2.2.1). (a) For three circular objects, repulsive potential fields surround their central points and add value to the cells of an occupancy grid, with a higher cost (y-axis) the closer to each object centre. (c) An attractive potential field gradually decreases the cost of cells from the top-right start position of the AMR to the bottom-left goal. (d) An overall potential field map is produced by combining the two fields (a and c), for a path to be found by constantly moving to an area of lower potential/overall cost. (e) A number of trajectories are plotted that accurately replicate the AMR's dynamics for the dynamic window approach. Each potential trajectory incorporates information about the AMR's acceleration and turning potential, and current velocity, so that realistic trajectories can be produced.

Dynamic Window Approach This approach [128] was developed for one of the earliest AMRs to move within a crowd [36], discussed in Section 2.4.1, and adaptations have been used in many since (e.g. [41, 134]). Unfortunately, like potential fields this reactive approach can still become trapped in local minima. However, the technique works well as the planned movement is directly related to the AMR's mobile dynamics, using a "dynamic window" that consists of the velocities reachable within a short time interval.

Firstly, the unreachable velocities are eliminated due to the AMR's acceleration limits. Secondly, a combination of translational and rotational velocity pairs that cannot stop before colliding with obstacles are eliminated. Finally, an optimum velocity pair is chosen from the remaining candidates, Fig. 2.6e, by maximizing the objective function. This is a measure of the AMR's progress towards the goal, the forward velocity of the robot, and the distance to the next obstacle on the trajectory. The approach predicts the results of each velocity pair candidate in terms of final heading angle, minimum distance to obstacles, and linear velocity values. Therefore, real-time collision-free motion trajectories can be predicted and executed.

2.2.2.2 Globally Proactive Searches

Search algorithms are a cornerstone of global path planning. They allow for routes to be planned via graph/tree searches through various configuration space solutions of node networks formed from occupancy grids, Section 2.2.1.1, or roadmaps, Section 2.2.1.2. Traditionally AMRs would travel through static environments, and the preferred path would be the one with the shortest distance to the goal. However, for an AMR to move in real-time within a dynamic environment often heuristics are added in order to increase processing speed and find appropriate solutions within an expected time limit, sacrificing optimality, completeness, accuracy, etc. These heuristics could also be utilised in order to modify the type of paths the AMR takes, and generate paths that could be deemed "socially acceptable". Searches can be either 1) Depth-first, where space and time complexity is $O($longest path length searched$)$ and $O(b^d)$, respectively; or 2) Breadth-first, with a space and time complexity are $O(b^d)$ and $O(b^d)$, respectively, where for both cases b is the branching factor and d is the maximum depth. Both forms of search can find global solutions[3] using a brute-force algorithm, guaranteed to find the best path possible by evaluating every single path combination it is possible to make within the network of nodes. Unfortunately for that very reason it is also the least efficient algorithm possible. Path planing search algorithms are often optimised by including a heuristic element:

[3] assuming a finite branching factor, b

$$f(n) = g(n) + h(n) \ . \tag{2.1}$$

where

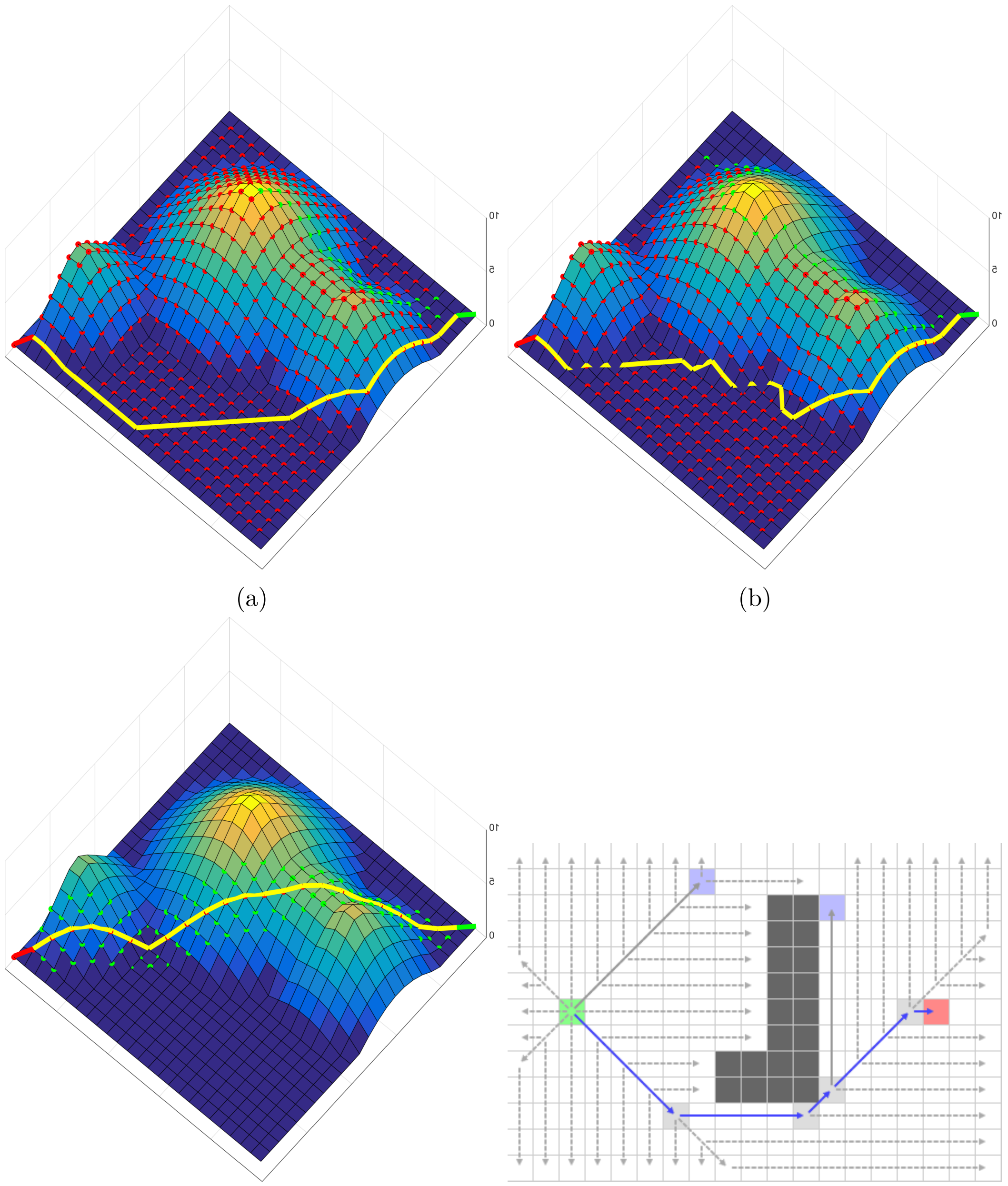
$$\begin{cases} f(n) = \text{the evaluation function} \\ g(n) = \text{total cost from the start node to the current node} \\ h(n) = \text{heuristic to encourage a search that moves toward the goal} \end{cases}$$

Dijkstra's Algorithm This algorithm [135], Fig. 2.7a, has been a classical approach to path finding for over half a century, employing a breadth-first search from the source node, located at the AMR's current position. If the node network is unweighted the front-end of the search (the set of nodes available for expansion known as the "frontier" [136]) will appear as a wavefront moving outwards along all connected node vertices uniformly, finding a global solution of the least number of vertices traversed. As the algorithm expands in all directions, solutions to multiple destinations can be found. The algorithm may also require exponential memory growth if the environment is unbounded.

For Dijkstra's algorithm each node is weighted based upon whatever cost-function is applied, resulting in a uniform-cost search that finds a global solution of a path with the lowest weight. The cost function for robot path planning is movement related, and to encourage an AMR to move along certain types of paths a larger weight is applied to nodes in areas that are undesirable and smaller weights in more preferable areas. This will result in the frontier appearing to move between vertices at different rates: faster in more preferable areas and slower in others.

The algorithm does not have a heuristic element and only includes the cost $g(n)$ of the optimal path from start to n, i.e. $f(n) = g(n) + 0$. However, priority queues can be used in order to reduce the operation time of a naive implementation from $O(n^2)$ to $O(n \log n)$ [137]. An alternative implementation is the bidirectional search, whereby the algorithm is simultaneously executed forward from the source node and backward from the goal node [137]. The search is terminated as soon as a node is visited from both directions, and the overall path is stitched together from the front and back segments.

Best-First The A* algorithm [138], Figs. 2.7b and 2.7c, is an extension of Dijkstra's algorithm that uses a best-first search heuristic, optimised for a single destination. In its simplest form the heuristic is a Euclidean distance from the current search location to the goal. Rather than expanding the search along all connected vertices, only the vertices connected to the best node, $\min(f(n))$, are used: i.e. $f(n) = g(n) + h(n)$, where $g(n)$ is the actual cost of an optimal path from start to n, and $h(n)$ is the actual cost of an optimal path from n to a preferred goal node of



(a)

(b)

(c)

(d)

Fig. 2.7 Examples of global path planning search algorithms (Section 2.2.2.2). (a) In Dijkstra's algorithm the frontier expands equally in all directions simultaneously, increasing it exponentially until eventually it reaches the goal node. (b) A Euclidean distance to the goal heuristic of the A* algorithm initially directs the search in a straight line towards the goal. When the search hits the obstacle it explores alternative routes through nodes from the open set (green dots). The search area expands until eventually it moves sideways enough to pass the obstacle, and continue almost directly toward the goal. (c) An example of an A* search finding a sub-optimal path by using a heuristic 10 times greater than the admissible heuristic limit, therefore overestimating the cost of reaching the goal. (d) The jump point modification of the A* search reduces the number of nodes explored along the optimal path from 13 to 4. Jumps are made to nodes that are either vertically, horizontally, or diagonally visible from the current search node.

n [138]. This informed search is facilitated by the heuristic in order to expand nodes that are estimated to be closer to the goal. To find an optimal path, Fig. 2.7b, the heuristic must also be "admissible", by which it does not overestimate the cost of reaching the goal, otherwise a sub-optimal path will be found, Fig. 2.7c. If the cost of the optimal path from start to n is not factored, the algorithm becomes a greedy best-first search, i.e. $f(n) = 0 + h(n)$, which is far from optimal as it can become stuck in loops.

As A* is a best-first search, at any given time the 'best path' has been discovered, making it excellent for a deterministic system. The algorithm is very popular in a number of robot path planning challenges (e.g. [139–141]), as it reduces both the memory requirements and increases the processing speed due to a much smaller search area. However, if the path to the goal is not very direct then it has the potential to require exponential memory, similar to Dijkstra's algorithm. The popularity of the A* algorithm is apparent due to a large number of optimisations that have further been developed for it, such as:

Simplified Memory Bounded: This modification [142] will produce the same optimal path as the original A* algorithm, even if the available memory is not sufficient to store all branches of open nodes. If there is no memory left, the nodes with the highest cost $\max(f(n))$ is removed from the queue. The algorithm must store the cost of the best forgotten child node, $f(n+1)$, of the parent node, $f(n)$. If all explored paths are worse than any forgotten paths the path can re-generated from the saved data.

Iterative Deepening Depth-First: This modification [143] can be applied to searches in order to limit the memory requirements, offering a depth-first search space-efficiency and breadth-first search completeness. Using the same cost, $g(n)$, and heuristic, $h(n)$, as the standard algorithm a depth-first search is performed, cutting off a branch when the total cost exceeds a given threshold, t_{max}, i.e: $f(n) = g(n) + h(n) > t_{max}$. As A* potentially stores an exponential queue of unexplored nodes, memory can fill up quickly. However, the algorithm differs from the standard A* as a depth-first search requires only linear memory, storing only the nodes on the current path. Unfortunately due to this the search may revisit the same node whilst exploring various paths.

Jump Point: This modification [144] is designed to increase the processing speed of the A* algorithm by making assumptions about a current node's neighbours on a uniform grid. Unlike the standard A* that expands its search along all connected neighbouring cell nodes, this algorithm jumps either vertically, horizontally, or diagonally to nodes that are further away. To make a jump the destination node must be visible from the the current node. Although the jump will cost more, there will be fewer of them as the search jumps across any neighbouring nodes between the current and destination node, Fig. 2.7d. This modification claims to have very few disadvantages: "it is simple, yet highly effective; it preserves optimality, yet requires no extra memory; it is extremely fast, yet requires no preprocessing" [144].

Incremental Heuristic The A* algorithm is a very efficient informed best-first search. However, the heuristic element that guides the search direction is pre-defined before execution. Whilst this is fine for a static environment, within a dynamic environment the search requirements may be required to change with it.

The generalized adaptive A* algorithm [145] updates the heuristic $h(n)$ used in a previous search, so that the next search is more informed, e.g. a moving goal. The lifelong planning A* algorithm [146] reuses information from previous searches so that the path can be quickly updated, e.g. moving obstacles. The initial search is a standard A*, however consequent searches repeatedly recalculate the shortest paths between start and goal node as the edge costs $g(n)$ of a graph change. The D* Lite algorithm [147] is based on the lifelong planning A* algorithm. The initial search is again a standard A*, which originates at the goal node and terminates at the AMR's node. However, the second search of Lite expands only a subset of those cells whose goal distances changed or have not been calculated before. This makes the algorithm excellent for unknown environments as the path only need be updated if the AMR identifies a hitherto unknown obstacle blocking the path.

Dynamic weighting, $w(n)$, is another heuristic modification that can be performed on-the-fly [148]: $f(n) = g(n) + w(n) * h(n)$, where $w(n) \geq 1$. This could be used to change the AMR's necessity to reach the goal as directly, e.g. if the environment becomes more densely populated with dynamic agents the weight can be decreased so that actual cost of the path $g(n)$ becomes more important. This can allow more consideration to be paid to how the AMR interacts with the moving agents, as the planned path will focus more on their costs.

2.2.3 Collision Avoidance

Dynamically reacting to any immediate collisions is inherent in all local path planning techniques, Section 2.2.2.1, whilst global path planning relies on the proactive search for collision free paths, Section 2.2.2.2. Therefore, collision avoidance strategies are required for an AMR to make adjustments as it moves along a global path prediction, due to any prediction uncertainty in a dynamic environment. Collision avoidance is also required in static environments in case any localisation errors occur as the AMR moves.

A basic collision avoidance system would treat all obstacles the same. For static obstacles only a simple geometric proximity check is required, whilst for dynamic obstacles a surrounding "buffer zone" [149] should be added to account for any uncertainty of sensor data. As this thesis focuses on navigation within a pedestrian environment, proxemics must be incorporated in order to factor in the amount of space that people feel it necessary to set between themselves and others, discussed later in Section 2.3.1.1. Pedestrians also expect certain navigational styles to be adhered to, discussed later in Section 2.3.1.2, which the literature often regards to as "social navigation".

2.2.3.1 Behavioural Collision Avoidance

The behaviour an AMR exhibits when moving is essential in order to avoid collisions, as it helps other dynamic agents predict the AMR's response to a upcoming potential collision [67, 139]. For an AMR to navigate in a particular dynamic environment it should also adopt the same behaviour as the other dynamic agents. When considering these dynamic agents as pedestrians, the closer an AMR passes the more uncomfortable the pedestrian becomes, especially for non-tech groups. Therefore, a static proxemics field can be introduced around each pedestrian [150], in order to influence how the AMR moves within the pedestrian's immediate vicinity. By surrounding a pedestrian with a dynamic circular costmap, the relative motion prototype [151] constantly updates the "high-cost radius" over time in order to produce a path for the AMR that resembles human motion behaviour. However, these results are compared to a circular proxemics field rather than an elliptical one [150] which should be elongated along a pedestrian's velocity axis, discussed later in Section 2.3.

Passing a pedestrian on a corridor requires the aforementioned "social navigation", as pedestrians attribute to the AMR: "a rational mind similar to their own, capable of making decisions and interacting in traditional human ways" [68]. This behavioural collision avoidance extends beyond geometric proxemics and relies on the optimistic assumption [67] that the pedestrian expects the behaviour of the AMR to be the same as theirs. Although pedestrians do statistically behave in certain ways, discussed later in Section 2.3.1, when calibrating the parameters of the collision avoidance criteria for an AMR there are not always readily-found values that result in the desired behaviour [68]. Especially as humans will interact differently towards robots than other humans [134].

Collision avoidance strategies also assume just that: collisions will be avoided. However, these behavioural collision avoidance strategies focus on individual collisions between one AMR and one pedestrian. In an environment with multiple pedestrians the AMR may be in an inevitable collision state [152], whereby a collision is suddenly unavoidable. Although global path planners search to avoid these potentialities, they may also be impossible in an overcrowded environment. Therefore, overcrowding may lead to the freezing robot problem [69], a path that may have to be planned requiring obstacles to be moved if a clear path is not available [153], or a "minimum-conflict" path [154] where a path the the minimal number of collisions is returned , addressed in Section 2.4.2 when evaluating human-aware path planners.

2.3 Pedestrian Crowds

> "Crowds are the elephant man of the social sciences. They are viewed as something strange, something pathological, something monstrous. At the same time they are viewed with awe and with fascination. However, above all, they are considered to be something apart." [155]
>
> (Professor Stephen D. Reicher)

This quote summarises the difficulties in how one can successfully evaluate a crowd. The movement of a crowd, to the movement of pedestrians, to the individual interactions between people and their individual psychology, makes it difficult to accurately anticipate how a pedestrian will behave on their own, or within the presence of others [156]. Unpredictible pedestrian behaviour also extends to social navigation for AMRs [157], whilst "predictable behaviour" of humans is only achieved for simple and limitied movements, such as arm movement exercises [158].

However, by reducing the elements to focus upon, and simplifying the interactions that occurs, movement models can successfully predict crowd movements with moderate success. Human crowds can be described in many different ways, depending on what kind of crowd is being assessed [159]. In order to plan paths within crowds, appropriate models must be used in order to describe motion, as well as behaviours, so that the AMR can move accordingly [55].

2.3.1 Path Planning Methodology and Interaction

The manner in which a pedestrian behaves and interacts with others and their current environment, depends on factors concerning [55, 160]: the environment (present and future); population density; gender; and personal preferences of individuals. Such a large number of contributing factors makes it difficult to successfully predict how an individual will behave. For example, certain environments or age groups will express more likely behaviours, whilst the customs and traditions of individuals will potentially limit the number of possible outcomes [160]. It is however difficult to predict how everyone will react in a given situation as pedestrians exhibit free will.

Despite being able to potentially estimate movement through these stereotypes, there are many more parameters concerning the complex nature of the individual, including [55]: will; communication; fitness; geographical knowledge, many of which are conflicting. Pedestrian route choice and walking processes are mostly subconscious, however it can be assumed that the route choice is based on utility optimization [12] with the desirability of a route dependent upon factors such as:

f_1 Distance or travel time between origin and destination, which is applied to all environments

f_2 Proximity of obstacles or other physical obstructions; closeness to walls, which is applied to basic maps, such as static environments

f_3 Stimulation of environment, and attractiveness (e.g. ambience conditions, shopping windows, shelter in case of poor weather conditions). Factors that are determined by the individual pedestrian preferences

f_4 Number of sharp turns and rapid directional changes (route directness), which is applied to path planning optimisation

f_5 Expected number of interactions with other pedestrians (level-of-service), relevant to a pedestrian only environment

Empirical studies have shown these are not mutually consistent, whilst their importance will vary between each pedestrian. To facilitate the development of the AMR, these factors can be reduced to consider a pedestrian only environment. The walking behaviours and interactions of individual pedestrians within a crowd can be assumed and summarised as the desire to achieve the following, for Item f_1:

g_1 Arrive at their destination with a certain time (e.g. [161])

g_2 Move at a constant preferred speed as they walk (e.g. [14, 162])

and for Item f_5:

h_1 Avoid collisions with other pedestrians (e.g. [73, 163])

h_2 Find the easiest route through the crowd to their destination (one that involves the least potential collisions) (e.g. [162, 164])

These points are the cornerstone of a pedestrian's navigation in a pedestrian-only environment [165], with the other points removed for the sake of simplifying the number of unknown variables.

2.3.1.1 Proxemics

The most desirable trait would be for a pedestrian to avoid any direct collisions, as well as to a lesser extent prevent violating the personal space of others [23]. Table 2.4 provides an example of the commonly desired spatial ranges that pedestrians prefer to remain between each other, dependent upon their relationships. For a scenario that involves walking in a public environment it can be safely assumed that the distance between individuals will be within a minimum of the "personal" zone, as "close" and "intimate" zones are reserved primarily for stationary embraces. However, groups of acquaintances moving together often walk within these proximities [43]. In order to best avoid collisions, violation of the 'personal' zone should be avoided, which should also help prevent collisions with groups. Any gap between two pedestrian of less than the 'personal' zone may mean cutting through a couple

Spatial Zone	Range	Situation
Close Intimate	0 - 0.15m	Lover or close friend touching
Intimate Zone	0.15m - 0.45m	Lover or close friend only
Personal Zone	0.45m - 1.2m	Conversation between friends
Social Zone	1.2m - 3.6m	Conversation to non-friends
Public Zone	3.6m+	Public Speaking

Table 2.4 Proxemics for human-human personal spatial zones [22, 23]. The desirable zones regarding the proximity of one human to another. Study carried out on mainly urban English-speaking parts of the world.

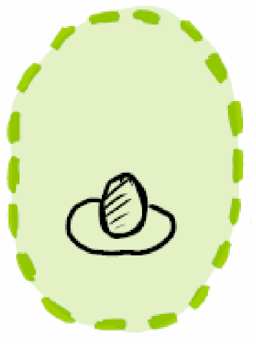


Fig. 2.8 Elliptical proxemics distribution around a human, elongated in front of them (Reproduced from [11]).

of friends, but also will create a higher collision potential on both sides of the pedestrian cutting through. For a moving pedestrian proxemics are often represented with an elliptical distribution, Fig. 2.8, due to their current velocity. This is observed in empirical studies discussed later in Section 2.3.2.4.

2.3.1.2 Navigation

Crowds can be unpredictable due to the heuristic methods pedestrian use to plan their paths, which are excellent for complicated scenarios as it aims to improve strategies in order to create better decisions [166]. Humans strategically plan a route to their ultimate goal, but navigate along this route using a heuristic approach [167]. The "Categorization by Elimination" heuristic [168] uses successive visual cues to reduce the set of possible categories objects may belong to until only a single category remains, so that only necessary information is processed (e.g. when a pedestrian walks along a pavement the heuristic eliminates everything except what is infront of them on the pavement. For instance, cars on the road are of no concern until the pedestrian desires to cross it, in which case heuristic will eliminate everything except the cars passing and the distance required to cross the road. Once the pedestrian has crossed, the heuristic eliminates everything except the upcoming pavement, as before. This highlights the fact when undertaking a specific task, a human will be very selective on what information to process, using only the minimal information required. Complex behaviour can occur when responding to objects based on only a few simple rules, which has been previously utilised for crowd modelling [169]. How a pedestrian chooses to behave when path planning is split into three levels [12]:

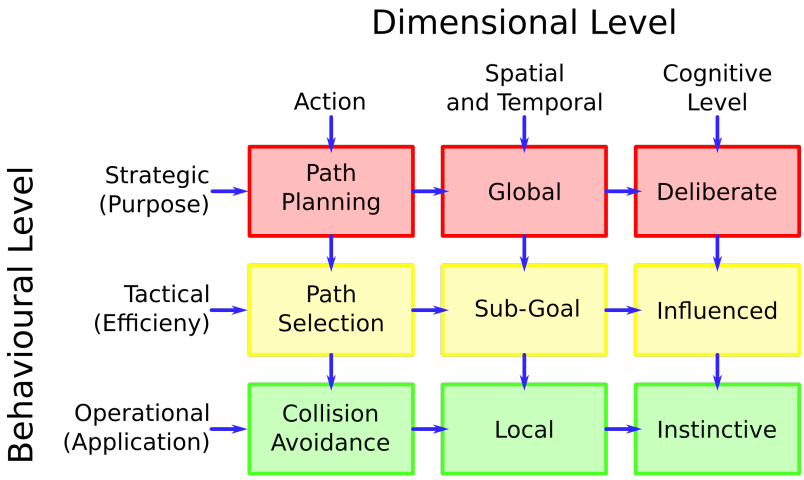


Fig. 2.9 Visualisation of the pedestrian path planning choices. Partially derived from the text in [12], and inspired by the basic framework for applying a social force model found in [13]. The diagram shows the three behavioural levels of how a pedestrian plans their path, along with the three dimensional levels of how a human operates. Within the "Dimensional Level" the "Action" level taken by pedestrians is the same as the AMRs' general autonomy cycle, Fig. 2.1b. Also, the "Spatial and Temporal" level employs the same navigation strategies as an AMR, Section 2.2.

i_1 **Strategic level:** Departure time choice, and activity pattern choice.

i_2 **Tactical level:** Activity scheduling, activity area choice, and route-choice to reach activity areas.

i_3 **Operational level:** Walking behaviour.

Fig. 2.9 highlights the modelling framework for pedestrian behaviour when path planning. Similar to robotic path planning [96] both global and local path planning is used. The "strategic" level (Item i_1) involves deliberate global path planning that finds the most desirable paths that go from start to finish. The "tactical" level (Item i_2) involves selecting the best path en route that is influenced by constantly acquired knowledge, updating the global path. The "operational" level (Item i_3) involves instinctive local collision avoidance, which is the immediate response to something unpredictable in the immediate vicinity.

As mentioned in Section 2.3.1, movement and path planning is dependent upon physical issues of crowd density, groups, and traffic direction. When focussing on path planning and collision avoidance for pedestrians in a dense urban environment, the most helpful and significant factors are ones that involve mutual interaction, especially when trajectories intersect. To establish the significance of this particular pedestrian interaction and collision avoidance scenario a number of empirical studies are analysed, along with an evaluation of any associated model attributed to the findings.

2.3.2 Crowd and Pedestrian Modelling

Humans effectively utilise the same autonomy cycle as AMRs, Section 2.1.1, implementing *path planning* and *adaptive path control* strategies. However, unlike the majority of AMRs, which are designed for specific aims, environments, and path planning techniques (e.g. *local path planning* in an *urban environment*, using *macroscopic modelling* of external dynamic agents), pedestrians use all of these techniques for all situations. Global and local path planning, as well as macroscopic and microscopic movement analyses, are used both heuristically and intuitively, despite pedestrian models only choosing a select few of these.

Models are critical as they represent how a system may behave based on predefined data, preferably empirical in order to represent the correct behaviour, with crowd modelling being key to ensure successful path planning [55]. Crowd and pedestrian models are mathematical, using equations to describe likely dynamics and interactions. Designing an appropriate model that estimates where each pedestrian will be helps constrain any uncertainties by assessing probabilities of movement. There are many different forms of model available [159] with each attributing to a specific purpose, as no one model is able to account for all movement possibilities. An appropriate crowd model relies on the choice between two forms of crowd motion: microscopic and macroscopic [170, 171], each requiring a different form of analysis, whilst mesocopic is a combination of the two [172]. The choice of model can be made by asking the question: Is the model intended to describe the crowd (macroscopic) or the pedestrians (microscopic)?

2.3.2.1 Macroscopic Modelling

This perspective of crowd modelling focuses on pedestrians being a continuum. The flow of the crowd is typically modelled using partial differential equations, based on fluid dynamics [58]. This approach uses few parameters as the individual agents are ignored, making the model particularly good for optimization [173] and evacuation testing [64]. A crowd of pedestrians can generally be treated as a continuum provided the distance scale between pedestrians is significantly less than the distance scale of the area the pedestrians are moving [64]. Therefore, the distance between individual agents is considered negligible, with the hydrodynamic approach referring to locally averaged quantities and ignoring local velocity fluctuations [174].

The macroscopic model is an excellent analysis tool in order to determine and predict the general movement of a large mass of people. In a high density environment it allows environmental factors that increase congestion to be identified, such as bottlenecks, corridors, tight corners, etc, allowing the route with the easiest flow to be identified, and provide information on specific areas to avoid. This is clearly beneficial when designing queing and exit systems for arenas where large volumes of people will move in confined and direction orientated spaces [175]. However, as the individual is ignored it is not so good for analysing the interactions between each pedestrian. Macroscopic approaches can model the average properties of the system directly, without any explicit reference to its underlying microscopic nature. A central assumption of macroscopic models is that no, or sufficiently little, significant information is lost when the microscopic details are averaged out [13].

Modelling the crowd using continuum dynamics would be a poor choice for creating unique path predictions as the behaviour of the pedestrians is considered a continuous mass rather than discrete particles. An AMR would be unable to plan a path as its movement would be dictated only by the crowd. Therefore, plotting a path *through* the pedestrians would be improbable to achieve if using a macroscopic approach.

2.3.2.2 Microscopic Modelling

This perspective of crowd modelling represents pedestrians as individuals, which are capable of interacting with one another. Each pedestrian is represented as an agent with their dynamics defined by a continuous dynamical equation, or discrete cellular automata.

To design a model there are multiple parameters available to describe the complex nature of the individual [55], and so predict their behaviour. Microscopic pedestrian models are based on [176]:

- j_1 A detailed representation of space, to employ a uniform movement algorithm
- j_2 The representation of individual persons, which takes into consideration personal abilities and characteristics

To understand the collective behaviours observed in crowds lies in the different natures of local interactions among individuals [177]. The mechanisms of crowd dynamics requires understanding the behaviour of pedestrians during local interactions. This is any kind of social influence that motivates an individual to change or adapt their behaviour, based on social cues originating from neighbouring individuals. In pedestrian crowds there are at least five different types of interaction [178]:

- k_1 Collision avoidance: the strategic adaptation of walking speed and direction to avoid an upcoming collision with another person
- k_2 Physical interaction: when people are in physical contact with one another, at high density levels
- k_3 Social interaction: the behaviour of social groups of pedestrians, such as friends going together to the same place.
- k_4 Imitation: modifying their walking destination to move in the same direction as other surrounding individuals

k_5 Indirect interaction: adapting their walking behaviour on the traces left in the environment by other pedestrians who are not present any more.

These cannot be performed using a macroscopic approach. As stated in [179]: "Real pedestrians influence each other in their walking behaviour either with mutual or reciprocal action. They need to avoid or overtake each other to be able to maintain their speed, they need to change their individual speeds and direction and sometimes they need to stop and wait to give others the chance to move first". Assessing each individual separately creates a model to anticipate their next move, allowing paths to be created that move around individual pedestrians. Unlike macroscopic models the speed and direction of each individual is dictated by themselves, and only influenced by the pedestrians directly surrounding them. The microscopic model aims at describing the behaviour of individuals, which can become complex when many individuals are interacting simultaneously and in close proximity [167, 180]. For this type of model a specific aim of how pedestrians should behave is initially established, assuming a specific form of crowd behaviour. As pedestrians exhibit free will it would be impossible to encompass all of the potential behaviour and interactions. However, it allows unique path predictions to be made for each individual pedestrian. Many variations of both agent-based models and Cellular Automata exist, and each can be used to try and mimic a specific form of microscopic pedestrian behaviour.

2.3.2.3 Phenomenological Microscopic Models

Cellular Automata (CA) could be regarded as similar to a form of cell decomposition used in AMR path planning (Section 2.2.1.1). Pedestrians are represented by cells in a grid that transition to empty neighbouring cells according to a set of simple rules [181]. As the CA method is a discrete state system and the transition rules are simple, the computational cost is far more efficient than solving partial differential equations, such as with the SFM, however movements are not as smooth. CA can be used to produce macroscopic behaviour for traffic analysis [181] by using microscopic collision avoidance, which would be a mesoscopic mix of the two. Agent-based models (ABM) are typically used to describe each pedestrian, simulating the actions and interactions of each specific person [1, 182, 183]. The goal of an ABM is to search for explanatory insight into the collective behaviour of agents (which do not necessarily need to be "intelligent") obeying simple rules, often within the immediate vicinity.

The Intuitive Crowd Behaviour Model (ICBM) is a collision avoidance system, similar to CA, that uses rules based on [184]: distance (close / near / far), behaviour (walking / waiting), and type of collision (front / following / perpendicular) that may occur between pedestrians. This forms 16 reactions for normal conditions and 12 for traffic congestion. The movement model forms direction decisions by using data ‘left behind’ by others, which creates a macroscopic model effect by following pedestrians *en masse*. However, local laws allow for the presence of individuals, similar to microscopic models, even if the overall outcome produces macroscopic behaviour. The attempted crowd behaviour is to queue, rather than overtake.

The Social Force Model (SFM) is an established method that mimics Newtonian mechanics, where the forces between pedestrians represent an impetus on their motion [185]. The SFM was initially inspired by Boltzmann-like and Boltzmann-Fokker-Plank equations [186]. The forces between pedestrians arise from a sensory stimulus, and create a behavioural reaction depending on a set of behavioural alternatives attributed to personal aims. When formulating the SFM three main effects that determine the motion of a pedestrian are described as:

l_1 To reach their destination as comfortably as possible. i.e. the shortest path

l_2 Their motion is influenced by others, which is dependent on the crowd density and their desired speed

l_3 They are sometimes attracted to other people, e.g. friends, street artists, etc, and objects, e.g. window displays

These points are similar to the "utility optimisation" route choices made by pedestrians, as described in Section 2.3.1.

The SFM is commonly used to describe microscopic pedestrian behaviour when simulating AMRs (e.g. [42, 187]). Simulation results of the SFM demonstrate self-organization concerning lane formation and oscillatory changes in two crowds walking in opposite direction through narrow a passage. However, each pedestrian is modelled as a homogeneous agent and so a level of cooperation between each pedestrian must be assumed, as defined by mutual Newtonian forces between each agent. In reality crowds contain heterogeneous individuals with different behavioural constraints, many of which cannot be predicted. This would make the SFM unreliable for anything more than assessing basic crowd phenomena at the operational level. The forces between SFM agents only interact locally around their current positions in a form of reactive collision avoidance path planning, as seen in the AMRs autonomy cycle, Section 2.1.1, and local path planning methods, Section 2.2.2.1. The SFM does not consider how pedestrians may use strategic path planning techniques to avoid certain areas and potentially take longer routes, as the SFM assumes the shortest path is always used (Item l_1). This is again similar to the AMR’s autonomy cycle, Section 2.1.1, as well as roadmaps used to find a suitable path, Section 2.2.1.2.

This level of additional pedestrian navigation that must be considered is discussed further in Section 2.3.3.

The Predictive Collision Avoidance Model (PCAM) is an excellent expansion of the SFM, which considers what potential interactions may occur between pedestrians [169]. Unlike the SFM, which produces relative forces between pedestrians' at their *current* positions, the PCAM creates its reactive force based on pedestrians' *predicted* future positions. The applied forces relate to where the pedestrians are likely to meet, in order to move them off collision trajectories rather than applying forces when they are about to collide. The reactive SFM adapts an agent's motion rather late and so paths have a high curvature and unnecessary movements, whereas the PCAM's predictive approach allows the agents to plan early for collisions and avoid detours. In factors of time, path length, average speed, etc, the predictive PCAM is able to outperform the SFM. The naturally forming crowd phenomena of lane formation was more aesthetically pleasing, whilst the predictive model demonstrated two pedestrian flows crossing very elegantly. (Please refer to Fig. 4 in [169] for visual comparison to the SFM).

Conclusion Movement models do not seem to capture the social behaviour of pedestrian movement. The SFM relates to *social* as a reference to a 'public collective', whilst the social aspect regarding unique interpersonal behaviours, as well as the individual idiosyncrasies of each person, are ignored. This is no doubt due to a high level of unpredictability in crowds, as movement models often admit to making significant assumptions about the chosen pedestrian behaviour in order to achieve a desired outcome. The academic *continuum crowd simulator* [64] specifies that it is not suitable for all crowd behaviour, and also includes the limitation that pedestrians move with a common goal. The *a priori* knowledge of pedestrian goal orientation is a common assumption (e.g. [64, 69]), with the models unable to consider people "wandering aimlessly" without a purpose.

One issue regarding crowd models, that does not translate into the real-world, is often the lack of spatial seperation between agents. Many of the aforementioned models allow pedestrians to overlap, which is something that could never occur in real-life. The Reciprocal Velocity Obstacle model [188] (not a pedestrian model, however is discussed later in Section 2.3.3.2 when analyisng pedestrian navigation as it does replicate empirical pedestrian interaction) removes this assumption and more accurate microscopic movements are simulated. Although it works well at low densities, when compared to the high density crowds it produces unrealistic motions and causes the pedestrians to constantly alter speed or change orientations in order to reach their goals [169]. It seems a model that respects the physicality of each pedestrian can only operate for low density crowds. The combination of macroscopic and microscopic approaches is ideally required, unifying global navigation and local collision avoidance into a single framework [64], respectively.

All models use homogeneous agents, assuming all pedestrians will behave the same, with the same desires, strategies, etc. However all pedestrians in a real-life crowd will behave differently at a microscopic level, moving around differently. In order to develop a path planner, the assumption that the prediction of a pedestrian's movement will be to a specific location must be avoided. A pedestrian's unpredictability must be effectively introduced into the model, preventing errors occurring in the model's prediction to detrimentally affect planning collision-free and considerate paths. In order to establish what would be an effective pedestrian model to use, real pedestrian interactions need to be studied.

2.3.2.4 Empirical Microscopic Models

In addition to using microscopic models to predict the individual movements of pedestrians, models can be trained to estimate a pedestrian's potential trajectory and short-term path choice. By evaluating empirical data of pedestrians moving and interacting with each other within a particular environment, models can be designed that use probability distributions to predict the likelihood of them choosing a particular path. However, it must be assumed that the training data used for the model must cover all meaningful behaviour, so as to make an informed decision. A Markov Decision Process (MDP) is often used in order to predict the trajectory of a pedestrian (e.g. [189–191]), however the main drawback to this approach is that human motion towards other humans differs from the motion towards robots [134]. Therefore, typical human motion patterns cannot be learned from the motion patterns of human-human interaction.

Robot-Human Hindrance Problem A simulation focusing on a robot-human hindrance problem [189] used a *soft-maximum* MDP model to demonstrate the benefits of trajectory forecasting. A dataset of 83 pedestrian trajectories for training and 83 for evaluation was used to create the model. 200 hindrance-sensitive planning problems, corresponding to 22 different trajectories where naïve planning causes hindrances, were then generated. By ignoring casual influence of robot's action on a person's trajectory the trade-off between robot efficiency and interference with pedestrian trajectory is made. The movement of pedestrians is predicted over a floor grid in order to reason probabilistically where they will move to. Bayesian inference is used over a *prior* distribution using the observed trajectories, and through incorporation of uncertainty the shortest path may not be taken even though the learned cost function may be higher. Although the model's probability distribution areas are successful, with a 64% success rate for predicting the correct trajectory for each pedestrian, they cover large areas and do not accurately predict individual trajectories - only probability distributions form their current location.

The cost function is designed to best explain previous behaviour and therefore predict future pedestrian positions. To make the cost function transferable to other environments, six parameters are used. The first is a constant feature for every grid cell of the configuration space. The second is an indicator for if an obstacle exists in a particular cell. The third to sixth are four "blurs" of obstacle occupancies that dissipate the cost in the cell into neighbouring ones (please see Fig. 4 in [189] for visual demonstration). Using a small number of parameters is shown to be more effective, as when comparing the results to a variable-length Markov model, the paper for this model concludes that "the variable-length Markov model (and similar directed graphical model approaches) are generally much more difficult to train and do not generalize well because their number of parameters is significantly larger than the number of parameters of the cost function employed in our approach" [189].

Predicting Pedestrian Trajectories A mixed Markov model [190] was also developed to predict pedestrian trajectories, evaluated using a museum environment. A training dataset of 691 people was used, with an average participation time of 1h 31m per person. The model can predict the next discrete movement of a pedestrian from a probability-distribution obtained when training, using a maximum likelihood function. This can provide an informed approximation towards how individual pedestrians may move by making discrete destination predictions. However, these discrete predictions are not specific coordinates, but are topological points at exhibitions in a museum. Using a pedestrian-flow simulator [192] the model produced a 74.1% success-rate, as the more available history of transitions from exhibition to exhibition was available the better it faired. However, this level of accuracy may not be achievable if predicting movement trajectories of individual pedestrians. As a provided comparison the more basic Markov model and Hidden-Markov Model achieved much lower success rates of 16.9% and 4.2%, respectively. The model also achieved a 64% success rate for the prediction of real pedestrian movements in a real-life exhibition.

Learning Navigation Policies The Hamiltonian Markov Chain Monte Carlo model [191] is designed to learn pedestrian navigation policies from 230 observations of two pedestrians and one wheelchair user crossing trajectories, which can then be inferred onto an AMR under similar scenarios. To decipher if the "perceived" pedestrian navigation policies really could be transferred into an AMR, ten human subjects were asked to differentiate between recoded human behaviour and behaviour generated from the algorithm [193]. The subjects were shown images of completed trajectories, and each provided answers for what they considered human in order to generate a corroborative opinion. Unlike many other models which routinely evaluate the models success based on number of collisions, completed trajectories, etc, this paper [193] evaluates its success based on what is considered social motion. Out of the ten human subjects 79% correctly identified any recorded human behaviour and 68% of the behaviour generated by the algorithm was mistaken for human behaviour. However, It can be assumed that an overall success rate is actually closer to 54%, as only 79% of real human behaviour was correctly identified ($0.79 \times 0.68 = 0.537$).

The SFM [185] is also evaluated using the same overlapping trajectories and only 45% of the SFM's behaviour was perceived as human. As this study was focused on individual pedestrian crossings, how the system would respond to multiple overlapping pedestrian trajectories within a free-flowing crowd is unknown. It could be assumed that within a high density crowd the total number of clashing pedestrian trajectories may lead to the freezing robot problem [69]. Predictions based on machine learning result in worse predictions when compared to model-based approaches, as they are based on observations of one agent at a time and so the motion of several agents is not taken into account [105].

Conclusion It is observed that taking the shortest distance path is not always the most effective, as more potential collisions may occur along it. Using a *soft-maximum* MDP, the movement of pedestrians is predicted over a floor grid in order to reason probabilistically where they will move to. Bayesian inference is used over a *prior* distribution using the observed trajectories, and through incorporation of uncertainty the shortest may not be taken even though the learned cost function may be higher. Although the model's probability distribution areas are successful, they cover large areas and do not accurately predict individual trajectories - only probability distributions form their current location.

2.3.3 Analysis of Individual Collision Avoidance

One of the main conclusions taken from the literature thus far is that the most inefficient aspect of pedestrian models is how to effectively replicate the movement of individual pedestrians that have conflicting trajectories. When considering the general manner of how pedestrians path plan, Section 2.3.1, a model using microscopic empirical evidence as training data can inform an AMR of a pedestrians statistical and topological movement within a shared environment [113, 194]. This allows informed global path planning strategies to be made that represent deliberate cognitive reasoning, Fig. 2.9 (top row). Current pedestrian models such as the SFM account for instinctive local collision avoidance of pedestrian behaviour, Fig. 2.9 (bottom row). However, how individual pedestrians cognitively influence one another is a behavioural level, Fig. 2.9 (middle row), which is omitted from well known models such as the SFM. These individual interactions are not based on physical Newtonian mechanics between each pedestrian, but involve cognitive reasoning and anticipation from each pedestrian about how best to interact with one another and mutually avoid a collision. This form of cooperation is something required for an AMR to successfully move within a crowd, by anticipating how individual pedestrians will mutually interact.

2.3.3.1 Empirical Studies

The microscopic models covered in Section 2.3.2.4 all attempt to train a model to select trajectories that are similar to pedestrians. Only the final model [191] evaluates how "human" these trajectories are when cooperation between pedestrians is required, by deliberately employing scenarios where two pedestrians must cross in front of one another. However, this study does not evaluate the specifics of how the pedestrians respond to one another, simply that the model learns when to replicate specific movement patterns. Two/Three studies are now presented that evaluate the responses of two pedestrians crossing one another's path, using a total of 1,350 trials. Studies such as these can help quantify the tactical path planning stage level of pedestrians, Fig. 2.9 (middle row). The cognitive evaluation of potential upcoming collisions, influenced by neighbouring pedestrians' movements, can allow for pro-active rather than reactive collision avoidance to be developed.

Perpendicular Path Crossing A study focusing on two walkers crossing paths at a perpendicular angle [195] concludes that collision avoidance is collaborative. From 450 trials a sub-set of 260 were used that measured a risk of collision. Evasion was performed by both pedestrians with a reaction occurring 3s before crossing paths with a preference to change trajectory over speed. However, collision avoidance behaviour is "role-dependent", i.e. the walker giving way contributes more to the avoidance than the one passing first. This is related to the elliptical personal space surrounding a moving pedestrian [196]. The asymmetry in collision avoidance also stems from a difference in the visual field of each pedestrian, as perceiving someone passing in front of oneself is very different to perceiving someone passing behind.

Non-Reactive Path Crossing Another study focuses on two walkers crossing paths where only one pedestrian is influenced by the other [18]. One pedestrian is non-reactive (interferer), continuing in a straight line, ignoring the pedestrian which responds to a collision potential (participant), evaluated over 900 trials. The two pedestrians are evaluated crossing trajectories at 45°, 90°, 135°, and 180°, as well as a static pedestrian blocking a moving pedestrian's path. For the participate to avoid a collision with the interferer, their walking trajectory is locally adjusted according to the interferer's crossing angle. Path adjustments were observed over *all* angles, whilst speed adjustments were consistently evident *only* in scenarios with perpendicular or less crossings.

Trajectories are locally planned and adjusted as pedestrians consider the dynamics of the interferer to predict a hypothetical collision, avoiding this position. However, it is suggested that pedestrians also consider the uncertainty about this hypothetical collision. They assume a higher uncertainty along the speed parameter of the obstacle, as this would explain the braking behaviour for acute crossing angles. Therefore, an elliptical uncertainty is placed around the potential collision area. The participants chose a collision avoidance strategy of smooth spatial adjustments, which was initiated early enough to render speed adjustments unnecessary. These observations support the assumption of path adjustments as a default collision avoidance strategy in the presence of sufficient space. The results in this study also claim that their results are "in line with the predictions of the smoothness-optimization model of locomotion [197, 198], and are even more consistent with an advanced model of this theory, which adds an extra term that penalizes large variations of the speed [199]. As speed variations are penalized, turning should clearly be a favoured strategy."

Conclusion Both studies conclude similar results, including the fact that a variation in trajectory is preferable to a change in speed. Also, the personal space surrounding a pedestrian and the area of uncertainty surrounding the anticipated collision point are both represented elliptically, elongated along the velocity axis. Regardless of either *collaborative* or *considerate* interaction between two pedestrians, there is still a consistent *you go first* narrative based on the orientation between each pedestrian, with the pedestrian further from the collision point moving behind the other. These are results drawn by other papers on simillar subject matters (e.g. [7, 65, 66]). Further to the remarks of how agent based models do not replicate human movement in Section 2.3.2.2, the second study [18] concurs that "models based on repulsive potentials principle (e.g. [169, 185]) are not suitable for explaining general human locomotion and collision avoidance behaviour", and that there has been no conclusive study to state exactly how humans do behave [18].

2.3.3.2 Predictive Collision Avoidance Models

Once conclusions on how pedestrians interact under various scenarios have been drawn, they can then be mathematically expressed and developed into an appropriate model. All the pedestrians that participated in the *Empirical Studies* in Section 2.3.3.1 behaved similarly, employing a similar collision avoidance system is used by each pedestrian that focused on approximately anticipated points of collision in the future. The avoidance strategy is one where pedestrians considerately collaborate. The pedestrian furthest from the point of collision adjusts their trajectory the most, moving behind the other pedestrian and allowing them to move mostly uninhibited.

The Reciprocal Velocity Obstacle Model This algorithm [188] does not use pedestrian dynamics, but does employ locally reactive collision avoidance in a simillar manner. Cooperation is observed between multiple agents, mutually adjusting their trajectories in order to avoid collisions. All agents choose the same side to pass each other, creating "smooth and realistic motions". It is implicitly assumed that each agent makes similar collision-avoidance reasoning. Due to this it can be guaranteed that both collision-free and ocsillation-free navigation will be achieved. The anecdotal evidence of agents directly crossing paths shows optimal smooth and collision-free paths, especially when compared to the original velocity obstacle model of 10 years prior [200]. The use of mutual cooperation highlights how this can achieve more effective path planning and collision avoidance results. However, if non-similar collision avoidance strategies are used for each agent, it can be assumed that these navigation guarantees cannot be assured.

The Interaction Gaussian Process Model This algorithm [69] suggests that within a dense enough crowd the robot will believe the environment to be unsafe, i.e. "every path is expected to collide with an agent in the crowd due to massive uncertainty". Within densely populated environments, avoiding collisions will naturally become more difficult, therefore the robot will either freeze due to the freezing robot problem (FRP), making no progress, or make extremely evasive manoeuvres. In order to combat this, the robot and pedestrians must cooperate between one another using "joint-collision avoidance". Using a human dataset, this is observed as they move asymmetrically parallel to one another, with either a 0 or 180° relative angle between them. However, to achieve this the model assumes *a priori* knowledge regarding the pedestrians' destination, and then Bayesian inference calculates *a posterior* of movement. As the pedestrians do not exhibit complex behaviour or even turning, just forward motion, the model is able to calculate trajectories that deviate based on upcoming collisions with oncoming pedestrians, resulting in cooperative collision avoidance. To evaluate the models success the robot replaces a pedestrian in the dataset, therefore the robot's movements can be compared to the pedestrians. The model claims that over 10 *runs* 70% of the robot's movements performed better in distance travelled and in safety margin distances passing pedestrians compared to the pedestrians themselves. However, the anecdotal results show the robot moving 1.2 seconds behind the pedestrian. From the Empirical Studies in Section 2.3.3.1, pedestrians begin to react 3s before a collision, and so it is dubious that the model is working as well as claimed. When examining the video footage the pedestrian that the robot replaces seems forceful as he/she moves through the crowd. The other pedestrians *cooperate* by moving out of his/her way, as he/she moves in a straight line. The crowd parts for the pedestrian, the robot follows, and it can be assumed that similar results could be yielded using a simple pedestrian following robot.

Conclusion Both predictive models rely on cooperation in order to avoid collisions, which for human-aware motion planning should be regarded as an "optimistic" quality [67], covered earlier in Section 2.2.3.1 as well as later in Section 2.4.2.2. The RVOm was designed to move an AMR around a dynamic environment with other similar AMRs. Therefore, the cooperation between two AMRs is essentially guaranteed, as the algorithm will adjust each AMRs movement in order to accommodate any predicted intersecting trajectories the same way for both. For multiple AMRs with multiple trajectory intersections the exact response of each AMR may not be completely predictable. In an experiment were a circle of 250 AMRs directly crossed the circle's circumference, the emergent behaviour involved pockets of clustering and lane formation between AMRs that moved together towards similar arcs on the opposite side of the circle where their goals were located. This macroscopic phenomena reflects pedestrian behaviour observed in the academic *continuum crowd simulator* [64].

Comparably, the IGPm is very optimistic at assuming cooperation between the AMR and pedestrians. When employed on a real-robot platform [41], set to move along a in a crowded cafeteria, the AMR is unable to move much at all. Pedestrians successfully manoeuvre past it, however the AMR is unable to progress far along a path, only really being able to rotate to a different heading. Although the algorithm is able to plan a path, the fact that the AMR does not move well an argument can be made that the model does not overcome the FRP it attempts to address. The downfall of this implementation may be related to humans responding differently to robots [134], but also due to the fact the assumption pedestrians will cooperate is wrong, discussed further in Section 2.4.2.2, as at one point a pedestrian deliberately tries to block the AMR from moving at all.

2.4 Human-Mobile Robot Interaction

The initial aim of human-robot interaction (HRI) was to have a robot capable of some form of communication between human and robot [201]. The original Turing test could be considered the first assessment of HRI, providing direct interaction through means of textual communication [202]. Since its inception HRI has developed into a multi-field area and now covers a much broader spectrum, e.g. socially assistive robots and socially interactive robots, robotic human perception/gestures, cognitive models, and motion planning. The current research areas of HRI primarily concern assistive and interactive robots. The assistive robots [203] work alongside care givers (e.g. [204]), physical therapists (e.g. [205]), autistic children (e.g. [206]), etc, through physical contact. Thus providing benefits toward rehabilitation, cognitive disorders, and education [207]. Interactive robots entertain through direct interaction, recognise other agents, use natural cues, exhibit a distinctive personality, etc, [208], providing educational tools, and research platforms. Three perspectives on social interaction with robots [209] can be made:

m_1 **Visceral factors of interaction:** instinctual reaction, e.g. emotional response

m_2 **Social mechanics:** higher-level communication and social techniques

m_3 **Social structures:** development of changes to the social relationship and interaction between robot and human

The robot can become anthropomorphic and develop from a standard task-orientated machine into a relationship-orientated one. Regarding social structures (Item m_3), people would rather fix their robot than replace it, becoming saddened by the latter alternative. Table 2.1 highlighted the fact that an autonomous system's capacity is defined by the HRI level, with many surveys that highlight the complexity and expanse of the HRI arena (e.g. [105, 201, 208, 210, 211]), as well as HRI performance metrics [212].

2.4.1 Mobile Robots in Human Environments

One of the original and most highly cited AMRs specifically designed to operate within a human environment began with the museum tour guide RHINO ('97) [36], and its second generation incarnation MINERVA ('98) [37]. These were deployed in *Deutsches Museum Bonn*, Germany, and *Smithsonian's National Museum of American History*, USA, respectively. Both proved to be very successful with RHINO claiming a success rate of '99.75%', based on the ratio between number of requests (2,400) and number of collisions (6), and MINERVA claiming to be superior still. This was due to the added facial features of MINERVA [213], as the anthropomorphism results in people preferring to interact co-operatively. The navigational success of these two projects was only to move to their required locations without colliding with pedestrians. Treating people as simple objects to locally avoid, rather than incorporating social dynamics into a global path planner, would allow effective collision avoidance, but does not address the mimicry of pedestrians.

Both RHINO and MINERVA used the DWA, Section 2.2.2.1, to navigate their environments , which is a method still implemented in current H-Mobile-RI systems such as seen in the STRANDS project [214]. It is long-term autonomy projects such as STRANDS that focus on simple collision avoidance approaches towards pedestrians, involving the local DWA planner and a global costmap, that focus on the long-term dynamics of mapped space rather than behavioural navigation. The project operates at the Akademie für Alterforschung at the Haus derBarmherzigkeit in Vienna, and the AMR uses topological maps, Section 2.2.1.2, in order to evaluate the probability that someone will interact with the robot at a given location and time, based on these long-term observations, which allows the robot to adapt to working routines. This long-term autonomy is vital concerning the probability of success, as only over large time periods can all faults be detected.

This thesis focuses on developing a navigational strategy and path planner that results in an AMR employing considerate behaviour towards pedestrians. This is an area of research that is being developed in robots with much more short-term autonomy levels. In 2013 Obelix [40] travelled 3.2km in 1.5h from the Faculty of Engineering, University of Freiburg, to the Bertoldsbrunnen in the inner city, Frieiburg, Germany. However, despite Obelix being developed 17 years after RHINO there has been little change in how it interacts with pedestrians, and it moves much slower than any pedestrian it passes. Obelix does not employ any form of pedestrian model when it navigates through the streets, and it never needs to evaluate how it should respond to individual pedestrians as they move around the robot. Obelix relies on SLAM in order to navigate and employs an unknown collision avoidance strategy to avoid crashing. In addition to this, no moving pedestrians ever obstructed its path[4], and so even its limited collision avoidance could not be evaluated.

In a more localised indoor environment [42] demonstrates that a simple microscopic pedestrian model is vital to recreate similar social movements. For a robotic navigation system to operate like a pedestrian within a crowd, the pedestrians themselves must be appropriately modelled. This allows their movements to be predicted so the manner in which the robot should behave and move can be correctly designed. The social force model [185], is recalibrated from human-human to human-robot situations. This adaptation allows the robot to use a human-like collision avoidance system in a real-world pedestrian environment. The robot recreates a collision avoidance strategy that can be perceived as safe and natural by humans, with 96.4% in agreement. The density of the crowd was however very low, with the robot only interacting with a sparse number of pedestrians head-on, and moving a significant distance of 8 metres before a collision potential. In a more crowded environment such an early divergence from its trajectory may cause problems by moving into the path of other pedestrians, as pedestrians only begin to path correct at an average distance of 0.38 to 0.86m [195]. The negative effects of the original social force model may then develop, where unnatural oscillation occur between neighbouring agents.

[4] https://www.youtube.com/watch?v=gPzC88HkgcU

2.4.1.1 Vehicles

Aside from the aforementioned AMRs, which are human sized, larger autonomous vehicles (AVs) tend not to deal with the intricacies of social navigation as they are larger, move faster, and do so with more limited movements. Also, AVs are often confined to roads, e.g. self-driving cars, whilst pedestrians remain on pavements. Due to this the main interaction AVs have with pedestrians is when a person crosses the road (e.g. [65, 66]). Any form of collision avoidance occurs as a "stop-and-wait" policy, which is the only available method when an travels along a road lane. However, some human-aware path planners still employ this method in both path planning and collision avoidance strategies, discussed in Section 2.4.2.3 and Section 2.4.2.1, respectively. If the stop-and-wait policy was to be employed in a crowd this would increase congestion and cause movement oscillations, as observed in microscopic pedestrian models, Section 2.3.2.2. Therefore, AVs must be able to successfully predict if a pedestrian is to cross in front of it, and implement a navigation strategy that will ensure the safety of: the pedestrian crossing in front of the AV; the passenger of the AV (if there is one); and if the speed of the AV must be reduced or stopped, the safety of any vehicle moving along behind the AV.

To ensure the safety of anyone crossing the raod, the intention of the pedestrian must be evaluated to decipher if they are to cross or not. For methods that assume the intentions of pedestrians do not change, and that they aim to move toward their goal with a constant trajectory [65, 66], the AV will simply come to a stop if a pedestrian is on the road, and continue to move once they are on the opposite pavement. However, predicting when a pedestrian will cross at specified crossings require a fast and reliable process, as they can "perform instantaneous changes in motion behaviour following changes in intent." [215]. A "changepoint" must be identified whereby "observed data better fits a new behaviour model than the current model to which it is being compared", as "agile dynamic agents such as pedestrians may exhibit new behaviours or mid-trajectory changes in intent". Although a stop-and-wait strategy is best, being able to detect a pedestrian's crossing intent should prevent collisions by reducing the AV's velocity before the pedestrian begins to cross.

The "Intention-Aware Online partially observable Markov decision process Planning" method is used to autonomously drive a golf buggy AV through a sparse crowd [216]. An A* search, Section 2.2.2.2, is applied to find a minimum-cost path to the goal. The positions of all detected pedestrians are anticipated over a 2 time-step prediction horizon, and the path is planned using a simple 4-stage reward model:

n_1 To ensure safety a large penalty is applied if any pedestrian gets within a small distance of the AV.

n_2 To encourage the AV to reach its goal the reward increases with a decreasing distance to it.

n_3 To encourage the AV to maintain a high speed, when safe to do so, a greater penalty is applied the slower it moves.

n_4 A small penalty is applied for any +/- acceleration in order to encourage smooth driving.

The AV is able to move along relatively smooth trajectories, however the maximum speed of the AV is $1.5ms^{-1}$ and an emergency brake is triggered when a pedestrian gets within 0.5m of the buggy. As a result the AV still stops and starts, however does not simply wait, but re-plots alternative paths.

For road bound AVs, the stop-and wait navigation strategy is not suitable to be used in pedestrian crowds. Also, for all of these AVs the physical limitations do not allow sufficient adaptability in order to manoeuvre like a human. However, the latter system [216] does employ a reward strategy that is used in a crowd, although with its increased speed it is suitable for sparse crowds rather than much denser crowds where the CPP is intended to be implemented.

2.4.2 Human-Aware Path Planning

It can be assumed that the autonomy cycle of all AMRs follow the standard autonomy cycle structure, Fig. 2.1b. The data processing within each stage, as well as the interaction between each stage, is then modified for the selective purposes of each AMR. As mentioned in, Section 2.1.2, careful consideration must be made towards what type of dynamic agents the AMR will share the environment with. The movement of an AMR around humans is very important [217] and should reflect human-like motion [218]. To improve robot acceptance the AMR's behaviour must include naturalness, comfort, and sociability, as defined in [105]. An AMR cannot simply plan global paths that find the shortest distance to the goal, using geometric local collision avoidance as the AMR moves. Instead it must plan global paths that are considered "human" [219], using proxemics influenced local collision avoidance that respect socially acceptable human personal boundaries [150]. To generate "legible" robot behaviour, whereby humans can understand clearly what an AMR is doing, assumptions made in the literature include:

o_1 Model human-like behaviour [139, 220, 221]

o_2 Generate stereotypical motions [220, 222]

o_3 Generate efficient motions [139, 221, 223]

o_4 Take into account social constraints, human abilities, and preferences [224–226]

o_5 Robot motion must be as visible as possible [227, 228]

o_6 Add complementary motions (gestures) in order to clarify intentions (e.g. gaze, pointing, use animation principles) [223, 229–231]

all of which can be constrained to the path planning stage, bar the final item which would be a bi-product of the planning stage. These all aim to make the AMR appear to move as humanly and therefore as predictably as possible, such as making the AMR move ‘stereotypically’. This elaborates on the requirements for human-aware navigation discussed in Section 2.1.2.2, List *d*.

2.4.2.1 Planning Paths

An AMR will often be designed to navigate within a specific type of environment and interact with a specific type of obstacle. For instance, the Kiva robot system [54] mentioned in Section 2.1.2.1 is designed to operate within a "Human-Exclusion Zone", and the environment is kept relatively certain as the path for each robot is calculated using a centralised architecture. For an AMR to navigate within a pedestrian environment an appropriate model must be designed, Section 2.3.2, in order to anticipate what any one pedestrian may do next. However, no model can predict with 100% accuracy and so an AMR must continually re-plan multiple paths, each belonging to a different homotopy class in order to aide diversity. This strategy is superior to a greedy shortest-path as it allows the AMR to consider multiple alternative choices as well as interactions with neighbouring pedestrians [154]. Despite the AMR only executing a small fraction of each path, the remaining unexecuted path serves two important functions [232]: 1) To guarantee safety the planner needs to look beyond the AMR’s minimum stopping distance; 2) The remaining path can approximate what future paths may be, although there is no guarantee the next planned path will contain along the remaining current path.

In a complex and changing real-world it is not possible for a programmer to anticipate all the situations a robot may encounter [233]. Therefore, the pedestrian model and the robot’s behaviour would be limited to set scenarios. As discussed in, Section 2.3.3.1, in order for pedestrians to move fluidly their collision avoidance strategies must be collaborative, with both pedestrian taking on a role and then cooperating. As AMR human-aware path planning is akin to real human path planning, this is essential in minimising the confusion that could occur if more rudimentary collision avoidance was implemented. In more conservative planners a passive reaction is preferred, implementing a "stop-and-wait" procedure and allowing the pedestrian to move before the AMR continues [7]. However, within a crowded environment this may add to congestion as a "traffic jam" scenario could develop, as well as confusing pedestrians moving up behind the AMR.

2.4.2.2 Assuming Cooperation

The cooperation paradigm is an optimistic assumption whereby potential collisions will be mutually resolved between the robot and pedestrian as they get closer together, via trajectory and velocity adjustments [67]. This assumption will constrain the uncertainty in the pedestrian model, as well as allowing paths to be predicted that simulate human paths and behaviour. The now more 'predictable' pedestrian paths will allow more potential space to become available for the AMR to predict its own path. By assuming pedestrians cooperate it will be more efficient for the AMR to continue along a currently occluded path than it will be choosing another [67].

For an AMR to move towards its goal, a global path must be planned from its current position to the goal, Section 2.2.2.2. When assuming cooperation between robot and pedestrian this becomes much more feasible in a crowded environment due to the increased available space, and so also eliminating the potential "freezing robot problem". However, within such a dynamic environment it may not be possible for the AMR to predict a collision free path, in which case the AMR should pick one that provides the least number of collisions. Even if collision free paths are available, it may be preferable for the AMR to select a less "comfortable" path, as long as the path length is considerably shorter than a path with minimal conflict, and as long as it does not interfere with a pedestrian early on [154]. As the AMR should only partially move along its path before recalculation, Section 2.4.2.1, there is no necessity for the path to be collision free in the relative future.

The IGPm, discussed in Section 2.3.3.2, implemented a "joint-collision avoidance" strategy for cooperation. During simulations the method claimed to execute paths that were 'safer' than its pedestrian counterparts. However, when implemented on a real-life AMR platform [41] the resultant motion did not replicate the results due to a different dynamic structure of the crowd, as well as pedestrians individual reactions to the robot itself. Cooperative strategies rely on pedestrians choosing similar navigation principles, which also only occur under more structured environments [193], such as moving past one another in a corridor. Lane-based pedestrian traffic flow is required for the IGPm to be effective, with cooperation occurring between pedestrian and AMR meeting head-on. As discussed when analysing the empirical evidence of individual pedestrian collision avoidance, Section 2.3.3.1, pedestrians will behave differently when approaching from different angles. Therefore, all reactions that occur due to different angles of approach must be considered when developing a cooperative path planner.

2.4.2.3 Avoiding Collisions

The IGPm fails as it does not consider all potential angles of approach between the AMR and pedestrian. To prevent an AMR colliding with a pedestrian many papers (e.g. [7, 65, 66]) claim that the best method is for the AMR to "stop-and-wait" until the pedestrian has passed before continuing along its path. Although this is practical in a one-on-one scenario, this is clearly not suitable to a free-flowing dense crowd. Although this method is considerate to an individual pedestrian, *considerate braking* should be extended to *considerate motion* so that the AMR can replicate empirical human collision avoidance, Section 2.3.3.1. Taken from various studies (e.g. [18, 195]), generally when two human subjects cross paths they assume a role of either a reactive or non-reactive agent. The pedestrian furthest from the hypothetical collision point will remain non-reactive, whilst the other will adjust their trajectory to move behind the other pedestrian as trajectory variations are preferable to speed changes.

For successful HRI in this context, the pedestrian's acceptance towards the AMR plays an important role [140]. The acceptance of the AMR is influenced by: trust; anxiety; perceived usefulness; and perceived enjoyment [234]. As well as "perceived safety" [140]. Safety is a basic human need [235] and can be designated as a key requirement of HRI, with the definition of perceived safety as: "Perceived safety describes the user's perception of the level of danger when interacting with a robot, and the user's level of comfort during the interaction" [236]. Due to this it can be assumed that extra care and consideration be taken by the AMR, and so AMR-human collision avoidance strategies must vary slightly from human-human ones even if attempting to replicate human behaviour.

The *ContextCost* Costmap This algorithm [157] is implemented on an AMR for a 90° path crossing scenario with a single pedestrian. However, as it is only capable of slowing the AMR down and not alter its trajectory. This was a choice based on the author's previous publication [139] whereby analysis of 10 different pedestrians crossing each other in the same scenario, over 4 trials, resulted in only velocity deviations. This is a contradiction to far more comprehensive studies, Section 2.3.3.1, which concludes there is more significant evidence for preferences of trajectory changes instead. As a result the paper states that a "robot using cost model *ContextCost* would not deviate from the straight line, but instead reduce it's velocity" and therefore a "robot path direction and human direction are considered incompatible if the human and robot could frontally run into each other". This will limit the application of the costmap to the specific scenario chosen, and unless the context of the scenario is know by the pedestrian *a priori* if the robot stops then its navigational intentions cannot be inferred by the pedestrian.

The Layered Social Costmap This algorithm [7] also behaves in the same way for the same path crossing scenario. Similarly, when the resultant AMR's behaviour is compared to a path predicted using a static costmap, the same erratic motions are produced as the AMR tries to cross in front of the pedestrian. The social costmap incorporates a constant velocity prediction model that projects an elliptical proxemics distribution in front of the for several time-steps. As this clearly makes passing in front more costly, if the AMR simply moved behind the pedestrian no interference would occur. This would not only allow the AMR to maintain its speed it would reach its goal faster and more efficiently. This is again why it is preferable to change trajectory over speed, which is a more common pedestrian trait, and would allow the model to be applied to more varied scenarios.

Conclusion The "legibility" in path planning discussed at the start of this subsection is so that another pedestrian can correctly interpret the AMR's intentions. For a "Human-Robot Path Crossing Task" [140] a video of an AMR approaching a moving pedestrian from the front at 3 angles: right-diagonal; head-on; and left-diagonal trajectory, is evaluated. Using a "Human-Aware" path planner that predicts paths based on "social costs", the resultant behaviour taken by the AMR was perceived as more human than with a more traditional path planner. Although this reduced any confusion for the pedestrian, and it can be assumed that a ‘human-like’ collision avoidance was observed, no resultant paths the AMR took were provided. Two AMRs working in close proximity [67] also adapt their trajectories in similar ways to pedestrians, by both adjusting their trajectories early on. However, detailed analytical comparisons between empirical pedestrian-pedestrian collision avoidance, Section 2.3.3.1, and robot-pedestrian collision avoidance for multi-angle path crossing scenarios has not yet been undertaken. Only social collision avoidance strategies, Section 2.2.3.1, have been developed to replicate human behaviour, focusing on the AMR replicating pedestrian trajectories rather than evaluating an AMR's interaction when directly crossing paths.

2.4.3 A Taxonomy for Interactive Robot-Pedestrian Navigation

A standardised taxonomy of something is an excellent way to classify and evaluate it against similar things. Taxonomies for autonomous robots, Table 2.1, evaluate how much human input is required in order for a robot to complete a task, whilst the task itself can be arbitrarily defined. However, with increasingly diverse forms of robotic autonomy, involving direct interaction with people, these previous definitions cannot accurately translate into these areas. Traditionally robot autonomy was evaluated for a single robot and single human interaction [19, 33, 85]. However, it is possible for HRI to incorporate upto 8 possible combinations of single or multiple humans and robots, acting as individuals or in teams [237].

Despite this update to include the ratio of humans to robots interacting, as well as the consequent level of shared interaction, the autonomy evaluation still splits into the "autonomy level" and the "intervention time" [237]: "The autonomy level measures the percentage of time that the robot is carrying out its task on its own; the amount of intervention required measures the percentage of time that a human operator must be controlling the robot. These two measures sum to 100%". Unfortunately this cannot be the case for an AMR navigating through a crowd of pedestrians, as humans will indirectly influence an AMR's behaviour. For human-aware AMRs influence from humans is required in order to increase its level of autonomy, rather than the traditional decrease.

What all these taxonomies still have in common is the concept of the human interaction element as *direct*, rather than *indirect*. The information shared between the human and robot is always considered a relationship where the human and robot are working directly together to complete a task. However, for an AMR to navigate through a crowd of pedestrians it must indirectly interact with all neighbouring humans as well. For an AMR in a HRI scenario, in order to achieve full autonomy the robot must take input from humans in order to allow for cooperation and social mechanics, which is contrary to standard HRI taxonomies. Due to this a new taxonomy is proposed, Table 2.5, specifically for evaluating robot autonomy with indirect passive interactions from pedestrians, rather than a direct interaction from a human controller.

The proposed taxonomy, Table 2.5, is divided into 10 levels. The AMR's autonomy level increases from "no autonomy" (level 1) to "full autonomy" (level 10), in regards to the level of HRI it has with 'passive' surrounding pedestrians on the crowd. However, the top and bottom 5 levels are divided into the input the AMR receives from a direct operator and non-direct near-by pedestrians, respectively. The first 5 levels are similar to traditional taxonomies, where the direct influence of a human operator is inversely proportional to the autonomy level of the robot; i.e. "These two measures sum to 100%" [237]. The latter 5 levels would be regarded by traditional taxonomies as all fully autonomous. However, the more influence taken from pedestrians results in AMR becoming more autonomous and independent through cooperation. The resulting direct interaction will allow the AMR to behave more like an agent similar to a human.

2.5 Chapter Summary

This chapter evaluated the necessities required for human-aware robot navigation. Both the path planning process of AMRs and pedestrians parallel one another, and the path planning choices made by an AMR should reflect the same behaviour as a pedestrian. Global path planning and predictive modelling should be undertaken by the AMR. This *look ahead* feature is not just for safety, but will aide in the AMR achieving human-like motion by anticipating pedestrian motion so that paths are

	Influence on AMR Movement				
#	**Direct**	**Passive**	**Level of Autonomy**	**Description**	**E.g.**
1	🟦	🟪	Remote Control Operation	The operator remotely controls the robot, which has no mobile autonomy whatsoever.	N/A
2	🟦	🟪	Expression of Intent	The robot can use simple automatic cues to facilitate the pedestrians' awareness of its movement, such as brake lights and indicating.	[238]
3	🟦	🟪	Reactive Collision Avoidance	The robot can adjust its position in order to prevent collisions, based on the proximity of neighbouring pedestrians.	Section 2.2.2.1
4	🟦	🟪	Proactive Path Planning	The robot will suggest paths within its immediate vicinity, of which the operator can select their preferred option.	Section 2.2.2.2
5	🟦	🟪	Autonomous Direction Control	The robot can move in a general direction stated by the operator, choosing its own path and using reactive collision avoidance.	N/A
6	🟥	🟪	Macroscopic Crowd Modelling	The robot is able to model the overall dynamics of the surrounding crowd, allowing it to move along with the crowd flow.	Section 2.3.2.1
7	🟥	🟪	Microscopic Path Planning	By modelling the individual interactions of surrounding pedestrians, the robot can manoeuvre within the crowd and plot a unique path.	Section 2.3.2.2
8	🟥	🟪	Implicit Cooperation	It is assumed that both the robot and the pedestrians will contribute towards collision avoidance of each other in the same way.	Section 2.4.2.2
9	🟥	🟪	Explicit Cooperation	The robot directly communicates with the pedestrians in an easily understandable manner, so that they can mutually move along collision free paths together.	N/A
10	🟥	🟪	Hypothetical AI Singularity	The robot is able interact with other pedestrians in all the same ways as a real human, moving towards an 'Asimovian future' with no environmental boundaries.	N/A

Table 2.5 A novel taxonomy proposed to highlight autonomy levels specific to human-robot navigation within a crowd; each increasing stage develops upon the previous. Based on a hierarchical system of influence from humans initially outlined in List *a*, Page 2. The *direct* influence on the movement of the AMR is from input procured from humans that interact with the AMR on purpose. Level 1-5 reflects a traditional autonomy scale, Table 2.1, with a decreasing human influence increasing the autonomy level. Level 6-10 reflects the proposed modern autonomy scale that with a decreasing human influence increases the autonomy level, due to an increasing level of direct cooperation between robot and human. The *passive* influence on the movement of the AMR is from data obtained from anonymous crowd members that indirectly have an affect on the task the AMR plans, based on their current behaviour. The autonomy level continually increases from level 1-10 by predicting what each pedestrian may do next, as well as due to a increasing level independent communication the AMR can achieve with pedestrians.

planned that foresee potential upcoming interactions. It is clear that cooperation is not always available between AMR and pedestrian to the same extent as between pedestrians only, and so a considerate approach must be undertaken instead. However, from empirical studies on pedestrian collision avoidance, considerate behaviour generates implicit cooperation itself. The next chapter will use the acquired knowledge in order to generate the model and requirements for the proposed considerate path planner.

Chapter 3

Developing the Considerate Path Planner

"And Now for Something Completely Different."

— Monty Python

THE aim of this thesis is to design a considerate navigation strategy (CNS) for an autonomous mobile robot (AMR), moving *considerately* through a crowd of pedestrians. Therefore, this chapter focuses on developing the novel considerate path planner (CPP), and the key problems and challenges faced in the area of human-aware robot navigation are first addressed, Section 3.1.

An appropriate crowd model must also be implemented in order for the CPP to successfully plan a path within a dynamic pedestrianised environment. This can describe pedestrian behaviour mathematically and allow predictions of their individual movements to be made. A pedestrian movement model is presented, Section 3.2, which uses novel *uncertainty ellipses* that cover pedestrians' movements over a prediction time-horizon, so that global paths can be planned through an evolving environment that will contain morphing pedestrian positions. The uncertainty of each prediction must be contained enough in order to prevent the freezing robot problem (FRP) [69]. It is also essential that the behaviour of each pedestrian is modelled [55], in order to assess how the system works if it is impossible/impractical to create experimental conditions in real life.

Finally, the design of the CPP is presented, Section 3.3, which provides the basis for a considerate navigation strategy (CNS). This includes the design of the novel Voronoi diagram-visibility graph hybrid (VD-VG) roadmap; the type of search algorithm appropriate for the model robot is also presented; as well as the *considerate weights* that modifies the search algorithm in order to find the considerate paths and fulfil the aims and objectives, Section 1.3.2.

3.1 A Novel Human-Aware Navigation System

Following on from the reviewed literature in the previous chapter, Human-Aware Robot Navigation is a diverse research area. From developing an autonomous robot that is capable of implementing suitable navigation strategies, Section 2.2; to modelling pedestrians in a way that can most accurately predict their behaviour, Section 2.3; to robot behaviour that can implement legible behaviour for human-robot interaction, Section 2.4; each is fraught with assumptions and limitations that accumulate with each separate stage, resulting in increasing uncertainty. For the CPP this uncertainty is to be embraced, allowing for paths that can be planned and result in the AMR's behaviour being both consistent and legible to others.

3.1.1 Problem Statement

In crowds people expect other pedestrians to move in similar ways, moving both as a macroscopic member of the crowd and as a microscopic individual Section 2.3.1. The vision for a human-aware AMR, which moves within a pedestrian crowd, is that it should also behave like another human. The AMR's speed, manoeuvrability, path planning, and interaction with others, should mimic as close as possible to other pedestrians. This way any of the pedestrians will not be confused or unsure about what the AMR may do next. Making the AMR's behaviour legible to other pedestrians Section 2.4.2 will make the AMR safer and easier to integrate into the crowd.

Many current human-aware navigation systems, Section 2.4.1, are only capable of interacting in limited and structured environments. Often, assumptions are made about how a pedestrian will cooperate with the AMR, Section 2.4.2.2. However, when a pedestrian does not behave as assumed, the AMR's navigation system is not robust enough to perform consistent behaviour under these alternative scenarios. This will result in a reduction in legible behaviour of the AMR, making its behaviour more confusing and thus less safe to the pedestrians.

To address this issue a CNS is proposed, which employs the novel CPP. Rather than rely on pedestrians to behave in a certain way, and cooperate with the AMR, the CPP will plan paths that do not assume the pedestrians will do anything other than continue to move as they are. The CPP will exploit gaps that are already present in between pedestrians in order to move past them, rather than requiring the pedestrians to move as well. Even so, these movements of the AMR towards gaps in the crowd will result in implicit cooperation between the AMR and pedestrians. Just as how the AMR sees a pedestrian moving towards a new space in a certain direction, the pedestrian will see the same for the AMR. As it is reasonable to assume pedestrians will not want to walk into anything, the pedestrian will avoid the AMR in the same way as the AMR avoids the pedestrian, by moving into their own free space. By having the CPP plan paths that result in the AMR moving along similar paths as a pedestrian would, the AMR's behaviour would be clearly legible to any pedestrian and any collisions mutually avoided.

3.1.2 Current Challenges

As established in Section 2.1.1 the path planning stage of an AMR's autonomy cycle must also contain crowd modelling, Fig. 2.2. These two factors are dependent upon each other due to the dynamic nature of crowds, in order to continually predict pedestrian movements and plan suitable paths accordingly. The planned paths must make the AMR's movement legible to neighbouring pedestrians, Section 2.4.2. This facilitates the overall flow of the crowd and eases individual interaction, by allowing the pedestrians to infer what the AMR may do next. To achieve this the path planner must also be able to infer what each pedestrian may do next.

3.1.2.1 Pedestrian Models

A suitable microscopic pedestrian model will allow the system to achieve this, however significant assumptions are currently made when modelling human behaviour, Section 2.3.2, which include the following:

- a_1 Goal knowledge of pedestrians are known a priori, e.g. [64, 69]. This is used in order to make the pedestrians' intentions more predictable, but is something that can only be known by each individual pedestrian.
- a_2 The chosen microscopic model accurately captures all pedestrian movement dynamics and potentials.
    - $a_{2.1}$ Agent Based Models, Section 2.3.2.3, such as Newtonian Mechanics, describes all interaction based on balancing forces between other pedestrians, objects, and goals. Attractive and repulsive forces allow exact movements to be predicted for individual pedestrians.
    - $a_{2.2}$ Empirical Models, Section 2.3.2.4, for inferring human-human collision avoidance trajectories, provides a probability distribution for different potential trajectories. Exact predictions are not made, only likelihoods, using additional layers to the Markov model that pedestrians reflect.

The assumed a priori pedestrian goal knowledge forces the model to organise the movement of pedestrians, and without it many well known pedestrians simulators admit to failing. Many microscopic models also make this assumption, which leads to limitations for the AMR by reducing the number of scenarios the AMR can be used in. In an unstructured environment, e.g. wandering aimlessly at an exhibition, no goal can either be known or even inferred. Only the pedestrian's next movement can be estimated based on their currently observed position and velocity, due to a pedestrian's Markovian behaviour [63]. Furthermore is the assumption that one model is used by all pedestrians. Of the pedestrian models evaluated in Sections 2.3.2.3 and 2.3.2.4, each produces different predictions, however all are valid at describing pedestrian motion in some form. No mixture of models is ever used as it is not possible to establish what form of navigation any one pedestrian will undertake.

An assumption is now made that a much simpler pedestrian movement model should be used, which can be continually modified based on known occurrences, including a pedestrian's: statistical walking deviations and their desired proxemics, discussed later in Section 3.2.3.2 and Section 3.2.3.3, respectively. A model that does not assume a priori goal knowledge of pedestrians, and creates a balance between exact movement predictions and probability likelihood distributions. Embracing the uncertainty and confidently confining it to within an elliptical distribution.

3.1.2.2 Legible Path Planning

To plan legible paths, current strategies seemingly either lack confidence in predictions or are overtly optimistic in exactly how pedestrians will respond to the AMR. These two opposites include:

b_1 AMRs that exhibit cautious behaviour use very simple models, if any at all.

  b_1 The *polite* "stop-and-wait" policy, e.g. [7, 139, 157], is the wrong strategy for a crowd, as any pedestrians moving along behind the AMR will be blocked, causing congestion and interrupting pedestrian movement.

  b_2 Strategies that are more *cautious*, by providing a wide berth to upcoming pedestrians, adjust the AMR's trajectory too much when avoiding upcoming pedestrians. As a result, avoiding a hypothetical collision with one pedestrian can put the AMR on a hypothetical collision course with another.

b_2 AMRs that exhibit optimistic behaviour use more complex models, with far greater assumptions. If "optimistically" assuming pedestrians will cooperate with the AMR, Section 2.4.2.2, by using the same collision avoidance strategy, the AMR cannot move if pedestrians do not cooperate.

Similarly to the two types of pedestrian model in Section 3.1.2.1, a balance must be made between the conservative and hesitant, and the assuming and optimistic navigation strategies. To generate legible behaviour the considerate navigation strategy must firstly always be moving, so that the pedestrians can infer the AMR's intentions. Secondly, the AMR should not move into scenario that assumes cooperation, but can cooperate if the pedestrian chooses. Finally, if in an identical scenario the pedestrian chooses not to cooperate, the AMR can just as easily plan a considerate path that does not require any direct cooperation from the pedestrian.

3.1.3 Considerate Navigation Strategy

Considerate navigation addresses these challenges by exploiting potential space within a crowd. Rather than the "optimistic" assumption of Human-AMR cooperation, Section 2.4.2.2, consideration does not assume the pedestrian will react at all. Cooperation assumes that both parties will work together to adjust their paths so that they do not physically collide at the intersection points of their trajectories. Conversely, consideration is designed to exploit gaps in between pedestrians so that the pedestrian need to react at all, but still produce similar behaviour. This is reflected in the newly proposed "Interactive Robot-Pedestrian Navigation", Table 2.5, for which considerate path planning is level 8, whilst "optimistic" cooperation would be level 9.

3.1.3.1 Pedestrians

When the pedestrian movement model is designed in Section 3.2, it will represent a pedestrian's walk, therefore only the x and y dimensions will be used. The model environment will be represented by a 2D Euclidean plane, assuming the pedestrians' walking surface has a flat topography

$$(x \times y) \in \mathbb{R}^2 \tag{3.1}$$

The central position of each pedestrian is represented using Cartesian coordinates, (x_p, y_p). The dimension of each pedestrian will be represented by a geometric circle with a radii of their largest cross-section, r_p,

$$(x - x_p)^2 + (y - y_p)^2 = r_p^2$$

Following the assumptions of how to physically represent pedestrians means that path planning strategies can now be devised.

3.1.3.2 Considerate Path Planning

Empirical studies of pedestrian collision avoidance, Section 2.3.3.1, show that even if one does not react to the other, the same collision avoidance strategy occurs as when both can react: one pedestrian will adjust their trajectory and move behind the other. This slight readjustment of their trajectory avoids the hypothetical collision without negatively affecting their overall path efficiency. Utilising this strategy, the CPP can plot smooth paths around multiple pedestrians in a crowd without assuming cooperation. By plotting entire paths that reflect the CNS over a prediction time-horizon, the AMR can move towards areas where gaps between passing pedestrians are likely to occur, even in a high density crowd. As the CPP will not rely on cooperation, considerate path planning will reduce potential path conflict between pedestrians and the AMR. However, if the pedestrian does choose to react, implicit cooperation would occur regardless as a gap will be made available for the CPP to exploit through the design of an appropriate pedestrian model.

3.2 Designing a Pedestrian Movement Model

The freedom of a pedestrian allows them to alter their speed, direction, and behaviour almost instantly [239]. However, many current pedestrian models assume a priori knowledge of each pedestrians goal in order to generate more predictable behaviour, Section 2.3.2. Theoretically all pedestrians could begin to move in any direction at their maximum speed in a moment. If this was accounted for in movement models the model's prediction would guarantee to cover the pedestrians' future positions, however the "uncertainty explosion" [69] could easily result in the FRP. Any currently available space would become too costly to move through as the prediction time-horizon grew.

To restrict the uncertainty in predictions, pedestrian behaviour is often modelled as a Markov process with various 'hidden-layers' added to more easily predict their behaviour. However, the prediction of random pedestrians in unrestricted environments, and without a priori goal knowledge, can only be made on what is currently observed. This makes them Markovian [63], as the probability of a pedestrian moving to another position only depends on their present state, as this includes data from all previous states. The prediction is therefore applied using their currently observed velocities. The result is a linear dynamic system commonly known as the constant velocity model (CVM). Considerate paths are achievable when using the much simpler CVM, as microscopic agent based movements are dominated by short time-horizons, and crowds are partially stochastic [240]. Therefore, sophisticated models are not required since displacements are so small that linear extrapolation is sufficient [14]. To evaluate this claim a dataset of real-life pedestrian trajectories is analysed in order to establish how any deviations from the CVM can be *embraced* by the model.

3.2.1 Pedestrian Movement Analysis

In order to effectivly generate a model that describes real-life pedestrian motion, an analysis of how a pedetrian's velocity changes over time is made. The BIWI Walking Pedestrians dataset [14] is a manually annotated .txt file of pedestrian positions, Fig. 3.1 and Table 3.1, obtained from a video recording made from the top floor of the ETH main building in Zurich, which covers pedestrians moving around outside as they enter and leave the building, Fig. 3.2a. The video is 8 minutes and 38 seconds long, and contains a total of 367 pedestrians travelling a velocity range of

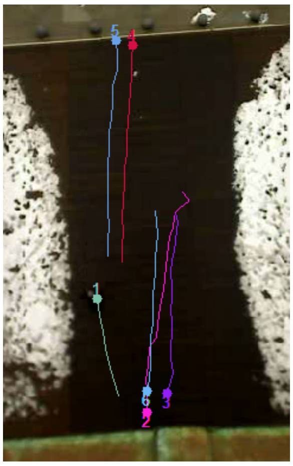
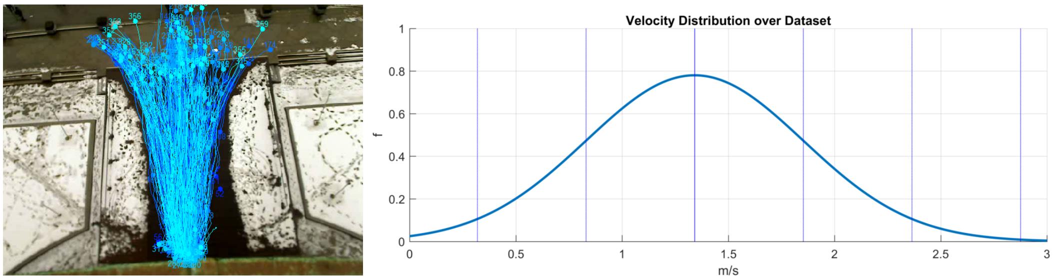


Fig. 3.1 Pedestrian trajectories over first 25 time-frames of the dataset.

frame #	ID	$\mathbf{P}_x$	$\mathbf{P}_y$	$\mathbf{V}_x$	$\mathbf{V}_y$
780	1	8.457	3.588	1.672	0.17630
786	1	9.126	3.659	1.663	0.32670
792	1	9.787	3.849	1.683	0.37110
798	1	10.470	3.955	1.599	0.26480
804	1	11.070	4.061	1.575	0.45640
	2	13.020	5.783	−2.324	−0.07661
810	1	11.730	4.321	1.644	0.54440
	2	12.090	5.752	−1.589	−0.06597
816	1	12.380	4.497	1.624	0.44060
	2	11.750	5.730	−1.141	0.10540
⋮	⋮	⋮	⋮	⋮	⋮

Table 3.1 Pedestrian data for the first 7 time-frames of the video, stored in a .txt file.



(a)

(b)

Fig. 3.2 Visualisation of the BIWI Walking Pedestrians dataset [14]. (a) All pedestrian trajectories within the dataset. (b) Pedestrian velocity distribution of the dataset, with a mean velocity of $1.34ms^{-1}$.

$0 - 2.84ms^{-1}$, $\mu = 1.34ms^{-1}$, Fig. 3.2b, over an average distance of $10.4m$. The frame-rate of the video file is $2.5Hz$, with pedestrian positions recorded every $0.4s$. The dataset provides the following variations in pedestrian movement:

c_1 **Bidirectional Traffic:** This ranges from two pedestrians moving in either the same, or alternating, directions.

c_2 **Various crowd densities:** These range from single pedestrians, to very large groups of pedestrians.

The data will be evaluated over a prediction time-horizon of $1 \leq T \leq 10$, which would be a real-time span in seconds, s, of $0.4 \leq s \leq 4$. Therefore, the model is evaluated against all pedestrians that have their positions recorded in over 10 or more consecutive frames, f to $f + 10$.

As linear extrapolation of movement is sufficient [14] the initial assumption is that the pedestrian model is based upon the CVM. Therefore, the analysis will be of how velocities of the pedestrians in the dataset change between each time-frame, Fig. 3.3. The speed and trajectory changes, Figs. 3.3a and 3.3c, can both be fitted with a Gaussian distribution, which conforms to the model's assumption. The 2D distribution results in an ellipse with semi-major and semi-minor axes the same as the covariances of their respective distributions, $a = 0.296ms^{-1}$ and $b = 0.249ms^{-1}$ respectively, highlighting that changes in speed occur slightly more than trajectory changes, which may be due to pedestrians not crossing in front of each other in the dataset. The mean of both distributions would be expected to be zero, however as multiple pedestrians converge on a single doorway they would need to slow down when entering the building, Fig. 3.2a, and so the mean speed change is marginally negative and the resultant ellipse off centre at $(-3.78\times10^{-1}ms^{-2}, -2.50\times 10^{-3}ms^{-1})$.

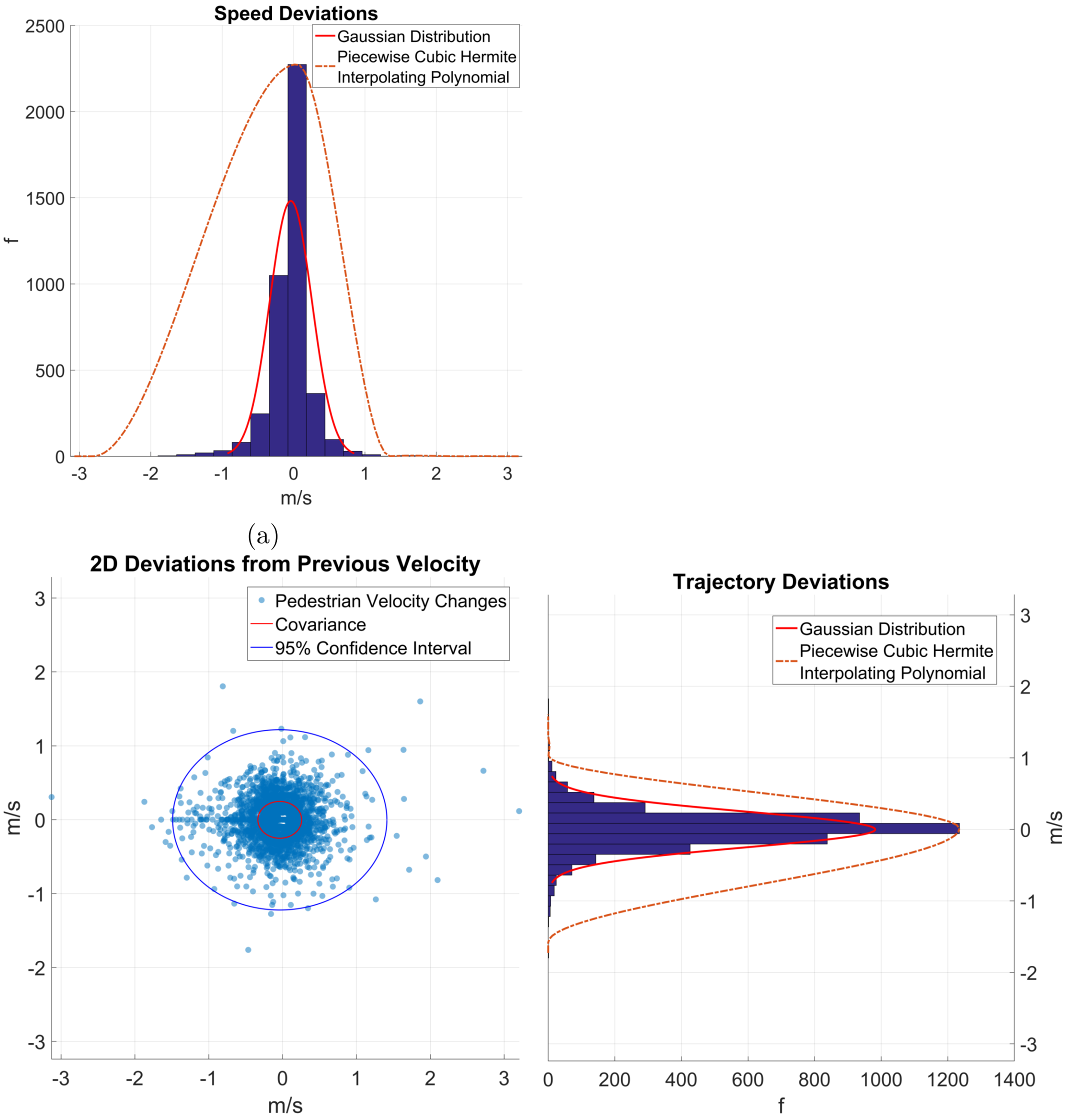
The Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) in Matlab is designed to interpolate all data points. Therefore, a negative skew develops for speed changes, Fig. 3.3a, due to pedestrians that have suddenly stopped and the resulting large negative velocity changes. These extreme outliers are insignificant in changing the overall Gaussian distribution of the data, however it does show that pedestrians will randomly stop on occasion. This initial evaluation reinforces elongating the uncertainty ellipses from the pedestrians' start positions in order to cover sudden stops in movement. Using a PCHIP to interpolate all trajectory deviations, Fig. 3.3c, a skew occurs to the right which may suggest a minor preference for pedestrians to cross paths of the right-hand side, although not enough to influence the design of the model. However, like the speed deviations these outliers are not significant enough to affect the overall Gaussian distribution.

3.2.2 Spatio-temporal Representation

The pedestrian's current velocity, v_p, is also calculated by observing them at two consecutive instances in time. The velocity of the pedestrian can then be easily found by comparing their current position at frame f to their position in the previous frame, $f - 1$

$$v_p = \sqrt{[x_p(f) - x_p(f-1)]^2 + [y_p(f) - y_p(f-1)]^2} \tag{3.2}$$

The prediction made for the future position of a pedestrian is made in two dimensions, whilst the CVM is a one-dimensional model. Due to this, the CVM can become inaccurate faster if the pedestrian makes any lateral trajectory deviations that occur orthogonally to their velocity.



(a)

(b)

(c)

Fig. 3.3 Analysis of the pedestrian velocity deviations within the BIWI Walking Pedestrians dataset [14], from consecutive time-frames. (a) The distribution of speed deviations can be modelled as Gaussian. (b) The change in velocity of each pedestrian in consecutive time-frames. The two-dimensional distribution generates an ellipse that supports the use of an uncertainty ellipse, Section 3.2.3, to contain the pedestrians' potential movements. (c) The distribution of trajectory deviations can be modelled as Gaussian.

The position of a pedestrian in the future, (x'_p, y'_p), can then be calculated

$$\begin{aligned} x'_p &= [x_p(f) - x_p(f-1)]\,t \\ y'_p &= [y_p(f) - y_p(f-1)]\,t \end{aligned} \tag{3.3}$$

where t is the time-step along the prediction time-horizon, $t \leq T$.

To also account for random sudden stopping of pedestrians all the space along the pedestrian's trajectory can be considered uncertain, from the pedestrian's start position to the CVM prediction at t. Other microscopic models attempt to either be very accurate and make their prediction at only one point, Section 2.3.2.3, or produce a likelihood distribution of potential trajectories from their current position to their predicted goal, Section 2.3.2.4. This model *embraces* the uncertainty by covering the potential area the pedestrians could have moved to within a set prediction time-horizon, and marry the benefits of the accuracy of the exact prediction with covering of inaccuracy with the distros.

A estimation of a pedestrian's position can be made that will encompass the prediction using a confidence interval to include velocity and lateral prediction errors, assumed to be a normal distribution, based on the observations in Section 7.2. The Gaussian distribution is an isocontour of an ellipse, and when mapped over two perpendicular dimensions it creates an elliptical confidence interval, Fig. 3.4a, that will more likely contain pedestrian's position at any time-step along the prediction time-horizon.

As an ellipse, Fig. 3.4b, is defined as a quartic, it satisfies the implicit equation [241]:

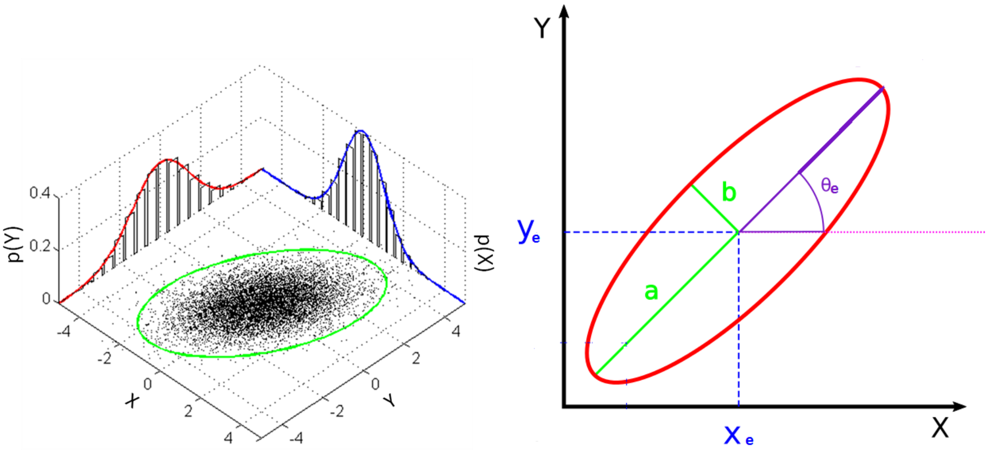
$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{3.4}$$

providing

$$B^2 - 4AC < 0.$$

The general equations coefficients are defined using the ellipse's semi-major axis, a, semi-minor axis, b, centre point, x_e, y_e, and angle of rotation, θ_e, as:

$$\begin{aligned} A &= a^2(\sin\theta_e)^2 + b^2(\cos\theta_e)^2 \\ B &= 2(b^2 - a^2)\sin\theta_e\cos\theta_e \\ C &= a^2(\cos\theta_e)^2 + b^2(\sin\theta_e)^2 \\ D &= -2Ax_e - By_e \\ E &= -Bx_e - 2Cy_e \\ F &= A{x_e}^2 + Bx_ey_e + C{y_e}^2 - a^2b^2 \end{aligned}$$



(a) (b)

Fig. 3.4 (a) Pedestrians may deviate laterally (through trajectory change) and longitudinally (through speed change) from their current velocity, with a Gaussian distribution. A Gaussian distribution is an isocontour of an ellipse, and so a two-dimensional Gaussian distributed multivariate system would forms an elliptical probability range. This ellipse represents where the pedestrian may have moved to, and is the initial stage of the novel uncertainty ellipse. (b) The resulting ellipse is orientated along the original pedestrian's velocity, θ_e, with the centre, (x_e, y_e), at the pedestrian's observed location. The semi-major axis, a, represents speed fluctuation, and the semi-minor axis, b, represents trajectory deviations.

where

$$a > b, \ \ (x_e, y_e) \in \mathbb{R}^2, \ \ 0 \leq \theta_e < 2\pi.$$

For each pedestrian's uncertainty ellipse, Fig. 3.5a, θ_e remains constant based on the initially calculated velocity of the pedestrian, Eq. (3.2). (x_e, y_e) is dictated by a ratio of between the pedestrian's start position, (x_p, y_p), at $t = 0$ and their predicted position, (x'_p, y'_p), at $t \leq T$, Eq. (3.3). The semi-major, a, and semi-minor, b, axes will be defined by the confidence of containing all the uncertainty, due to statistical deviations, the pedestrian may make from the CVM when $t \leq T$.

3.2.3 Calculating Uncertainty

Observed through empirical data the lateral trajectory deviations of pedestrians are greater than their velocity deviations, Fig. 3.3. The uncertainty of the CVM will be embraced and confidently confined to the *uncertainty ellipse*, allowing paths around the pedestrians to be planned. The dimensions of the ellipse are modified by both the statistical walking deviations of the individual pedestrian [160], as well as the proximity of other pedestrians, based on standard proxemics, Table 2.4. These will either increase or decrease the size of the ellipse's axes based on four conditions:

d_1 Increase the semi-major axial length, a, based on the average walking speed of the pedestrian

d_2 Increase the semi-minor axial length, b, based on trajectory deviations

d_3 Decrease semi-minor axis length on the side another pedestrian is too close

d_4 Increase semi-minor axis length on the opposite side another pedestrian is too close

The modification of the ellipse due to these factors can result in a multivariate split Gaussian distribution of 2-dimensions along the semi-major axis, due to differeing left and right trajectory deviation predictions. This will form an uncertainty ellipse designed to encompass the possible locations of where the agents could be at a given future time-step, Fig. 3.5b. As pedestrians walk in a forward motion, a velocity model can be justified over a short time horizon. The uncertainty ellipse will adjust for any factors involving velocity and lateral movement, providing a solid model for any dynamic pedestrian's motion. The uncertainty ellipse is calculated using a 2-DoF Chi-square distribution [242] in order to confine these errors using 90% confidence interval, Figs. 3.5c and 3.5d, over the prediction time-horizon.

3.2.3.1 Reducing Uncertainty

Removing uncertainty completely is impossible [62]. However, uncertainty is one of the most commonly occurring problems in robotic path planning within a dynamic environment [243], and to prevent collisions the three following sources of uncertainty should be considered [244]:

e_1 The robot's own dynamics

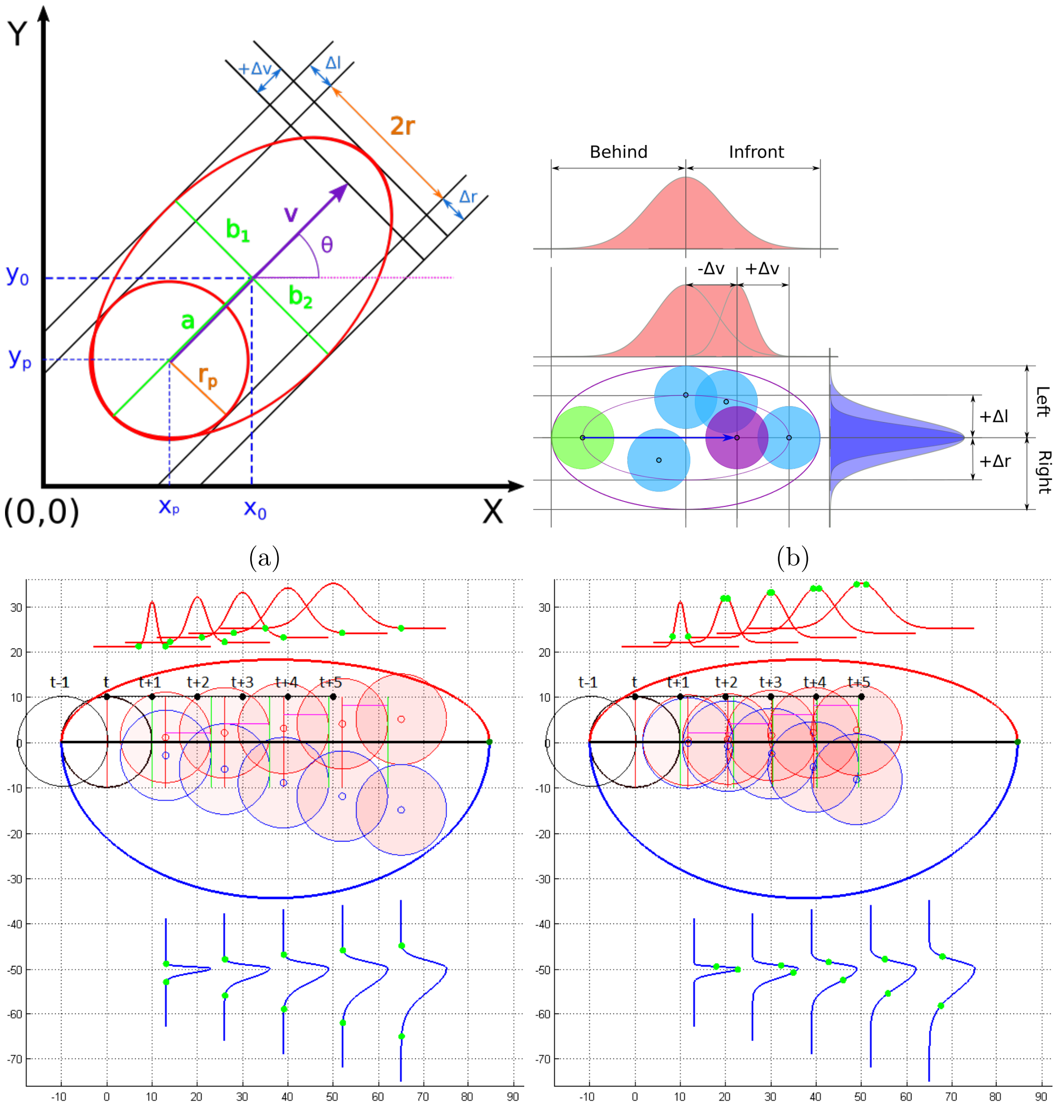
e_2 The environment's future conditions and positions of objects

e_3 Reasoning of the robot's motion over an infinite time-horizon, or at least a time-horizon required to reach the goal

One of the biggest contributors towards increasing collision potential is unreliable predictions of where other dynamic agents may be in the future (Item e_2). A prediction time-horizon (Item e_3) can be used to estimate their future positions though the appropriate model, however for a microscopic agent-based crowd an infinite time-horizon will undoubtedly contribute towards an uncertianty explosion. By limiting the predictive time-horizon to a reasonable length, a large enough window can occur for a considerate path to be plotted around the uncertainty ellipses.

The prediction time-horizon, T, is therefore limited due to:

f_1 The Markovian nature of pedestrians [63], where by predictions can only be made based on the pedestrian's current state. Therefore, predictions too far in the future would be assuming a continuation of past states.



(a) (b)

(c) (d)

Fig. 3.5 Demonstration of how the uncertainty ellipse of a dynamic pedestrian is calculated using their observed velocity, P_v. (a) The ellipse elongates along the same trajectory, P_θ, as the pedestrian from its initial position, (P_x, P_y). This encompasses the deviations of velocity, $+\Delta v$, and of lateral movement, $+\Delta l$ and $+\Delta r$, the pedestrian may make. This will dictate the semi-major, a, and semi-minor, b, axes of the ellipse. If the pedestrian's velocity is zero the ellipse degenerates to a circle, with radius P_r of a pedestrians shoulder width: $2P_r$, representing the pedestrian's position. (b) Visualisation of how the uncertainty ellipse covers positions of where the pedestrian starts (green circle at $t = 0$), and where they may be in the future (blue circles at $t > 0$). The arrow represents a CVM (measured from the centre points of the pedestrian's circle), with a purple circle representing the CVM at $t > 0$. (c-d) Examples of an uncertainty ellipse with a split-Gaussian trajectory deviation, set over a prediction time-horizon of $T = 5$. (c) Using the maximum deviations at each time-step, only a small portion of the left deviation prediction extend outside of the ellipse at the last time-step. (d) Using a random deviation occurring at each time-step, at the end of the prediction time-horizon all movements of the pedestrian remain inside the ellipse.

f_2 The growing uncertainty accumulating with each consecutive prediction, Section 3.2.3.1. Therefore, an "uncertainty explosion" can be avoided.

f_3 Pedestrian's only react to a hypothetical collision upto 3s before it occurs, Section 2.3.3.1. Therefore, predicting what a pedestrian may do on a larger time-scale is redundant.

f_4 Confirms the generalisation that pedestrians move directly towards their goal [2, 245]. Therefore, at the current moment a short extrapolation of a pedestrian's velocity is a reasonable assumption.

With a limited prediction-time-horizon, the model must attempt to contain the uncertainty as best possible. Making the predictions of where each pedestrian will be as *confident* possible, given the statistical likelihood of each pedestrian deviating from the CVM.

3.2.3.2 Statistical Uncertainty

Both the speed changes and trajectory deviations are modelled using statistical data, formed from the normal distribution of sampled pedestrian walking speeds when walking along sidewalks, wide sidewalks, and precincts [160], Fig. 3.6a. The range between the distribution peaks, μ_{min} to μ_{max}, is considered the uncertainty range of a pedestrian's velocity. However, the speed of a pedestrian can typically vary from $0 \rightarrow 2ms^{-1}$ [246], and the uncertainty would differ for each pedestrian's own walking speed preference. Although the aim of the model is to *embrace the uncertainty*, by increasing the uncertainty margin to cover all walking speeds ($0 \rightarrow 2ms^{-1}$) the resultant uncertainty ellipses would make the model environment become overcrowded, similar to an "uncertainty explosion". The uncertainty margin, u, is therefore calculated based on the pedestrian's observed velocity, v_p, in relation to the aforementioned walking speed distributions. This thesis postulates that the amount of uncertainty to add can be confined to three separate scenarios:

g_1 If the pedestrian is static, $v_p = 0$, then:

- $g_{1.1}$ No uncertainty is added.
- $g_{1.2}$ Uncertainty ellipse degenerates to a circle.

g_2 If the pedestrian is moving below the minimum mean, $0 < v_p < \mu_{min}$, then:

- $g_{2.1}$ Rate of change of uncertainty increases.
- $g_{2.2}$ Pedestrians more likely to speed up to the minimum level.
- $g_{2.3}$ Pedestrians more easily change direction.

g_3 If the pedestrian is moving above the minimum mean, $\mu_{min} \leq v_p < 2ms^{-1}$, then:

- $g_{3.1}$ Deviation uncertainty is minimal.

$g_{3.2}$ Pedestrians less likely to speed up any more, as they approach the maximum.

$g_{3.3}$ Pedestrians less likely to change direction.

This will create an ellipse that allows a higher uncertainty margin to occur if the pedestrian is below the average speed, as a change in speed and direction could be made much easier at lower speeds.

To ascertain real values of uncertainty, u, when $v_p = \mu_{min} \Rightarrow u = \mu_{max} - \mu_{min}$ (the aforementioned pedestrians' velocity uncertainty range). To describe this mathematically and create the model, if $v_p \leq \mu_{min}$ the uncertainty is be calculated using

$$\frac{(x - x_0)^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{3.5}$$

whilst, if $v_p \geq \mu_{min}$ the uncertainty is calculated using

$$y = \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ - \left[\frac{(x - \mu)^2}{2\sigma^2} \right] \right\} + v_p(x) \right) \frac{b}{\max(y)} \tag{3.6}$$

where for both Eqs. (3.5) and (3.6)

$$\begin{aligned}
x_0 &= \mu_{min} \\
a &= \mu_{min} \\
b &= \mu_{max} - \mu_{min} \\
\mu &= \mu_{min} \\
\sigma &= (\max(v_p) - \mu_{min})/3 \\
y &= u + v_p
\end{aligned}$$

Under the given conditions Eq. (3.5) describes the arc for a quarter of a translated canonical ellipse, whilst Eq. (3.6) is the RHS of a translated and normalised Gaussian distribution with a 3-sigma rule, Fig. 3.6a. Eq. (3.5) is designed to more rapidly increase the uncertainty that must be added, Item g_2, whilst Eq. (3.6) is designed to conform with standard pedestrian walking distribution, Item g_3.

The uncertainty calculated thus far is based on velocity uncertainty, $u_v = u$, however both trajectory and speed deviations will be different, Fig. 3.3. As trajectory deviations are assumed to be more likely when $v_p < \mu_{min}$, the associated trajectory uncertainty, u_θ, is ratio of u_v based on v_p. Therefore, if $v_p \leq \mu_{min}$

$$u_\theta = u_v \left[1 - \frac{v_p - \mu_{min}}{\mu_{min}} \right]$$

otherwise if $v_p \geq \mu_{min}$

$$u_\theta = u_v$$

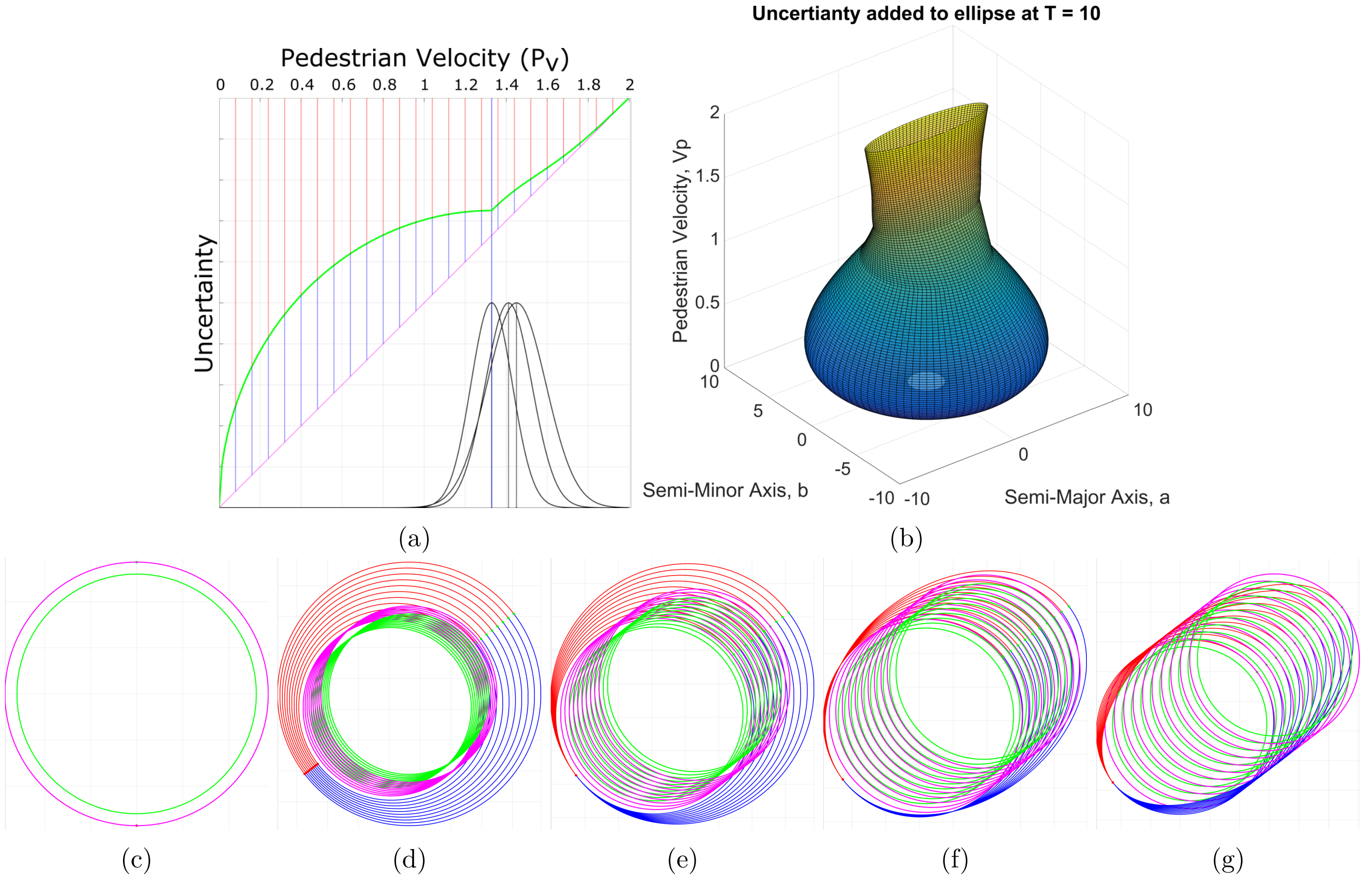


Fig. 3.6 Demonstration and examples of the level of uncertainty that creates the uncertainty ellipse, depending on the pedestrian's observed velocity, P_v. (a) Graph of how much uncertainty is added, dependent upon P_v.(b) Semi-Major, a, v Semi-Minor, b, axes change v Pedestrian velocity, V_p. (c) $V_p = 0$, the ellipse degenerates to a circle around a stationary pedestrian. (d) $V_p = 0.4ms^{-1}$, the low velocity results in a an uncertainty circle due to more likely trajectory changes. (e) $V_p = 0.8ms^{-1}$, as the velocity is below the minimum mean, μ_{min}, trajectory changes are still likely. However, the uncertainty ellipse extends further along the velocity line. (f) $V_p = 1.2ms^{-1}$, as the velocity is above the minimum mean, μ_{min}, trajectory changes are less likely. (f) $V_p = 2.0ms^{-1}$, as the velocity is at the perceived maximum no additional uncertainty is added to the velocity, and trajectory changes are considered not likely. The result is an ellipse that effectively represents the pedestrian's current velocity vector.

The semi-major, a, and semi-minor, b, axes of an uncertainty ellipse are defined by the current time-step, t, along the prediction time-horizon, T, as well as the initial radius of the pedestrian, r_p, and additional buffer zone, BZ, that surrounds the ellipse. Therefore, they are defined as:

$$\begin{aligned} a &= r_p + (v_p + u_v + bz)t \\ b &= r_p + (u_\theta + bz)t \end{aligned}$$

Fig. 3.6b visualises the velocity uncertainty, u_v, added to the semi-major axis, a, and the trajectory uncertainty, u_θ, added to the semi-minor axis, b, for the entire pedestrian velocity range, $0 \leq v_p \leq 2ms^{-1}$. Examples of final uncertainty ellipses are provided in Figs. 3.6c to 3.6g, over different prediction time-horizons, $\{T \in \mathbb{Z} \mid 1 \leq T \leq 10\}$.

3.2.3.3 Visual Proxemics

Humans favour distinct spatial zones that dictate how close they prefer to be to others, Table 2.4 in Section 2.3.1. However, a pedestrian can only respond to a *violation* of this space if they can see the *violator*. Therefore, a proxemics field of view (PFoV), Fig. 3.7, is used to modify the uncertainty ellipse of a pedestrian by evaluating the visible proximity of other pedestrians. The PFoV surrounds a pedestrian's uncertainty ellipse, combining six visual zones with the preferred proxemics of pedestrians. The shape of the PFoV will be anisotropic and extends father in front of a person [247] where the pedestrian's gaze focuses, as the situation in front of a pedestrian has a larger impact on their behaviour than things happening behind then [248].

Any other uncertainty ellipses that intersect the PFoV, as the positions of multiple pedestrians are predicted simultaneously, will adjust the lateral trajectory uncertainty by modifying the semi-minor axis of the uncertainty ellipse for that pedestrian. Adjusting the ellipse through this method is representative of the pedestrian marginally adjusting their trajectory if a collision is soon approaching, as observed in empirical studies, Section 2.3.3.1.

The six zones are reflected along the semi-major axis, so they occur on both the right and left of the ellipse with 12 areas in total, Table 3.2. Each area combines an aspect of both the spatial area around the ellipse, as well as the pedestrian's ability to see other pedestrians. A weighting system has been devised based on what form of collision could occur, ranging from insignificant (1), where little action need be taken, to very significant (6), where a trajectory change is likely. A simple cross-section between the ellipse and the PFoV polygons for each zone is taken when an intersection occurs, with the assigned weighting factor applied. If another ellipse intersects the outer layer of the PFoV (zones 4,10/6,12) the semi-minor axis on

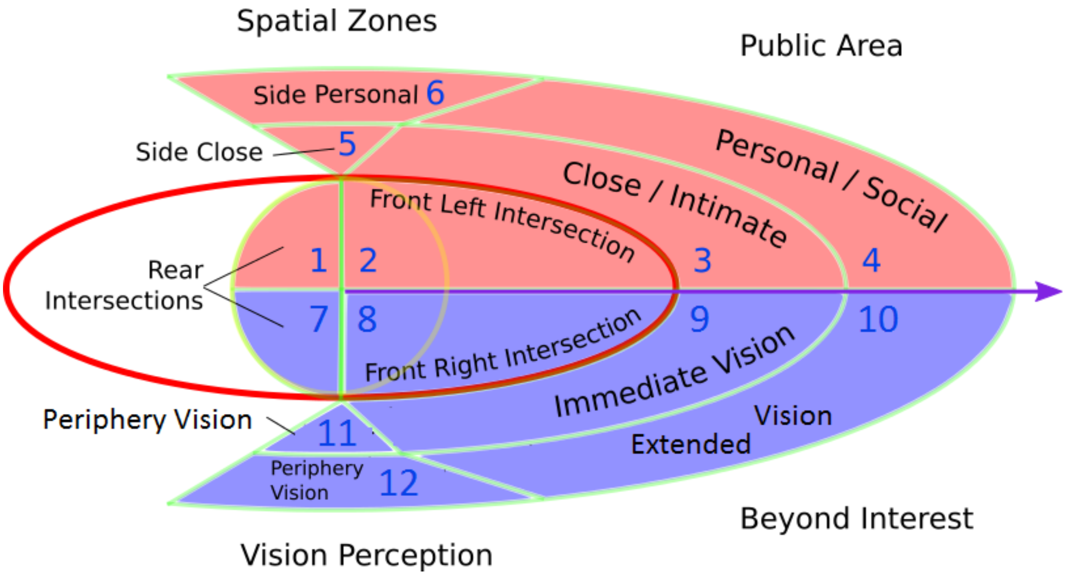


Fig. 3.7 Visualisation of the PFoV set-up placed around all currently visible pedestrians. Generated from a combination of the "Human-Human spatial zones" criteria, Table 2.4 in Section 2.3.1, and the visual perception range of pedestrians. Definition of the spatial zones within the PFoV, displayed on top red half, and the visual components of the field, displayed on the bottom blue half.

that side will decrease, as the pedestrian is less likely to move towards a potential collision. If the inner of the PFoV (zones 3,9/5,11) is penetrated, the semi-axis of that side reduces, whilst the axis on the opposite side increases, as the pedestrian may potentially move away from an upcoming collision. If the actual uncertainty ellipse is intersected (zones 1,7/2,8) both semi-minor axes increase, as the pedestrian may move either side to avoid a potential collision, as there is likely to be a direct collision.

A final uncertainty ellipse is generated that: 1) Predicts the potential locations of pedestrians in the near future; 2) Covers any statistical deviations a pedestrian may make over that time-frame; and 3) Modifies the trajectory uncertainty of the pedestrian if their proxemics is violated, so that the ellipse can account for any potential trajectory changes the pedestrian may make to avoid a collision. Now that the pedestrian model has predicted where each pedestrian may move, the CPP can exploit any gaps in between these ellipses and plan considerate paths through the crowd.

3.3 The Considerate Path Planner

To behave considerately, the CNS must direct an AMR along paths that fulfil the thesis' aims and objectives, set out in Section 1.3.2. The CNS is capable of using whatever environmental mapping or global path planning search techniques previ-

#	Zones		Weight	Justification of Weighting
	Spatial	**Visual**	**(1-6)**	
1,7	Rear Intersection	Rear Intersection	5	Intersecting the uncertainty ellipse, however SFoV relies on pedestrian's perception. From behind the collision would not be visible; requires physical contact.
2,8	Front Intersection	Front Intersection	6	Exactly where the pedestrian should be, causing a direct collision; highest weighting.
3,9	Close / Intimate	Immediate Vision	4	Directly in front of pedestrian; would want to move around obstacle.
4,10	Personal / Social	Extended Vision	3	Ahead of pedestrian; not close enough to be an immediate issue, only potential.
5,11	Side Close	Head Movement	2	Roughly perpendicular to the pedestrian; could see someone next to them and may want to move slightly away.
6,12	Side Personal	Periphery Vision	1	No collision path or any proximity issue; a precautionary measure to be 'aware' of something next to them.

Table 3.2 The impact weights attributed to each area of the PFoV, Fig. 3.7. The further another pedestrian enters the PFoV, the greater the weight becomes which readjusts the semi-minor axes of the uncertainty ellipse.

ously reviewed in Section 2.2.1 and Section 2.2.2.2, respectively. The novelty of the CPP lies in the *consideration weights*, presented at the end of Section 3.3.4, which allows the chosen search algorithm to plan considerate paths by evaluating the current search position in much the same way as a standard heuristic would, Eq. (2.1).

For the particular implementation of the CNS in this thesis, additional measures are included so that the CPP can plan paths that help assist in avoiding collisions with pedestrians. This is because prediction models cannot guarantee complete accuracy [62], and so these extra elements help account for model errors. The search uses the *consideration weights* and an AMR model to evaluate the current path at each open search node, to predict how an AMR may interact with pedestrians in the future.

The VD-VG is used in order to counteract the pedestrian model prediction errors. The VD element of the roadmap will limit the location of the search to distances that are equally the furthest distance from all neighbouring *uncertainty ellipses* simultaneously, helping to avoid path collisions with pedestrians. At the same time, the VG element will allow the paths to be planned that move more directly in between pedestrians, if required.

3.3.1 Modelling the Autonomous Mobile Robot

The elements of the robot model, Fig. 3.8, have all been selected to share similar attributes with the pedestrian model. The movement simplifies the robot's movement to be holonomic, where controllable degrees of freedom equals total degrees of freedom. As the robot operates within a 2-dimensional space the mechanics of motion only require the robot position and velocity. Comparable to the pedestrian model, the robot will move along a velocity vector with instantaneous acceleration. The robot model is defined as follows:

h_1 Spatial Representation:

$h_{1.1}$ Cartesian coordinates, (x_r, y_r), for the position of the AMR's centre within the modelled environment

$h_{1.2}$ Dimension of the AMR to be represented as a circle, with radius r_r

$$(x - x_r)^2 + (y - y_r)^2 = r_r^2 \tag{3.7}$$

$h_{1.3}$ Current bearing of the AMR, θ_r, in relation to the x-axis

h_2 Forward Kinematics:

$h_{2.1}$ Velocity , v_r, with axial components v_{rx} and v_{ry} over the 2 DoF

$$
\begin{aligned}
v_{rx} &= v_r cos\theta_r \\
v_{ry} &= v_r sin\theta_r
\end{aligned} \tag{3.8}
$$

h_3 Input Sensors:

$h_{3.1}$ The AMR's field of view (FoV) has both a distance range, r_{fov}, from the AMR's centre, and scanning range, θ_{fov}, equally spaced around the AMR's current bearing

$$\text{FoV scan} = \theta_r \pm \frac{\theta_{fov}}{2} \tag{3.9}$$

$$\text{FoV area} = \frac{{r_{fov}}^2\theta_{fov}}{2} \tag{3.10}$$

$h_{3.2}$ A uniform exclusion zone, EZ, is added around the robot . This added for the safety of pedestrians in order to prevent direct collisions, by immediately stopping the AMR if a pedestrian enters it

$$(x_r - x_p)^2 + (y_r - y_p)^2 \leq (r_r + EZ)^2 + {r_p}^2 \tag{3.11}$$

The robot model has been designed to have a windowed FoV, Item $h_{3.1}$. This more accurately replicates input sensors such as laser range-finders. Also, due to the dynamic nature of the pedestrian environment, predictions made outside the window will be unnecessary due to the increasing unreliability of the pedestrian model further into the future.

3.3.2 Environmental Mapping

A roadmap, Section 2.2.1.2, has been selected as the form of environmental representation, due to the customisation the resultant node network can provide [119]. As the safety of the pedestrians is paramount, the Voronoi diagram (VD), Fig. 2.5a, is chosen for the basic roadmap structure. The VD guarantees a roadmap of connected nodes that provide a maximum distance safety margin between objects, Fig. 3.9a, described by the vertex point (VP) in the centre of an Apollonius circle between 3 neighbouring objects and centre point of the minimum distance (MD) between each of the neighbouring ellipses. Connecting these describing points will generate an approximated VD of vectors, Fig. 3.9b, allowing the search algorithm to plan paths that keep furthest away from any ellipses and potential collisions. Unfortunately, the VD does not generate very direct paths when moving between objects, and so the visibility graph (VG), Fig. 2.5b, is also used. The VG generates a roadmap of connected nodes that ensures the most direct path can be taken when moving

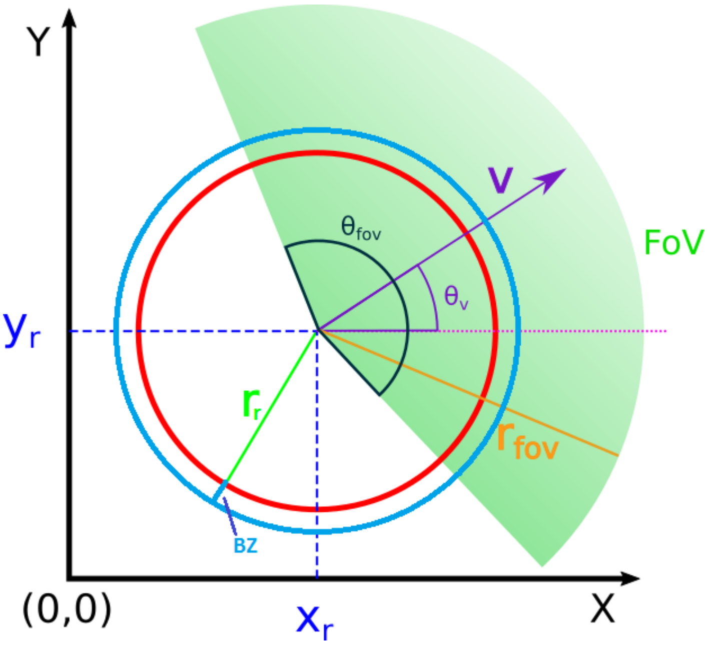


Fig. 3.8 The model of the AMR. The robot's spatial representation is modelled using a circle with centre (R_x, R_y) and radii R_r. Its trajectory, R_θ, is orientated relative to the x-axis, along which its velocity, R_v, is directed. The FoV the AMR uses to detect objects within has a distance range, R_{fov}, beginning from the centre of the AMR with a angular range, FoV_θ, equally spaced either side of its front. The buffer zone, BZ, surrounds the AMR as a safety precaution which stops the AMR moving if a pedestrian enters it.

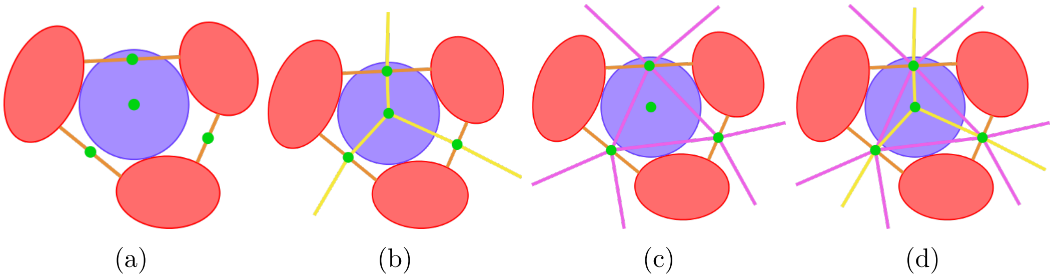


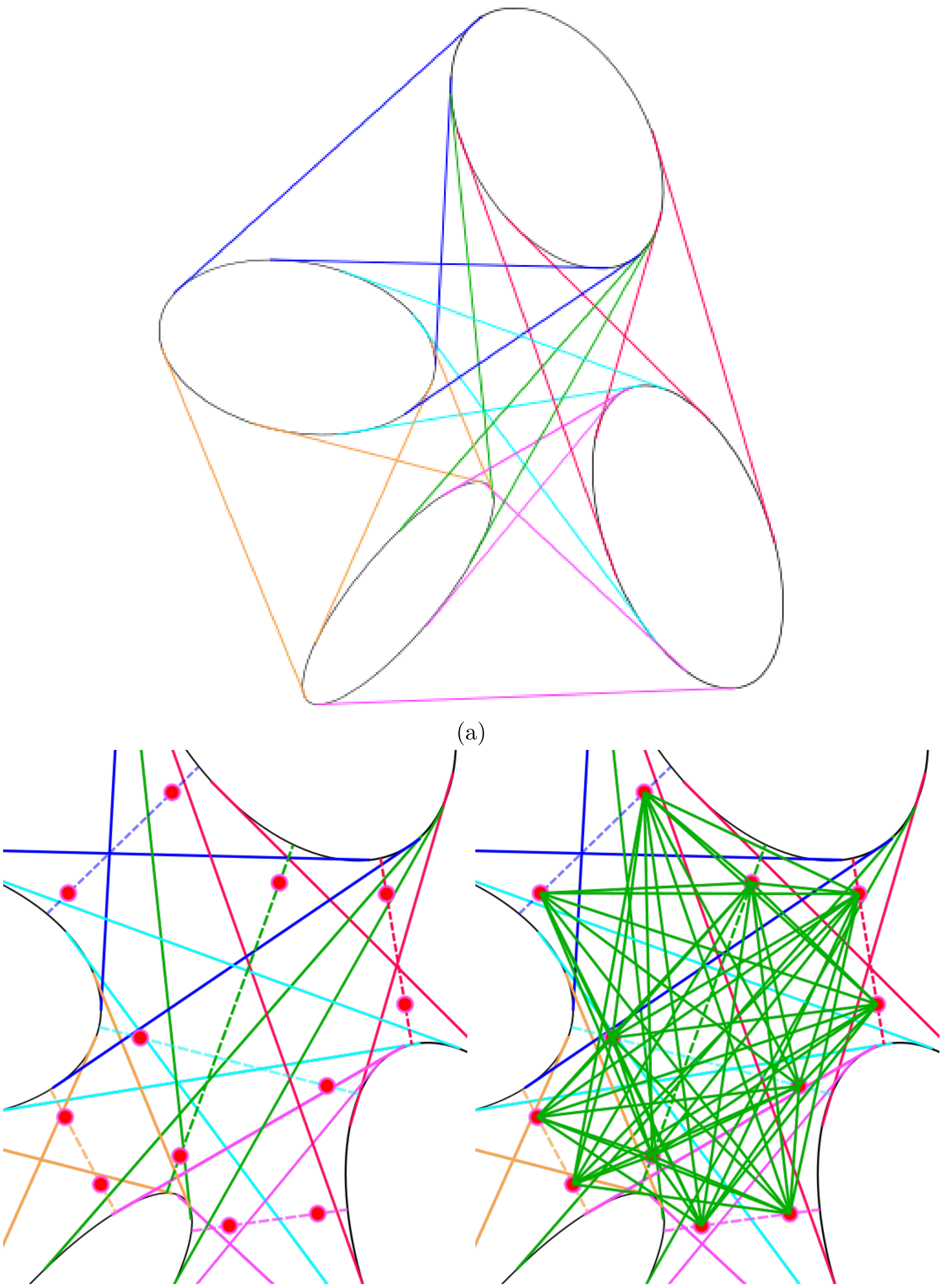
Fig. 3.9 Visualisation of the construction of the Euclidean vector roadmap. The blue circle is the Apollonius circle, which has its centre at the Voronoi vertex point. The orange lines are the MD between neighbouring ellipses. (a) The green nodes of the VP and MDs are the connectivity graph, from which the roadmap is formed. (b) The yellow lines are the approximated VD vectors. (c) The purple lines are the shortcuts that occur between MDs, in order to create a more natural flow. (d) The complete VD-VG roadmap of all interconnected nodes.

between objects, allowing more efficient paths to be planned. These connections assist in aligning the search more with the direction of the crowd. Shortcuts via MD connections are made that can jump between ellipses when planing paths, Fig. 3.9c, not confining paths that detour along the VD connections. Therefore, for the CPP to plan paths that are both safe and efficient a hybrid roadmap between the VD and VG (VD-VG) has been developed, Fig. 3.9d.

3.3.2.1 Modifying the Visibility Graph for Closed Curve Conic Sections

A traditional VG roadmap is formed between multiple 2D polygons, by evaluating if there is a direct line of sight between each polygon's corners and any other polygon's corners. Unfortunately, as a circle or ellipse is a closed curve conic section, there are no corner nodes to be connected between each shape. However, there are 4 distinct tangent points on each shape that can be connected to one another, Fig. 3.10a. These provide the most direct connections between objects, either moving alongside (non-intersecting) or crossing between (intersecting) them. However, these connections not only result in paths that are too close to the pedestrians, the tangent points that create the nodes are unique for each conic sections pair, and therefore each node is not guaranteed to connect to another.

To ensure nodes can connect to one another, an MD connection is made between two neighbouring ellipses. Two nodes close to both ellipses are created along this MD, Fig. 3.10b, which is also a describing point of the VD. By observing the resultant node network, Fig. 3.10c, connections can be made that represent similar VG connections, as seen in Fig. 3.10a, when travelling between ellipses. The result is a large interconnected network that requires a large search in order to plan any path. However, this approximated VG (aVG) can now have its node network reduced and be combined with the VD.



(a)

(b)

(c)

Fig. 3.10 A demonstration of modifying a visibility graph of points to use with ellipses. (a) Demonstration of up to four points connecting one ellipse to another, which are tangential to both. Those connections directly visible from one ellipse to another are comparable to traditional visibility graph connections. Multiple ellipses are unlikely to share the same points with other ellipses, making a connected node network not possible. (b) Proposed node network for the visibility graph of ellipses, formed from the unobstructed connections that form the minimum distance between ellipses. (c) A standard visibility graph can be generated from the proposed node network in (b). The network produces many connections that are similar to those seen in (a).

3.3.2.2 Hybrid Voronoi Diagram-Visibility Graph Roadmap (VD-VG)

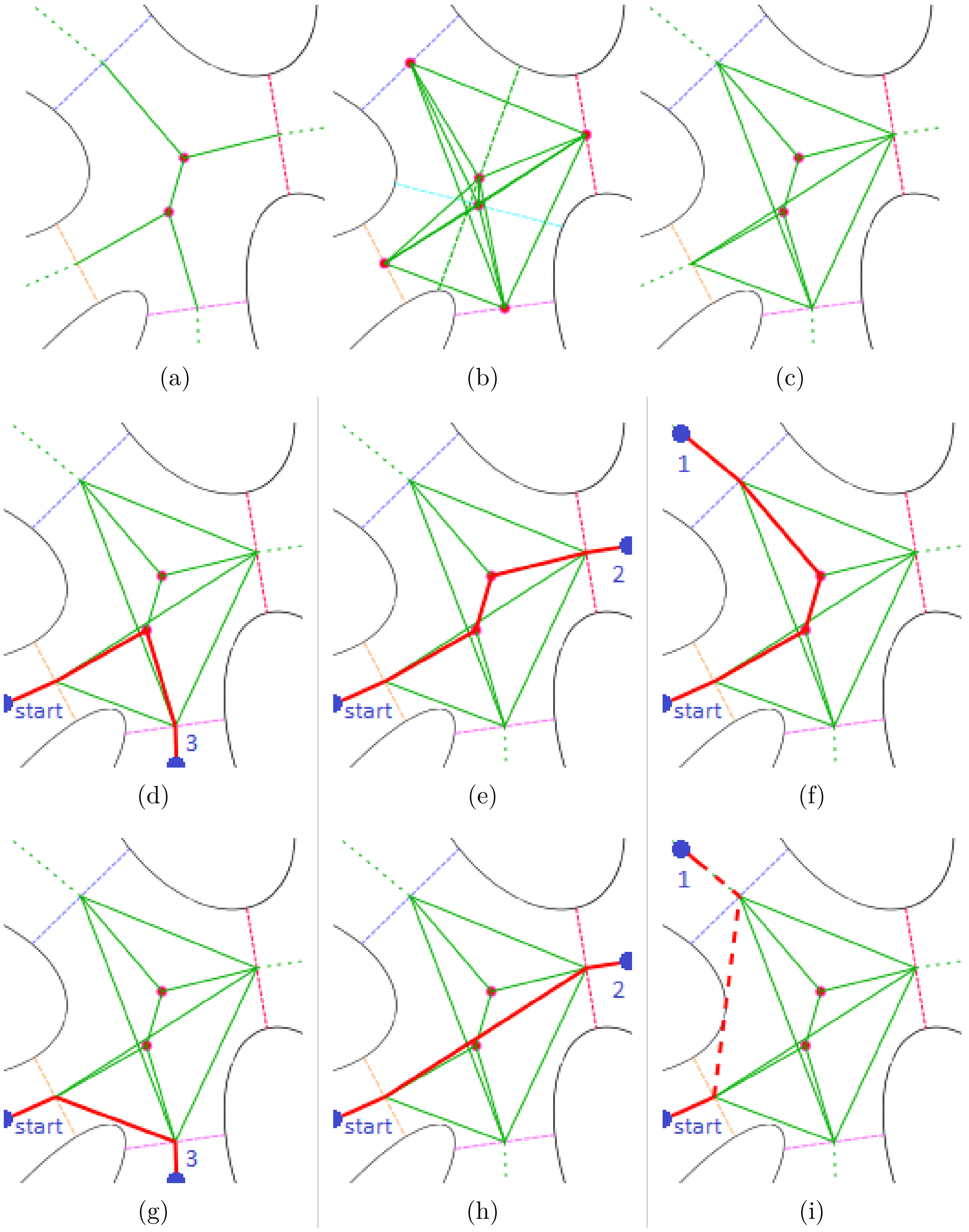
The aVG will have a lot of path connections, Fig. 3.10c, and provides a lot of potential path diversity, which is beneficial in path planning [154] in order to evaluate the potentials of different path possibilities, Section 2.4.2.1. Comparatively the VD, Fig. 3.11a, provides more limited but very specific path connections that ensure *safe* paths, by connecting vertex points with their associated MD connections. The aVG uses the points near the ends of the MDs, whilst the VD uses the MD's mid-points. If the aVG used these mid-point connections instead, Fig. 3.11b, it would still create connections that passed between all neighbouring ellipses, but with less path diversity and a larger clearance of the ellipses. However, by only using the MD mid-points the aVG can be fused with the VD to create created the VD-VG roadmap hybrid, Fig. 3.11c, which would possess the *safety* of the VD and path efficiency of the VG.

For whatever path finding search algorithm to be used, Section 2.2.2.2, it would be able to plan a route that could mix both elements of the VD-VG and change between safety and path efficiently at different moments along the overall path. Figs. 3.11d to 3.11f highlight an example path through the VD element of the VD-VG, from a start point at the bottom-left to three goals. Figs. 3.11g to 3.11i highlight the VG element of the VD-VG for the same scenario. The effectiveness of the VD-VG will be evaluated in Section 6.3 as an overall roadmap, as well as how effective the VD and the VG elements are when used as separate roadmaps.

3.3.3 Search Algorithm

Choosing an appropriate environment model is very important for path planning as it influences how the search algorithm is able to find a path to the goal [115]. As discussed in "World Perception" of the autonomy cycle, Section 2.1.1, the sensor detection range limits how much of the environment can be detected, which also limits how much of the environment can be searched for a path. The robot model, Section 3.3.1, has been designed with a windowed FoV, which for a dynamic environment is desirable as only what is within the FoV can be known for sure.

The windowed approach is beneficial at increasing search efficiency due to a smaller search area. Commonly used search algorithms, Section 2.2.2.2, are heuristic best-first searches such as A*. Variations of the A* algorithm have been developed for cooperative path planning [249] including "hierarchical" and "windowed hierarchical" additions. These all have a priori knowledge of the other agents' destinations, which can be considered as a form of non-verbal communication and lead to the assumption of cooperation. The cooperative windowed approach [249] limits the search to a fixed depth specified by the current window and their results show this has the highest destination arrival success rate. The benefits of this include the



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 3.11 Demonstration of how the Voronoi diagram of ellipses is combined with the modified visibility graph for ellipses, to create the Voronoi diagram-visibility graph roadmap. (a) A VD for ellipses that forms by connecting the vertex points and minimum distance connections between neighbouring ellipses. (b) A modified visibility graph for ellipses, Fig. 3.10c, which uses only one MD connection between ellipses rather than two. This is the same point shared by the VD. (c) The VD-VG formed from the original VD and the visible connections between the MDs in (b) (d-h) Example of the different red paths that start from bottom-left and move through the VD-VG to the 3 end points. (d-f) Demonstrates the VD paths possible. (g-h) Demonstrates the VG paths possible. (i) Shows the missing VG connection that was blocked by the ellipse on the left.

sophistication of a global path planning technique, confined to a local vicinity.

As the CPP focuses on moving the AMR *considerately*, the search should be encouraged to explore the environment. A path that is not desirable at the beginning, may be better overall come the end of the prediction time-horizon. The pedestrians between the AMR's sensor limits and the destination will be completely unknown. Therefore, an A* search heuristic that is heavily influenced towards moving to the goal does not benefit the path that the AMR may take. Rather the search should explore every direction within the FoV equally, selecting the final path based in the predicted interactions the AMR would have with pedestrians.

Therefore, a breadth-first Dijkstra's algorithm with no search heuristic is preferable to the best-first A* algorithm (as well as all other variants, Section 2.2.2.2). Despite the fact heuristic searches are often much faster, a windowed Dijkstra's algorithm is best for global exploration of an unknown windowed environment. It will search all of the area within the AMR's FoV quickly without searching unnecessary areas of the environment, which it will do in an unconstrained environment.

3.3.4 The Considerate Weights

The frontier of the windowed Dijsktra's algorithm is a representation of all the potential positions the AMR could be at a given moment. To plan a considerate path the search is weighted with a resistance, Ω, that affects the speed of the search as it moves along the VD-VG roadmap. Along the frontier of the search the model robot's dimensions are factored into the search as it is performed. This is to include the impact the AMR may have on the pedestrians as it moves within the crowd. From any point, p, within the Euclidean plane of the model environment Ω is calculated, which predicts how considerate the AMR will be towards the surrounding pedestrians from that point, $\Omega(p)$. The frontier of the search algorithm, S, moves along the roadmap connections at the model robot's maximum potential speed, $\max(v_r)$, and is slowed down based on $\Omega(p)$,

$$S_v(p) = \max(v_r) - f(\Omega(p)) \tag{3.12}$$

where S_v is the current speed at which S moves along a roadmap vector. A function of $\Omega(p)$, covered later when implementing the algorithm in Section 5.2.1, therefore reduces the speed at which the frontier travels along the roadmap. The resistance is based on the relationships between the robot, R, described by:

i_1 Position, $R_{x,y}$

i_2 Radius, R_r

i_3 Orientation, R_θ

and the pedestrian, P, described by:

j_1 Position, $P_{x,y}$

j_2 Radius, P_r

j_3 Orientation, P_θ

j_4 Velocity, P_v

Ω is calculated based on three separate elements, which include:

k_1 Ω_p - Euclidean proximity of the robot to neighbouring pedestrians

k_2 Ω_θ - Relative trajectory collision potential between robot and neighbouring pedestrians

k_3 Ω_{bz} - Collision potential between robot and buffer zones that circle static pedestrians

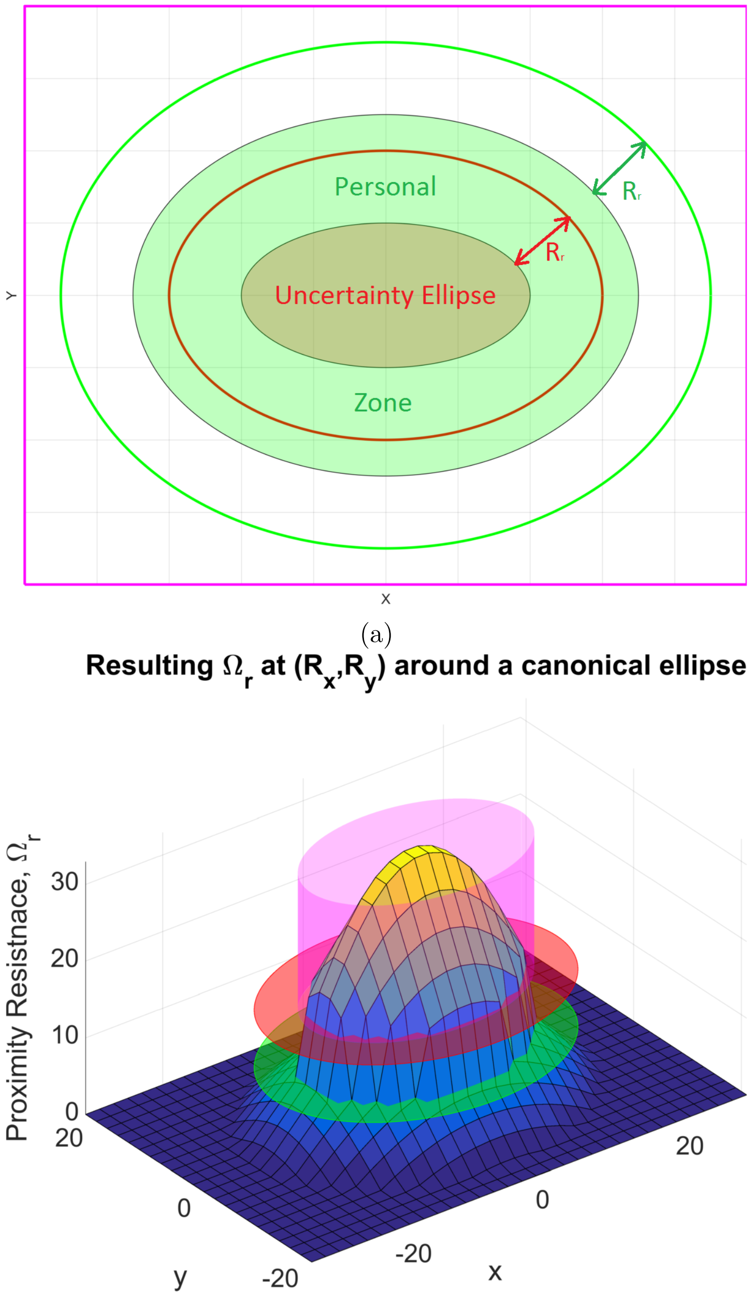
$$\Omega = (\alpha\Omega_p \times \beta\Omega_\theta) + \gamma\Omega_{bz} \tag{3.13}$$

where α, β, γ are user defined weights.

3.3.4.1 Proximity Resistance, Ω_p

Assessing the proximity of the robot to other pedestrians is an important factor towards considerate movement for two reasons: 1) Moving too close to a dynamic agent will increase the collision potential, as there will be less room to manoeuvre away if the agent behaves unexpectedly; and 2) From a social point of view there are certain distances that humans prefer to maintain between one another, which also occurs between AMR's and humans [218, 250]. Both of these will help to achieve the objective of not increasing crowd congestion.

The resistance component for proximity, Ω_p, is calculated based on the Euclidean distance between the search and pedestrians. This concerns the desired spatial ranges that surround people, Table 2.4, which will surround the entire uncertainty ellipse, Fig. 3.12. The *personal zone* (PZ) has been selected for the robot to ideally remain outside of, as it is assumed that the robot is not considered a "friend". If the search is outside of the PZ, no resistance will be added. If the search enters the PZ the Ω_p will equal the distance to the PZ's edge. If the search enters the uncertainty ellipse Ω_p will equal double the distance to the PZ's edge. This is so the search can incur an immediate penalty as the potential for a collision increases dramatically, due to the model's assumption that a pedestrian is inside the uncertainty ellipse.



(a)

(b)

Fig. 3.12 Demonstration of how an AMR's proximity to the uncertainty ellipse of a pedestrian affects the related resistance, Ω_p, due to the proxemics of the pedestrian's personal zone, Table 2.4. (a) The uncertainty ellipse is highlighted red and the personal zone surrounding it in green. To measure the proximity of the AMR to the ellipse the distance is taken from the AMR's centre, and so the ellipse's axes are increased by the AMR's radius, Section 3.3.1, to the red and green lines in order to perform the calculations. (b) A visualisation of how the resultant resistance, Ω_p, is affected by the distance of the AMR to the uncertainty ellipse. If the AMR is inside the green personal zone Ω_p is calculated as the distance from the AMR to the ellipse edge. If the AMR passes into the uncertainty ellipse Ω_p is immediately doubled, in order for the AMR to incur an immediate additional penalty. The uncertainty ellipse is highlighted magenta, and the green and red ellipses in the horizontal plane highlight the immediate increase of Ω_p as the AMR enters the uncertainty ellipse.

Zone	Description	Impact
Spatial_r	Radius r_r of robot behind pedestrian	Collision potential rises as the robot begins to intersect the rear of the ellipse
Rear	Rear S-Ma axis of the ellipse	Collision potential increases as pedestrian is most likely in the front S-Ma axis of the ellipse
Front	Front S-Ma axis of the ellipse	Maximum likelihood of a collision as most likely area to collide with pedestrian
Spatial_f	Radius r_r of robot in front of pedestrian	Area to avoid as much as a real collision, as would be cutting directly in front of a pedestrians trajectory
Velocity	The predicted movement of the pedestrian in the next t_s	Collision potential decreases the further ahead of the ellipse

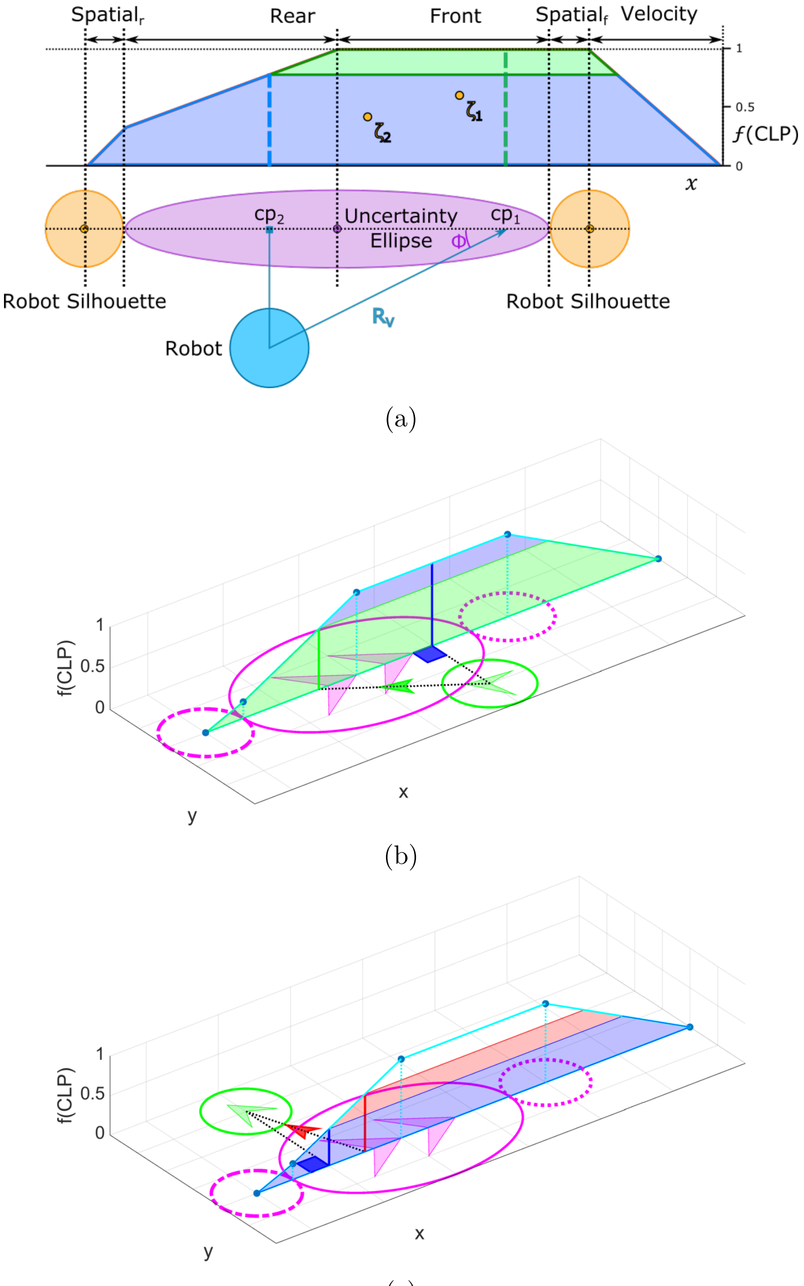
Table 3.3 The five zones shown in Fig. 3.13a. The zones are used to calculate the CoG, based on collision potential between robot and pedestrian.

3.3.4.2 Relative Trajectory Resistance, Ω_θ

Assessing the relative trajectory between the robot and the agent will allow the CPP to assess the impact of any upcoming potential collisions. The relative trajectory can establish whether or not the search is moving in the same direction as the pedestrian. By normalising the relative angle so that movement in the same direction is zero, and movement in a head-on collision is one, the severity of a potential collision can be established. For example, a path moving with the flow of a crowd would generate minimal resistance, whilst a path moving against would generate a considerable resistance. This will contribute towards achieving the objectives of, moving with the flow of the crowd; and not cutting in front of pedestrian trajectories.

The resistance component for collision potential, Ω_θ, is calculated based on the relative trajectory and position of the search w.r.t. the pedestrian's uncertainty ellipse. There will always be a collision point between the trajectory of the model robot, at the frontier of the search, and the pedestrian's velocity line at c_p. This point then maps onto a collision likelihood profile (CLP), Fig. 3.13. The CLP is divided into zones regarding factors that are likely to contribute towards different forms of collision potential, Table 3.3. Where c_p maps onto the CLP will determine what kind of collision may occur, whilst the intersection angle, c_θ, will establish the intensity.

3.3.4.3 Buffer Zone Resistance, Ω_{bz}

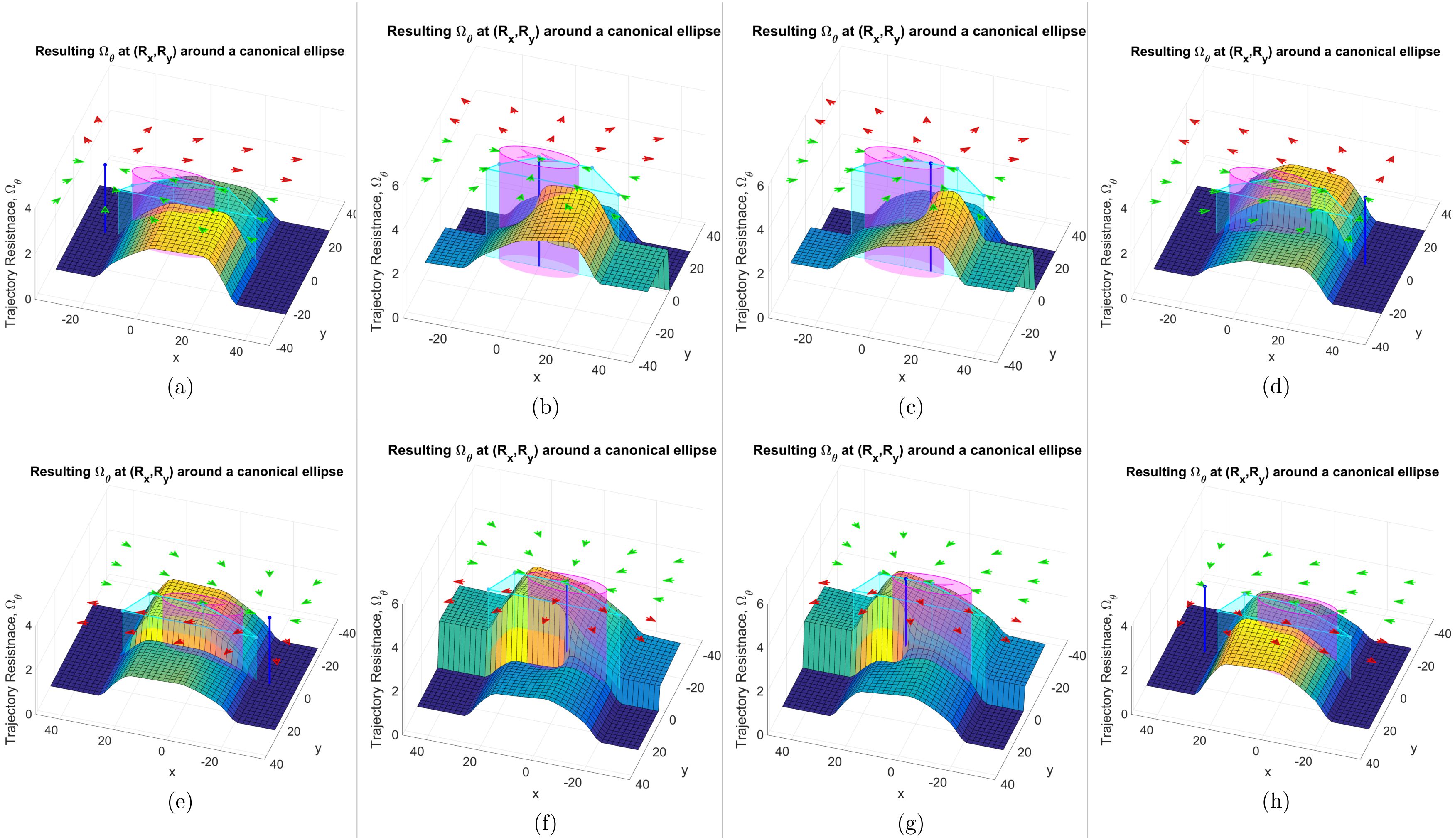


(a)

(b)

(c)

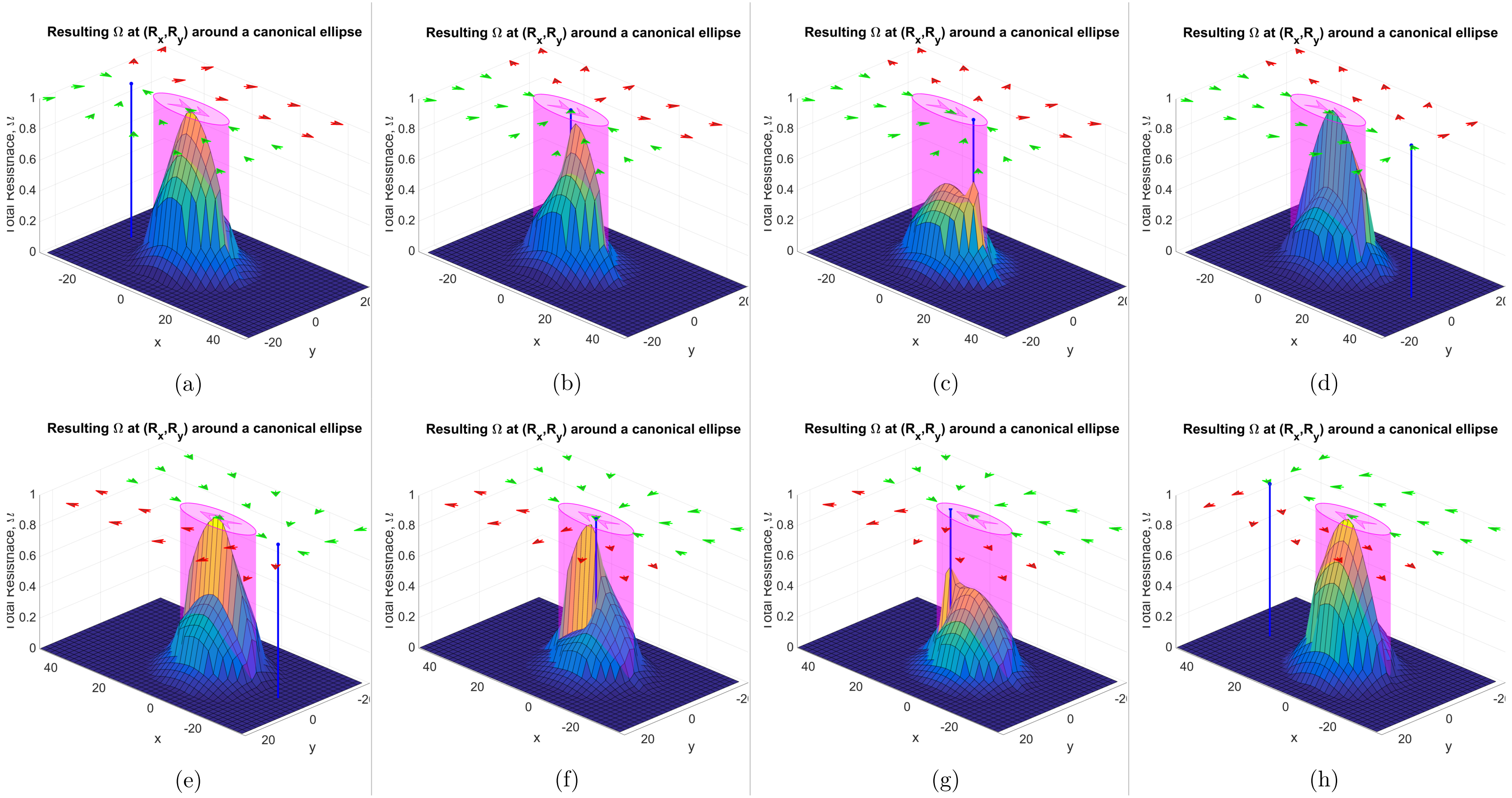
Fig. 3.13 Demonstration of how the CLP is used to calculate the likelihood, and severity, of a collision. (a) The CLP is outlined red, with five distinct regions along the x-axis, Table 3.3, and a range 0 to 1 along the y-axis, $f(\text{CLP})$. The CoG is calculated based on the intersection between the ellipse and the robot trajectories, c_t, and the AMR's proximity, c_p, calculating the CoG average from the area underneath the CLP. The CoG will assess the likelihood of a potential collision, from least likely (0) to most likely (1). The relative trajectory, θ, of the AMR to the ellipse dictates the severity of the potential collision, by evaluating how direct a head-on collision may be. It will quantify the impact level between robot and agent, which ranges from *head-on*, $\theta = 180^o$ (1), to *moving-with*, $\theta = 0$ (0). Additional *Spatial* areas are added to the rear and front of the ellipse, which correspond to the the robot's radius, to include how much of the AMR may collide. (b) An example of the AMR moving towards a hypothetical collision with the *rear* of the ellipse, with a CLP value of $\frac{\theta}{180}(c_t + c_p)$. (c) An example of the AMR moving away from a hypothetical collision with the *rear* of the ellipse, with a CLP value of $\frac{\theta}{180}c_p$.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3.14 Example of the "Relative Trajectory Resistance, Ω_θ" function. The graphs show Ω_θ (z-axis) from all positions surrounding the ellipse. The magenta ellipse shows the actual ellipse and the arrows highlight the direction of the agent. The blue vertical lines highlight examples of 4 hypothetical collision points, c_p, that the search/model robot intersects the CLP, Fig. 3.13a (which is also shown above the ellipse in cyan). The side of the graph with green arrows above it shows examples of Ω_θ as the search moves towards c_p. The side of the graph with red arrows above it shows examples of Ω_θ as the search moves away from c_p. (a-d) Show the side of the graph where the search is moving towards c_p. (e-h) Show the same graphs as (a-d), but from the other side where the search is moving away from c_p.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3.15 Example of the combined total resistance of "Proximity", Ω_p (Fig. 3.12), and "Relative Trajectory", Ω_θ (Fig. 3.14), as defined by Eq. (3.13). All sub-figures reflect the same perspective as the sub-figures of Fig. 3.14.

As mentioned in Section 2.4.2.2, within a dynamic environment a path should always be planned even if there are potential collision en route. This is achievable thus far for the CPP as both Ω_p and Ω_θ only slow the search down, but do not stop it. However, if pedestrians are stationary and the uncertainty ellipse degenerates to a circle, this area should be avoided entirely due to the obvious risk of collision. These are specific no-entry exclusion zones (EZ) that are also added around any points where a collision has recently occurred as the AMR moves along in the real-world. The latter is intended to avoid another collision again with the same pedestrian when moving along the next planned path. Surrounding these EZs are buffer zones (BZ) that attempt to discourage the search moving towards any EZ by increasing the resistance it encounters, the closer and more directly it moves towards it.

The resistance component for the BZ, Ω_{bz}, is calculated based on the position and trajectory of the robot w.r.t. the EZ. Similarly to Ω_p, the search must be within a certain distance of the edge of the EZ for Ω_{bz} to be calculated and is dependent upon distance the search has entered the BZ, Fig. 3.16. The relative angle, θ_{bz}, between the model robot at the frontier search and the center of the EZ, is normalised. Therefore, when the robot directly faces the EZ $\theta_{ez} = 1$, and any angle $\pm > 90°$ from the EZ $\theta_{ez} = 0$. This will help prevent the search colliding with the EZ, as a path that moves away would incur less resistance.

3.4 Chapter Overview

This chapter has presented the requirements for developing a *considerate path planning strategy. One of t*he key elements to path planning in a dynamic environment is suitable modelling of the dynamic agents [55]. As pedestrians are agents, and agent-based models are dominated by short time-horizons, the use of a CVM is justifiable for short-term trajectory extrapolation. The uncertainty in the CVM predictions can then be confidently calculated, so that the end prediction result becomes confident enough. Collisions can then be avoided, and the CPP can predict considerate paths that will avoid all the potential positions of each pedestrian.

The direct movement of a pedestrian is dictated by what they can visually perceive. Their Markov behaviour also means that a prediction of their future movements can only be made on their current movement. The use of the CVM over a limited time-horizon is reinforced by this, whilst the novel uncertainty ellipse surrounding the pedestrian's potential positions *embraces* the uncertainty. Applying the ellipse to all individual pedestrians will cover their walking deviations, based on their current velocity, and their adjustment for collisions due to the proximity of other pedestrians within their FoVs, based on empirical proxemics.

The *considerate weights* are dynamically added alter the speed of the windowed Dijkstra's algorithm in order to search for a suitable path within the AMR's FoV.

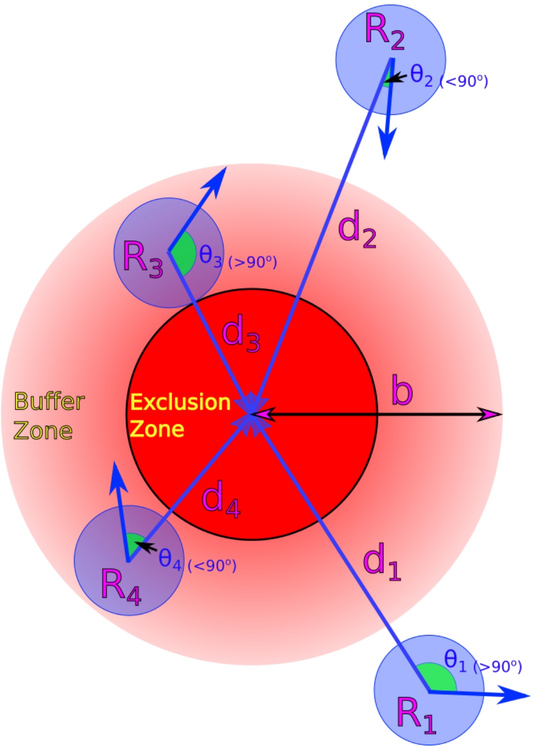


Fig. 3.16 Visualisation of how the buffer zone (BZ) that surrounds an exclusion zone (EZ) affects resistance, Ω_{ez}, with four example robots, R_1 to R_4. Only if the current trajectory of the robot will cause a potential collision will it affect Ω_{ez}. R_1 is *outside* the BZ ($d_2 > b$) therefore $\Omega_{ez} = 0$. R_2 is moving *towards* the BZ ($\theta_2 < 90^o$), however it is too far away to generate a resistance. R_3 is *inside* the BZ ($d_3 < b$), however it is moving *away* ($\theta_3 > 90^o$), therefore there is no collision potential. R_4 is *inside* the BZ ($d_4 < b$) and is moving *towards* ($\theta_3 < 90^o$), therefore $\Omega_{ez} = \frac{\theta_4}{d_4}$, as a collision is more likely to occur. The BF is half the radius of the robot for a static EZ, and the diameter of the entire robot when a collision EZ.

This modifies the search to include how the AMR may interact with pedestrians through proximity and relative trajectory between them both. The next chapter will present the algorithmic implementation of the VD-VG roadmap for ellipses.

Chapter 4

A Novel Voronoi Diagram-Visibility Graph Hybrid Roadmap

"Nothing behind me, everything ahead of me, as is ever so on the road."

— Jack Kerouac, *On the Road*

THE roadmap to be used by the considerate path planner (CPP) is a Voronoi diagram-visibility graph (VD-VG) hybrid. As established in Section 3.3.2, a Voronoi diagram (VD) provides an excellent fundamental roadmap for the CPP, as it creates a roadmap that will always be the *safest* distance from any object, whilst the visibility graph simultaneously generates more direct connections of many shared nodes. The VD-VG uses the 2 describing points of the VD, the vertex points (VP) and the minimum distances (MD) between neighbouring objects.

The pedestrian movement model developed in Section 3.2 represents each dynamic pedestrian using a *uncertainty* ellipse. However, a true VD of fully-intersecting ellipses has currently not been solved, only a VD of partially-intersecting ellipses [251] has been obtained. To combat this problem, approximations of the pedestrians' uncertainty ellipses are formed using tangentially aligned circles, hereby referred to as pseudo-ellipses (PE). Using the VD of circles sweepline algorithm [72], a novel modification is made that can generate the VD of PEs. The VD of these pseudo-ellipses (PE) generates a very similar VD to the original ellipses, using an algorithm with logarithmic time complexity.

This chapter begins with the novel adaptation of the VD of circles, Section 4.1, used as a precursor to approximate the VD of real ellipses later. As the VD of PEs will be only an approximation of the VD of ellipses, some VPs will not be discovered. To try and discover these missing VPs, estimation methods are then used, Section 4.2, in order to increase the level of completeness when forming the VD of real ellipses. Only a few iterations are required to find the VPs of real ellipses to within a high degree of accuracy. The VD of real ellipses is then approximated, Section 4.3, by converging the VPs of PEs onto the locations of the VPs of the uncertainty ellipses. The associated MDs are then asymptotically converged upon from the newly discovered ellipses' VPs.

Finally, using the uncertainty ellipses' VP and MD describing points of the associated VD, the VD-VG roadmap is constructed, Section 4.4. The roadmap will be confined to within the autonomous mobile robot's (AMRs) field of view (FoV). As covered by the model robot in Section 3.3.1, a roadmap within a dynamic environment should only be constructed using data that can be immediately observed by the AMR.

4.1 A Novel use of Circles to Approximate the Voronoi Diagram of Ellipses

As mentioned in Section 2.2.1, there are numerous methods of environmental segmentation. The novel application of *consideration*, presented in Section 3.3.4, relies on *resistance* being added to the path, which can be applied to any of these methods. However, the VD inspired roadmap method was chosen for this thesis as it extracts the equal spaces in between objects, allowing the robot to navigate effectively *through* the crowd using a path that can maintain the *safest* distance.

Unfortunately a VD of arbitrarily placed and fully-intersecting (4 unique intersection points) ellipses has not yet been solved. Analytically it is intractable as there are more unknown variables (9) than equations (3)[1]. However, the VD of *partially*-intersecting (2 unique intersection points) ellipses has been solved by using "smooth convex pseudo-circles" [251]. Unfortunately, this is insufficient for the CPP as the pedestrian movement model, Section 3.2, can easily produce *fully*-intersecting ellipses. To overcome this problem a novel adaptation of the VD of circles [72] is proposed. Four circles are tangentially aligned to form PEs, for which the VD can be easily found.

Sweepline algorithms are efficient methods for computational geometry in Euclidean space [252]. The sweepline itself is a self-balancing binary tree, which generates a worst-case time complexity of $O(n \log n)$, where n = number of events to assess. For a general sweepline algorithm the sweepline translates/sweeps a line in a constant direction across a plane in 2D Euclidean space, pausing at points of interest. Geometric operations are performed at every pause, describing objects the sweepline is intersecting, as well as objects in the immediate vicinity of the line that it has

[1] the minimum distance between two ellipses requires a 12-dimensional polynomial

already passed. The sweepline algorithm for the VD of circles [72] can be created in this way by using a parabola, which forms a locus between the sweepline and a circle. This allows the distances to be equated accurately wherever the sweepline is positioned [253]. The VD is mapped according to the points of intersection between neighbouring parabolas: two intersecting parabolas create vertex lines that trace a line of equal distance between the circles, whilst three intersecting parabolas create the VP between three neighbouring circles, Fig. 4.1. However, due to representing the VD using particular geometric shapes, the sweepline algorithm must include three assumptions [254], which apply to the VD of circles in the following ways:

a_1 Numerical computation must be carried out with high precision arithmetic, as parabola-parabola intersections are used to trace the VD lines. As a result, the unique intersection point of 3 parabolas describes a VP.

a_2 No four Apollonius points (points on the circles surface which correspond to creating a VP between 3 circles) align on a common circle (assuming non-degeneracy). This will cause two VPs to occupy the same space simultaneously.

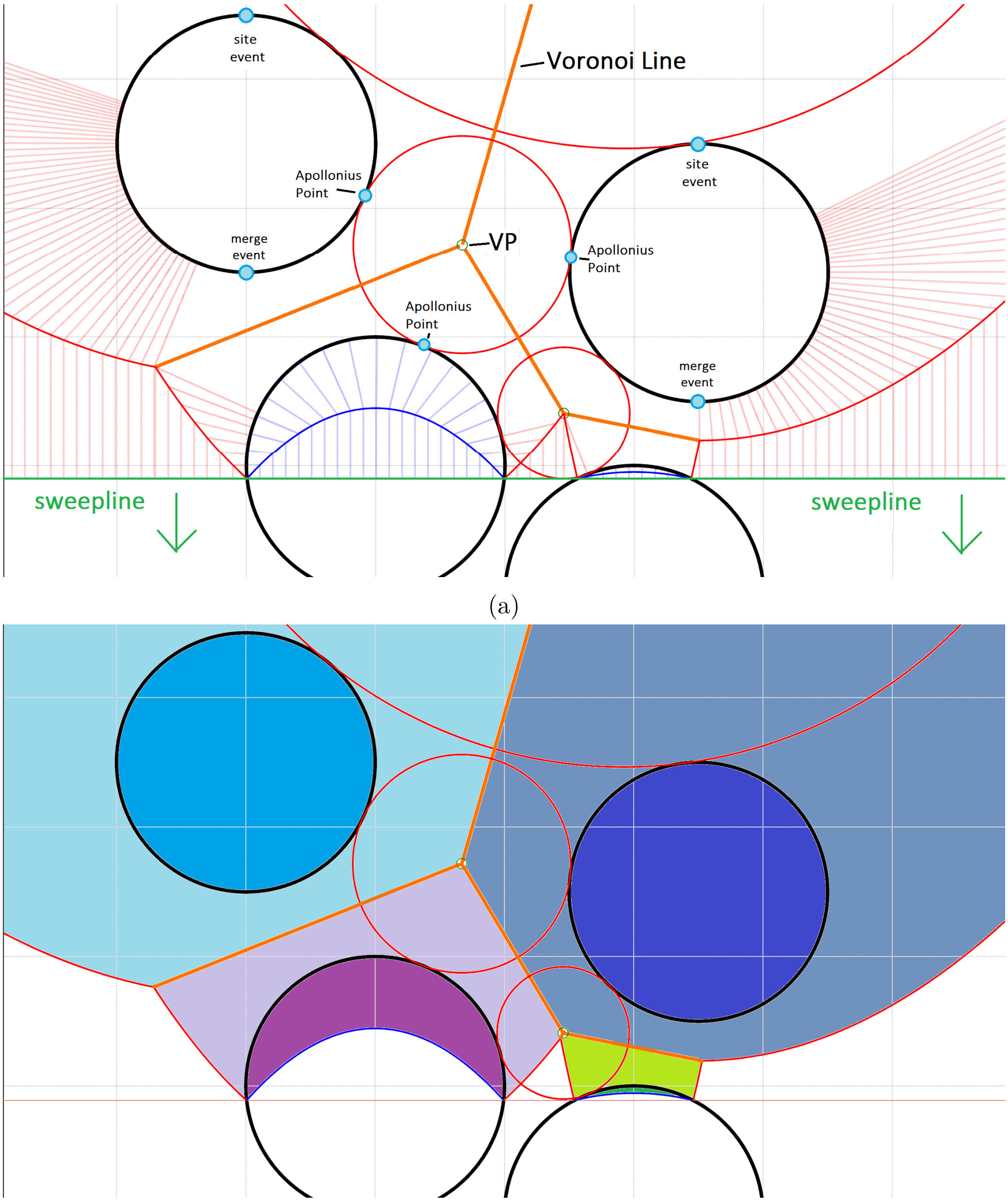
a_3 No two event points (points at either the top of the circle, bottom of the circle, or circle-circle intersections) occupy the same space.

a_4 The three circles that create a VP do not have horizontally aligned site events (points at the top of the circle). Therefore, an Apollonius circle (circle which aligns with 3 Apollonius points of 3 neighbouring circles, with the VP at its centre) will not exist.

All circles have equidistant radii from their centre points, which is where the parabolas' focus points are positioned. However, as the event point that creates the parabola is at the top of the circle, this creates a shifted parabola relative to the circle's centre and radius. The outside of the circle is described using a standard parabola, whilst the inside is described by an inverted parabola around the same point, Fig. 4.1a.

This thesis presents a novel adaptation of the VD of circles, which takes advantage of the nature in which these parabolas describe each circle, in order to estimate the VD of ellipses. Approximations of real ellipses are formed using four tangentially aligned circles, which create PEs, Fig. 4.2. Using the addition of a novel *tangent* function, the VD of PEs can be found with only a minimal increase in time complexity when compared to the original VD of circles: $O(n^2 \log n)$ to $O([n^2 + 4n] \log n)$, where n = number of circles.

An ellipse can be approximated using four tangentially aligned circles that produce the same semi-major (S-Ma) and semi-minor (S-Mi) axes as the original ellipse, creating a PE, Fig. 4.2. A novel *tangent* function is then implemented, which introduces a new point-of-interest to the sweepline algorithm at this tangential intersection. It is at this point that the parabolas of these tangential circles are themselves



(a)

(b)

Fig. 4.1 Demonstration of how the sweepline algorithm uses parabolas to describe the size and position of circles, using the circles' directrices as the green sweepline that translates down the page. (a) Visualisation of the original sweepline algorithm for the VD of circles. The sweepline translates down the page, manipulating parabolas to describe the equal perpendicular distances from sweepline to parabola, and parabola to circle edge (dashed lines). When two neighbouring parabolas intersect, orange VD lines are traced. When three neighbouring parabolas converge on the same point a VP is created, shown with its red Apollonius circle. (b) The solid coloured areas above the parabolas highlight the 'safe' areas, which will remain the same regardless of any future event. All of the white space below the parabolas may be changed by a future event.

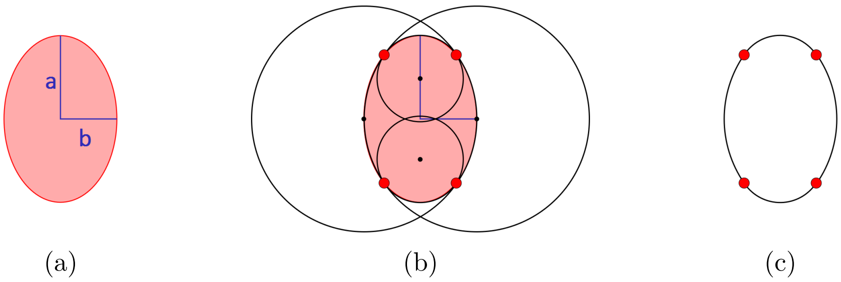


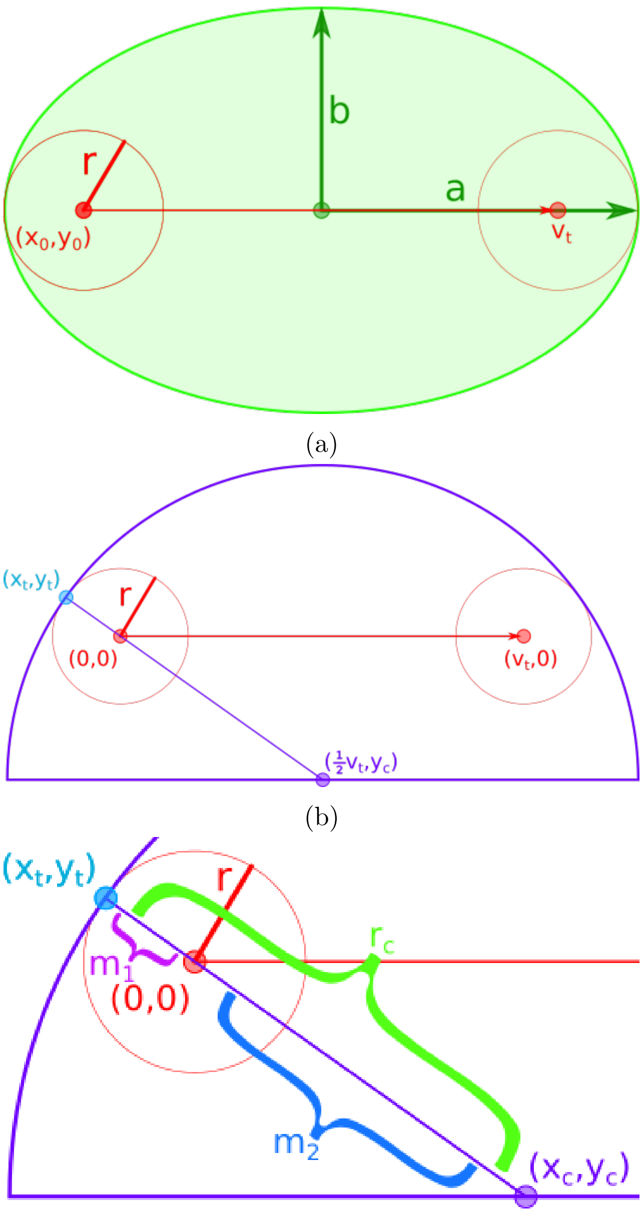
Fig. 4.2 Visualisation of how a PE is created, by tangentially aligning four circles. (a) The original ellipse, with S-Ma axis a and S-Mi axis b. (b) Four circles are aligned so that each one is tangential to an axis limit of the ellipse. When correctly aligned, and appropriate radii are appointed to each circle, each circle will tangentially align with its neighbour (red dots). (c) The PE is created by only using the circle arcs that align with the ellipse axes. Each circle arc will tangentially align with its neighbour at the red dots, creating the smooth convex shape of the PE.

tangential. This remains true as the sweepline continues to translate, allowing both parabolas to describe their respective circles, but also the point at which they tangentially align. The execution process responds in the same way as the original VD of ellipses, using parabolas to describe the outline of the 4 circle arcs that form the PE. The VPs of the VD found through this method, provide excellent initial approximations to the VPs of the original ellipses. Using the bi-section method the VPs of the original ellipses can be found in only a few iterations.

4.1.1 Approximating the Uncertainty Ellipse with Circles

To convert the uncertainty ellipse into a PE, the edges of four circles must tangentially align with the axes of the ellipse, as well as each other, Fig. 4.2. Two smaller circles represent the S-Ma axis limits, whilst the positions and radii of two larger circles, representing the left and right S-Mi axes, can easily be calculated using the mid-point between the two smaller circles, Fig. 4.3. Algorithm 1 provides the generic pseudo-code to describe mathematically how to convert an ellipse into an approximated PE.

As the PE is an approximation of the ellipse there will be some errors, as the PE will always be inside the ellipse. However, these errors are small, Fig. 4.4, and when comparing the area of the ellipse to the converted PE, the area difference ranges from zero to 15% smaller. This maximum error occurs regardless of the ratio between the axes, whilst the axis limits of the ellipse remain flush with the PE. The tangent points that align the four circles together now become individual events that the sweepline must pause at and consider, as it translates across the workspace plane.



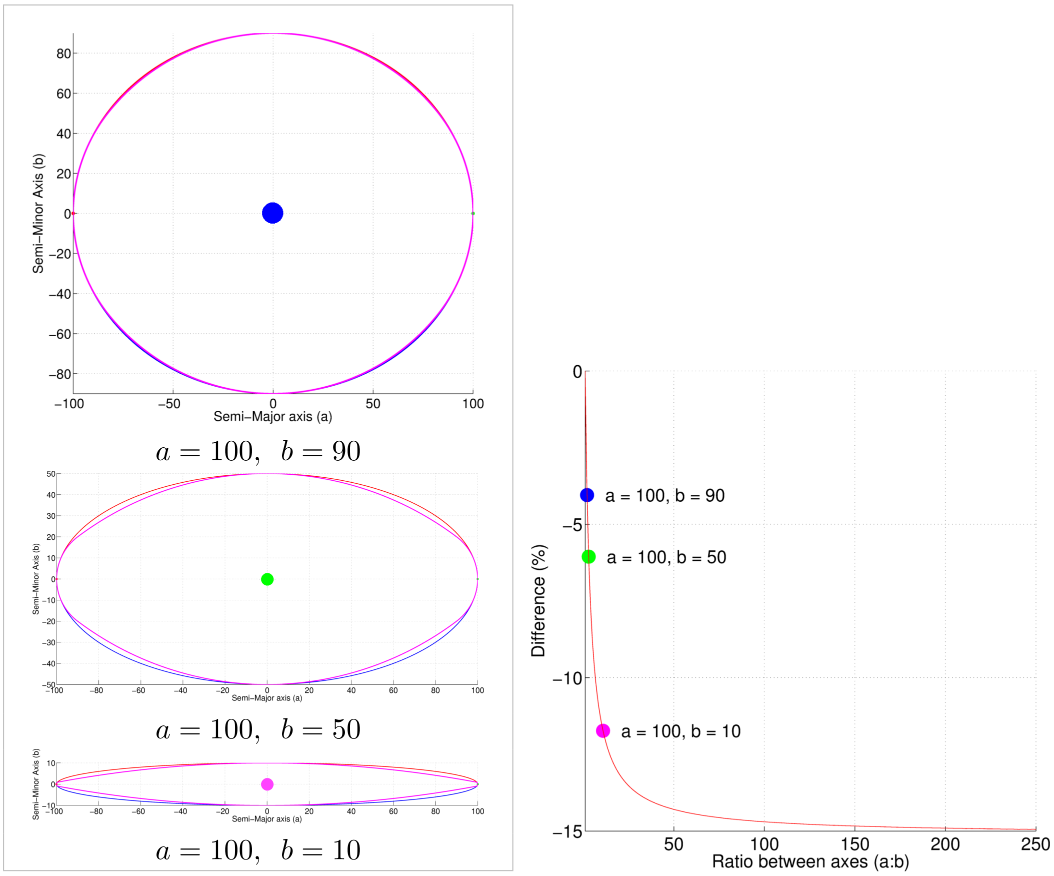
(a)

(b)

(c)

Fig. 4.3 Calculation steps required to compute the pseudo-ellipse, as performed by Algorithm 1. (a) From the pedestrian's observed position at $t = 0$ (x_0, y_0), S-Ma axis, a, is calculated based on the predicted velocity, v, at current time-step, t (v_t). The S-Mi axis, b, is calculated based on the uncertainty profile, Fig. 3.6. (b) (x_0, y_0) is positioned at the origin, $(0, 0)$, and the S-Ma axis is aligned with x-axis, for ease of calculation. Using Algorithm 1 (Line 1) the centre point of the circle is found, which will tangentially align with S-Mi axis of ellipse, (x_c, y_c). The tangential intersection point between the larger S-Mi and smaller S-Ma axis circles forms at (x_t, y_t). The x-coordinate of the S-Mi circle, x_c, is the mid-point between the agent circle at $t = 0$ and t. (c) Using Algorithm 1 (Line 3 to Line 7) the y-coordinate of the S-Mi circle, y_c, is calculated and its radius, r_c, is easily found. The coordinates that tangentially align the circle (x_t, y_t) are found my comparing the comparing gradients of m_1 and m_2.

```
Algorithm 1: Ellipse to Pseudo-Ellipse conversion
/* Fig. 4.3 */
/* Assume original position of agent $(x_0, y_0)$ is at origin $(0, 0)$, and velocity is aligned
   along the x-axis. This will make make calculation easier, and the final elements should
   be rotated and positioned relatively to the real-data. */
Calculate the mid-point of the ellipse along S-Ma axis, which is the centre
  point of the S-Mi axis circle, $x_c$;
$\therefore$ if calculating the PE using ellipse parameters then
    $x_c = \frac{a^2 - b^2}{a}$ ; // convert real ellipse into a pseudo-ellipse
else if calculating the PE directly from the agent velocity observation then
    $x_c = \frac{v_t}{2}$ ; // create pseudo-ellipse straight away
$x_0 = 0$; $y_0 = 0$; // from initial assumption
for $i = -1$ to $1$ do
    // when $i < 0$ calculate left, when $i > 0$ calculate right
    find the remaining elements of the S-Mi axis circle, y-coordinate, $y_c$, and
      radius, $r_c$;
    $r = a - x_c$ ; // radius of smaller S-Ma axis circle
    $x_c^2 + y_c^2 = (r_c - r)^2$ ; // distance from S-Mi axis circle to tangent point
    $r_c = b - y_c$ ; // radius of the S-Mi axis circle
    $\therefore y_c = \left( \frac{r^2 - {x_c}^2 - 2rb + b^2}{2(r - b)} \right) \times i$;
    $\therefore r_c = b - (y_c \times i)$;
    /* circles tangentially align */
    calculate the tangential point $(x_t, y_t)$ between the 2 circles at $t = 0$ (left
      S-Ma axis);
    /* substitute following equation (1) into (2) */
    (1) $\frac{y_t - y_0}{x_t - x_0} = \frac{y_t - y_c}{x_t - x_c}$; // gradients are equal
    (2) $(y_t - y_c)^2 + (x_t - y_t)^2 = {r_c}^2$; // distance from the circle centre to the
      tangent point is also the radius
    calculate the tangent between the 2 circles at current time-step, $t$ (right
      S-Ma axis);
    $\therefore x_0 = 2x_c$;
    $i = i + 2$;
rotate and translate tangent points to their respective co-ordinates to the
  ellipse's original position and angle;
return;
```



$a = 100, \quad b = 90$

$a = 100, \quad b = 50$

$a = 100, \quad b = 10$

(a)

(b)

Fig. 4.4 Error between the original ellipse and the pseudo-ellipse. At the axis limits of the ellipse the dimensions of the real and pseudo-ellipses are identical. (a) Three examples of PEs over the original ellipses. The coloured centre points are displayed in (b), to show their error values. (b) Quantitative error with respect to the differences in area between the ellipses and the PEs. The error converges at −15%, regardless of the ratio between S-Ma and S-Mi axes.

4.1.2 The Novel Tangent Function

Similar to the original VD of circles sweepline algorithm, Fig. 4.1, the novel *tangent* adaptation concerns the four parabolas associated with each of the four circles that form the PE. The parabolas used to describe two tangential circles will also be tangential at the same relative points as the circles themselves. The two parabolas are merged together at this tangential point, Fig. 4.5a, clipping them both so that the two segments of the parabolas describe their respective segments of the circles Fig. 4.5b. This merged parabola forms a *super* parabola, Fig. 4.5c, allowing the convex PE to be described in a same way as a circle. As the sweepline moves past the *tangent event* the parabolas will continue to remain tangential at the point that describes it on the PE, Fig. 4.6.

4.1.3 Executing the Modified Sweepline Algorithm

The novel modified sweepline algorithm for the VD of PEs executes in exactly the same way as the original sweepline algorithm for the VD of circles [72], Fig. 4.1, with comparable efficiency. The convex shape of the PE is described using exactly the same principle as with the VD of circles, and can be easily modified to describe any convex shape constructed from tangential circles; the corresponding time complexity would be $O([n^2 + t.n] \log n)$, where $t =$ number of tangent points. Although the algorithm continues to execute in the same way as the original, the PE only uses segments of differently sized circles. Due to this a specific problem may occur along the *velocity line*, which connects the two smaller S-Ma axis circles.

4.1.3.1 The *Velocity Line* Problem

During the execution of the novel VD of PEs, a problem may occur as certain parabolas intersect along the *velocity line* of the PE. The velocity line connects the two smaller S-Ma axis circles of the PE, and in the case of pedestrian prediction is their predicted velocity. In the original VD of circles sweepline algorithm all the inverted parabola intersections occur due to different circles, which in turn trace the VD lines. This is the result of the circle having only one focus point, and so all points on the inverted parabola will always describe the inside of the circle. However, the shape of the PE is described by four separate parabolas, and the PE is again analogous to a real ellipse by also having two *pseudo*-focus points. These points occur at the centres of the smaller S-Ma axis circles. Also, the arcs of the PE that represent the S-Mi axes of the ellipse are tangential to that smaller circle. Therefore, effectively the focal points form a VP between these three circles. Due to this the S-Mi axis parabolas intersect one another along the velocity line, Fig. 4.6f. In the standard sweepline algorithm of circles this would create a Voronoi line between the two circles. However, as both circle arcs belong to the same PE, no Voronoi

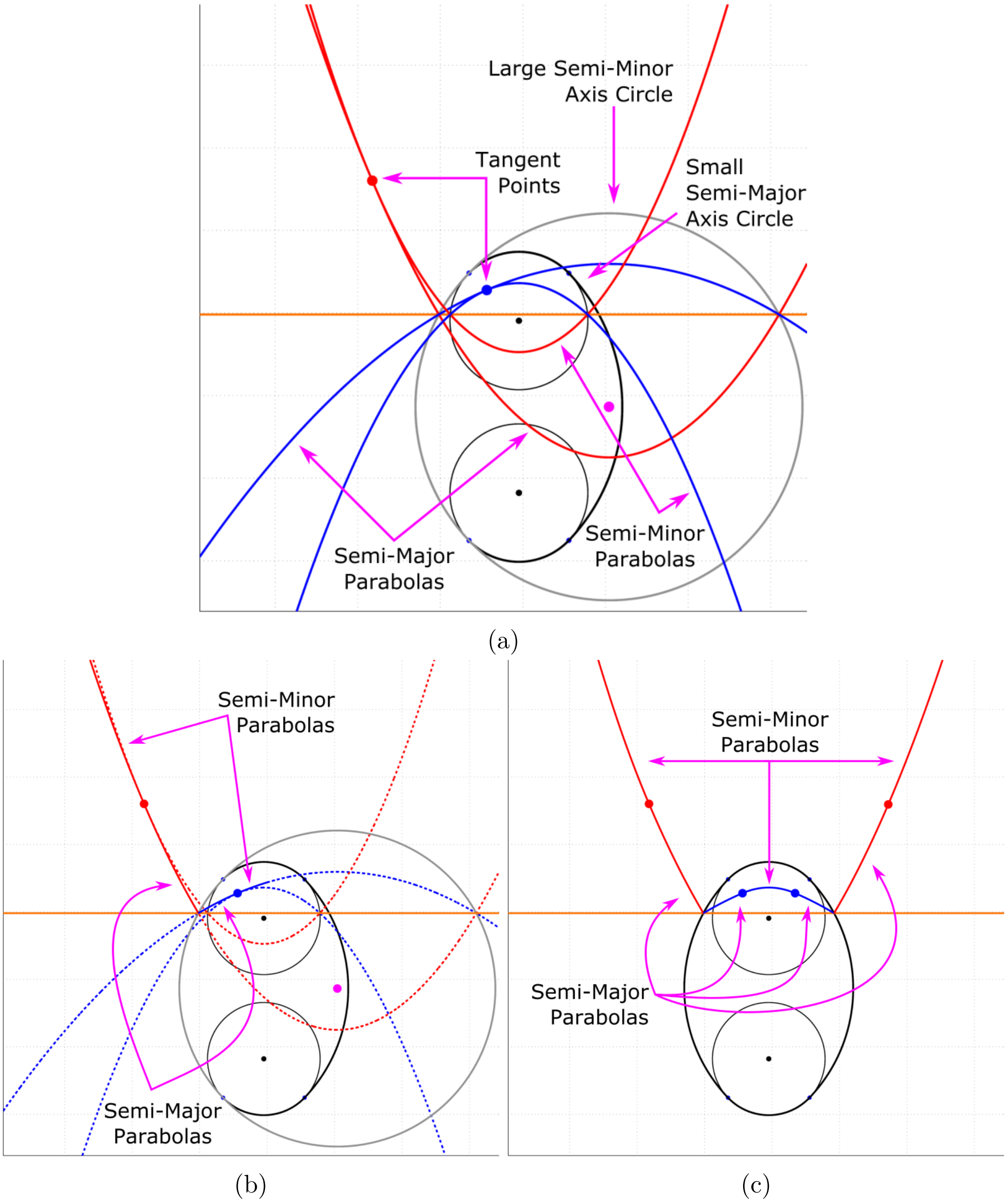
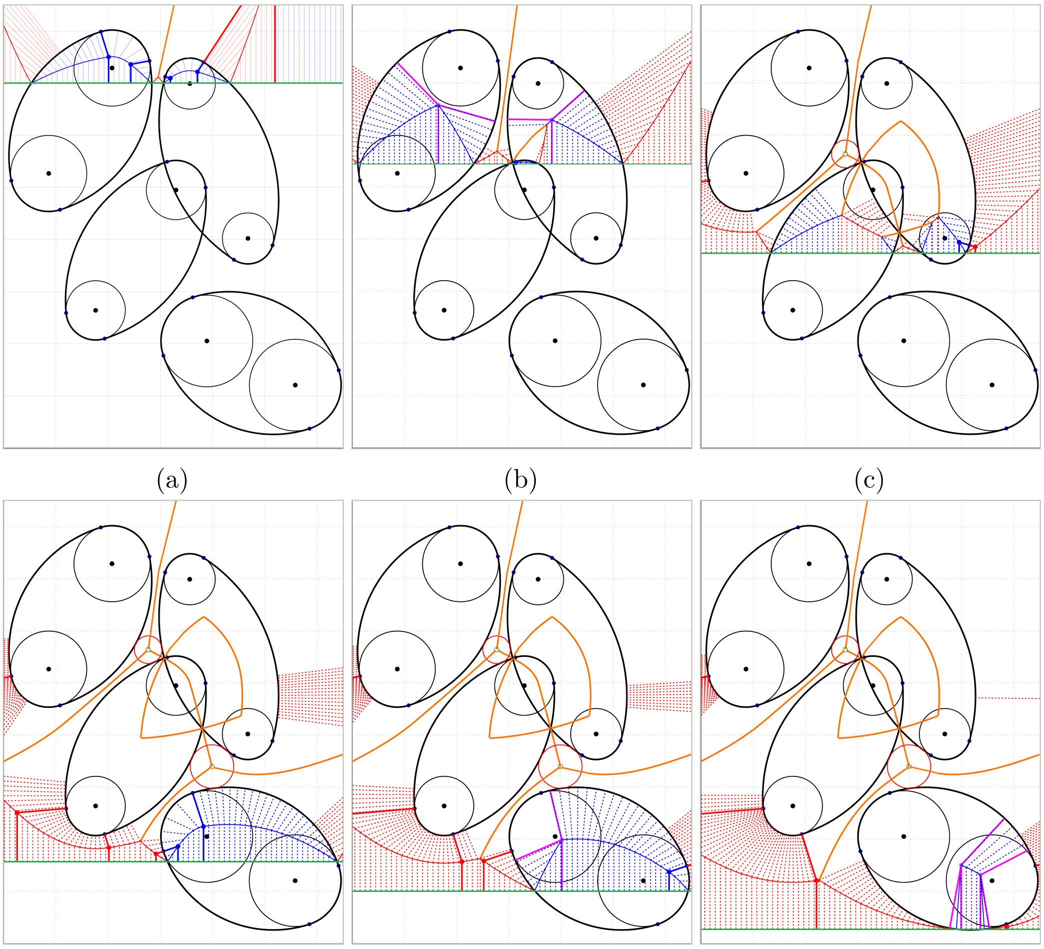


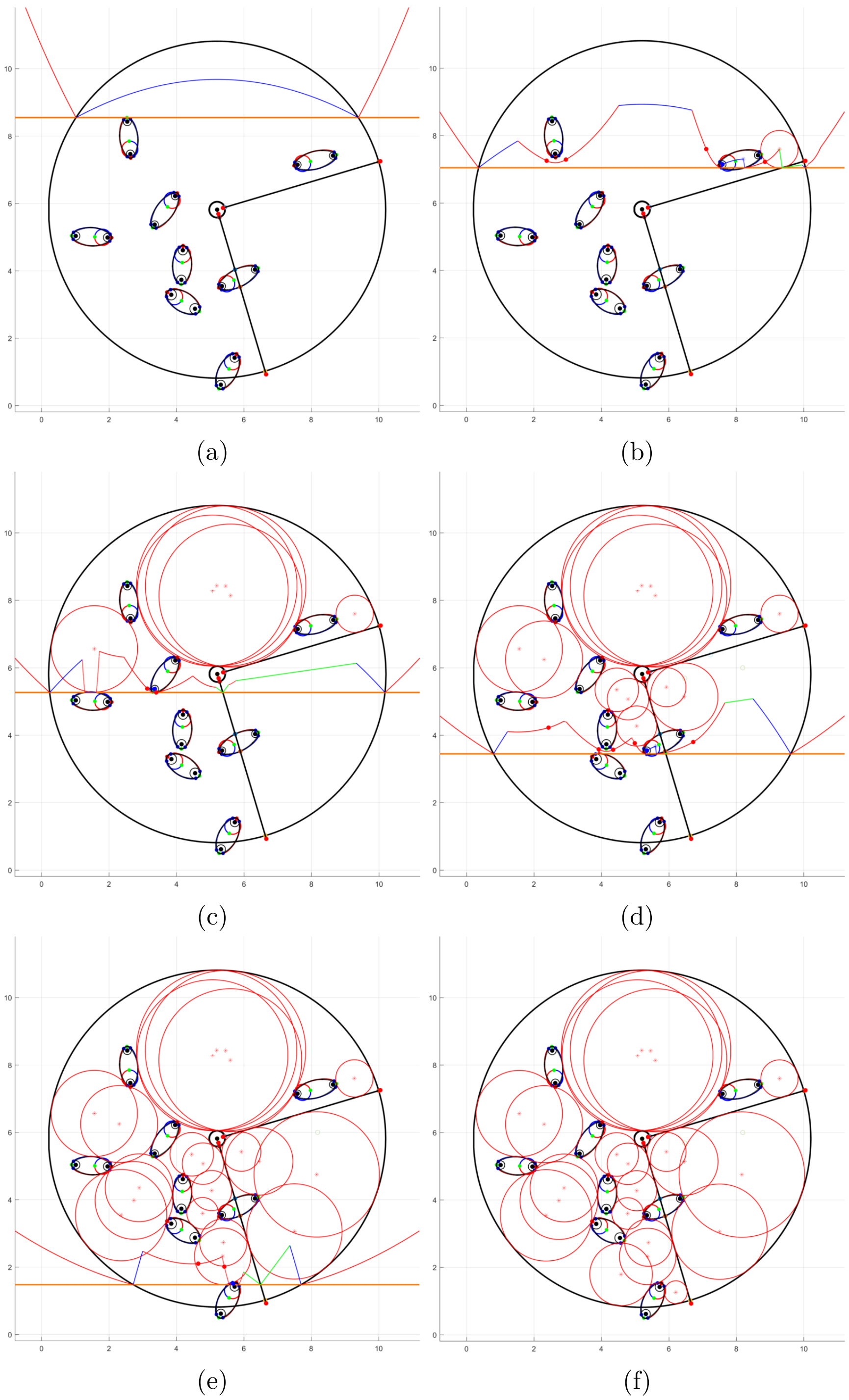
Fig. 4.5 Demonstration of the novel 'tangent' function, Section 4.1.2, merging two parabolas together to form a *super*-parabola, used to describe the shape of the PE. (a-b) Until the sweepline reaches the tangent point only one parabola is used to describe the PE, which is the circle the sweepline intersects. When the sweepline arrives at the tangent point the parabolas associated with both circles tangentially align. As the sweepline continues past the tangent point the two parabolas continue to remain tangentially aligned. The parabolas each side of the alignment describe their respective sections of the PE. The dashed sections of the parabolas are then discarded. (c) The *super*-parabola is formed by merging the remaining solid lines of the parabolas in (b). By discarding the sections of the parabolas that are not describing the circle segments of the PE, the *super*-parabola describes only the edges of the PE.



(a) (b) (c)

(d) (e) (f)

Fig. 4.6 Demonstration of how the sweepline algorithm uses parabolas (red external, blue internal) to describe the size and position of PEs, using the circles' directrices as the green sweepline. The sweepline translates down the page, manipulating parabolas to describe the equal perpendicular distances from sweepline to parabola, and parabola to PE edge (dashed lines). When two neighbouring parabolas intersect, orange VD lines are traced. When three neighbouring parabolas converge on the same point a VP is created, shown with its red Apollonius circle. (a-f) Six snapshots as the sweepline traces out the VD of PE using the novel tangent function, Section 4.1.2. The novel *super*-parabola, Fig. 4.5, describes the edge of the PEs. The solid lines amongst the dashed highlight the tangent point where the super-parabola merges two parabolas together. As the two S-Mi parabolas of a PE intersect along the velocity line, (f), the point is treated as a pseudo-tangent point to prevent a VD line being traced between the PE's own edges.



(a) (b)

(c) (d)

(e) (f)

Fig. 4.7 Demonstration of the novel adaptation to the sweepline algorithm finding the VPs of the PEs within the robot's FoV. The sweepline is utilised in the same manner as the original algorithm, moving down the workspace, whilst using the novel *super*-parabolas to describe the circle segments that form the PEs. (a-f) Each consecutive sub-figure shows the sweepline moving down the figure, revealing the VPs as it goes.

line should be traced. The problem is easily addressed by simply treating this intersection point as a *pseudo*-tangent point, and the two intersecting parabolas as one *super*-parabola.

The pseudo-focus points effectively increase the eccentricity, e, of the PE from a standard circle ($e = 0$) to one similar to an ellipse ($0 < e < 1,\ \ e = \sqrt{1 - \frac{b^2}{a^2}}$). This causes the two inverted S-Mi axis parabolas of a PE to intersect along the velocity line if there is nothing inside, as their corresponding circles are not tangentially aligned. When merging all four individual PE parabolic segments together to form the *super*-parabola, the intersection point between S-Mi axis parabolas are treated as if it were a tangent point. This prevents the algorithm drawing an incorrect Voronoi line along the velocity line, as the two parabolas are treated as a complete parabolic curve.

The velocity line problem could also be used as an advantage if non-convex shapes need to be described. Intersecting parabolic segments that describe the same shape would not trace Voronoi lines, and instead would be treated as one parabolic curve. Although not used in this thesis, this implementation could be used to describe other aspects of the environment, such as irregular objects.

4.2 Enhancing Accuracy

The PVPs of the PEs are guaranteed to be successfully calculated by the sweepline algorithm, provided the computational accuracy required can be achieved (Section 4.1, Item a_1). However, as PEs are approximations of ellipses, contained within the ellipse, Fig. 4.4a, discrepancies between PVPs and EVPs may occur. Additional EVPs may occur as a result of intersecting ellipses, Fig. 4.8, by creating more unique neighbours between ellipses than there were between intersecting PEs. By comparing the internal and external orientations of PVPs to its PE and associated ellipse, it can be determined if an additional EVP may be required.

4.2.1 Accuracy Required for the Considerate Path Planner

The important EVPs occur externally to the ellipses, as this is the safest area of the roadmap, allowing the CPP to generate a path that moves around potential collisions. All external EVPs can be found through this novel algorithm as the PE is enclosed within the ellipse, and so an external EVP will always begin with an associated external PVP. Fortunately the errors that do occur involve only internal VPs, and only arise over very small spaces, Fig. 4.8. Therefore, it does not affect the performance of the CPP in any significant way, however it does mean that a completely accurate VD may not be possible.

Measures are now introduced to find additional EVPs that cannot be found through direct convergence of PVPs. These additional EVPs should occur approx-

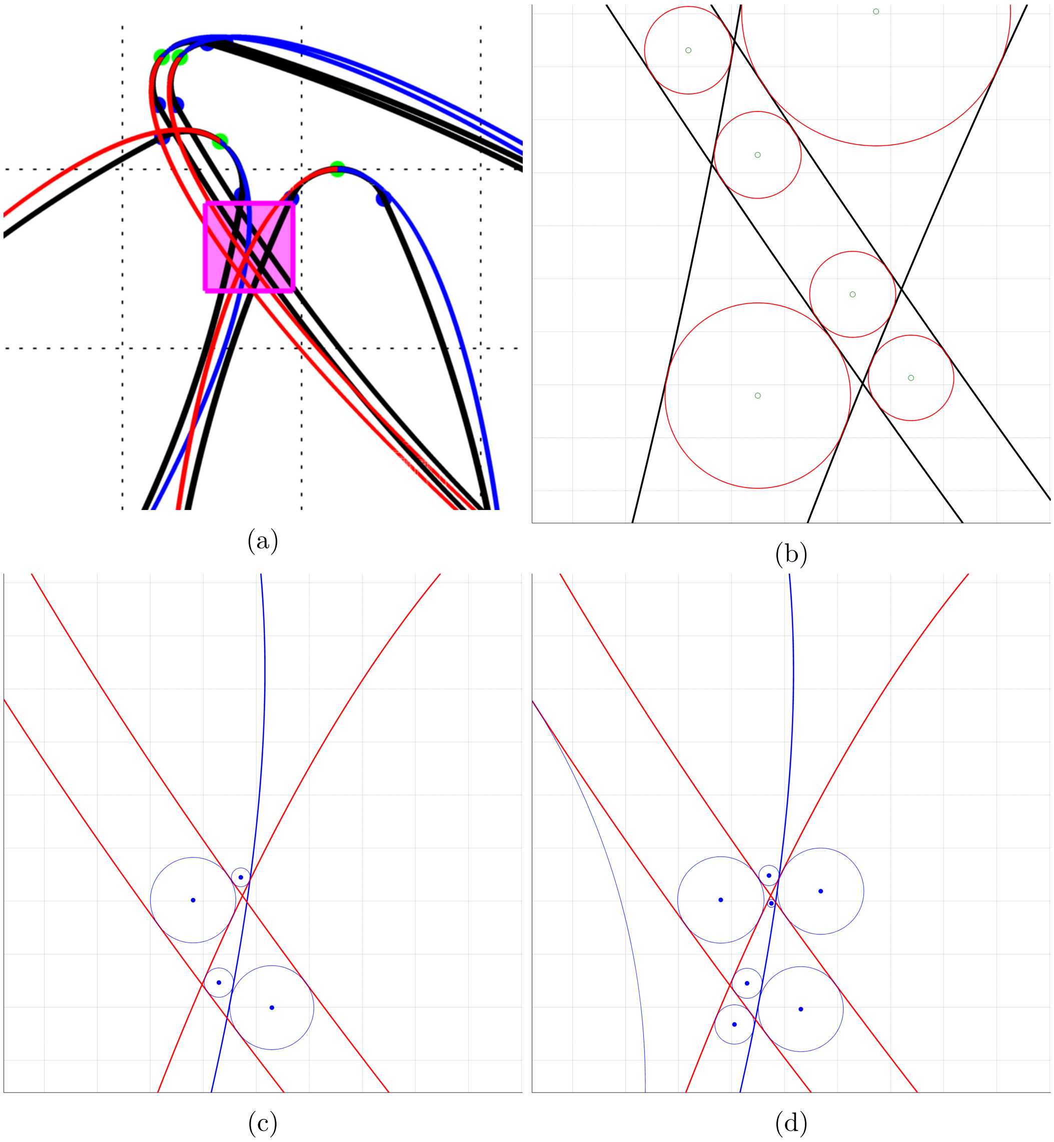


Fig. 4.8 As the PEs are only approximations of ellipses they will always be smaller in area when they do not degenerate to a circle, Fig. 4.4. Therefore, over small areas (pink square in (a)), there may be more EVPs than PVPs, and so not all EVPs can be found. To increase the chance of finding all the EVPs additional points are calculated, Section 4.2, that should converge on EVPs not represented by PVPs. (a) Using four ellipses there are ellipse-ellipse intersections that are not represented with similar PE-PE intersections. The pink square highlights small area where there are more EVPs than PVPs. (b) PVPs that are found for the VD of PEs. (c) EVPs that are found for the VD of ellipses, when using only the PVPs as initial approximations. (d) EVPs that are found for the VD of ellipses, when using additional approximations to the initial PVP approximations, Section 4.2.1.1.

imately along the first internal edge of the ellipses. This additional accuracy will therefore allow the VD to be approximated externally to the ellipses, as well as just within their internal perimeter. Any VD further inside the uncertainty ellipses is not beneficial, as the very nature of the 'safest path' that the VD provides would form paths that actually increase the collision likelihood.

4.2.1.1 Estimating Additional Ellipse Vertex Points

To establish if an additional EVP may occur, the PVP must be external to the PE and internal to its elliptical equivalent, or vice versa. This quick check will establish if neighbouring ellipses have intersected when their PE approximations have not. The bi-section method is used to converge on the EVP from an initial approximation.

Approximating the new EVP If determined that an additional EVP may exist, a normal intersection point, p_p, is created between the PVP and the closest edge of the ellipse that the PVP is on the opposite side of. From p_p, normal intersections are made between it and the other two ellipses that are represented by the other two PEs. The intersection point between the perpendicular bi-section of these connections then creates an additional PVP to be tested, this time on the correct side of the ellipse.

Converging onto the EVP As a PVP converges on an EVP root, it may cross an ellipse edge. This will be due to the orientation between three PEs differing to their equivalent ellipses. If this occurs, an additional start point is created, to test for alternative EVPs that occur due to additional ellipse-ellipse intersections.

Once the EVP is successfully converged upon, if it occurs on the opposite edge of an ellipse edge, then a new start point is created to test for an additional EVP. As with *Approximating the new EVP*, a normal intersection point, p_p, is created between the PVP and the closest edge of the ellipse the PVP is on the opposite side of. This is repeated for the new EVP and creates another point, p_e. The intersecting perpendicular bisectors between PVP - p_p and EVP - p_e generate the new start point.

When comparing the difference between enhancing the accuracy and not, Figs. 4.8c and 4.8d, the enhanced accuracy appears to allow additional EVPs to be discovered that cross the opposite edge of an ellipse. This level of accuracy is acceptable for the CPP, as ideally the CPP will never venture far into an ellipse when predicting a path. This will aide in avoiding paths that are likely to form collisions when executed. In order to generate a *perfect* VD of ellipses, the construction of the PE will have to be re-evaluated in order to be more similar in shape.

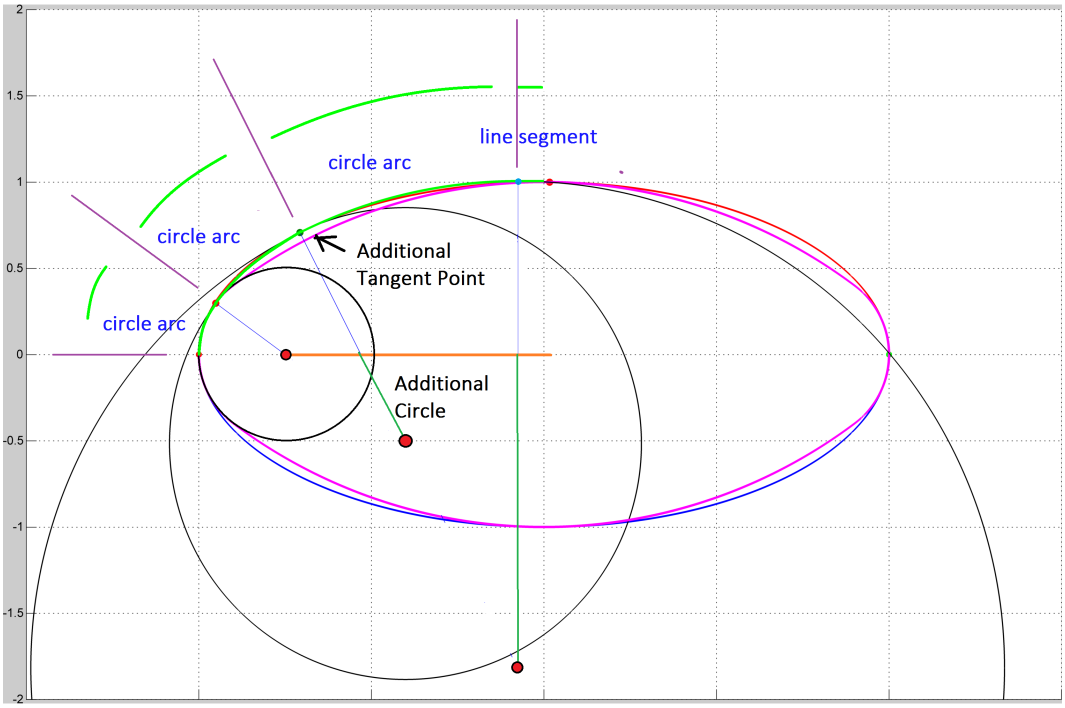


Fig. 4.9 Visualisation of re-creating the PE with additional circles. An additional circle is introduced, which is tangential to the ellipse in between the S-Ma and S-Mi axis limits. This will help to create a PE that is more similar in shape to the original ellipse, resulting in a VD of PEs that has a higher fidelity to the VD of ellipses. The addition of the extra circle will increase the processing time of the sweepline algorithm, however it will still operate with a logarithmic time complexity, Section 4.1.3. The introduction of a line segment must also be included at the S-Mi axis limit, as the additional circle cannot tangentially align with the axis and the new point of the ellipse. However, the line segment is tangential to the circle, and so the novel tangent function, Section 4.1.2, can still be applied.

4.2.2 The Potential for Complete Accuracy

When using just four circles to construct the PE, the accuracy is at worst 15% less than the area of the ellipse, Fig. 4.4b, regardless of the ratio between the ellipse axes. To create a better PE approximation, more tangential circles must be used. Fig. 4.9 provides an example of introducing an additional point along the ellipse that a circle can align tangentially with. The four circles of the PE used in this algorithm align tangentially with the ellipse axes, with parametric values: $0, \frac{1}{2}\pi, \pi, \frac{3}{2}\pi$. Introducing another tangent point half way between the ellipse axes, with parametric values: $\frac{1}{4}\pi, \frac{3}{4}\pi, \frac{5}{4}\pi, \frac{7}{4}\pi$, causes the accuracy of the PE to increase by becoming more comparable in shape.

Regarding the performance costs, the time complexity still remains logarithmic, Section 4.1.3, however each new tangent point causes another event to be evaluated. This version of the PE must also connect the two circles either side of the S-Mi axis with a line, as it is not possible to tangentially align the circle with the S-Mi axis as well. Fortunately, the line segment is tangential to the tops of both of these circles, and the same novel tangent function can still be used.

No additional problems should occur if using a different manifestation of the PE. The *velocity line* problem, Section 4.1.3.1, would still be processed in the same way, as the circle that represents the S-Ma axis limits remains the same. This short example highlights the versatility of the novel tangent function presented, Section 4.1.2. It will allow any shape to be traced that can be formed of tangentially aligned circles and lines.

4.3 Approximating the Voronoi Diagram of Ellipses

The major components of the VD are the VPs and MDs. The VP describes the single point equidistant to all three neighbouring ellipses, and the MD describes the closest point between two neighbouring ellipses. By connecting either VP → VP or VP → MD appropriately, a complete VD approximation can be created.

Initially the EVPs between three ellipses must be found, using the PVPs as initial approximations. The convergence from PVP → EVP requires only a few iterations. Secondly the MDs between neighbouring ellipses must be found, using the EVPs as initial approximations. The convergence from EVP → MD occurs at an asymptotic rate.

4.3.1 Converging on Vertex Points Between Three Ellipses

To approximate the VD of ellipses requires a conversion of the EVPs into VPs for the real ellipses. To achieve this the problem of Apollonius must be solved [255], which is to construct a circle that is tangential to three other neighbouring circles in the same plane, Fig. 4.10. The VPs of a VD will always be the centre of an Apollonius circle, which is true for the VD of circles, the VD of PE, and the VD of real ellipses. However, as mentioned in Section 4.1, unlike an Apollonius circle that is tangential to another circle, an Apollonius circle that is tangential to an ellipse *cannot* be calculated analytically. Instead the tangential alignment of the Apollonius circle must initially be estimated, and the closest root converged upon to within an acceptable tolerance level. This is done by repositioning the PVP so that it aligns with a standard conical ellipse.

From the canonical implicit equation of an ellipse,

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{4.1}$$

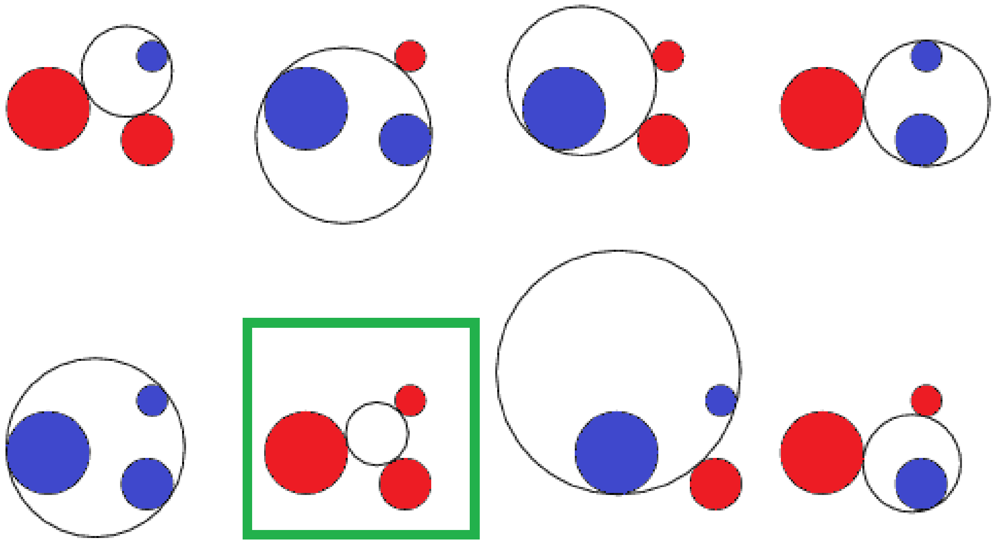


Fig. 4.10 Example of all eight possible Apollonius circles (black) to three circles (red external, blue internal) [15]. Only the Apollonius circle that is external to all other circles is valid for a VP (green box).

where

$$\begin{cases} a > b \\ (x, y) \in \mathbb{R}^2 \end{cases}$$

the normal intersection, n, at any point on an ellipse can be calculated through implicit differentiation of Eq. (4.1),

$$\frac{2x}{a^2} + \frac{2y\frac{\Delta y}{\Delta x}}{b^2} = 0$$

therefore the normal gradient at any point around the ellipse is

$$n = -\left(-\frac{b^2 x}{a^2 y}\right)^{-1}$$

The perpendicular intersection of the line from the PVP, (x_v, y_v), and the ellipse has a gradient n, which has an identical gradient, m,

$$m = \frac{y_v - y}{x_v - x}$$

Through substitution of both gradients into the standard linear equations,

$$y = mx + c$$

and

$$y = nx + c$$

The y-axis intersection, c, can be eliminated and a suitable expression for y can be found,

$$y^2 = \frac{b^2 y_v x}{(b^2 - a^2)x + a^2 x_v}$$

Substituting y^2 back into Eq. (4.1), the real-roots of the line-ellipse normal intersection from the VP can be expressed as

$$Ax^4 + Bx^3 + Cx^2 + Dx + E = 0 \tag{4.2}$$

where

$$\begin{cases} A = (a^2 - b^2)^2 \\ B = -2a^2 x_v (a^2 - b^2) \\ C = a^2 \left(a^2 {x_v}^2 + b^2 y^2 - (a^2 - b^2)^2 \right) \\ D = 2a^4 x_v (a^2 - b^2) \\ E = -a^6 {x_v}^2 \end{cases} \qquad \begin{cases} a > b \\ (x_v, y_v) \in \mathbb{R}^2 \end{cases}$$

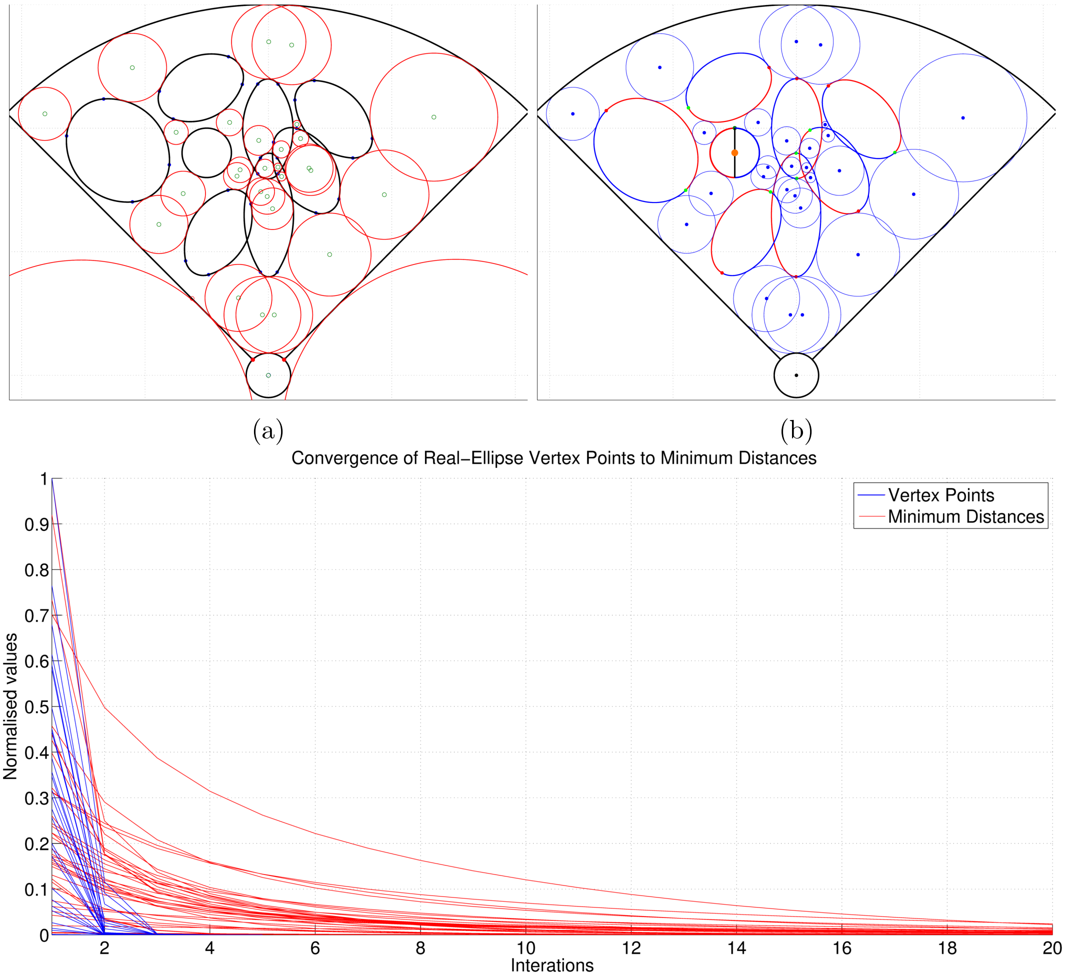
From this quartic equation, there can either be 2 or 4 real roots, depending on whether the point (x_v, y_v) resides outside or inside the ellipse's convolute, respectively. The corresponding y-values are calculated by substituting back into Eq. (4.1),

$$y = \pm \sqrt{b^2 \left(1 - \frac{x^2}{a^2} \right)}$$

and the process must then be repeated for the remaining two ellipses.

By relatively orientating a VP of a PE, so that the corresponding ellipse is canonical, the tangential alignment of a circle from that point can be calculated using the aforementioned process. After repeating this process for all three ellipses the real VP can be found in only a few iterations, using the bi-section method between perpendicular intersections of two ellipses. This continues until the radius of the circle is within an acceptable tolerance and the circle becomes a true Apollonius circle to the real ellipse.

The PVPs obtained from the VD of PEs provide excellent initial approximations to the EVPs of the real ellipses, Fig. 4.11. However, due to the fact that the PEs will never be the same as an ellipse (unless degenerated to a circle) some PVPs may not translate to a corresponding EVP. Due to this an initial comparison is made between the internal and external position of a PVP to the PE, and associated ellipse, Section 4.2.1.1.



(a) (b)

(c)

Fig. 4.11 Comparison between an example of PVPs of PEs, found using the novel tangent function, with their equivalent EVPs of ellipses. The positions between each PVP and EVP varied by only +0.00898% and +0.00591% for x and y, respectively, highlighting the accuracy of the PVP used for initial approximations. (a) The PVPs of PEs, found using a novel adaptation of the Voronoi diagram of circles, Section 4.1.2. (b) The EVPs that the PVPs converge upon, and the MD lines that the EVPs converge upon. (c) Graph that shows the convergence rate of the PVPs to EVPs (blue), and EVPs to MDs (red).

4.3.1.1 Collision Checking

As mentioned in Section 4.2, the difference between PEs and ellipses may generate different PVPs to EVPs. Over high densities neighbouring PEs may not accurately represent neighbouring ellipses. Therefore, once an EVP is found the Apollonius circle must be collision tested with all other ellipses. This will clarify if the ellipses are just as neighbourly as their PE counterparts. If the Apollonius circle is only intersected by three tangential alignments of its associated ellipses then it is confirmed as an accurate VP. If the circle intersects another ellipse the VP is deleted, as its circle is not an Apollonius circle. The VP would therefore not be a point of equal distance from the three closest ellipses.

4.3.2 Converging on Minimum Distances Between Two Ellipses

Once an EVP is found the associated MDs can also be found with the same bi-section method used when converging on the EVP's root. Using the EVP as the initial estimate the normal intersections from the EVP to the surface of the neighbouring ellipses intersect along the Voronoi line, and converge asymptotically, Fig. 4.11c, to the MD. The CPP does not require a smooth VD, as movement predictions for the robot will not require that level of accuracy. Therefore, only VPs and associated MDs are used to create a VD of vectors between these points[2]. Using these components the novel VD-VG roadmap can be created, which will allow the CPP to find a considerate path through the crowd.

4.4 Creating the Roadmap

Using the EVPs and MDs of the ellipses found thus far, the area within the robot's FoV can be appropriately transformed.

4.4.1 Transforming the Robot's Field of View

The roadmap must be confined to within the robot's FoV. The area inside the FoV is segmented into neighbouring polygons that form by tracing around ellipses and MD connections, Fig. 4.12. Additional points are added around the perimeter of the FoV to generate possible roadmap vectors that will allow the search algorithm to explore *all* of the model environment, and not just the areas in between ellipses.

It is possible that an MD connection may intersect other ellipses, Fig. 4.13, in which case it would be divided into additional ellipse-ellipse connections. It is also possible for MD connections to intersect each other, which is not desirable as an

[2] directly intersecting ellipses have their MD at those intersection points

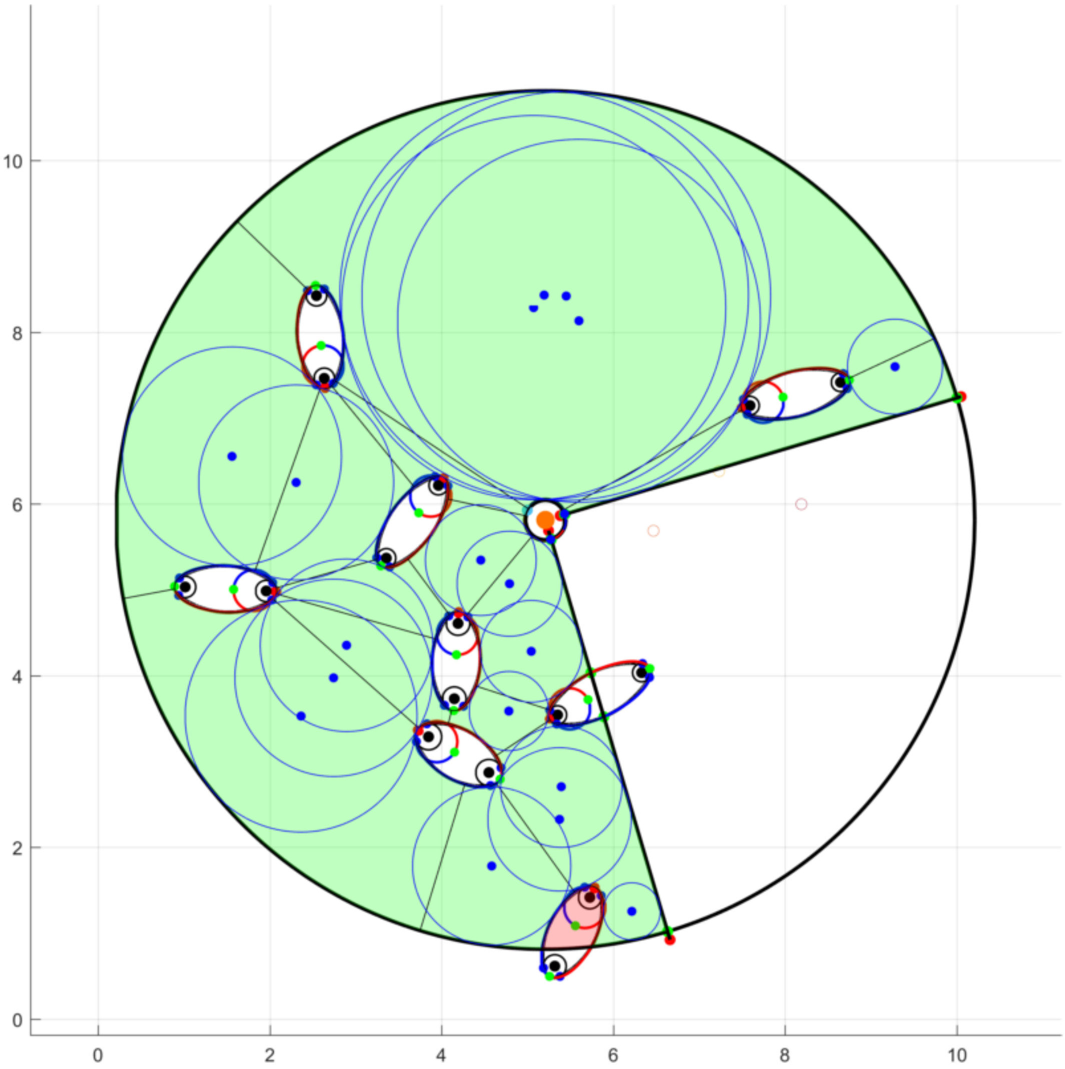
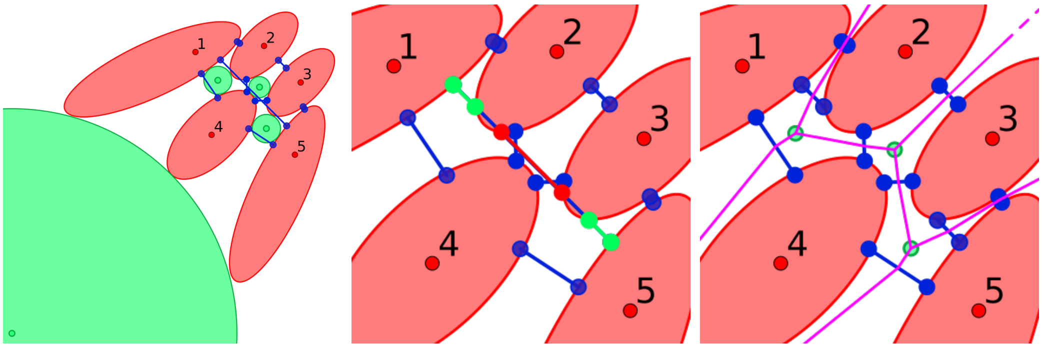


Fig. 4.12 Visualisation of the polygons that form between the MD and ellipse-ellipse connections. The polygons divide the model environment into a configuration space, for the CPP to plan the AMR's movement around. The white areas inside the ellipses, that do not form polygons, are areas inaccessible by the CPP. Green polygons are the *safest* polygons, representing open-space areas of the virtual environment. They are formed by tracing around ellipses, which connect to one another with MD lines. The valid VPs between ellipses are also highlighted blue, with their corresponding Apollonius circle.



(a) (b) (c)

Stages	Lines (# - #)	Description
0	1 - 5	Initial MD connection
1	1 - 2 / 2 - 3 / 3 - 5	Segment MD from collisions with neighbouring ellipses
2	1 - 2 / ~~2 - 3~~ / 3 - 5	Delete line segments that collide with neighbouring MD lines
3	1 - 2 / 3 - 5	Final line segments

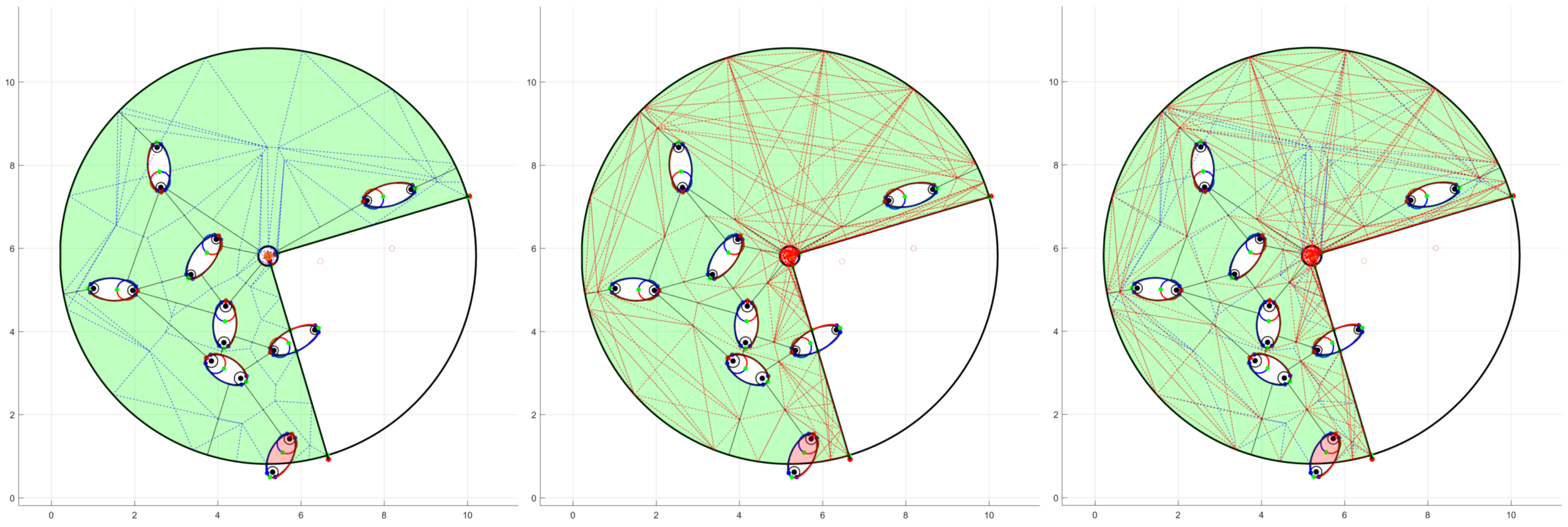
(d)

Fig. 4.13 Demonstration of collision testing MD lines so that they will only create external ellipse-ellipse connections, and none intersect other MD lines. (a) Visualisation of how a VP between ellipses 1, 4, and 5, causes resultant MD connection between ellipse 1 and 5 to intersect ellipse 2 and 3. (b) The MD between ellipse 1 and 5 is subdivided and collision tested with the neighbouring MDs. The green divisions are kept, whilst the red division is deleted. (c) The resultant VD that also uses the mid-point of the MD subdivisions. (d) Table of the execution stages.

intersection would interfere with the resultant polygons. If this scenario does occur, the longest connection is simply deleted. The mid-points of the new ellipse-ellipse connections will not be perpendicularly equidistant from the ellipses, however they do provide extra roadmap nodes, Fig. 4.13c. This will help prevent collisions by creating additional movement vectors around ellipses that are in close proximity. From the segmentation of the model environment within the robot's FoV, the roadmap of vectors can be constructed by connecting all of these neighbouring points together.

4.4.2 Connecting Roadmap Vectors

To create the full VD-VG roadmap, the VD of vectors, Fig. 4.14a, and the network of shortcut vectors, Fig. 4.14b, must be formed. The correct VD connections can be found using a simple decision tree. The VG shortcut connections are formed by connecting a polygons sides together.



(a) (b) (c)

Fig. 4.14 Demonstration of the novel VD-VG roadmap construction stages, within the FoV. (a) Through appropriate connections between VPs and MDs, a vector based network is created that represents a VD roadmap. Connections are made regardless of whether they intersect with any ellipses. (b) By creating connections between MD lines, which cross their own polygons, the 'shortcut' roadmap is created. This roadmap creates a network of vectors that move more in-line with ellipses, allowing paths to be found that considerately move behind pedestrians. Only vectors that do not intersect any ellipses are included. (c) The final VD-VG roadmap will allow the CPP to move between the 'safest' VD and the more 'efficient' VG shortcut network at each node.

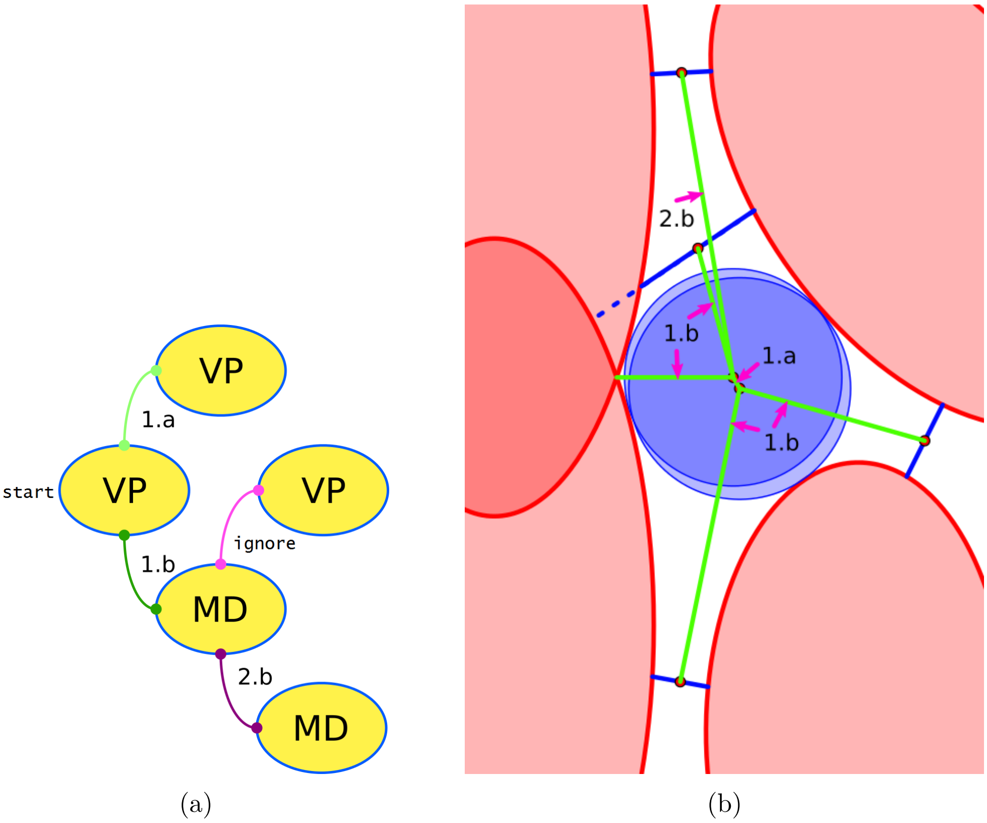


Fig. 4.15 Visualisation of the decision tree required for calculating the VD. The process begins from a point on an ellipse tangential to an Apollonius circle of a VP. Searching either side of this point the next intersection with the ellipse will either be another tangential Apollonius circle, from another VP, or the connection of a MD between the ellipse and a neighbour. (a) The decision tree can make connections with upto two intersections either side of the initial VP. All the first potential connections form simple Voronoi vectors. If connection 1.b is made, then there may be another MD to connect the original VP to. (b) Example of the various connections made by the decision tree.

4.4.2.1 Voronoi Diagram

To create the VD of vectors a simple decision tree is used, Fig. 4.15, which demonstrates which connection must be made. Vectors are formed by assessing the neighbouring intersections around an ellipse. On either side of the tangential alignment of the Apollonius circle, the neighbouring intersections are assessed.

4.4.2.2 Visibility Graph Shortcuts

Shortcuts comprise of MD → MD connections, which allow more direct paths to be taken by the robot as it passes other agents. The connections are made by joining the edges of a polygon, within the FoV, together. The shortcuts can align more directly with the S-Ma axis of an ellipse and hence with the agents movement, allowing the robot to move more easily with the flow of the crowd. The connections can also move directly in front or behind the ellipses, perpendicular to the S-Ma axis, allowing consideration to be implemented by moving behind an agent if the robot must cross its path.

4.5 Chapter Overview

This chapter has presented a novel method for approximating the Voronoi diagram of fully-intersecting ellipses, from which the novel VD-VG roadmap can be formed. As no methods currently exist for accurately generating the VD of ellipses, a novel adaptation of the VD of circles [72] was presented to approximate the VD VPs. The VPs of a VD are the essential element, which allows the unique point of equal distance from three neighbouring objects to be identified. From this the roadmap is produced that creates a VD of vectors, creating the safest path, along with shortcut vectors that encourage flowing and considerate movement.

To begin the VD of ellipses approximation, the ellipses were approximated into PEs using four tangentially aligned circles. This allowed a novel adaptation of the VD of circles to be implemented. The VD of circles uses a sweepline algorithm to manipulate parabolas, describing their size and position, using the circles directrix as the sweepline. As the parabolas calculate the distance from the edges of the circles, the novel adaptation presented exploits the parabolas behaviour at the points of tangency between the circles. Due to the circles being tangential, the parabolas that describe them must also be tangential. Therefore, by creating the PE from circle arcs between these tangential points, a novel *super*-parabola was created by removing the sections of the parabolas for the discarded circle arcs. The remaining parabolic section can be merged together, and used to describe the PE in the same way as a circle.

The algorithm's execution is proven to be very efficient, with logarithmic time complexity. The VPs of PEs that were found provide a very good approximation of the VPs of the real ellipses. The PEs were at most 15% smaller in area, whilst maintaining the same axial dimensions as the ellipses. Therefore, the VPs of the real ellipses were converged upon in only a few iterations, due to the PVPs providing excellent approximations to the correct root. Although not all of the EVPs may be found, for the purposes of the CPP algorithm the accuracy was more than sufficient. However, by utilising the novel tangent function, a PE described by more tangential circles could easily provide a more accurate initial estimate of the EVPs.

The roadmap was easily generated by also finding the MDs between neighbouring ellipses. Connecting EVPs and MDs appropriately easily generates a vector based VD of ellipses. Whilst connecting MDs together can create shortcuts that will aide the CPP to generate considerate paths. As the CPP is designed to plan paths within crowded and dynamic environment, the need to generate a completely accurate VD was superfluous. When the robot moves in the real-world, even the most considerately planned path will still need to be adjusted due to errors in predictions. Due to this the vector roadmap is more than sufficient to have a considerate path planned along it.

The next chapter will present the implementation of the CPP, using the roadmap generated in this one. The chapter will cover all of the elements required for the robot to manoeuvre considerately, from acquiring input data to executing the robot's movement in the real world.

Chapter 5

The Considerate Navigation Strategy Algorithm

In algorithms, as in life, persistence usually pays off.

— Steven S. Skiena, *The Algorithm Design Manual*

THE considerate path planner (CPP) is designed to plan a number of exploratory paths through a model environment, before the most desirable path is chosen and executed by an autonomous mobile robot (AMR) in the real-world. This chapter covers the considerate navigation strategy (CNS) algorithm, and is structured to follow as closely as possible the order in which it executes, Fig. 5.1.

The CPP's input data, Section 5.1, is obtained by the AMR's input sensors, which for this implementation will be the AMR's field of view (FoV), Section 3.3.1. The input data must be able to uniquely identify each pedestrian, and their positions within the FoV, for every observation made of the real-world environment. Once the positions of each pedestrian is established the pedestrian model, Section 3.2, is used to generate their uncertainty ellipses. A deterministic implementation of the windowed Dijkstra's algorithm, Section 3.3, is then used to search the model environment for one time-step, Section 5.2. The Voronoi diagram-visibility graph (VD-VG) roadmap, Chapter 4, is used, and the consideration weights, Section 3.3.4, are dynamically calculated on-the-fly as the search propagates through the roadmap. At the end of the time-step the frontier of the search is converted into path vectors that connect back to the search's start position, and are ranked by their *desirability*.

A number of path filtering techniques are employed to help remove paths that belong to the same homotopy class, Section 5.3. As well as encouraging path diversity, Section 5.4, by selecting paths that explore the uncertain environment, as pedestrian movements are predicted further into the future. Upon the completion of a search, and the selection of the most desirable path vectors, the uncertainty ellipses are updated in accordance with the pedestrian model. From the end of each of the selected path vectors another deterministic implementation of the windowed Dijkstra's algorithm is performed for another time-step. This is a process that repeats until the end of a prediction time-horizon, in order to add the predictive element, Fig. 2.2, required for a CNS. At the end of the prediction time-horizon a number of possible global paths will have been planned, from which the most desirable can be chosen for the AMR to execute.

Finally, the AMR will move within the real-world environment, Section 5.5, along a movement vector the length of a time-step. At this stage local collision avoidance is required in order to prevent the AMR impacting a pedestrian. The local movement vector is calculated using the overall desirability of the chosen global path and the immediate collision potential over the next time-step. The turn of the AMR is limited to a 45^o cap, and the speed of the AMR is proportional to the turning angle. The AMR will change trajectory as it moves along the movement vector if a collision is imminent, otherwise will continue uninhibited for the length of the time-step. The entire algorithm repeats until the AMR reaches its destination.

5.1 Evaluating Input Data

As mentioned in Section 3.2.2, the movement of each agent within the environment can be established through two consecutive observations of the real-world environment. The central positions of the agents over these two consecutive observations are the only requirements for obtaining the necessary input to successfully execute the CPP. As the CPP will plan paths along the ground, the data regards only their 2-dimensional (2D) Cartesian co-ordinates in a plan view Euclidean plane.

As stated in Section 3.3.1 only data within the robot's FoV can be extracted from the real-world environment. These data will be the positional data of any observed agents, obtained from an observation frame of the real-world. The agents will include pedestrians and other similar robots, and the algorithm will not differentiate between what kind of agent each one is. The active agents within the FoV have their data stored in a simple dynamic 3D matrix $(n, 3, 2)$, which is measured over two consecutive observation frames of the real-world environment: number of agents (n), agent data (ID and Cartesian coordinates), and environmental frame (f)

$$A_f = \begin{pmatrix} ID_1 & x_1 & y_1 \\ ID_2 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ ID_n & x_n & y_n \end{pmatrix} \quad A_{f+1} = \begin{pmatrix} ID_1 & x_1 & y_1 \\ ID_2 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ ID_n & x_n & y_n \end{pmatrix}. \tag{5.1}$$

All individual agents must be assigned a unique identification number, allocated by

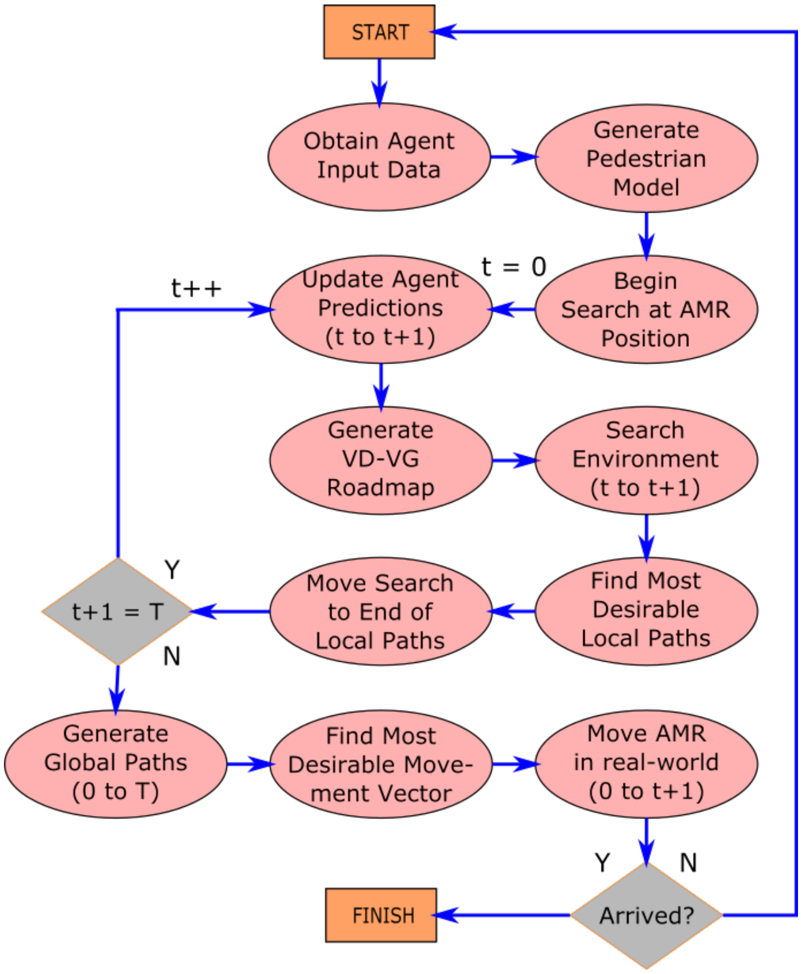

START
Obtain Agent Input Data
Generate Pedestrian Model
t = 0
Begin Search at AMR Position
t++
Update Agent Predictions (t to t+1)
Generate VD-VG Roadmap
Search Environment (t to t+1)
Find Most Desirable Local Paths
Move Search to End of Local Paths
Y
t+1 = T
N
Generate Global Paths (0 to T)
Find Most Desirable Movement Vector
Move AMR in real-world (0 to t+1)
Y
N
Arrived?
FINISH

Fig. 5.1 Overview of the CNS's execution stages. Once data is obtained from the real-world environment, the CPP begins its predictive exploration over a series of model environments within the prediction time-horizon, T. A roadmap is created over the model environment, detailed in Chapter 4. Considerate paths of one time-step in length are then found. Within the model environment a model robot is also moved along these one time-step paths. The predictions of the agents are updated to $t+1$. Once T has been reached, the most desirable global path across the model environment is executed in the real-world environment. The position of the robot is then evaluated to see if it has arrived at the destination. If so the CPP has completed its mission, else the algorithmic process repeats.

the data collection procedure, in order for them to have their velocities calculated between frames.

Additional exclusion zones (EZ) are included that encircle either static agents, or areas where a collision has been avoided within the previous frame. The EZ of a static agent is remembered by the robot, and included in the following environmental frame even if it is now outside the FoV. Unlike a dynamic agent, which is assumed to be always moving, static agents are assumed to be less volatile and more likely to remain in-situ, therefore extra precautions are taken to avoid these areas. The EZs are stored in a dynamic matrix (n,3): number of EZs (n), and Cartesian coordinates and radius

$$EZ_f = \begin{pmatrix} x_1 & y_1 & r_1 \\ x_2 & y_2 & r_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & r_n \end{pmatrix}, \tag{5.2}$$

which is updated at each environmental frame.

All data within the workspace that lies outside the FoV (excluding previously observed static agent EZs) is ignored when obtaining data in each environmental frame. The process of converting the agent objects into usable data is now presented.

5.1.1 A Robot's View of the World

As mentioned when defining the model robot in Section 3.3.1, the roadmap for the model environment is limited to within the robot's FoV. Only agents within the robot's FoV are included when creating the model environment. Therefore, if no agents are within the FoV, the robot can move as directly as possible towards its destination, Fig. 5.2, as there is no fear of a collision. As the robot does not have a 360° view of the real-world environment, the robot must only move within it's FoV limits if the destination is not in the robot's line-of-sight, Figs. 5.2b and 5.2c. This is to prevent the robot from moving into an unknown area of the real-world environment, as the only certain knowledge of the real-world is confined to within the FoV's angular, FoV$_\theta$, and distance, FoV$_r$, range limits.

The FoV is represented within the same 2D Euclidean perspective as the real-world will be modelled on, Section 3.3.2. If agents are detected within the FoV then the CPP algorithm is activated. The most desirable paths are then assessed through a series of predictive model environments, described in the subsequent sections. If the FoV is empty, the chosen path is one that moves as towards the destination as directly as possible.

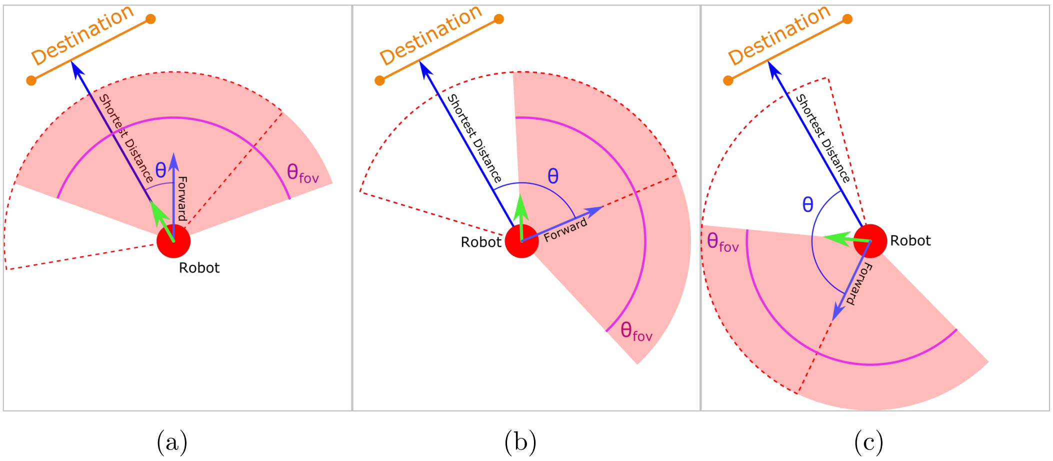


Fig. 5.2 Visualisation of robot's movement when no agents are detected within the robot's FoV. As the real-world environment can only be observed within the FoV the robot must also move within the same area. If the robot moves outside of the FoV a collision with an unknown agent could easily occur. (a) The robot can move directly towards the destination as the turn will not cause the robot to leave the FoV, $\theta < \frac{\theta_{fov}}{2}$. (b,c) The robot cannot move directly towards the destination as the turn will cause the robot to leave the FoV, $\theta > \frac{\theta_{fov}}{2}$. Therefore, the robot moves along the FoV's angular boundary that most directly moves toward the destination, preventing it from entering unknown territory.

5.1.2 Determining Agents' Status

To determine each agent's state (static or dynamic), a simple comparison between two consecutive measurements of the environment is made, Fig. 5.3. If the position of an agent appears at two different locations in consecutive environmental frames, the agent is dynamic, otherwise it is regarded as static. The velocity of a dynamic agent is calculated from the distance travelled between the two frames, as seen in Eq. (3.2). Both the *pedestrian* and *robot* agents are observed in the same manner, with the same method used to establish location and initial velocity.

At any moment it is possible for an agent to stop moving and become static. The EZs are simple perimeter circles that surround either a static agent, or an agent that caused the robot to stop and avoid a collision. They prevent the robot from plotting any paths that may cause the robot to enter into the EZ, preventing any collision with a static agent, and dramatically reducing the likelihood of re-colliding with the same agent in the next environmental frame.

If an agent transitions from static to dynamic the EZ associated with that agent is erased from memory, if observed within the FoV. If the robot moves and a previously calculated EZ is outside the FoV, the EZ data is retained until that space can be checked again. This is relevant for the robot as it moves towards its destination, as static agents that may have been blocking the destination will be remembered. Even if the FoV is currently clear of obstruction any blocked areas out of range

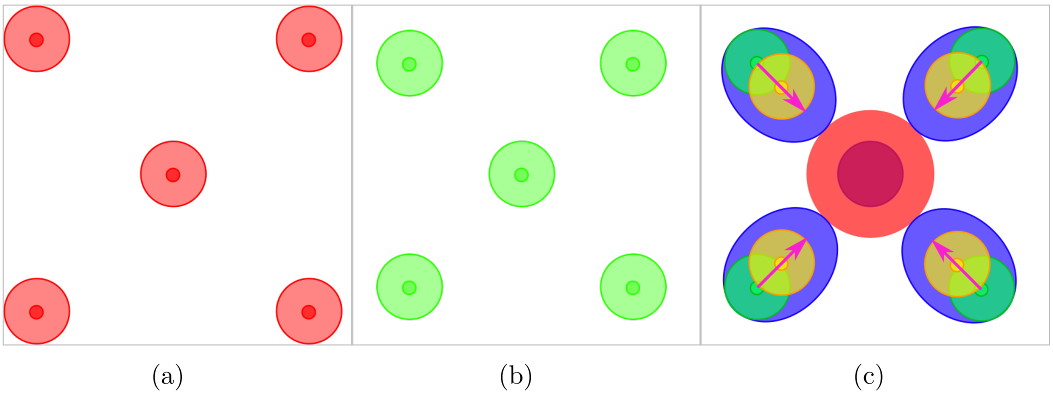


Fig. 5.3 Visualisation of how the agent data is obtained, by observing the real-world environment in two consecutive frames, f. (a) The positions of each agent in the first frame. (b) The positions of each agent in the consecutive frame. (c) The dynamic environment perceived by the robot, by using the positions of the agents in (a), and the velocity obtained by comparing their displacement in (a) and (b). As a result four dynamic agents, one in each corner, create uncertainty ellipses to be used by the CPP. The central static agent is surrounded by the EZ circle, as the CPP should avoid this area.

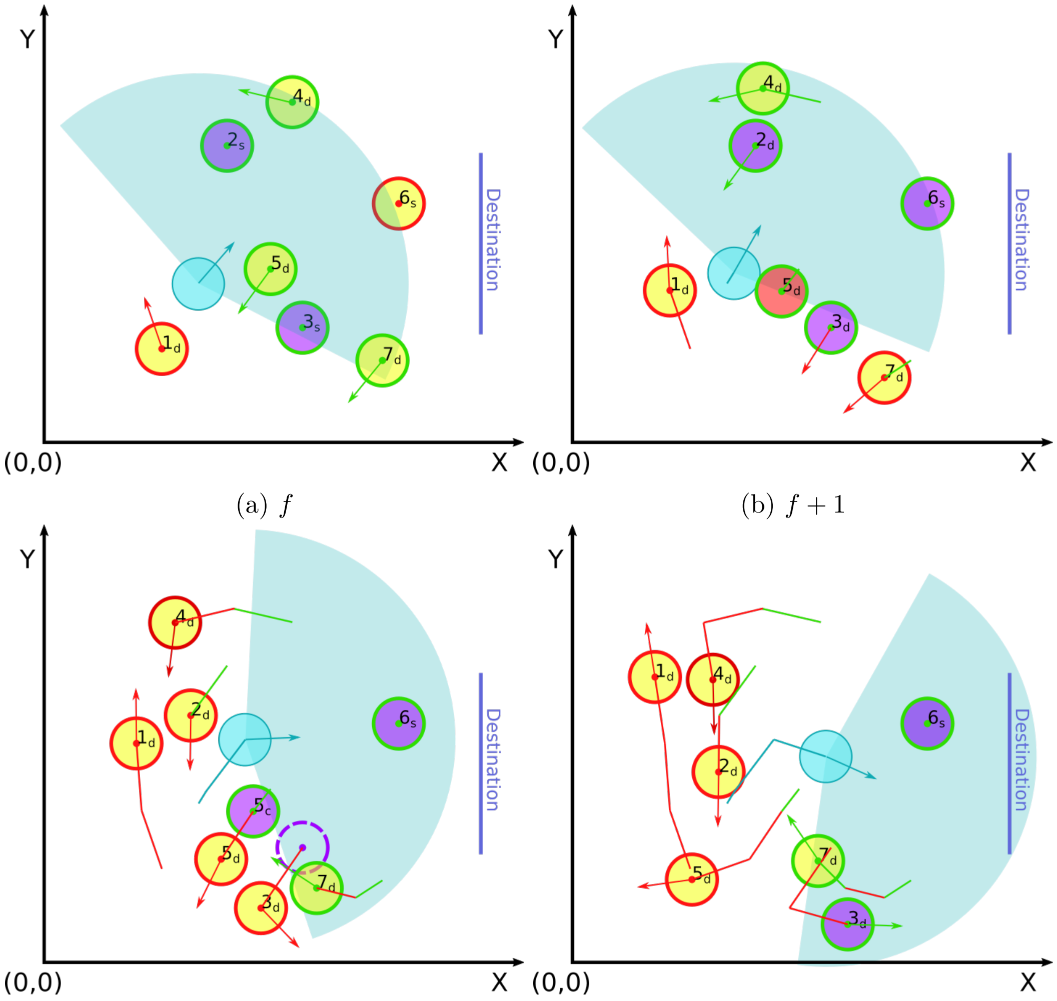
must still be taken into account. Fig. 5.4 provides a visual demonstration over four consecutive frames of the real-world environment. Initial detection of agents within the FoV is shown, and the CPP differentiates between static and dynamic agents. All instances of dynamic agents, static agents, and EZs are covered.

In order to contain the uncertainty of the agent's predicted future position, the ellipse will not extend behind the agent's initial position, only forward and laterally along the same orientation. Only the S-Ma axis, a, the S-Mi axis, b, and the ellipse's centre, (x_e, y_e), will be updated accordingly, Fig. 3.5a. To ensure the potential longitudinal and lateral deviations are confidently confined to within the ellipse a Chi-squared distribution is applied, using a 95% confidence interval [242].

A snapshot of the real-world environment has now been taken, containing all the objects within the robot's FoV. From the objects' data, obtained through two consecutive measurements of their positions, an appropriate uncertainty ellipse can be generated for their potential positions at each t within T. From a measurement of each agent's current velocity, a series of dynamic environments can be predicted during the prediction time-horizon. Using these movement predictions, and the uncertainty ellipses to ensure their confidence, the CPP can now plan paths around the uncertainty ellipses and EZs that have the highest chance of moving the AMR considerately.

5.2 The Windowed Dijkstra's Search Algorithm

As stated in Section 3.3.3, the search used to find suitable paths will be a dynami-



(a) f

(b) $f+1$

(c) $f+2$

(d) $f+3$

Fig. 5.4 Visualisation of which input data should be recorded by the CPP in each environmental frame, which covers dynamic agents and EZs, Section 5.1.2. The blue robot detects agents (5 out of a potential 7) within its FoV. Solid purple circles represent static agents. Solid yellow circles represent dynamic agents. Green circle outlines represent what is stored in the robot's memory. Red circle outlines represent what is *not* in the robot's memory. (a) The robot detects 5 agents within the FoV, 2 static, 3 dynamic. (b) The robot collides with 5_d causing the robot to stop. An exclusion zone is added around 5_d's position at the moment of impact. 3_s becomes dynamic, however this occurs outside the FoV so the robot keeps 3_s's original position in memory. (c) The exclusion zone of 5_d acts as an area to avoid in this environmental frame. The remembered position of 3_s is now observed to be empty, therefore it is removed from memory. (d) Even though the exclusion zone for 5_d is outside the FoV it is removed, as the collision as passed.

cally weighted Dijkstra's algorithm, windowed to within the FoV of the AMR. The start of the search begins at the AMR's current position (R_x, R_y) at $t = 0$. The search algorithm's frontier simulates the potential movement of the AMR, in order to evaluate how the AMR may interact with agents in the future. As a path must be planned through a dynamic environment, the search will be synchronised with each time-step of the prediction time-horizon, T, used by the pedestrian model, Section 3.2. The search will travel through the roadmap and explore the immediate area for one time-step. The pedestrian model will then update the uncertainty ellipses and the search will continue, repeating until $t = T$.

In the same way as a conventional search algorithm, there will be a list of open and closed nodes that are being/have been evaluated. Each node is connected by a vector, V, to at least one other node, in accordance to the VD-VG created for the current configuration of uncertainty ellipses. Along each of these vectors the calculated value of Ω acts as a dynamic weight that will impede rate of the search, Section 3.3.4. As the function for calculating Ω is continuous, any point along a VD-VG vector, V, can produce a unique value of resistance. Also, only a fragment of the AMR's FoV will be searched before the pedestrian model updates the uncertainty ellipses. Therefore, the resistance for each VD-VG vector, V_Ω, will be calculated on-the-fly for all the unexplored vectors connected to each node the search arrives at.

As the search algorithm's frontier is designed to represent the AMR moving through the environment, it will propagate at the same equal pace along all vectors at the AMR's maximum speed, $\max(R_v)$. Due to the continuous functions used to calculate Ω any point along a VD-VG vector can produce a unique value of resistance. To therefore limit the number of Ω values to calculate, each vector is separated into manually defined unit measures, Fig. 5.5. Ω calculations can then be made along equally sized intervals down the vector, regardless of its length.

5.2.1 Dynamically Calculating Consideration Weights

The frontier moves along each active search vector simultaneously, and proportionally to the amount of resistance encountered along each. To slow down the pace at which the frontier travels along each vector, Ω is added to the length of the VD-VG vector, V_d, the search is to travel along, Fig. 5.6:

$$V_d{}' = V_d + V_\Omega$$

The increased length of the vector does not physically affect the VD-VG, and is used purely for calculating the amount the search should move along V. Rather than subtract Ω from $\max(R_v)$ to slow the wave down, as generically formulated in Section 3.3.4 (Eq. (3.12)), increasing the distance of the vector from V_d to $V_d{}'$

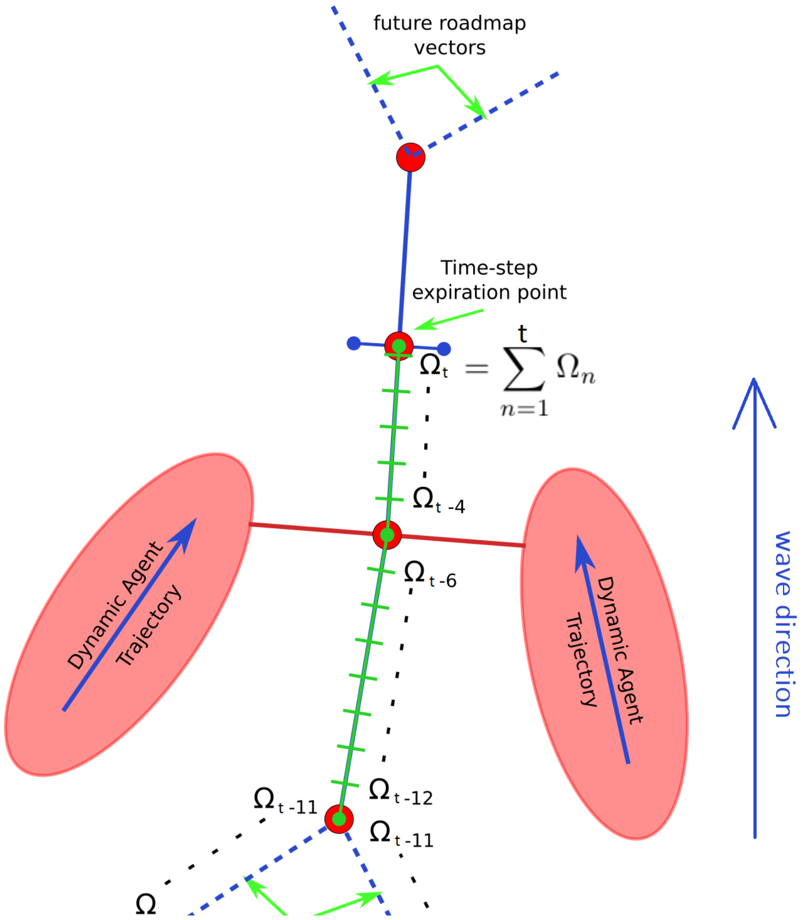


Fig. 5.5 Visualisation of how a roadmap vector is separated into unit measurements for calculating resistance, Ω. Ω is calculated from each segmentation of the vector and is summed along the complete length the search has travelled, up until either the time-step has expired or the robot's destination line is reached.

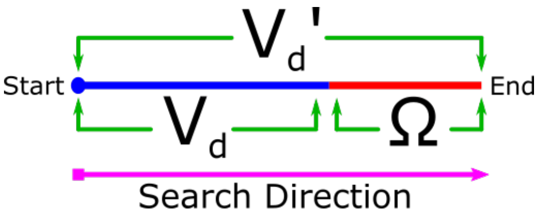


Fig. 5.6 The resistance for each vector is added to it, therefore the search will slow down along the real vector.

will cause the time taken for the search frontier to travel along that vector, V_t, to increase,

$$\frac{{V_d}'}{\max(R_v)} > \frac{V_d}{\max(R_v)} \Rightarrow {V_t}' > V_t \ \text{, if } V_\Omega > 0$$

Therefore, no matter how large Ω may be the speed of the search will never be negative. Whereas for Eq. (3.12), if $\Omega > \max(v_r)$ then $S_v < 0$, which would theoretically send the search backwards along the vector. By adding Ω to the original vector length, the speed of the search, S_v, will decrease, proved as follows:

$$\frac{{V_d}'}{{V_t}'} > \frac{V_d}{{V_t}'} \Rightarrow \max(R_v) > S_v \ \text{, if } V_\Omega > 0 \tag{5.3}$$

In areas of increasing collision likelihood (increased Ω), the frontier will slow down the model AMR to be more *cautious*. The total resistance encountered along these vectors is also considered when calculating the overall desirability of potential paths, discussed later in Section 5.2.2.

5.2.1.1 Propogating the Frontier along Roadmap Vectors

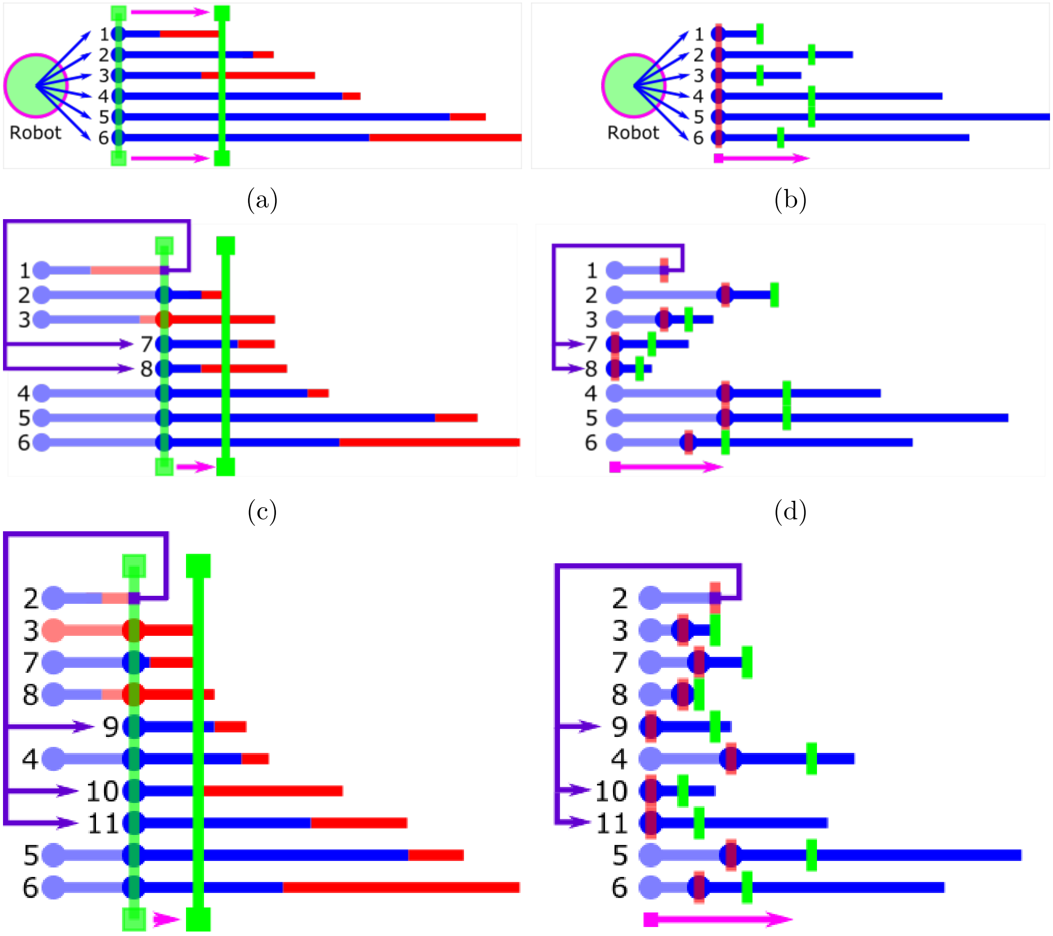
To avoid excessive calculations V_Ω is calculated by summing all values of Ω, at each unit measure along V, prior to the search moving along it. The active vectors are sorted by their relative lengths, ${V_d}'$, from $0 \rightarrow N$ in order to eliminate the need for the frontier to be continually updated at each interval point along each active vector. Instead the frontier is propagated at a discrete interval, determined by the remaining distance to the node at the end of $V(0)$. The frontier simultaneously moves the same distance along all other active vectors.The distance the search must move along all other active vectors is a simple ratio between the real and relative vector lengths, Fig. 5.7,

$$S_d(n) = {V_d}'(0)\left[\frac{V_d(n)}{{V_d}'(n)}\right]$$

where $\{n \in \mathbb{Z} \mid 1 \leq n \leq N\}$, and ${V_d}'(0)$ is the remaining distance the frontier must move along to reach the vector's end. $V(0)$ is now removed from the active list, as the node at its end has been reached.

When a node is reached the nodes at the end of the associated new vectors are compared with the currently active ones. This will determine if the destination nodes of the new vectors share any destination nodes of the active vectors, i.e. to check if the frontier is propagating toward the same node along two separate vectors. As each node should only be visited once, Algorithm 2 is used to establish what actions should be taken when the end of a vector is reached.

.



(a) (b)

(c) (d)

(e) (f)

Fig. 5.7 Visualisation of how resistance applied to the roadmap affects the speed of the search algorithm as it travels along n vectors, $V(n)$. (a) All vectors connected to the start point of the search, ordered by increasing values of $V_d{}'$. The search moves uniformly at $S_v = \max(R_v)$ along all active $V_d{}'$ vectors, by the smallest value in the list, $V_d{}'(1)$. (b) The relative distance the search moves along all associated vectors of length V_d, which form the actual VD-VG. The time period remains the same and so the speed of the search is altered along each vector accordingly, $t = V'_d(1) \div \max(R_v) \therefore S_v(n) = V_d(n) \div t$. (c) The node at the end of $V(1)$ is connected to $V(7,8)$ and adds them to the ordered list of V'_d vectors. Similarly to (a) the search moves uniformly along all vectors by the remaining distance to the end of $V'_d(2)$. (d) $V(7,8)$ vectors are added to the active list and highlights the same process shown in (b). (e) $V(9,10,11)$ vectors are added to the active list and highlights the same process shown in (c). (f) $V(9,10,11)$ vectors are added to the active list and highlights the same process shown in (d).

Algorithm 2: Adding New Vectors

```
$V \leftarrow$ active vector list;
$V^* \leftarrow$ new vector list;
if node at end of $V^*$ has not been previously visited then
    if node at end of $V^*$ is shared with node at end of $V$ then
        if ${V_d}'^*$ is less than the remaining ${V_d}'$ then
            the $V^*$ replaces $V$;
        else
            $V^*$ is ignored;
    else
        add $V^*$ to $V$ list;
else
    $V^*$ is ignored;
```

5.2.1.2 Avoiding Exclusion Zones

As the frontier moves along the roadmap vectors, it must also not enter an EZ. When calculating V_Ω, if the next unit measurement intersects an EZ, then the wave must be terminated to prevent the search from colliding with it. In order to circumnavigate the EZ an artificial node is positioned at the frontier, just outside the EZ, Fig. 5.8, and the current vector is terminated. New vectors are created from this artificial node to the nodes connected to the destination end of the terminated vector. If any collision free vectors can be made, providing it does not intersect the EZ, these new vectors are added to the active vector list.

5.2.1.3 A New Search at Each Time-Step

The search-time of the algorithm is synchronised with the update frequency of the pedestrian model. Therefore, the search terminates once the time elapsed reaches the value of the time-step, t, used in the pedestrian model. As the search executes the time taken for the search to move along each vector is accumulated, t_{acc}. When the search begins $t_{acc} = 0$, then for every discrete move along VD-VG the time taken is added

$$t_{acc} = t_{acc} + {V_t}'(0)$$

where ${V_d}'(0)$ is the distance the frontier must move to reach the end of the vector with the shortest remaining relative distance to its end.

Before each new vector is evaluated by Algorithm 2, a inequality check is made to determine if the end of the vector can be reached in the remaining time,

$$t - t_{acc} < {V_t}'^* \tag{5.4}$$

If this is satisfied, the search will terminate before the end node can be reached, in

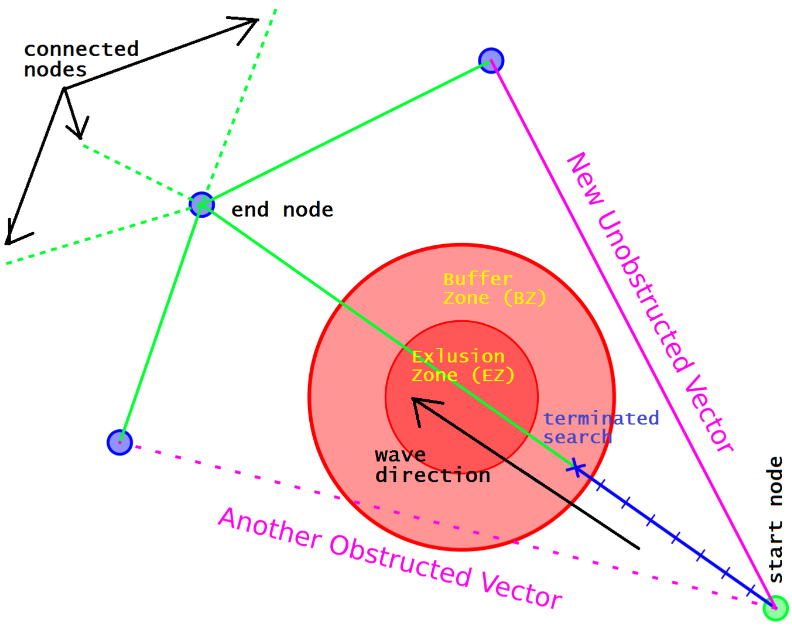


Fig. 5.8 Visualisation of how a new vector is created if the search enters an EZ. A new vector is created between the unit measurement just outside the EZ and an available node connected to end of the terminated vector. A new vector will only be formed if it also does not intersect the EZ.

which case the new vector will automatically be added to the active list. As the search will not explore the length of the entire vector, V_Ω is only calculated along a length up until Eq. (5.4) is no longer satisfied, Fig. 5.5. The cumulative time the search would take to reach each consecutive unit measure along V is evaluated. Therefore, any additional resistance that occurs when Eq. (5.4) is not satisfied will not inflate V_d' to include resistance for areas the search will not be conducted, and thus reduce S_v disproportionately. The frontier will make its final propagation along the remaining distance of $V_d'(0)$ when $t_{acc}+V_t'(0) = t$, and the search will terminate.

A selection of the most desirable points from the search frontier at t are converted into a selection of paths, $P_t(0 \rightarrow N)$. As linear predictions are suitable for dynamic environments [17], each P_t that traces back along the VD-VG to the start point of the search, P_0, is stored as a simple vector. All P_t vectors calculated for each consecutive time-step along T can then be connected to create global paths, P_T. The CPP will evaluate each P_T to predict the most considerate, which will then be executed by the AMR.

5.2.2 A Desirability Heuristic for Path Selection

Simply choosing a path that is of least resistance would create a very evasive AMR, as it would choose open space over moving into areas of any population. Therefore, the *desirability* of each path is calculated to encourage the CPP to choose paths that interact, rather then evade. A *desirability* heuristic, δ, will factor in the local displacement of the paths, and the relative distance moved towards the AMR's goal, in order rank the paths.

The relative displacement moved locally for each path within the time-step, t, is calculated using the ratio between the displacement the search has made in this time-step, and the maximum displacement the search could achieve in one time-step,

$$\delta_s(n) = \frac{|P_t(n) - P_0|}{(\max(R_v) \times t)} \tag{5.5}$$

where $\{n \in \mathbb{Z} \mid 0 \leq n \leq N\}$ and P_0 is the origin of the path at the start of the search. This contributes toward the CPP showing greater preference towards paths that move the AMR away from its current position, exploring more of the model environment.

The relative distance moved toward the AMR's goal, G, is calculated using the ratio between the distance from the search start to the goal, and the distance from the frontier to the goal,

$$\delta_d(n) = \frac{G - P_0}{G - P_t(n)} \ \ , \text{ if } P_t(n) \neq P_0 \tag{5.6}$$

where $\{n \in \mathbb{Z} \mid 0 \leq n \leq N\}$. This is a similar increasing reward for the AMR to move towards its goal as seen in [216], which will get stronger as the AMR approaches G. Planning considerate paths is more important than moving the AMR directly towards its goal. Therefore, δ_d, enables longer detouring paths to be planned in advance of the AMR converging on G, as the AMR will already have navigated considerately en route. δ_d simply helps to encourage a general movement of the AMR towards G, selecting paths that move in the general direction of the goal.

Paths are more desirable if they encourage the AMR to continually move at a constant speed, Eq. (5.5), and so help to increase legibility, Section 2.4.2. As well as paths that encounter less resistance, and so help promote considerate path selection. The desirability of each P_t path is determined by these relative movements, along with the accumulated resistance encountered along the search,

$$\delta_t(n) = (\Omega(n) + 1) \times \frac{1}{\delta_s(n)} \times \frac{1}{\delta_d(n)} \tag{5.7}$$

where $\{n \in \mathbb{Z} \mid 0 \leq n \leq N\}$. As with Ω, where less encountered resistance is preferable, the smaller the value of δ the more desirable the path. The addition of "1" as a precursor to the resistance, allows the relative distances, Eqs. (5.6) and (5.7), to decide the path desirability if $\Omega = 0$. As the AMR approaches the goal, the final movement to G is dictated primarily by the AMR's proximity to it, as $\delta_s \times \delta_d >> \Omega$. Although δ_s and δ_d provide an exploratory quality to path selection, explicit instructions are employed in order to add diversity to the final selection of paths before the search resumes at $t + 1$.

5.3 Avoiding Homotopic Path Repetition

In the same way as other roadmaps, the VD-VG forms multiple connections that converge on the same nodes. As mentioned in Section 5.2.1, the search along a vector will automatically terminate if another frontier is propagating along another vector that shares the same end node, and will arrive first. However, the paths converted from the frontier, once the search terminates, can belong to the same homotopy class by moving: between the same ellipses, in the same direction around an ellipse, and when moving through open space.

Moving Through the Same Gap Between Two Ellipses

All VD-VG nodes are either VPs, with 3 ellipses in common or MDs with 2 ellipses in common, which can either be inside or outside an ellipse, Fig. 5.9a. If any of the frontier paths have at least 2 ellipses in common, and point in the same direction with the same internal/external ellipse component, Fig. 5.9b, then only the most desirable is chosen, as they belong to the same homotopy class, Figs. 5.9c and 5.9d.

Moving Around the Same Ellipse

To achieve considerate movement, one of the main principles set out in the aims & objectives, Section 1.3.2, is to avoid interfering with a pedestrians' trajectories. A potential path can either collide with; pass on either side of; or move away from, an agent's uncertainty ellipse. As ellipse orientations represent agent movement vectors, the orientations of the frontier paths are compared with them. Comparisons between paths are made. Fig. 5.10, in order to determine if two paths are competing to move in similar directions:

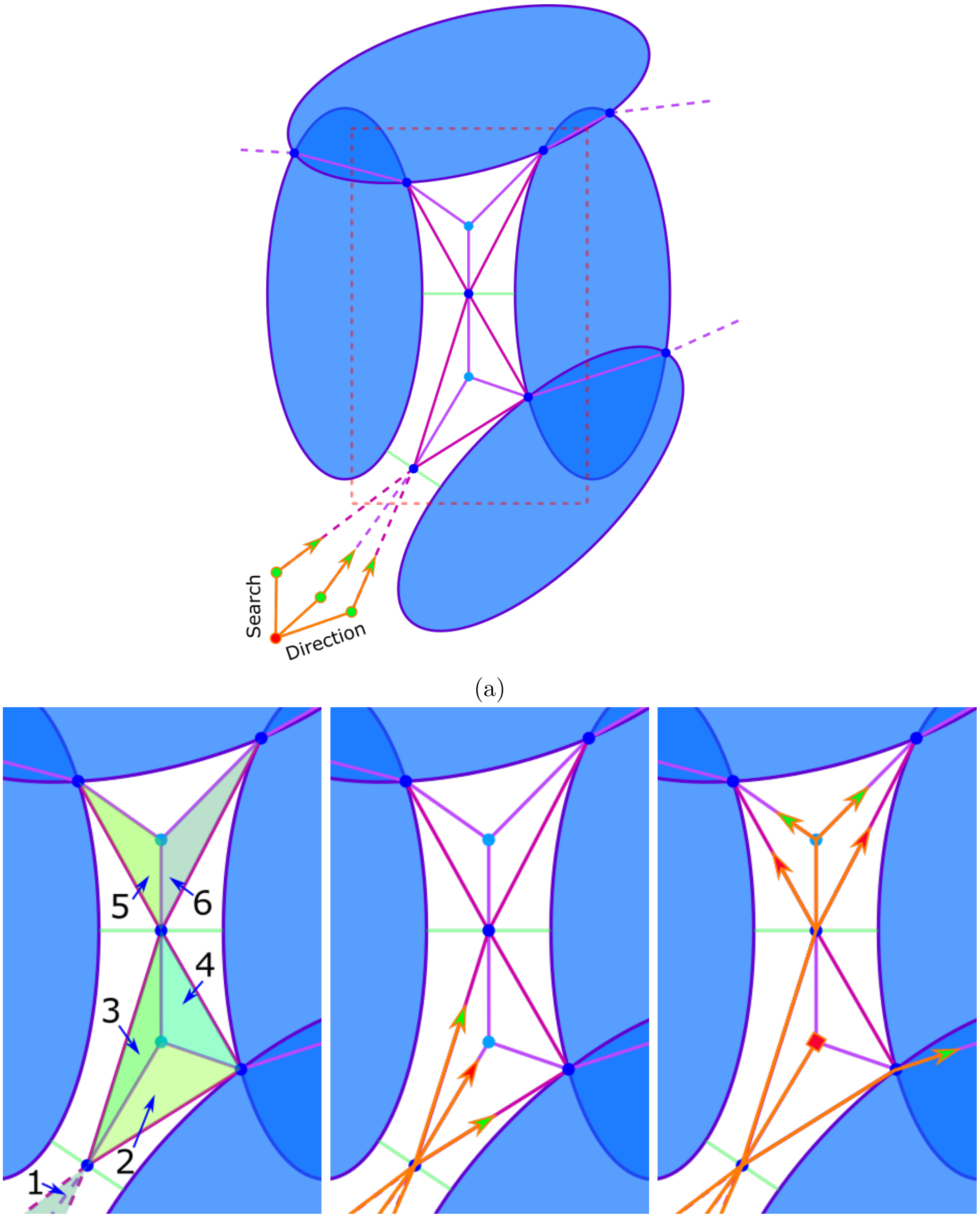
a_1 Only the most desirable path is chosen if one of the following three criteria are met:

$a_{1.1}$ Both paths collide with the ellipse, Fig. 5.10a.

- As a collision will likely occur for both paths, increasing the number of paths is not desirable.

$a_{1.2}$ Both paths pass by the ellipse on the same side, Fig. 5.10c.

- This is particularly effective if the CPP searches an environment where all agents appear only to one side of the AMR, as it will prevent multiple paths repeating on the same side of the crowd.



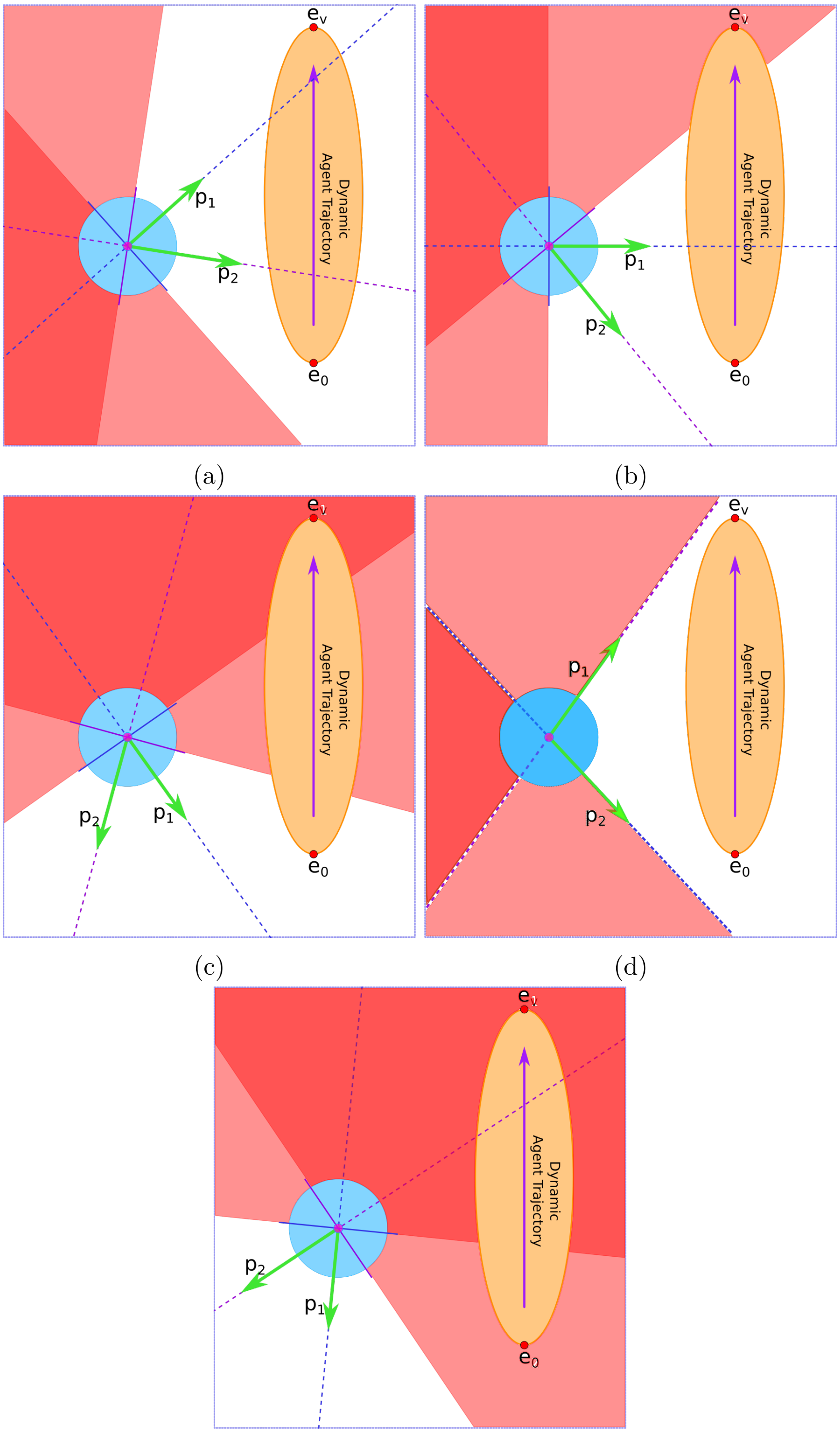
(a)

(b)

(c)

(d)

Fig. 5.9 Visualisation of how homotopic path repetition is reduced. (a) Example of the VD-VG roadmap for 4 ellipses, with the area of interest outlined with the orange square. The search will propagate along the roadmap beginning from the bottom-left. (b) There are 6 homotopy classes that are formed by the different vector orientations, highlighted in various shades of green. (c) If the search terminates here the 3 vectors share two homotopy classes; 2 are chosen (green markers), 1 ignored (red marker). The middle vector is more desirable than the lower vector, which are homotopic, however the middle is removed due to the upper vector and so the lower can remain. (d) If the search terminates here the 5 vectors belong to 3 homotopy classes; 3 are chosen (green arrowheads) and 2 ignored (red arrowheads). The search also terminated at the lower VP (red square marker) earlier on, as the other nodes connected to it would be reached by alternative vectors earlier, Algorithm 2.



(a)

(b)

(c)

(d)

(e)

Fig. 5.10 Visualisation of how similar paths are avoided. A comparison is made between the semi-major axis ends of the ellipse, e_0 and e_v, and the position of the potential paths, p_1 and p_2, as well as their orientation to the front or back of the robot (blue circle). (a) Choose the best path, as p_1 and p_2 both collide with the ellipse. (b) Keep both paths, as p_1 collides with and p_2 passes the ellipse. (c) Choose the best path, as p_1 and p_2 both pass on same side of the ellipse. (d) Keep both paths, as p_1 and p_2 pass on different sides of the ellipse. (e) Choose one path, as p_1 and p_2 both move away from the ellipse.

$a_{1.3}$ Both paths are moving away from the ellipse, Fig. 5.10e.

- A special condition where the orientation of the ellipse is irrelevant, and neither of the paths will interfere with the agent.

a_2 Both paths are kept if one of the following two criteria are met:

$a_{2.1}$ One path collides with the ellipse, whilst the other passes by, Fig. 5.10b.

- Both paths contribute differently, even if one has a much higher collision potential.

$a_{2.2}$ Both paths pass by the ellipse on opposite sides, Fig. 5.10d.

- Both paths will circumnavigate the agents via opposite routes, helping to explore the environment more.

The procedure to determine which frontier paths to use involves a simple comparison between which side of two frontier path vectors the ellipse's S-Ma axis limits, e_0 and e_v, lie ($+ \Rightarrow$ left, $- \Rightarrow$ right): **If** paths pass on the same side (the symbols are the same) **Then** keep both paths **Else** choose the most desirable path only. However, before this comparison can be made the orientation of the paths w.r.t. the ellipse must be checked: **If** e_0 or e_v is behind a path vector (the perpendicular line at its start) **Then** use the other vector to find the correct symbolinstead, e.g. Fig. 5.10c.

Moving into Open Space

A special condition arises if both paths move away from an ellipse, whereby the orientation of the agent is irrelevant. **If** both paths remain behind the ellipse, **Then** only the most desirable is chosen. No significant divergence is possible under this condition, as there is no potential for the AMR to begin interacting with the agent. This prevents multiple paths being created that would only ever move into open space, where the agent will not be.

5.4 Exploring a Dynamic Environment

Planning paths around dynamic agents means they must be calculated quickly, and in order to evaluate different potential paths they must also be diverse. In crowded environments, paths that belong to different homotopy classes may still move in very similar directions. Therefore, processing time would be wasted on planning paths that do not offer any diversity between them. Also, the dynamic environment results in any paths planned too far in the future becoming unreliable, due to escalating prediction errors in the pedestrian model as T increases.

Both of these issues are now addressed in order to ensure the selected paths move in different enough directions from one another. The number of paths will

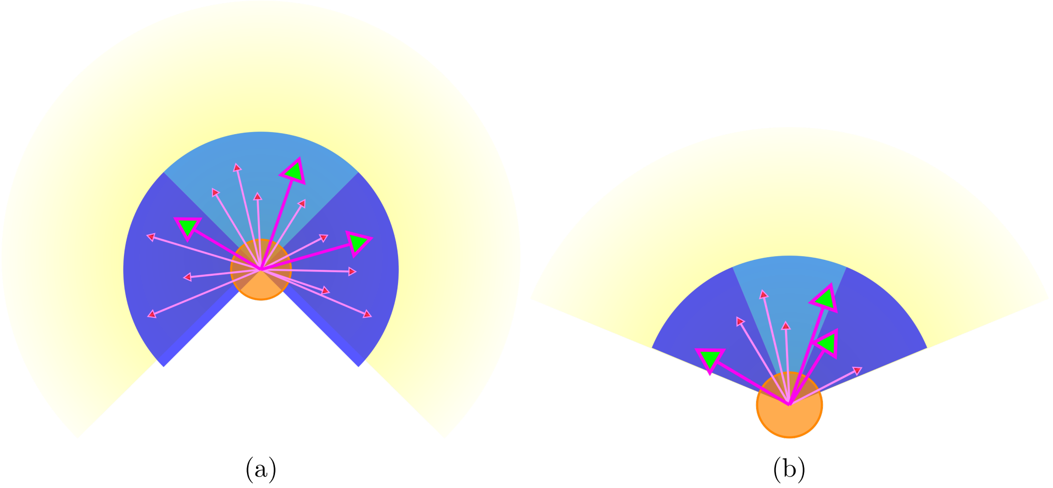


Fig. 5.11 Visualisation of how paths (arrows pointing from the AMR's centre) are chosen, in order to aide exploration of the model environment. A maximum of three paths (highlighted with green arrowheads) are selected, with only one occurring in the left and right (dark blue), and the front (light blue) quadrants of the circle. This path diversification will force the CPP to explore the environment, allowing for considerate paths to be discovered, and thus preventing three similar paths all occurring around the same point. The yellow has highlights the extension of the FoV beyond the path selection area. (a) An example of a FoV with a viewing angle of 270^o, with 3 separate 90^o quadrants. (b) An example of a FoV with a viewing angle of 135^o, with 3 separate 45^o quadrants.

also be limited, preventing too many potential paths being calculated, whilst also restricting the number to choose related to at what t along T, the CPP is planning paths.

Ensuring Path Diversity

After an initial reduction in the repetition of paths, a more explicit method is used to ensure the paths chosen are also divergent. Depending on the population density of the crowd surrounding the AMR the number of potential paths may be very large. Therefore, the maximum number of paths which can be chosen is limited to three, reducing the algorithm's processing speed and path redundancy. However, the three most desirable paths may be very similar, e.g. three paths that move left, in between, and right of two ellipses directly ahead. The three chosen paths are forced to diverge by ensuring they are selected to the left, the right, and ahead of the search's start point at t, by positioning three circular quadrants at these locations around the AMR, Fig. 5.11. Only the most desirable path is chosen from each quadrant, which will force the CPP to move to different parts of the model environment at $t+1$.

Limiting Paths Too Far in the Future

The three paths selected will aide in producing globally diverse paths, however the further into the prediction time-horizon, T, the CPP goes, the more unreliable and repetitive the overall paths may become. To prevent predicting multiple paths with less reliable data, the number of paths predicted at each time-step gradually reduced. T is equally divided into thirds so that for the first third of T three paths are selected; for the second third of T two paths are selected; and for the last third of T only one path is selected. The number of paths at each time-step, p_t, is dependent upon the current time-step as well as the overall T value, which satisfies the following inequality,

$$p_t = \begin{cases} 3, & \text{if } \Delta < \frac{1}{3} \\ 1, & \text{if } \Delta > \frac{2}{3} \\ 2, & \text{otherwise} \end{cases} \tag{5.8}$$

where $\Delta = \dfrac{t}{T} - \dfrac{1}{T}$. The maximum number of local path segments, P_t, is a summation series of nested product series:

$$P_t = \sum_{t=1}^{T} p_t = p_1 + (p_1 \times p_2) + \cdots + \prod_{n=1}^{T-1} p_n + \prod_{n=1}^{T} p_n \tag{5.9}$$

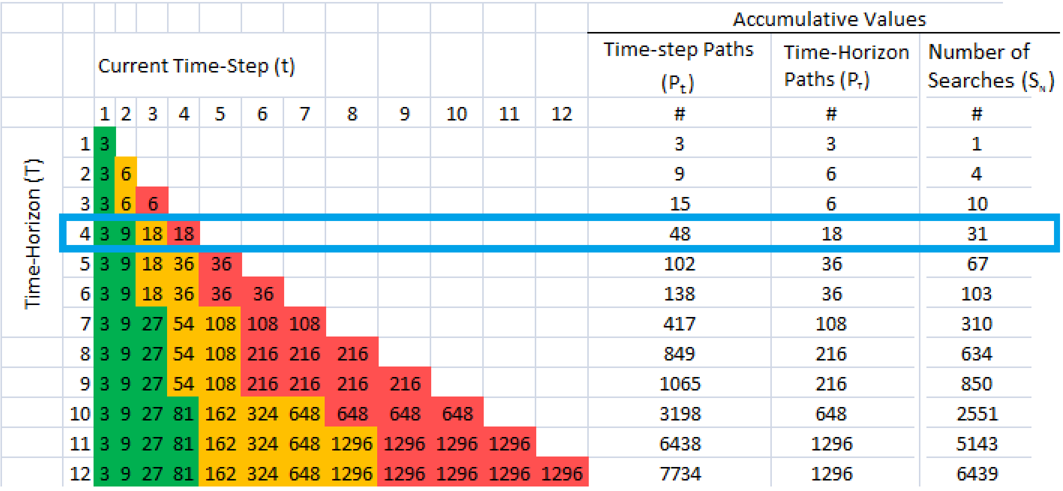
The maximum number of P_T paths, which occur by connecting consecutive P_t paths together for the length of T, is a simple product series, seen as the nested product series in Eq. (5.9):

$$P_T = \prod_{t=1}^{T} p_t = p_1 \times p_2 \times \cdots \times p_{(T-1)} \times p_T \tag{5.10}$$

The maximum number of searches required to create all the potential P_T paths is calculated using the difference between the maximum P_t and P_T paths:

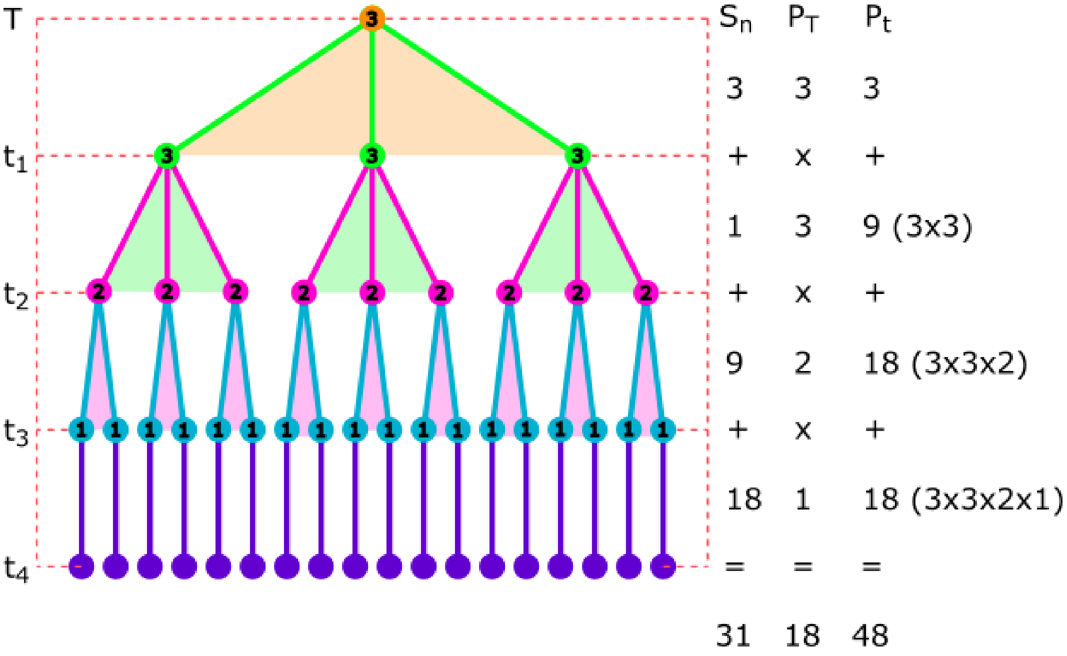
$$\begin{aligned} f(p) &= 1 + P_t - P_T \\ &= 1 + \Big[p_1 + p_2 + \cdots + p_{(T-1)} + p_T\Big] - \left[p_1 + (p_1 \times p_2) + \cdots + \prod_{n=1}^{T-1} p_n + \prod_{n=1}^{T} p_n\right] \end{aligned} \tag{5.11}$$

Fig. 5.12a shows the number of P_t paths, P_T paths, and separate searches required when increasing T from 1 to 12; Fig. 5.12b shows an visual example using a tree structure when $T = 4$.



	Current Time-Step (t)												Accumulative Values		
													Time-step Paths (P_t)	Time-Horizon Paths (P_T)	Number of Searches (S_N)
Time-Horizon (T)	1	2	3	4	5	6	7	8	9	10	11	12	#	#	#
1	3												3	3	1
2	3	6											9	6	4
3	3	6	6										15	6	10
4	3	9	18	18									48	18	31
5	3	9	18	36	36								102	36	67
6	3	9	18	36	36	36							138	36	103
7	3	9	27	54	108	108	108						417	108	310
8	3	9	27	54	108	216	216	216					849	216	634
9	3	9	27	54	108	216	216	216	216				1065	216	850
10	3	9	27	81	162	324	648	648	648	648			3198	648	2551
11	3	9	27	81	162	324	648	1296	1296	1296	1296		6438	1296	5143
12	3	9	27	81	162	324	648	1296	1296	1296	1296	1296	7734	1296	6439

(a)



(b)

Fig. 5.12 Example of the maximum number of FGPs that can be calculated for various prediction time-horizon lengths, T. (a) Each coloured box represents the maximum number of branches that can be created at that time-step (Green: 3, Yellow: 2, Red: 1). The numbers within the boxes represent the maximum possible accumulating calculations required at each time-step.(b) Example of tree when $T = 4$.

Completing the Prediction Time-Horizon

By removing paths that share similar ellipses, limiting the number of P_t paths to a maximum of three, and ensuring that of those three paths they diverge to the left, the right, and forward from the search start, the CPP will be able to create multiple unique, diverse, and considerate P_T paths the length of T.

As can be seen in the CNS flow diagram, Fig. 5.1, if $t < T$ the pedestrian model will update the uncertainty ellipses according to the size and position they should be at $t + 1$. The AMR's FoV that limits what portion of the environment can be searched remains the same, as the AMR has not yet moved. The VD-VG roadmap is recalculated using the updated ellipses, and remains limited to within the FoV as before. However, the searches that find the next P_t paths at $t + 1$ begin at the end of the P_t paths selected at t. The search repeats for all P_t paths until $t = T$, along with the same path filtering techniques, e.g. Fig. 5.13. Once all possible P_T paths have been calculated, as defined by Eq. (5.9), they can be connected to one another to create a global path along the entire prediction time-horizon,

$$P_T(n) = P_1(n) + P_2(n) + \cdots + P_{(T-1)}(n) + P_T(n) \tag{5.12}$$

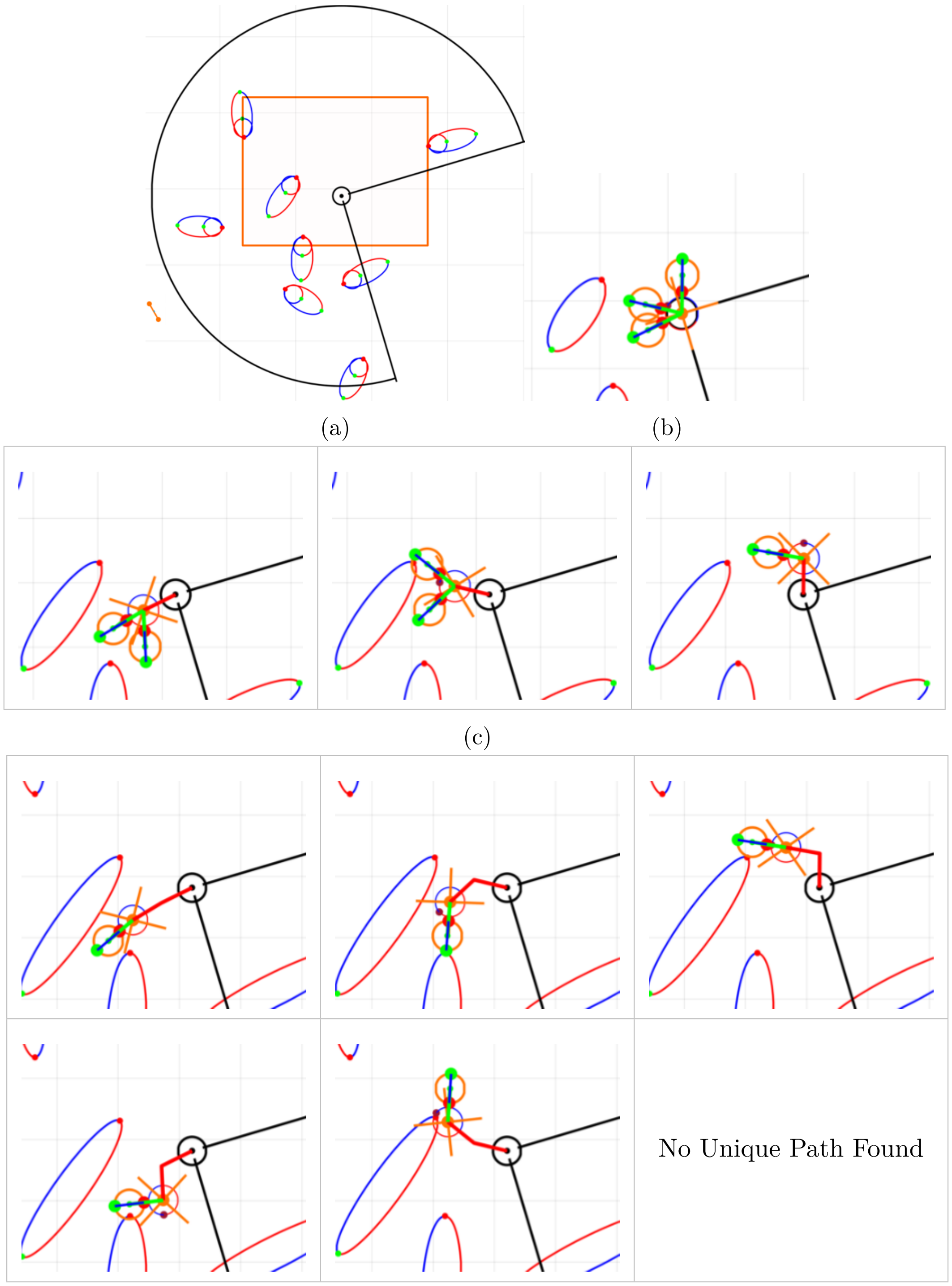
where N is the number of potential P_T paths defined by Eq. (5.10), and $\{n \in \mathbb{Z} \mid 0 \leq n \leq N\}$. The most desirable global P_T path can then be determined by the CPP for the AMR to move along, e.g. Fig. 5.14.

5.5 Executing a Planned Path and Moving the Robot

From the end of the most desirable local P_t paths the search repeats, and the same processes are repeated, Sections 5.2 to 5.4, until the consecutive number of path vectors reaches the prediction time-horizon. Unlike a heuristic best-first search, calculating multiple P_T paths means that a full evaluation of how different paths may be better suited at different times in the future can be made. P_T paths that have undesirable P_t path segments early on may have much less desirable paths further along T, and a sacrifice of desirability early on in fact leads to a better overall path the full length of T.

5.5.1 Finding the Most Desirable Global Path

Once the CPP has planned all P_T paths, it must select the most desirable. As mentioned in Section 3.2, the pedestrian model will become less accurate as T increases. Therefore the desirability, δ, is proportionately weighted along T in order to decrease

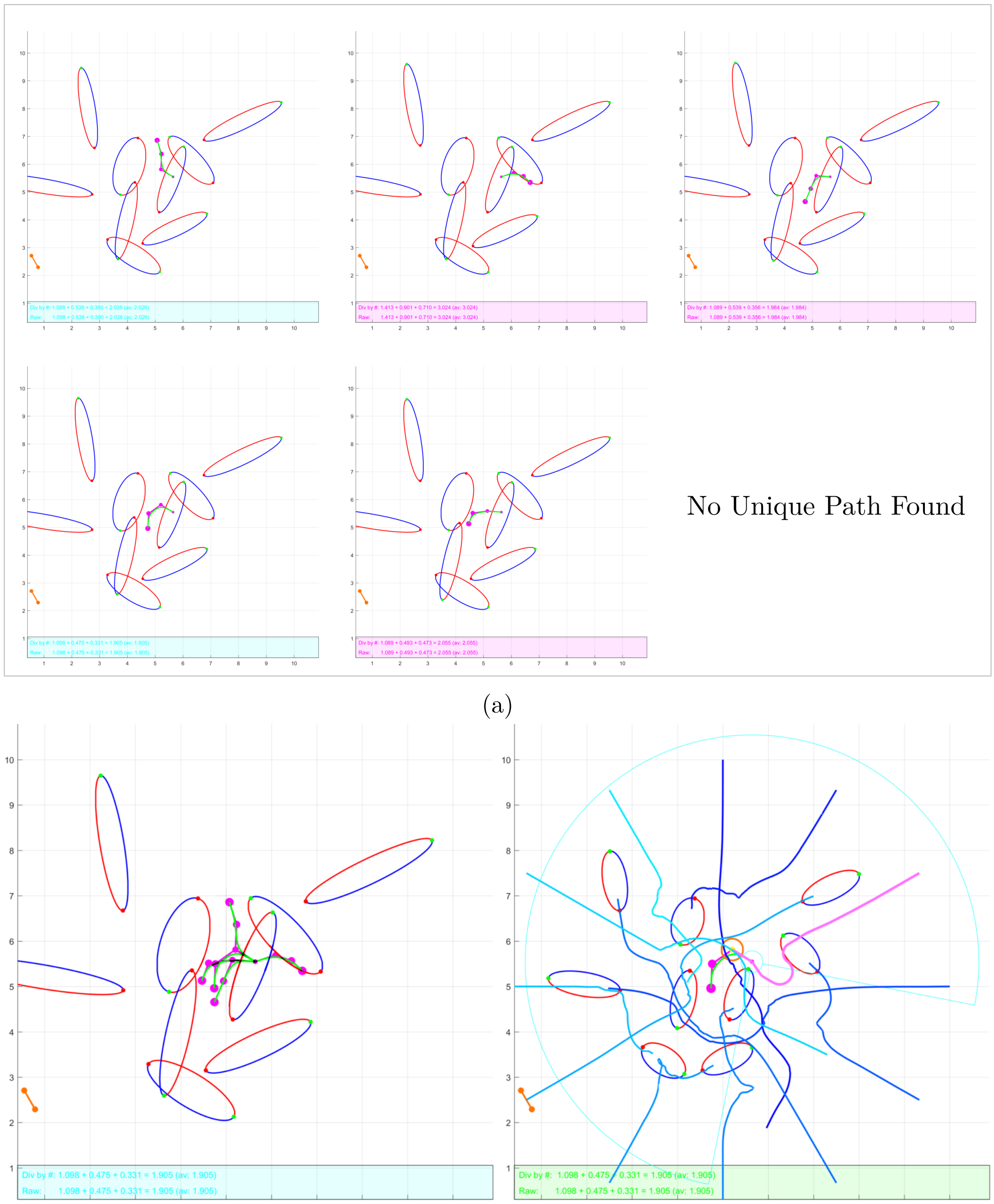


(a) (b)

(c)

(d)

Fig. 5.13 Demonstration of how the CPP moves along P_t paths at each time-step, t, which corresponds with Fig. 5.14. The example uses a prediction time-horizon of $T = 3$ with the number of potential paths and calculations at each t dictated by Eq. (5.8), and shown in Fig. 5.12. Eleven agents are all crossing a central point within the robot's FoV, in order to reach the opposite side of the environment (goal is orange line in bottom-left corner). (a) A full view of the agents within the AMR's FoV. The orange box signifies the area focused on in the proceeding subfigures. (b) At $t = 1$ three P_t paths are selected from the robot's current location. (c) At $t = 2$ two P_t paths are selected from the first two of the previous P_t paths. The remaining P_t only predicts one unique P_t path. (d) At $t = 3$ one intermediate path is selected from the 5 potential locations of the robot at $t = 2$.



(a)

(b)

(c)

Fig. 5.14 Demonstration of all the P_T paths predicted by the CPP for a prediction time-horizon $T = 3$, which corresponds with Fig. 5.13. Eleven agents are all crossing a central point within the robot's FoV, in order to reach the opposite side of the environment (goal is orange line in bottom-left corner). (a) All the individual P_T paths predicted by the CPP, shown with the uncertainty ellipses expanded over the entire prediction time-horizon. (b) All P_T paths superimposed into one image. (c) The most desirable P_T path is chosen to be the FGP. The FGP can also be seen to move in the same clockwise rotation as the other agents.

latter time-steps having as much influence on the final path selection,

$$\sum_{t=1}^{T} \delta_T(n) = \delta_t(1) + \frac{\delta_t(2)}{2} + \cdots + \frac{\delta_t(T-1)}{(T-1)} + \frac{\delta_t(T)}{T} \tag{5.13}$$

where δ_t is the desirability for each individual P_t segment of the associated P_T defined by Eq. (5.7), N is the number of potential P_T paths defined by Eq. (5.10), and $\{n \in \mathbb{Z} \mid 0 \leq n \leq N\}$.

The most desirable P_T path, with the lowest δ_T value, is selected to be the final global path (FGP), e.g. Fig. 5.14c. As global path planning allows many possible eventualities to be assessed before any real movement is executed, the CPP is capable of finding the most considerate paths possible. However, if $T > 1$ the entire FGP cannot be implemented, as the robot only moves one time-step in the real-world. Following this, a new model environment is created and the CPP recalculates a new FGP from newly observed real-world data. Therefore, a suitable final movement vector (FMV) for the AMR to move along is calculated, based on the FGP. Local collision avoidance techniques are implemented in order to ensure the AMR does not collide with any agents in the real-world as it moves along the FMV. Limitations are also applied to the AMR's velocity, R_v, and angular movement, R_θ, in order to simulate more likely real-world kinematics.

5.5.1.1 Calculating the Robot's Final Movement Vector

Rather than the AMR replicating the initial P_t path vector of the FGP, all connections from the AMR's position to all of the nodes along the FGP are evaluated, Fig. 5.15. A vector from the AMR's current position to an intermediate node along the FGP, is assessed for collisions and resistance. If the AMR can move directly toward an FGP node at $t > 1$, with no obstruction or collision potential, a more efficient movement may be possible for the AMR to make.

As the FMV is to be implemented in the real-world, collision assessment is made between vectors from the AMR's position and $P_t(1)$ to $P_t(T)$ of P_T, with agent uncertainty ellipses at $t = 1$, Fig. 5.15. Providing no collision occurs between the vector[1] and the ellipse, an appropriate resistance for the vector is calculated using the method described in Section 3.3.4. The vector chosen as the FMV will be the connection with the least resistance. The AMR can move more directly to where it should be later on, exploiting an immediate *shortcut*. Whilst the speed the AMR will move along the FMV is calculated in the same way as the the search speed is as it propagates through the VD-VG, Eq. (5.3), with the resistance calculated during the FMV's selection.

[1] including the AMR's radius

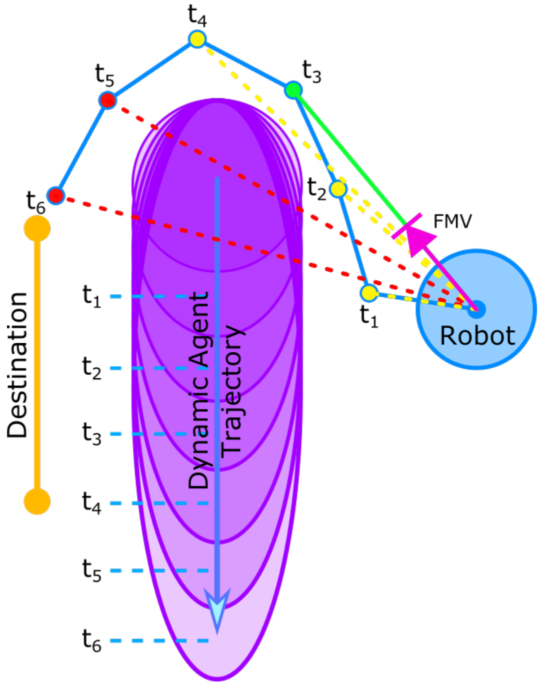


Fig. 5.15 Visualisation of selecting the FMV. The purple ellipse represents the uncertainty ellipse of a dynamic agent over one time-step. The blue path represents the FGP over a prediction time-horizon of $T = 6$. Connections are created between the robot's current location to all of the time-step paths along T, t_1 to t_6. The red lines are deleted due to a collision with the ellipse, whilst the yellow lines represent the potential FMVs available. The green FMV line is chosen as it is the most desirable, using Eq. (5.7).

5.5.1.2 Collision Checking

The FMV is determined by the FGP, and is designed to produce a considerate movement over one environmental frame of the real-world. However, despite how well planned the vector may be the dynamic nature of the environment means that there is no guarantee of avoiding any collisions, especially if the agents are not behaving as predicted. As mentioned in Section 2.1.1, the path from the AMR's current position to its destination should be planned using global path planning methods, whilst updates to the path as it moves should be performed by local collision avoidance. A simple proximity buffer zone (BZ), Fig. 5.16, is applied around the AMR as it moves, ensuring that a collision can never occur due to the AMR[2]. The orientation at which the BZ of the AMR is intersected will determine the appropriate action to take.

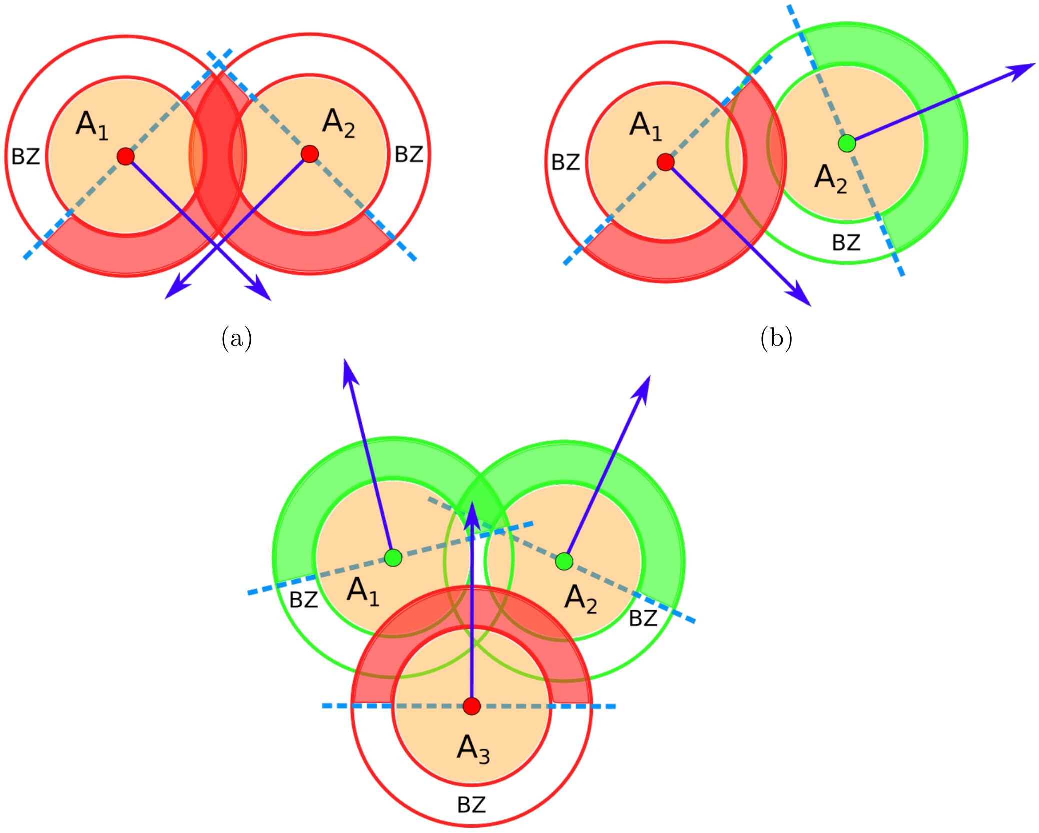
The *Collider* and The *Collidee* If the AMR's BZ intersects an agent, the relative orientation between the two agents will not only determine who collided with whom, but can help maintain crowd flow, Fig. 5.16. In order to prevent a complete halt to the AMR whenever their BZ is intersected, only if the AMR is responsible for the collision will it stop, Fig. 5.16b. Much like in a normal crowd, if someone walks behind a person only they can stop to avoid a collision, which helps the flow of traffic as those in front continue unhindered, Fig. 5.16c.

Avoiding a Collision Although the FMV is chosen to avoid any collisions that occur within the predicted model environment, this may not be reflected in the reality of the real-world. As the AMR moves along the FMV, collision checking is continually implemented to ensure that the BZ is not intersected. If an agent does enter the BZ, rather than have the AMR simply stop for the remainder of the time-step, the remaining segment of the FMV is adjusted to move away from the agent until it provides a collision free trajectory, Fig. 5.17. The *turn* of the FMV occurs incrementally in order to gradually move the AMR away, as aan instantaneous turn would be unnatural. Upon finding a collision free trajectory, the AMR will continue to move until the time-step expires.

5.5.1.3 Differential Drive Model

Although specific dynamics have not been applied to the AMR model used by the search, Section 3.2, in order to move the AMR within the confines of simulating the CNS limitations as to how it can move have been implemented to replicate the kinematics a differential drive, Fig. 5.18. The AMR's velocity, $R_v ms^{-1}$, is set by the length of the FMV (m), divided by the length of the time-step (s) used by the

[2] this is not to say a pedestrian would not collide with the AMR

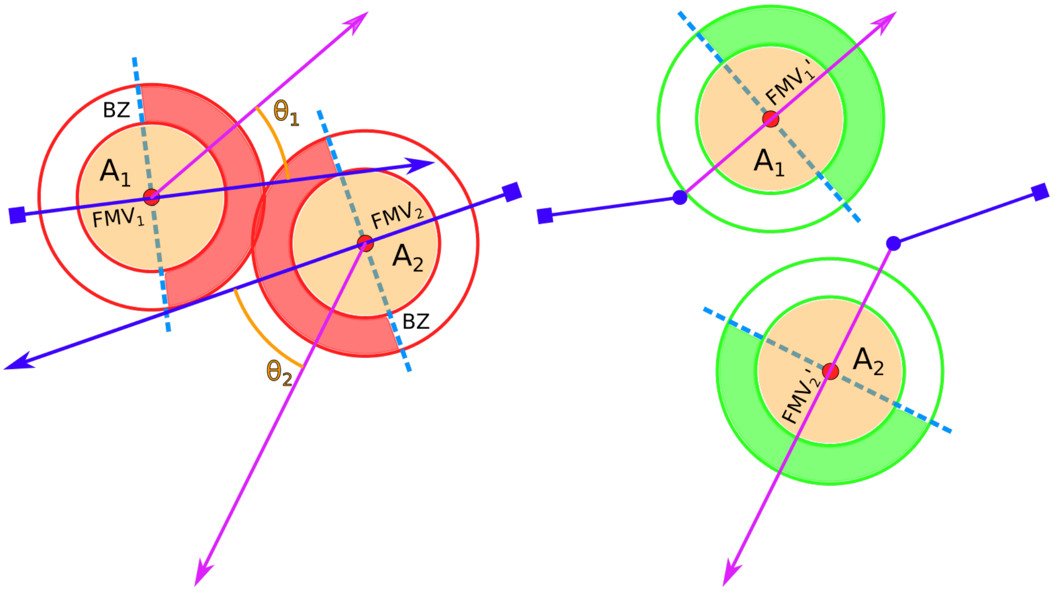


(a)

(b)

(c)

Fig. 5.16 Visualisation of how relative agent orientations determine who collided with whom, and therefore who is required to activate a collision avoidance strategy. For illustrative purposes assume this is an AMR only environment, with all AMRs loaded with the CPP algorithm. An AMR activates collision avoidance when an external agent enters into the buffer zone (BZ) that surrounds the front half of the AMR. The BZ is red when collision avoidance is required, and green when no action is required. (a) Both agents are mutually responsible for avoiding a collision and must stop. Both agents intersect the other's BZ and both their trajectories collide. (b) Only A_1 pauses to avoid a collision, as A_2 is within its BZ. A_2 does not take any action as its current trajectory will not create a potential collision. (c) Only A_3 pauses to avoid a collision, as both the other agents are within it's BZ. A_1 and A_2 do not take any action, despite them entering each other BZs, as their trajectories are diverging. This scenario helps maintain traffic flow as only A_1 needs to stop and manoeuvre around, preventing those in front from also stopping and adding to congestion.



(a) (b)

Fig. 5.17 Example of how the FMV the AMR moves along in the real-world, is updated if a collision is about to occur. For illustrative purposes assume this is a AMR only environment, with all AMRs loaded with the CPP algorithm. (a) As a collision is about to occur between both AMRs, both FMVs move *away* from the collision, θ_1 and θ_2. The remainder of the FMV is rotated away from the collision, using the AMR's current position as the point of rotation. This incremental rotation continues until the remainder of the FMV is aligned with a collision free path. (b) Both FMVs have rotated away from the collision, and so both AMRs can now continue to move to the remainder of the time-step.

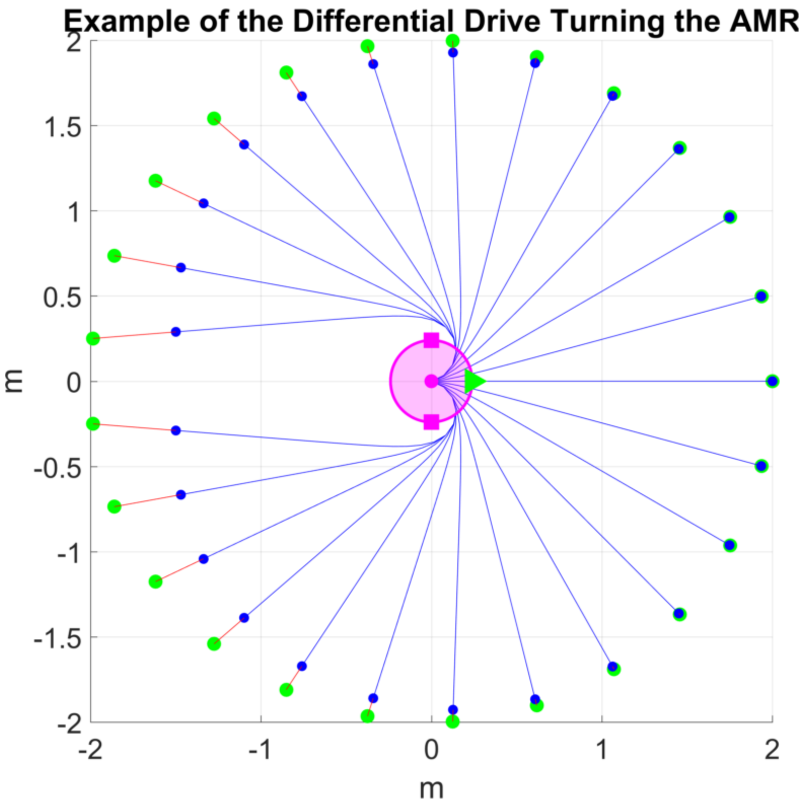


Fig. 5.18 Example of how the differential drive would move an AMR from the origin facing right, to various destinations (green dots) all at 2m distance from the AMR. The specifications of the model drive are: $R_v = 2.0ms^{-1}$, $\max(R_\omega) = \frac{\pi}{4}rad \cdot s^{-1}$, and $t = 1s$, whilst the AMR's position is updated every 0.1s $(10Hz)$. The blue lines represent the movement of the AMR for the length of t, as it moves towards the goals.

CPP. The differential drive is set to move the AMR to the end of the FMV with a maximum turning angle, $\max(R_\omega)$, of 45^o. Therefore, it is only possible for the AMR to arrive at the end of the FMV if its angle has not changed since from the previous.

5.5.2 Arriving at the Destination

The underlying aim of the CPP is to navigate an AMR considerately to its destination. Rather than using a traditional destination *point*, which the AMR navigates towards, a destination *line* is implemented. A line is used as it offers a more natural object to aim for, such as a door, wall, or the edge feature of a map. A point also requires a great deal of accuracy to position over, and if the AMR's turning ability is limited it may spend an inordinate amount of time unnecessarily re-positioning itself. Also, if the environment contains multiple agents all heading for the same point, collision potential will increase. Using the *line* method is preferable as it

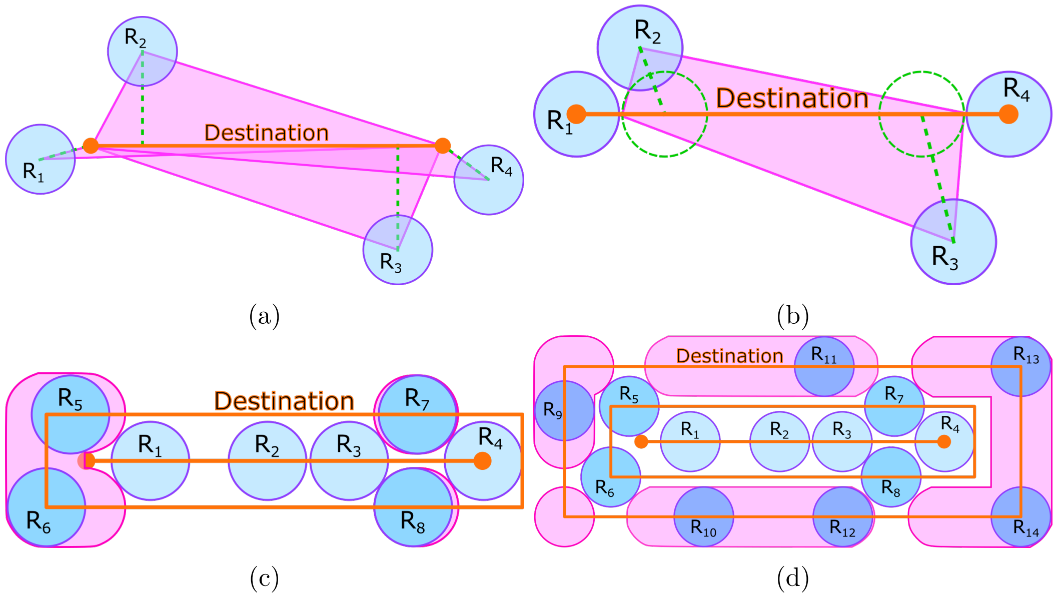


Fig. 5.19 Visualisation how AMRs ($R_1 - R_4$) move toward, and arrive at, the destination line. (a-b) The most direct route to the destination line is either perpendicular to it, or towards one of the ends. If other agents arrived at the same destination, and block a direct route, the approaching AMR moves along the shortest collision free path. (c-d) If there is not a gap of at least twice the AMR's radius along the destination line (less than just the radius at the end points), it is not possible for the AMR to reach its destination collision free. The destination line will therefore expand outwards by the AMR's radius, until a gap occurs that is suitably large enough for the AMR to reach. This new two-dimensional destination will allow the AMR to aim for a destination that is as close as it can get to the original, collision free. The pink bubbles represent the areas that the AMR can safely reach without colliding with agents that are already positioned along the 2D destination. Every time the destination becomes unreachable without collision it continues to expand until there is suitable room for the AMR to arrive.

is similar to a race finish line; the AMR will have reached its destination when it crosses the line, Figs. 5.19a and 5.19b.

As the AMR's destination line is known only to the CPP, at any moment the destination may become unknowingly blocked by another agent. It may also become blocked if other agents have already reached the same destination and are positioned over it. Under such circumstances the AMR should attempt to get as close as possible to the original line. If there are no gaps along the line that the AMR can safely fit into, the line expands outward in order to make gaps big enough for the AMR, Fig. 5.19c. Whenever an updated expansion of the line has no more room for a AMR to safely fit, it will continue to expand further, Fig. 5.19d. This ensures the AMR makes it as close as possible to the destination, if reaching it is impossible.

5.6 Chapter Overview

This chapter has presented the algorithmic implementation of CNS, which plans and executes the most considerate paths through a dynamic environment. The CNS models the real-world environment from what can be observed within the robot's FoV. The CPP then predicts a number of the most desirable paths at each time-step, each of which will consecutively connect and form a number of final global paths spanning the length of the prediction time-horizon. The most desirable paths are chosen primarily due to the consideration resistance, Ω, encountered as the search algorithm progresses. Ω inhibits the frontier's search speed, which represents the potential movement the AMR may take in the real-world. Two additional features modify the desirability of each P_t path to incentivise the AMR to move towards the destination, as well as encourage exploration of the model environment. The closer each P_t path end is to the destination, and the larger the path end's displacement from the start, results in additional desirability being added, respectively.

The FGP selected for the AMR to move along, is chosen using the cumulative desirability acquired at each P_t, within the selected P_T. By updating the model environment at each time-step, the CPP can evaluate potential paths that evolve *with* the crowd. As the model environment is a prediction of how the real-world environment might change over the prediction time-horizon, the likelihood of the robot experiencing a collision is reduced. Therefore, only simple local collision avoidance is used, which will adjust the AMR's trajectory if it is about to cause a collision.

The novel ability for the CPP to generate considerate paths is calculated using very simple but effective means. By *embracing the uncertainty* when using a simple CVM, in place of an alternative *complex* model, areas the CPP is unsure an agent may be can simply be avoided. Alternative path planners, which assume more accurate agent trajectory predictions, cannot handle much prediction error, which will may result in more potential collisions if the agents do not behave as expected. The CPP provides a novel and effective alternative, enabling the inherent unpredictability of a dynamic environment, such as a crowd, to be used to its advantage.

The next chapter will evaluate the CNS' performance within a simulated environment, using a microscopic pedestrian simulator. The movement of the AMR will be evaluated using a set of novel quantitative performance metrics that will quantify that amount of *considerate movement* the CNS exhibits towards an approximation of reality.

Chapter 6

Testing the Considerate Navigation Strategy

"Klaatu barada nikto"

— Edmunh H. North, *The Day the Earth Stood Still*

THE considerate navigation strategy (CNS) is intended too operate within a dynamic pedestrianised environment. This chapter is designed to evaluate specific design elements of the considerate path planner (CPP), and establish the optimal parameters for the CNS so that it can be successfully implemented as intended. Initial test variables and parameters are first established, Section 6.1, so that suitable simulations can be run. A virtual environment will be loaded with multiple dynamic agents, including similar autonomous mobile robots (AMRs) operating with the same CNS strategy, and microscopically modelled pedestrians from a simulator. The performance of the CNS will be evaluated against novel quantitative performance metrics (QPMs), Section 6.2. These will allow the specific objectives set out at the start of this thesis, Section 1.3.2, to be evaluated, and establish if the CPP can produce considerate paths that the CNS can then successfully execute.

An assessment for each of the design elements of the CPP is then made, Section 6.3. The reliability of the pedestrian model, presented in Section 3.2, is performed to see how accurately its predicts the movement of pedestrians, over an increasing prediction time-horizon, T. By recreating empirical studies discussed in Section 2.3.3.1, an evaluation of the novel Voronoi diagram-visibility graph (VD-VG) roadmap, presented in Chapter 4 is made, as well as the VD and VG elements when used as individual roadmaps.

Finally, the CNS is evaluated, Section 6.4, using simulated pedestrians and similar AMRs. A number of different scenarios are trialled, each of which will require multiple dynamic agents to interact with one another from various angles. Using the novel QPMs for consideration, the CNS is evaluated to see how well the AMR performs at various values of T, so that the optimal prediction time-horizon can be found. The results are also compared to a standard A* algorithm, as well a modified *considerate* A* (CA*) search, in order to highlight the CNS's improvements of pre-existing searches.

6.1 Establishing Test Variables and Parameters

This section presents the configuration requirements needed in order to implement the full CNS, which occur before the input and after the output of the CPP One of the most significant factors to impact the CPP's ability to predict global paths is the size of the AMR's field of view (FoV), as anything outside the FoV will be undetectable. It will also directly limit the maximum length of the prediction time-horizon, T, as predictions should not go beyond the FoV into unknown areas. How various prediction time-horizon lengths affect the behaviour of the CPP will therefore be investigated, to demonstrate its affect on global path planning.

6.1.1 Robot Model Parameters

The AMR model, Section 3.3.1, can now have appropriate parameters set so that the CNS can be simulated. The range and span of the AMR's input FoV will be based upon HOKUYO UST-10LXa high precision laser range-finder. The variable prediction time-horizon parameter of the CPP to be tested will be vary from the local vicinity to the range of the FoV. The AMR's differential drive movement kinematics will remain the same as described in Section 5.5.1.3, using the Milvus Robotics MRP2 platofrm as a basis.

a_1 **Input:** Real-world data obtained from within the AMR's FoV,

- $a_{1.1}$ FoV distance range, $\text{FoV}_r = 5\text{m}$ - also approximate human visual range of interest [2].
- $a_{1.2}$ FoV angular range, $\text{FoV}_\theta = 270°$ - allows the 90° left-forward-right areas in front of the robot, Fig. 5.11, to be equally divided.

a_2 **Processing:** The customisable CPP parameters that affect consideration to be tested,

- $a_{2.1}$ Variable prediction time-horizon, $T = 1$ to 7
- $a_{2.2}$ Variable maximum speed of the robot, $\max(R_v) \leq 2.0ms^{-1}$

a_3 **Output:** The AMR's kinematics, Section 5.5.1.3,

- $a_{3.1}$ Turn of the robot, $-45° \leq R_\theta \leq 45°$.
- $a_{3.2}$ Speed of the robot, $R_v \leq \max(R_v)$
- $a_{3.3}$ Wheeled platform operated using a differential drive

	Operational: Local $(0 < t < \frac{1}{3}T)$		Tactical: Sub-Global $(\frac{1}{3}T < t < \frac{2}{3}T)$		Strategic: Global $(\frac{2}{3}T < t < T)$	
t	1	2	3	4	5	6

Table 6.1 The division of the robot's FoV distance range, FoV_r, is representive of the spatial path planning ranges observed in humans, Fig. 2.9. For the FoV_r used in these simulations the maximum prediction time-horizon cannot exceed $T = 6$.

The range of T values is reflective of the spatial path planning ranges of a human, Fig. 2.9. By dividing the range of the FoV into thirds the three resultant ranges of T should provide the same strategic, tactical, and operational levels of behaviour, Table 6.1.

6.1.2 Adding Noise Variables

The CPP is to be implemented in the real-world environment, which will inevitably result in errors regarding sensor measurements, and the execution of the final movement vector (FMV), Section 5.5.1.1. To make the simulations more realistic, noise will been introduced to both the input and output stages of the CPP to corrupt the data [256]. To further replicate noise levels encountered in the real-world, the data will be corrupted in accordance to the accuracy levels of genuine products. The noise is expressed as a percentage of the values it is corrupting, assuming a uniform distribution.

6.1.2.1 Input Noise

Object detection errors, within the robot's FoV, will contribute towards errors in the positions of agents within the model environment. For the purposes of these simulations positional noise will be added to the positional Cartesian co-ordinates of all agents within the robot's FoV. The error will be in reference to radii of the agents detected, e.g. if applying a noise level of 1%, to an agent with a radius of 10, the agent's position will range from $\pm 10 \times \frac{1}{100} \Rightarrow (A_x \pm 0.1, A_y \pm 0.1)$. The uncertainty ellipse of each agent will also be affected, as the data used to produce it will be noisy.

6.1.2.2 Output Noise

The robot platform used by the robot model will have a differential drive and a maximum speed of $2.0ms^{-1}$. Noise is added to the robot's trajectory (direction of the FMV) and the the robot's velocity (length of the FMV). The noise has been added to the FMV as it is assumed that there will be no wheel slippage of the AMR platform.

Angular noise will corrupt the turn the robot makes for every FMV, e.g. applying a noise level of 5% to a robot with a turn of 45°, will result in the FMV differing from the original by $\pm 45 \times \frac{5}{100} \Rightarrow \text{FMV}_{\theta} = 45 \pm 2.25°$. Velocity noise will corrupt the length of the FMV, which will therefore affect the robot's velocity as the FMV represents the AMR's velocity trajectory in the next timestep, e.g. applying a noise level of 5% to a robot with a velocity of $2.0ms^{-1}$, will result in the robot's velocity ranging from $\pm 2 \times \frac{5}{100} \Rightarrow R_v = 2 \pm 0.1ms^{-1}$.

6.1.3 Dynamic Movement & Interaction

When the AMR executes the paths planned by the CPP the movement and interaction of the AMR with other dynamic agents will be evaluated. Scenarios are devised that involve multiple dynamic agents crossing each others paths, at different crowd densities. To test how considerate the overall CNS is, each scenario is repeated using dynamic agents of similar AMR's programmed with the CPP; and dynamic agents from the microscopic pedestrian simulator "PedSim", which uses the social force model.

6.1.3.1 Crowd Densities

Different levels of crowd density (people, p, per square metre, m^{-2}) will result in different behaviours due to the "level of service" [24], Table 6.2. In low density crowds ($p \leq 0.6$) pedestrian behaviour is typically to walk directly towards one's goal, reactively avoiding collisions en route. For the AMR, traditional object avoidance path planning strategies can be used. In medium density crowds ($0.6 < p \leq 0.75$) pedestrian behaviour can be replicated by developing non-disruptive path planning techniques that consider proxemics and respect each pedestrian's spatial zones. This is achieved via the accurate prediction of pedestrian intentions, allowing the AMR to avoid a hypothetical collision with a neighbouring pedestrian. In high density crowds ($0.75 < p \leq 2$), collaborative path planning between AMR and pedestrian can allow mutual collision avoidance of hypothetical collision points to be achieved together. For higher crowd densities ($p > 2$) the individual microscopic crowd dynamics are replaced by macroscopic dynamics, as the crowd becomes too dense.

6.1.3.2 The "PedSim" Microscopic Simulator

As it is not yet possible to implement the CNS onto a real-life robot platform, it must be simulated under an approximation of reality. This pedestrian simulator [16] is based upon the popular agent-based social force model, Section 2.3.2.3, often used for pedestrian simulation. To establish what level if social force, SF, is required (the accompanying documentation recommends a SF of 1 to 10), the positions of each

Crowd Density	People (p) per square metre	Flow Description	Example locations
Low	$p \leq 0.6$	Free	Public Parks
Medium	$0.6 < p \leq 0.75$	Impeded	Shopping Centres
High	$0.75 < p \leq 2$	Dense	Central Transit Centres
Very High	$2 < p$	Jammed	Not Recommended

Table 6.2 Values for various crowd densities, formulated from [24]

pedestrian that appear in 10 consecutive frames of the BIWI pedestrian dataset [17] are recorded. For each recorded pedestrian path the start and end points are loaded into the simulator. To see how closely the simulator directs the simulated pedestrians to their goal, in comparison to the dataset, the difference in the position between each individual pedestrian's datapoint and the associated simulated pedestrian are evaluated at each timestep, Fig. 6.1. From these observations it is deemed that $SF = 1$ is the closest approximation of reality, as the simulated pedestrians deviated the least from their real-life counterparts.

6.1.3.3 Similar Robots

This will test the hypothesis that two considerate navigation strategies can mutually respond to one another and become implicitly cooperative, which is observed to be the case between pedestrian interaction, Section 2.3.3. Robots will be set up with the same prediction time-horizons and speeds in order to evaluate them equally. Also, whenever more than one is tested together, each AMR will use the same T value for the CPP.

Using similar AMRs is also beneficial at aiding the CNS evaluation. When an AMR moves towards the end of the FMV, Section 5.5.1.1, it does so for the full timestep, without the CPP replanning any other path; readjusting only if a collision occurs, Section 5.5.1.2. Therefore, each AMR moves for upto $0.56m$ ($0.4s \times 1.4ms^{-1}$) along only one path planned by the CPP. However, the PedSim constantly updates each pedestrian's position using the social force model as the AMR moves within the environment. As the PedSim pedestrians' movements can change direction without turning limitations, by using similar AMR's bound by the differential drive kinematics, Section 5.5.1.3, this will help evaluate the CNS at manoeuvring around dynamic agents with more limited movements.

6.1.4 Assumptions & Limitations

As the CNS is evaluated by simulation, both the collection of input data to be used by the CPP and the movement of the AMR are all artificial. The introduction of noise, Section 6.1.2, helps to replicate real-world variability, i.e. input noise imitat-

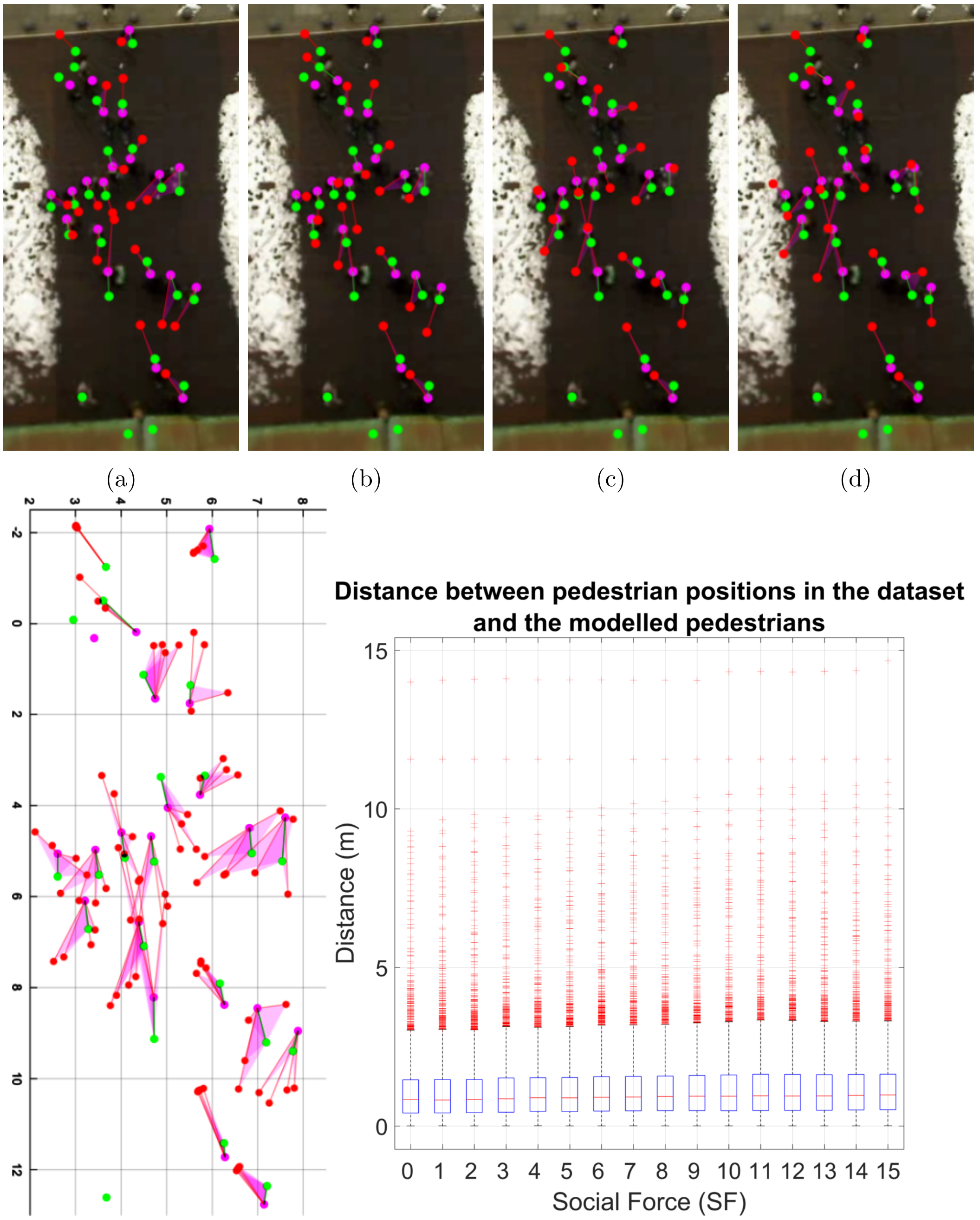


Fig. 6.1 Examples of how well the microscopic "PedSim" pedestrian simulator [16] replicates the trajectories of pedesrian from the BIWI dataset [17], using the social force (SF) model. (a-e) Examples of how much the simulator deviates from the dataset for the initial timestep, $t = 1$. The magenta dot is the location of the pedestrian in the dataset at $t = 0$. The green and red dots show the position of pedestrians at $t = 1$ for the dataset and simulator, respectively. Both green and red dots are connected to their respective initial pedestrian's position by a line of the same colour. (a) $SF = 0$. (b) $SF = 1$. (c) $SF = 5$. (d) $SF = 10$. (e) Superimposed image of all simulator results displayed in (a-c). (f) The boxplots measures how accurately the simulator recreates the movement of pedestrians, over a prediction time-horizon of $T = 10$, by measuring the distance between the datapoints and the model positions at each timestep. The PedSim documentation recommends a SF value anywhere between 0 to 10, and indeed the results plateau beyond this range. Although the difference is minimal between the various values of SF, $SF = 1$ does provide the most accurate reproduction, indicating that pedestrians deviate only minimally from a constant velocity.

ing distance measurement errors; and output noise imitating AMR wheel slippage. However, a number of assumptions are made concerning the dynamic agents, and how their relevant data is processed, given below:

b_1 **Dynamic Agents:** Behaviour

- $b_{1.1}$ All agents do not respond abnormally to the presence of the AMR
- $b_{1.2}$ All pedestrian data is reliably pedestrian in behaviour

b_2 **CNS Input:** Agent data

- $b_{2.1}$ All agents are individuals of the same size, based on the average shoulder width of a human
- $b_{2.2}$ All agents within the FoV range are guaranteed to be detected, with no occlusion
- $b_{2.3}$ All agents are uniquely identifiable, Section 5.1, which is required to calculate their velocity

b_3 **CNS Processing:** The CPP algorithm

- $b_{3.1}$ The timsetep used for these experiments is $0.4s$. This is assumed to be reasonable as the BIWI dataset of pedestrians used in this thesis is sampled at this rate [17].
- $b_{3.2}$ The path is calculated instantaneously, and the real-world does not change during the interim

b_4 **CNS Output:** AMR kinematics

- $b_{4.1}$ Instantaneous acceleration
- $b_{4.2}$ Velocity constant when moving during each time-step
- $b_{4.3}$ Localisation of the AMR is perfect when calculating the distance to the goal

Limitations occur when the CPP is processing, due to software/hardware specifications: as the CPP plans the path the world does not change in the interim. This would limit the application of the CPP in a real-world environment, due to processing speed. However, all assumptions and limitations are acceptable for evaluating how considerate the CNS will be, and now how to measure consideration is defined.

6.2 Measuring Consideration

The FGP is the most desirable path plotted within a model environment of the real-world, whilst the FMV is the robot's movement in the actual real-world. Novel QPMs for consideration are defined to evaluate how considerate the robot is when moving along the FMV, and determine the CNS's navigational performance.

Consideration can be observed through how the robot directly and in-directly interacts with pedestrians:

c_1 Direct (Active) Considerate Behaviour

- $c_{1.1}$ Move behind pedestrians if needing to move across their trajectory
- $c_{1.2}$ Avoid moving directly in-front of pedestrians
- $c_{1.3}$ Move out of a pedestrian's path if moving towards each other

c_2 In-direct (Passive) Considerate Behaviour

- $c_{2.1}$ Maintain a constant velocity
- $c_{2.2}$ Maintain a minimum distance between pedestrians
- $c_{2.3}$ Move with crowd flow

Although a quantitative assessment allows the most desirable FGP to be chosen, Section 5.2.2, the AMR's performance must be evaluated using QPMs as it travels along the FMV in the real-world environment. Novel QPMs must be explicitly defined for an AMR's movement along the FMV, as none yet exist to evaluate considerate behaviour.

Traditional metrics are first assessed in order to establish what is currently used by AMR's to evaluate their performance. Novel modifications are made based on *near-miss* collisions, so that each one is not considered the same. Finally, a list of novel QPMs for consideration is defined in order to asses how considerate the AMR actually is, as it moves along the planned considerate paths.

6.2.1 Traditional Metrics

Traditional robot path planners are optimized to find the shortest route (either time or distance) from the robot's current location to its destination, and for any AMR collision free navigation is essential. Instinctively this infers the four primary QPMs used in many systems [70]:

d_1 **Journey Success:** How many times the AMR arrived at its destination

d_2 **Journey efficiency:** ideal vs. executed

- $d_{2.1}$ **Time:** how quickly did the AMR arrive at its destination
- $d_{2.2}$ **Distance:** how long was the route the AMR took to its destination

d_3 **Collisions:** Number of collisions experienced or prevented en route

These basic QPMs are essential to evaluate an AMR's autonomy level. Additional QPMs can then be used, specific for each AMR's purpose.

6.2.2 Appraising Near Misses

As mentioned in Section 5.5.1.2, the AMR is surrounded by a buffer zone and will perform an *emergency stop* if intersected, so as not to collide with anything. This is also dependent upon whether the robot is responsible or not, i.e. has the robot moved along a path that has contributed towards colliding with a pedestrian?

Often collisions will be inevitable [152], especially in a crowded and dynamic environment. However an AMR may have only *grazed* past a pedestrian, making it unfair to regard each *emergency stop* as a simple near collision. Therefore, each is evaluated in order to establish the *collision intensity* that would have occurred had the AMR not stopped:

e_1 **Near-miss Velocity**: normalised velocity of the robot at the moment of emergency brake. This measures the potential force the robot would impart on the pedestrian. e.g. a faster the robot the more forceful the collision.

e_2 **Near-miss Incidence**: normalised impact angle between the robot and pedestrian at the moment of emergency brake. This measures the severity of the impact, based on how direct the collision was. e.g. the smaller the angle the harsher the collision.

e_3 **Near-miss Collision Intensity**: total magnitude of the emergency brake, as a product of the near-miss velocity and near-miss angle of incidence together.

By assessing the total intensity of a near-miss collision that intersects the robot's buffer zone, *minor bumps*, or *slight deflections*, will not carry the same gravitas as a more brutal head-on collision at full speed.

6.2.3 Novel Quantitative Performance Metrics for Considerate Navigation

At the beginning of this section the list of considerate behaviours, (Items c_1 and c_2), have not been previously defined by as QPMs. Therefore, a novel set of QPMs for considerate behaviour will be constructed, using the orientation between a dynamic agent and the AMR. All metrics are measured when the AMR is within an elliptical proxemics field that surrounds the agent, and orientated along the agent's current trajectory, which would make the AMR close enough to the agent for *interaction* to occur.

The metrics will quantify how the AMR crosses a pedestrian's trajectory or moves with the crowd flow, in order to evaluate the thesis' aims & objectives, Section 1.3.2. The rules are generalised to evaluate how the AMR crosses in front or behind a pedestrian, or *flows* with or against their movement, Fig. 6.2, within a $\pm 45^o$ viewing angle. The rule base is defined as follows, providing the AMR is within the proxemics field:

f_1 **Trajectory Crossing:** The AMR must be orientated ±45^o from a perpendicular intersection with the pedestrian's trajectory line.

$f_{1.1}$ **Behind:** *Considerate* movement in accordance with Item $c_{1.1}$.

$f_{1.2}$ **In front:** *Inconsiderate* movement in accordance with Item $c_{1.2}$.

f_2 **Flow Motion:** The AMR must be orientated ±45^o from the pedestrian's trajectory line.

$f_{2.1}$ **Behind:** *Considerate* movement in accordance with Item $c_{2.3}$.

$f_{2.2}$ **In front:** *Considerate* movement in accordance with Item $c_{2.3}$, and *inconsiderate* movement in accordance with Item $c_{1.3}$.

A tally will be made based on *considerate* or *inconsiderate* positions of the AMR at each updated position as it moves toward the end of the FMV.

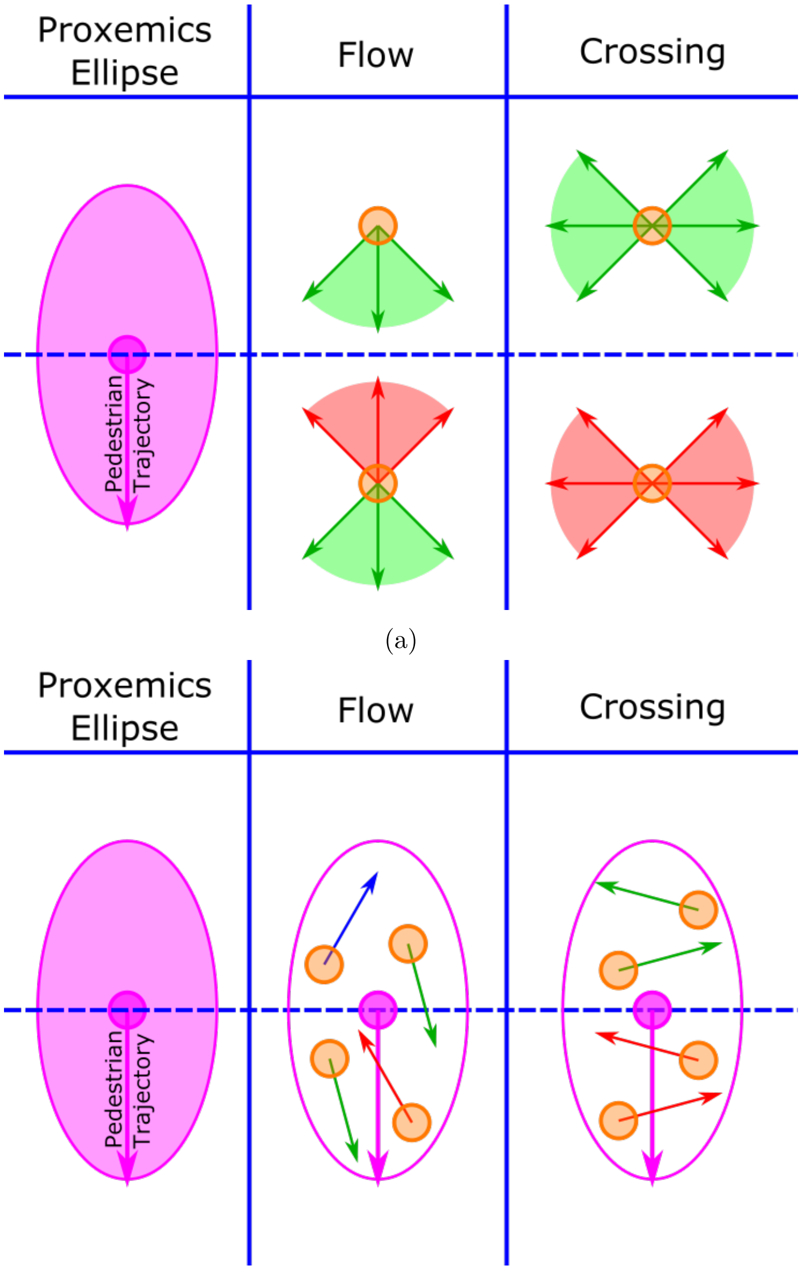
The metrics will be normalised so that the higher the value of each metric, and so the total combined value of all, represents the best performance. Therefore, it may appear counter-intuitive when tabulating, but larger values for distance and time taken, and number of inconsiderate movements, are actually preferable when presented in graphs. Establishing considerate behaviour is paramount, therefore traditional metrics will not be included as it is acceptable for the CNS to take a longer route to the goal if it means behaving more considerately. The *near-miss* metric, Item e_3, will be included to allow an evaluation of how accurate the CPP was at planning paths with minimal collisions.

6.3 Testing the Considerate Path Planner Design Elements

Before the CNS is tested on a crowd in the proceeding section, each of the CPP design elements are evaluated. A simple scenario designed that recreates the empirical studies of human-human collision avoidance in [18], with two dynamic agents crossing a hypothetical collision point at the centre of the environment, Fig. 6.3.

6.3.1 The Pedestrian Model

The uncertainty ellipse is designed to *embrace* any uncertainty in a pedestrians predicted movement, therefore as long as the pedestrian remains in the ellipse its purpose has been fulfilled. However, as mentioned in Section 3.2, by not limiting the uncertainty enough can lead to an *uncertainty explosion.* Therefore, to establish the effectiveness at estimating the level of uncertainty required by the pedestrian model, Eqs. (3.5) and (3.6) and Fig. 3.6, the distance from the pedestrian to the ellipse edges will be calculated at each t along T, Fig. 6.4. The results are compared to ellipses that are sized based on only the statistical deviations to a CVM, as observed in Section 3.2.1.



(a)

(b)

Fig. 6.2 Visualisation of how the 4 consideration metrics, Section 6.2.3, are calculated, using an ellipse orientated around each pedestrians position in the style of a standard proxemics field [11]. (a) The rules dictating considerate movement simply depends on the angle the AMR (orange circle) moves with respect to the pedestrian. If the AMR moves along a trajectory within the 90^o range of any green quadrant, the movement is considerate; i.e. Moving with the pedestrian flow, or crossing behind the pedestrian. If the AMR moves along a trajectory within the 90^o range of any red quadrant, the movement is inconsiderate; i.e. Moving against the pedestrian flow, or crossing in front of the pedestrian. (b) If the AMR enters the ellipse, its orientation will determine if it is moving considerately (green arrow) or inconsiderately (red arrow) If the AMR is moving against the flow behind the pedestrian, the movement is neither considerate or inconsiderate (blue arrow).

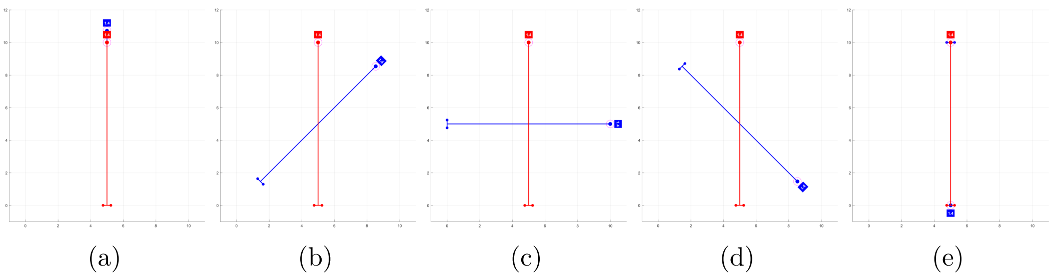


Fig. 6.3 Set-up used in the empirical studies of human-human collision avoidance [18]. The dynamic agent (red line) always moves down, and the AMR (blue line) moves across at 5 separate angles. (a) 0^o relative angle between the agent and AMR. (b) 45^o relative angle between the agent and AMR. (c) 90^o relative angle between the agent and AMR. (d) 135^o relative angle between the agent and AMR. (e) 180^o relative angle between the agent and AMR.

6.3.2 The Considerate Path Planner

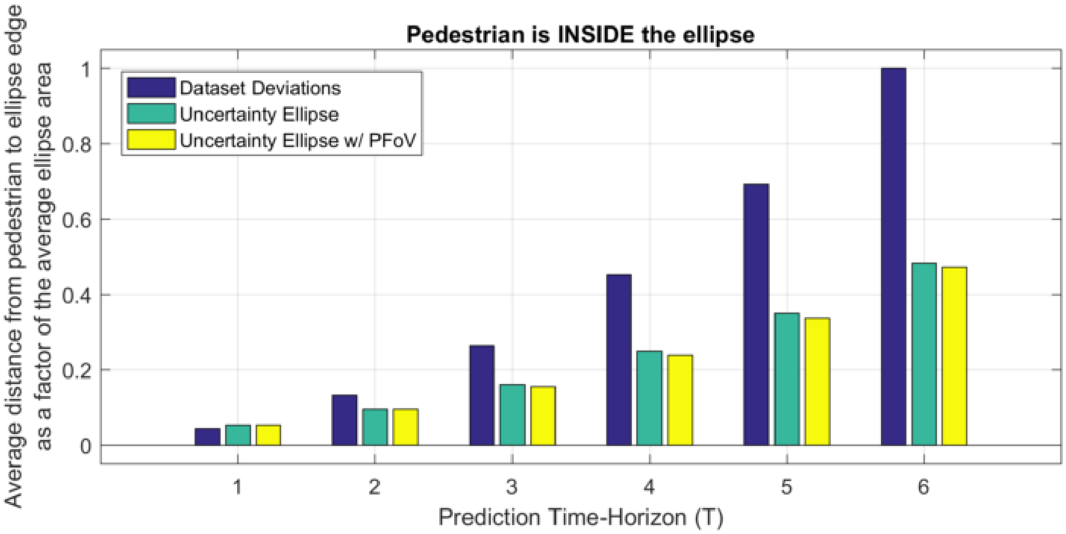
When recreating the empirical studies, the AMR will interact with non-reactive agents in order to evaluate its behaviour regarding *pure* consideration, as well reactive agents in order to evaluate its behaviour regarding implicit cooperation. Having the CNS move an AMR around only one dynamic agent will allow the baseline performance of the system to be analysed. The direct response of the CNS to a dynamic agent can be clearly observed, as multiple agents would not necessarily make it clear what aspect the CNS was responding to.

The dynamic agents (light blue paths) will move down the environment, whilst the CNS will move AMRs (dark blue paths) at angles of 0^o, 45^o, 90^o, 135^o, and 180^o relative to the initial agents position, Fig. 6.3. Within each example, the hypothetical collision point will be in the centre of the environment, and both AMR and dynamic agent will be set to travel at a maximum speed of $1.4ms^{-1}$.

6.3.2.1 A Demonstration of Consideration & Implicit Cooperation

To evaluate the very basic notions of consideration, a simple experiment is first presented of an AMR moving past a dynamic agent at 180^o, Fig. 6.5. The experiment is performed over all variable prediction time-horizon values of the CNS ($T = 1-6$), and is tested against a non-reactive agent; a similar AMR using the same value of T; and a simulated pedestrian from the PedSim simulator.

For all three dynamic agents in each experimental set-up, it is clear that increasing T results in a smoother movement of the AMR. The uncertainty ellipse of the non-reactive agent, Fig. 6.5a to Fig. 6.5f, is guaranteed to contain its predicted movement, as the agent will not deviate from its trajectory nor change in speed from the observed data used by the pedestrian model. Therefore, only a small amount of prediction is required, $T = 2$, effectively emulating local collision avoidance, Ta-



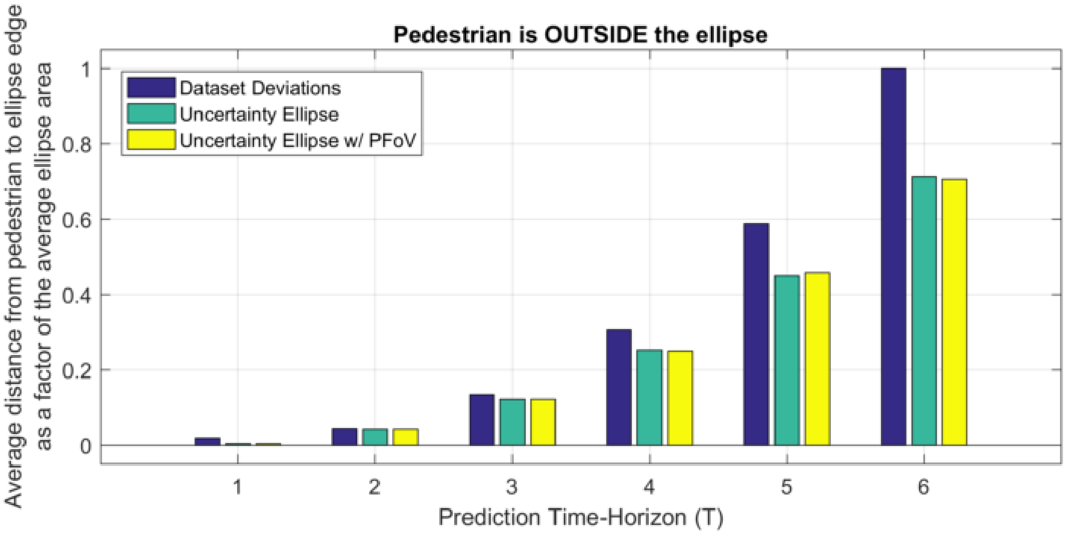
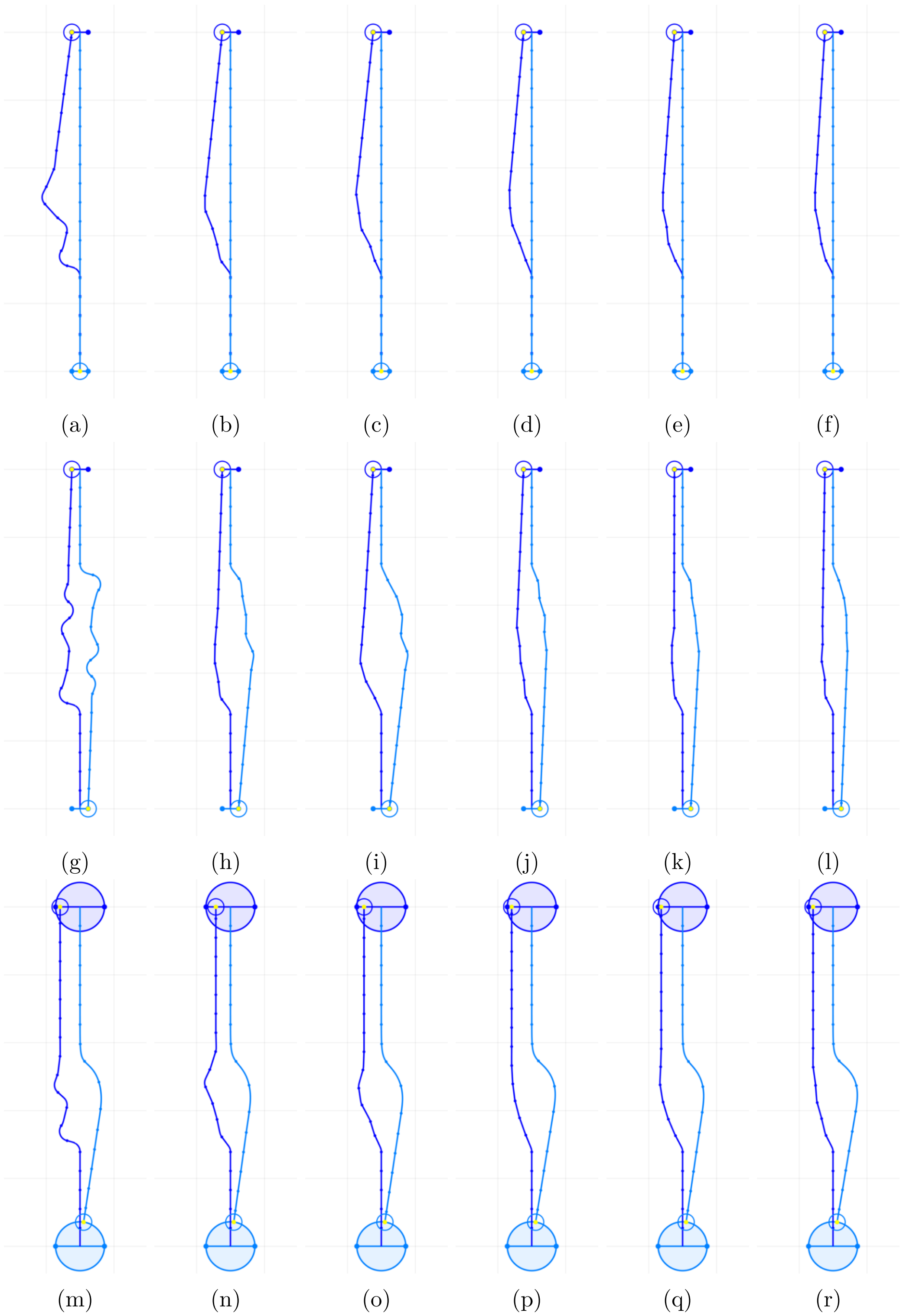


Fig. 6.4 Validation of the success of the uncertainty ellipse of the pedestrian model, Section 3.2, at containing the prediction of a dynamic agent's movement. The PedSim simulator is used to have dynamic agents cross each others paths (using all the scenarios designed to replicate empirical studies, Fig. 6.3, simultaneously). The accuracy of the pedestrian model is evaluated by calculating the average distance the agent is from the ellipse edge, using the 2 or 4 normal intersections to the ellipse from the agent, Eq. (3.4). The results are compared to an ellipse generated from the statistical deviations found in the BIWI pedestrian dataset, Fig. 3.3. The uncertainty ellipse clearly contains the agents more accurately (top graph), as the uncertainty profile limits the increases of ellipse sizes. If the agent does move outside the ellipse (bottom graph), again the uncertainty ellipse is superior. This is due its design to cover random stopping, and trajectory deviations.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

(m) (n) (o) (p) (q) (r)

Fig. 6.5 A basic example of the CNS demonstrating consideration and implicit cooperation. The CNS moves an AMR upwards, whilst various forms of dynamic agents move downwards, forcing the CNS to negotiate past a head-on collision. Each scenario is repeated consecutively for $T = 1, 2, 3, 4, 5, 6$, with path efficiency plateauing at $T = 4$. (a-f) A non-reactive agent moves down. The CNS is forced to instigate all collision avoidance and demonstrates considerate behaviour. (g-l) A similar AMR using the same CNS moves down. As expected both AMRs respond in similar ways, demonstrating implicit cooperation though mutual consideration. (m-r) A reactive agent from the PedSim simulator moves down. Both respond to the upcoming collision, demonstrating implicit cooperation without the requirement of "optimistically" assuming cooperation, Section 2.3.3.2.

ble 6.1. As the agent is non-reactive, the CNS must perform all collision avoidance, and so demonstrates considerate behaviour by not requiring a *pedestrian* to move out of the way.

When two AMRs using the CNS move past one another, Fig. 6.5g to Fig. 6.5l, they are both successful at passing considerately when $T = 2$. However, a larger prediction is required to make movement smoother, plateauing at $T = 4$, due to less easily predicted movement. As both agents are programmed with the same CNS, their movements at each timestep are very similar due to the CPP planning nearly the same paths for each. This experiment demonstrated that implicit cooperation can mutually occur when each agent behaves considerately.

When the CNS responds to a simulated pedestrian, Fig. 6.5m to Fig. 6.5r, similar results are also produced. The pedestrian responds identically each time, as the same social force is used for each, $SF = 1$. Whilst the AMR's path becomes smoother as the CNS increases the prediction time horizon, plateauing again at $T = 4$. As both agents are able to respond to the upcoming collision, this experiment demonstrates implicit cooperation can occur without the requirement of "optimistically" assuming cooperation, Section 2.3.3.2, requiring only consideration.

To create the smoothest and most efficient path, the CPP must plan paths with a prediction time horizon of $T \geq 4$. As the time-step used by the CNS is $0.4s$, this would be a $1.6s$ prediction window for $T = 4$ and $2.4s$ for $T = 6$. This compares to the evidence observed through empirical studies, which shows that pedestrians begin to respond to a collision between from $3s$ before [195]. The move lasts for $1.4s$, so the CNS should react within a similar time-window as normal pedestrians. This is a simple experiment with only one agent to avoid, a prediction time-horizon of $T > 4$ would likely be required if navigating amongst multiple dynamic agents. This hypothesis will be evaluated in the proceeding section, Section 6.4.

6.3.2.2 Path Selection

As discussed in Section 5.4, the number of potential paths is determined by the length of the prediction time-horizon. The larger T becomes the greater the upper limit of the potential paths. This adds to path diversity and contributes to exploring the environment more effectively. As just discussed, a minimum of $T = 4$ is required for smooth path planning. To provide a further example, using a non-reactive agent when $T = 1$, Fig. 6.6, the CPP performs a local search, Table 6.1, of the immediate environment. Consequently the CNS causes the AMR to behave as a reactive agent, by performing collision avoidance only in response to an immediate collision. As seen in Fig. 6.6d, this occurs too late and it results in a collision. At every attempt made by the CNS to pass the agent the movement is neither smooth, nor considerate.

When $T = 5$, Figs. 6.7a to 6.7d , the CPP performs a global search, Table 6.1, of the majority of the AMR's FoV. In comparison to $T = 1$, Fig. 6.6, the paths are adjusted to avoid collisions well in advance, and the overall paths are much smoother. Comparing Fig. 6.7c to Fig. 6.6b, the larger prediction time-horizon shows that deviating to move behind the agent well before it passes, results in no collisions along a minimally altered path. It is the path diversity used when exploring the environment, Section 5.4, that allows this to be successful. Comparing it to when the CPP uses a "best-first" path selection at each t, Fig. 6.7e, the CNS replicates a similar path as $T = 1$, albeit a lot smoother. An individual P_t path at each timestep may be better at t, but the consequent ones much worse. The most desirable P_T path the length of T, Section 5.2.2, may be slightly worse at an early stage, but overall much more suitable.

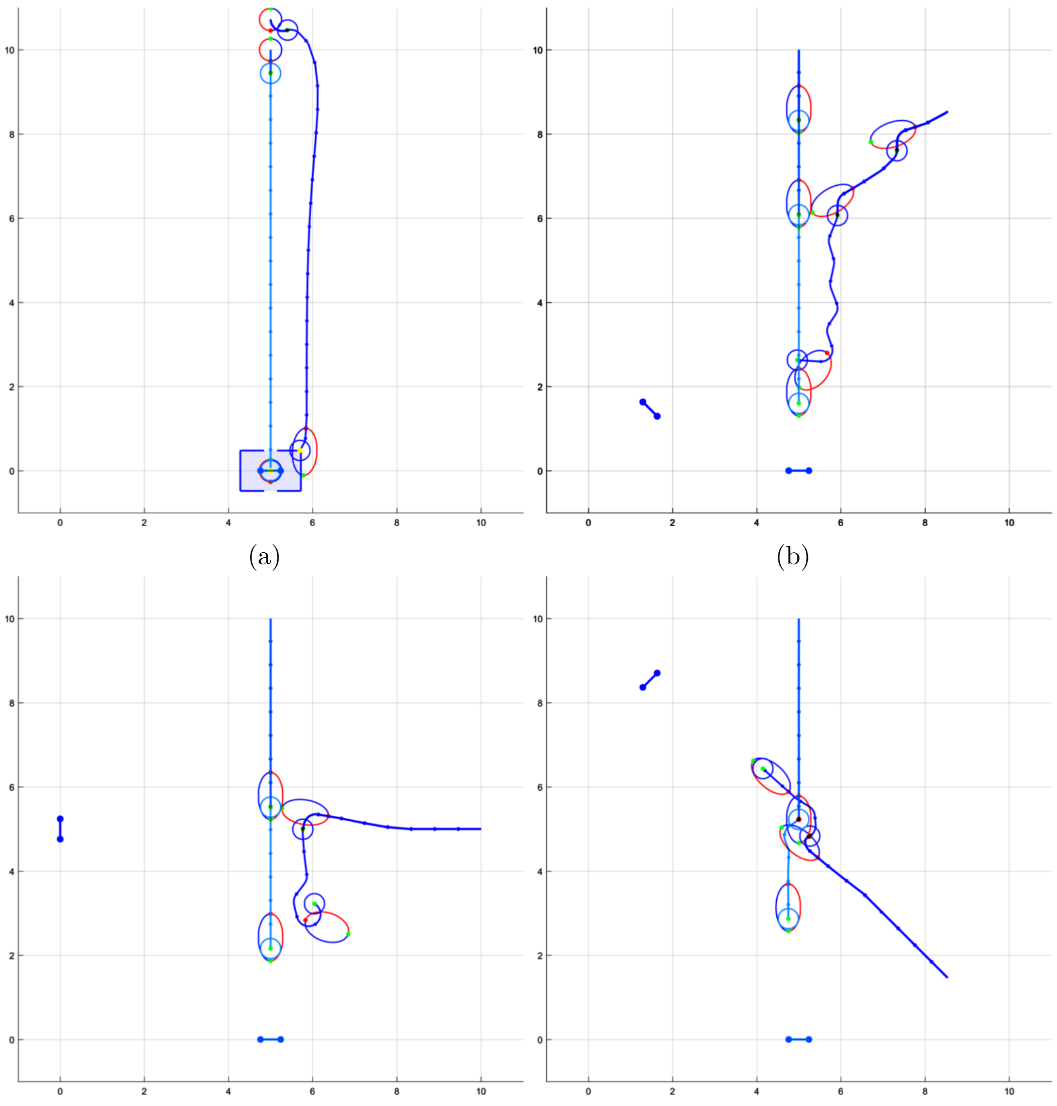
6.3.2.3 The Voronoi Diagram-Visibility Graph Roadmap

A global path that explores the environment is able to outperform a simpler "best-first" approach. However, the directions the search can move in is dictated by the roadmap, and the interconnectivity of its nodes. The previous examples all use the full VD-VG roadmap, which combines the VD and VG roadmaps to create an overall superior roadmap.

When using just the VD, Figs. 6.8a to 6.8d the paths become much more irregular. This is due to the vertex points of the VD, Fig. 6.8e, causing the CPP to select a much more limited number of paths that move either side of the agent. As seen in Fig. 6.8a, the path begins by snaking, due to the CPP selecting a path on the opposite side of the agent's uncertainty ellipse at each timestep. This limited number of P_t paths available at each timestep, which each diverge largely from each other, allows *safer* paths to be chosen by moving away from the agent, but are much less consistent.

When using just the VG, Figs. 6.9a to 6.9a, the paths become more similar to the VD-VG, Fig. 6.7. This is due to the greater number of interconnected vectors, Fig. 6.9e, making it much more likely to select a path along the VG. This is expected when moving though an open space, as the *saftey* of the VD is not required, rather the path efficiency of the VG preferable. When the AMR crosses the agent at a 90^o angle, Fig. 6.9c, the VG alone begins to perform worse. The VG does not provide the diverging path potential of the VD, for the CPP to select paths that move around the uncertainty ellipse. This can be seen in the form of the slant at the beginning of Fig. 6.8c. However, the path begins to snake again later for that very reason, as mentioned earlier, at which point when using the full VD-VG it would have switched back to a VG path.

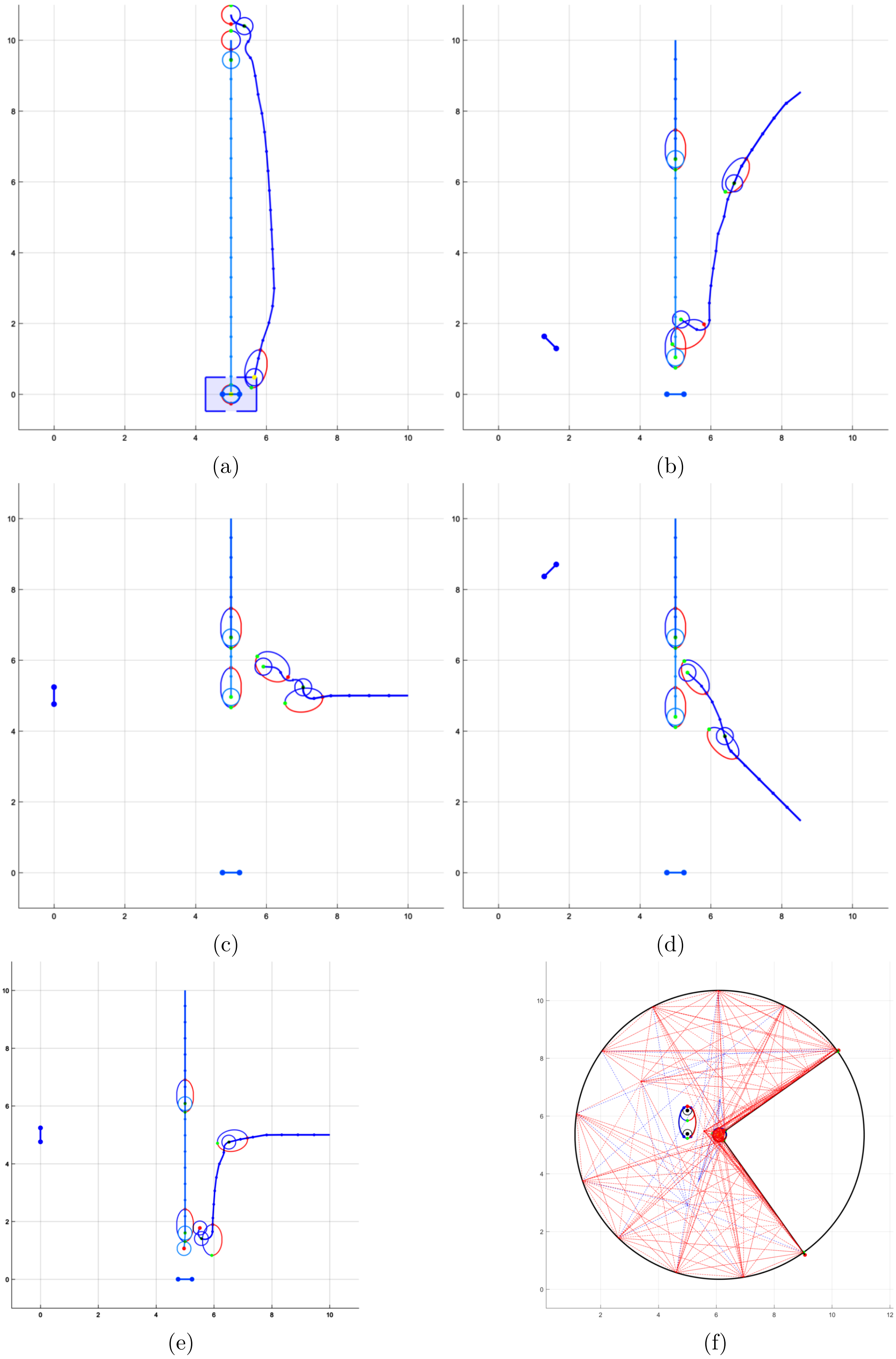
At the end of each time-step the diversity of the VD-VG roadmap means that many potential P_t paths for that time-step are available. By filtering the paths to avoid homotopic path repetition, Section 5.3, and to ensure the environment is explored, Section 5.4, the search effort is significantly reduced, Fig. 6.10a. The FMV,



(a) (b)

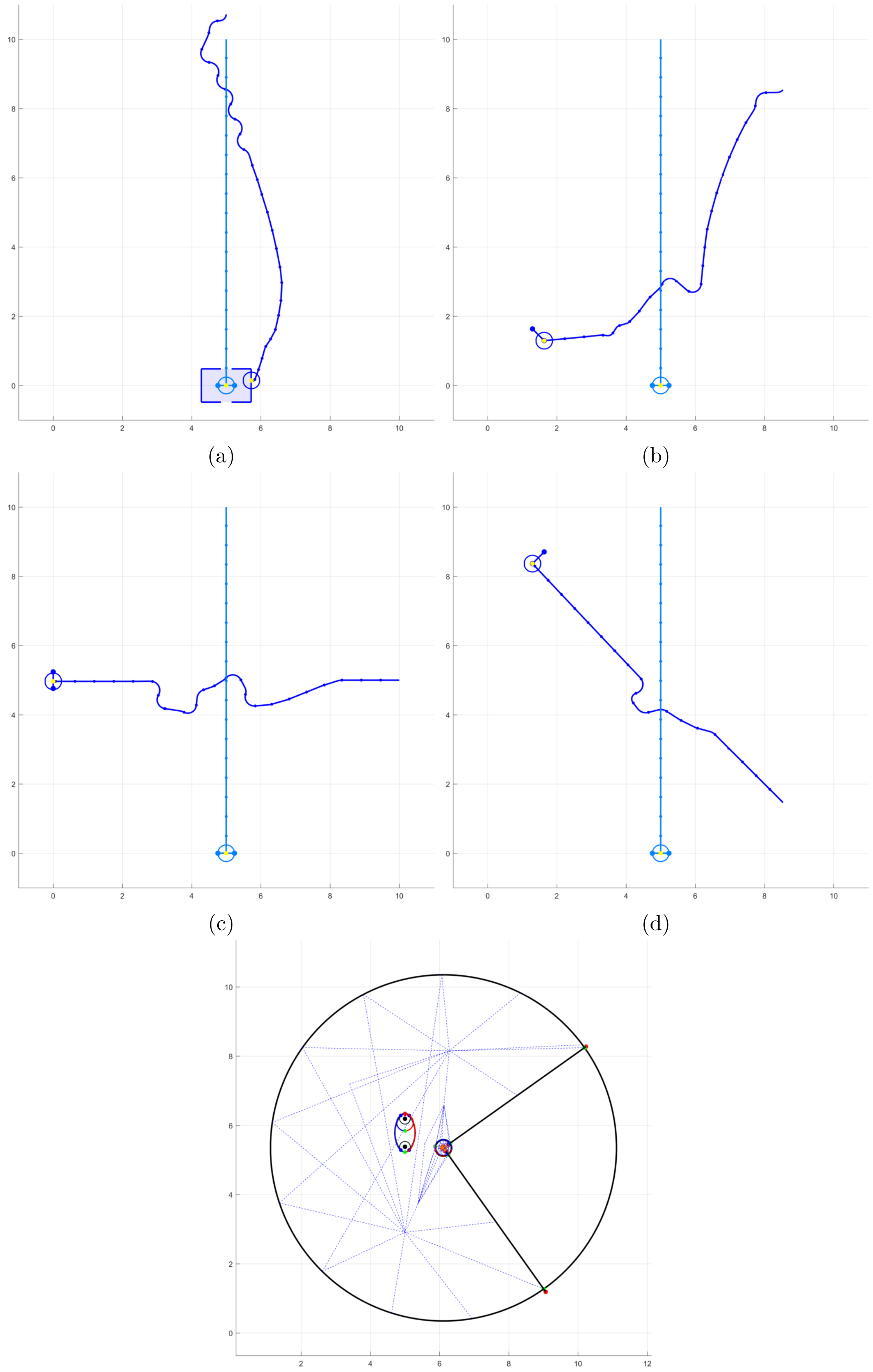
(c) (d)

Fig. 6.6 Examples of the paths taken by an AMR using the CNS, with a non-reactive agent moving down. The CPP uses the full VD-VG roadmap and a prediction time-horizon of $T = 1$. Both agents are shown with their respective uncertainty ellipses. (a) The AMR considerately moves with the pedestrians flow. (b) The AMR is constantly trying to move past the agent, before being moved away, creating the path's *wobble*. (c) The AMR moves with the pedestrian's flow, until it must turn back on itself in order to realign with its goal. (d) The AMR is unable to adjust its path soon enough, causing a collision with the agent. The agent is then forced to react, violating one of the main principles of the CNS.



(a) (b) (c) (d) (e) (f)

Fig. 6.7 Examples of the paths taken by an AMR using the CNS, with a non-reactive agent moving down. The CPP uses the full VD-VG roadmap and a prediction time-horizon of $T = 5$. Both agents are shown with their respective uncertainty ellipses. (a-d) The AMR moves considerately by moving behind each agent in all cases. (e) Using a "best-first" path selection, the CPP is unable to explore the environment sufficiently and move considerately behind the agent without colliding. (f) Example of the VD-VG roadmap used for (c).



(a)

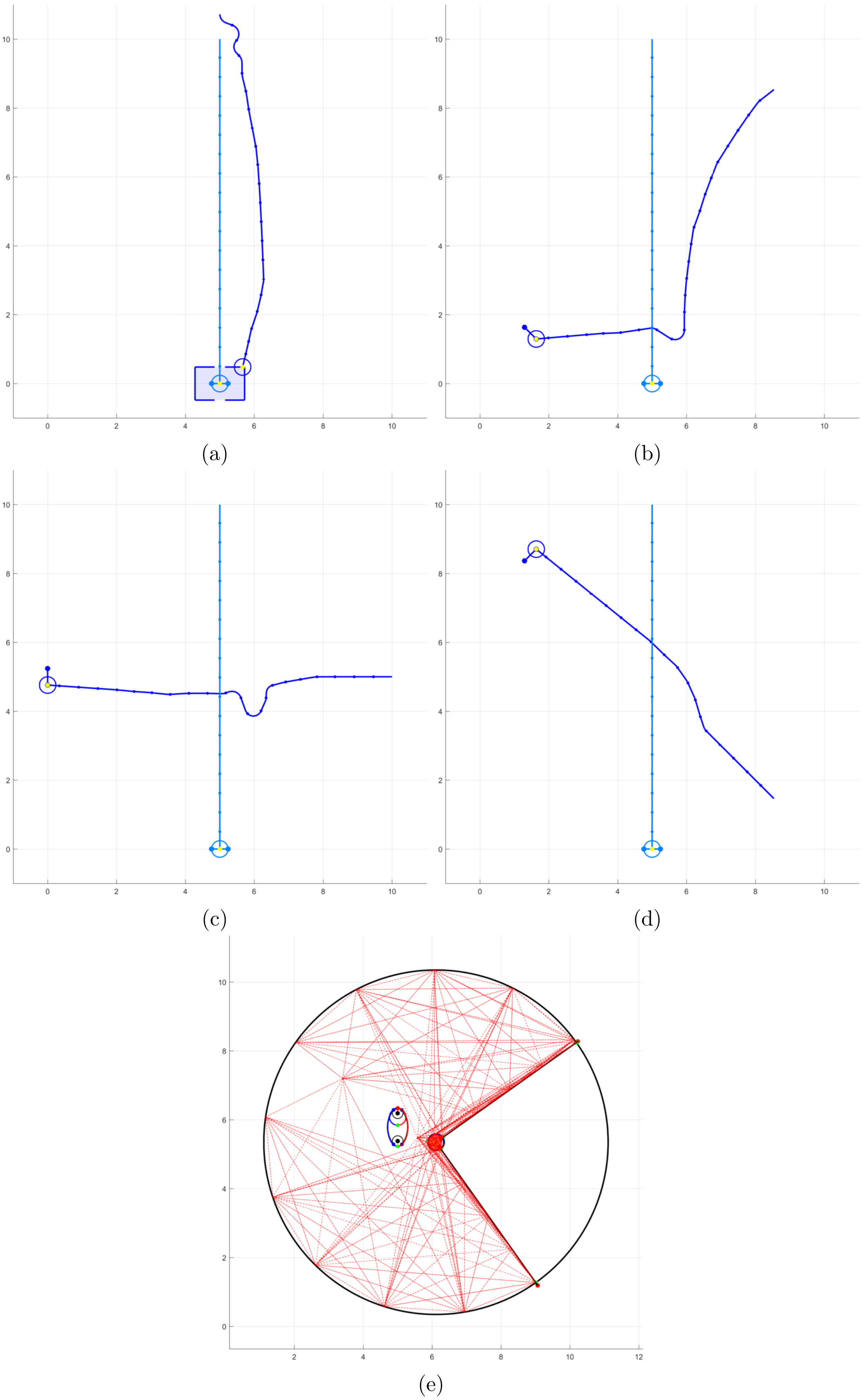
(b)

(c)

(d)

(e)

Fig. 6.8 Examples of the paths taken by an AMR using the CNS, with a non-reactive agent moving down. The CPP uses only the VD element of the VD-VG roadmap and a prediction time-horizon of $T = 5$. (a-d) The AMR moves along very inconsistent paths, due to the limited number of paths available by the VD. (e) Example of the VD roadmap used for (c).



(a)

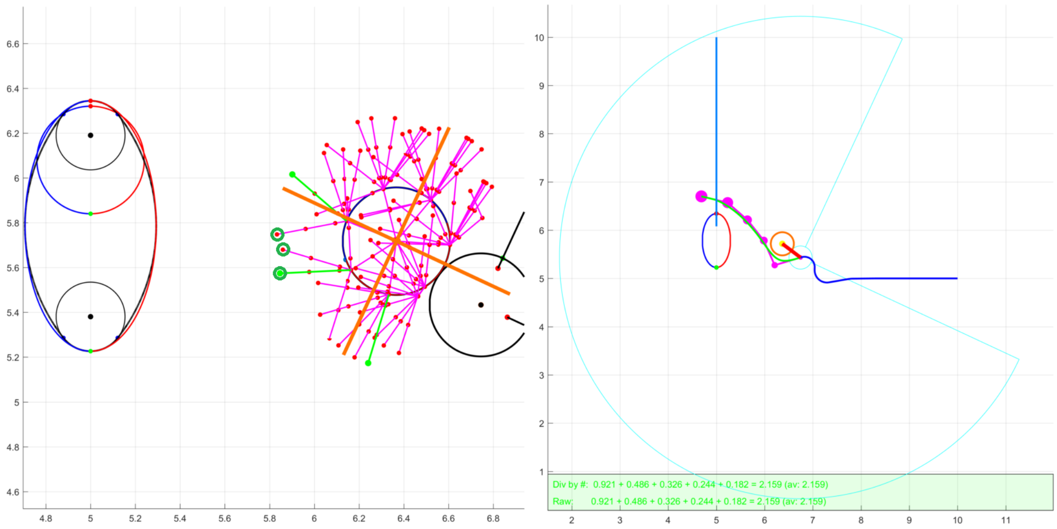
(b)

(c)

(d)

(e)

Fig. 6.9 Examples of the paths taken by an AMR using the CNS, with a non-reactive agent moving down. The CPP uses only the VG element of the VD-VG roadmap and a prediction time-horizon of $T = 5$. (a-d) The AMR moves much very smoothly, due to the large number of of paths available by the VG. (e) Example of the VG roadmap used for (c).



(a) (b)

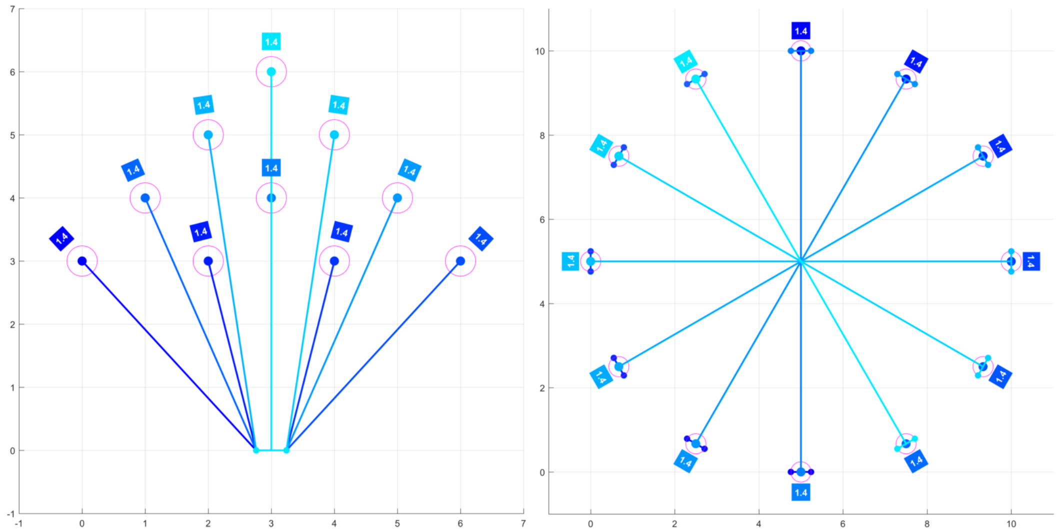
Fig. 6.10 An example of how avoiding homotopic path repetition, Section 5.3, and ensuring the environment is explored, Section 5.4, helps reduce the search effort and let the CPP plan more efficient paths. The experimental set-up is of Fig. 6.3c. (a) Frontier of the windowed Dijkstra's search at the end of $t = 1$ (red dots on magenta search paths). Out of the 64 frontier points, the 3 selected are highlighted green. Had the 3 circular quadrants, Fig. 5.11, around the search start (orange lines) not forced the P_t paths to diverge, the three chosen (highlighted with green circles) would all produce similar paths. (b) By selecting the FMV, Fig. 5.15, rather than the first P_t path segment of the P_T path, the AMR begins to move behind the agent early on.

Fig. 5.15, also enables the AMR to begin avoiding hypothetical collisions early on, and ensure the smooth path planning and movement, Fig. 6.10b.

6.4 Testing the Considerate Navigation Strategy in a Crowd

The empirical studies recreated in the previous section has allowed the design elements of the CNS to show its benefit, and similarities in recreating pedestrian collision avoidance strategy. In order to more thoroughly evaluate the CNS's effectiveness at navigating amongst multiple pedestrians, two scenarios have been devised, Fig. 6.11, that potentially require the avoidance of multiple hypothetical collision points between all dynamic agents simultaneously.

To establish the CNS's improvement of current alternative methods, each test will be ran in parallel with a traditional A* and a *Considerate* A* modified with a cost of the consideration resistance of the CPP, Section 3.3.4. The novel QPMs, Section 6.2.3, will provide consistent performance evaluations to establish how considerately the AMR navigates. To further demonstrate the requirement for prediction



(a) (b)

Fig. 6.11 Visualisation of the ideal robot-destination paths for each test scenario. (a) 10 robots converge on the same destination line. This will evaluate the CPPs ability to perform competitive collision avoidance over a small area.

of pedestrian movements and the accompanying path diversity of potential paths, each scenario will be evaluated over a prediction time-horizon of length $T = 1$ to 6, Fig. 6.12 and Table 6.1. For each scenario the PedSim maximum speed will be set at the average pedestrian walking speed of $1.4ms^{-1}$, and each scenario will be repeated twice, with the AMR's maximum speed set to the average and maximum pedestrian walking speed, $1.4ms^{-1}$ and $2.0ms^{-1}$, respectively. When using the PedSim simulator, each time the scenario is ran the position of each pedestrian in turn will be substituted with an AMR, so that the CNS can be trialled from all start positions within the crowd. The more traditional QPMs will also be used to evaluate the AMR's efficiency in completing its journeys.

6.4.1 A Considerate A* Path Planner for Comparison

The traditional A* path planner, Section 2.2.2.2, is common in many modern AMR navigation systems, including the field of HRI (e.g. [151]). The A* best-first heuristic is one of a Euclidean distance from the search's current position to the goal, which provides effective path selection of one that moves closer to the goal. However, the downfall of the A* is that the cost of the search's progresses is dictated only by the total distance moved from the start point to the search's current position. When the node network/occupancy grid, Section 2.2.1.1, is created the environmental obstacles provide a barrier that the search cannot penetrate. Therefore, although a clear path can be planned, if available, it only avoids any direct collisions with them.

The *Considerate* A* (CA*) search is a modified version of the A* search that

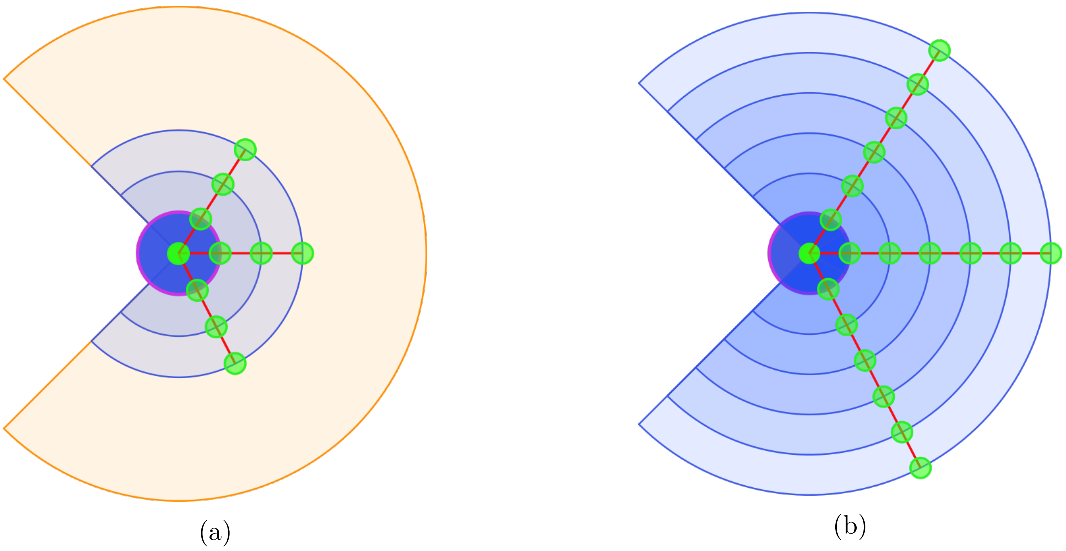
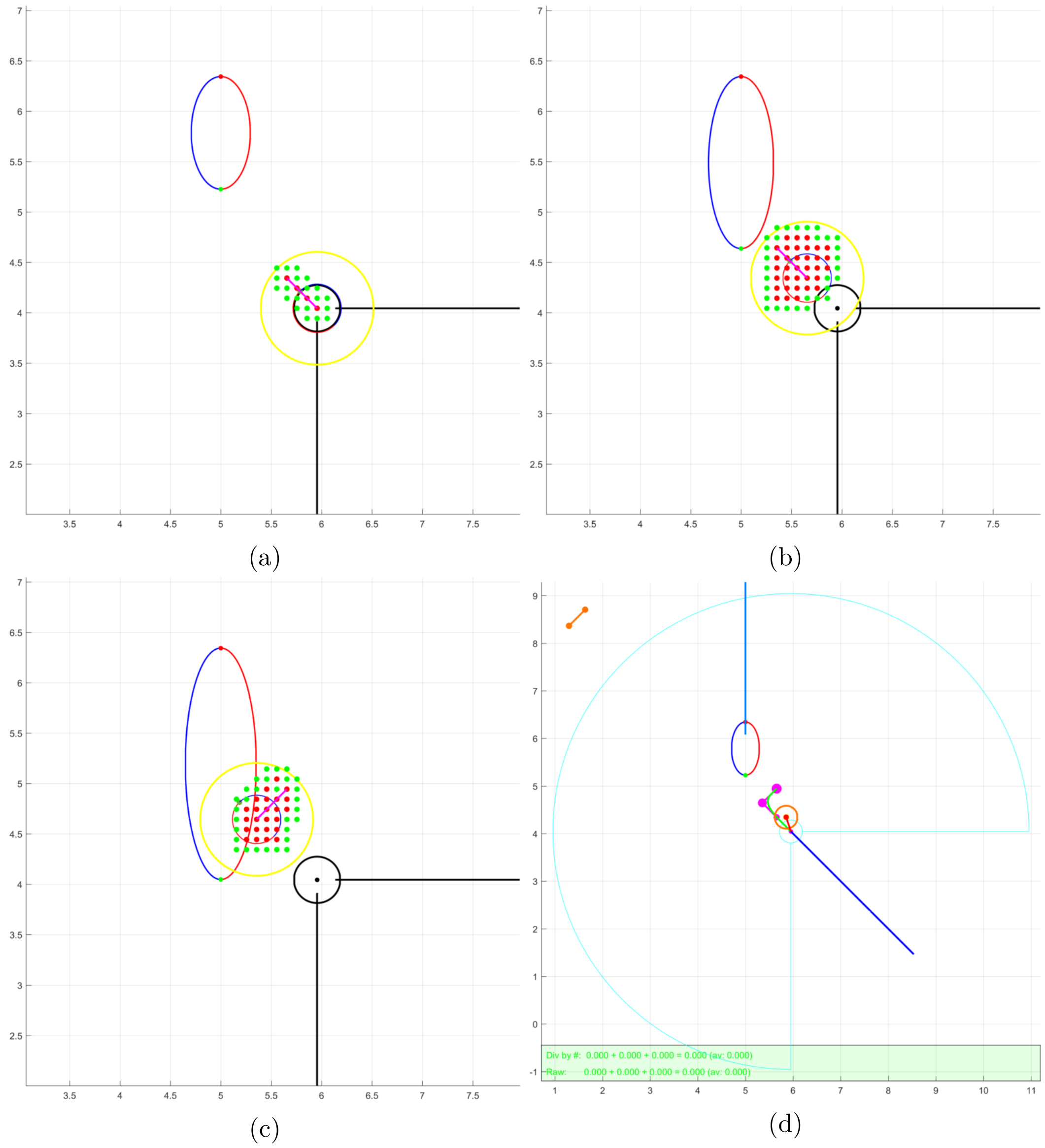


Fig. 6.12 Example of the maximum distance the CPP can predict paths into the AMR's orange FoV, dependent upon the prediction time-horizon used. Red lines indicate ideal straight paths outward from the AMR's initial position. Green dots give an example of the maximum distance the AMR could move for every time-step. (a) Medium prediction time-horizon, $T = 3$. (b) Maximum prediction time-horizon, $T = 6$.

uses the same *considerate weights*, Section 3.3.4, as used by the CPP. As the search explores the environment, the consideration weights are calculated as the search moves to the next node, in the same manner as the consideration resistance is calculated as it moves along the VD-VG of the CPP, Section 3.3.3. The total resistance calculated at each move is added to the total cost of the search point thus far, along with the total distance travelled thus far as with the standard A* search. As with the CPP, both alternative A* searches will move for either the distance possible within one time-step, or until the AMR's goal is reached. Once the best path is selected after each time-step, the pedestrian model, Section 3.2, will update the pedestrian's uncertainty ellipses, Fig. 6.13. The search will begin again for the consecutive time-step at the end of the previous time-step's search, as done so by the CPP.

6.4.2 All Agents Moving Towards the Same Goal

This scenario, Fig. 6.11a, analyses how much impact the AMR's have on contributing towards congestion, by setting all agents to converge on the same destination. When the AMR moves at $1.4ms^{-1}$, it will evaluate how well the AMR will follow the flow of the other agents to the destination. When the AMR moves at $2.0ms^{-1}$, it will evaluate whether the AMR will attempt to overtake other agents to reach its goal faster, or queue behind them and move with the flow of the crowd.



(a)

(b)

(c)

(d)

```
Div by #:  0.000 + 0.000 + 0.000 = 0.000 (av: 0.000)
Raw:       0.000 + 0.000 + 0.000 = 0.000 (av: 0.000)
```

Fig. 6.13 Example of the *considerate* A* search algorithm finding a path for three consecutive timesteps, using the empirical studies set-up shown in Fig. 6.3d. The yellow circle represents a distance possible to cover in one timestep, with close nodes shown in red and open nodes in green. Path selected at each timestep is shown in magenta. (a) P_t path selected for $t = 0$ to $t = 1$. (b) P_t path selected for $t = 1$ to $t = 2$. (c) P_t path selected for $t = 2$ to $t = 3$. (c) P_T path selected for $t = 0$ to $t = 3$. The AMR's final movement vector, Section 5.5.1.1, is highlighted red.

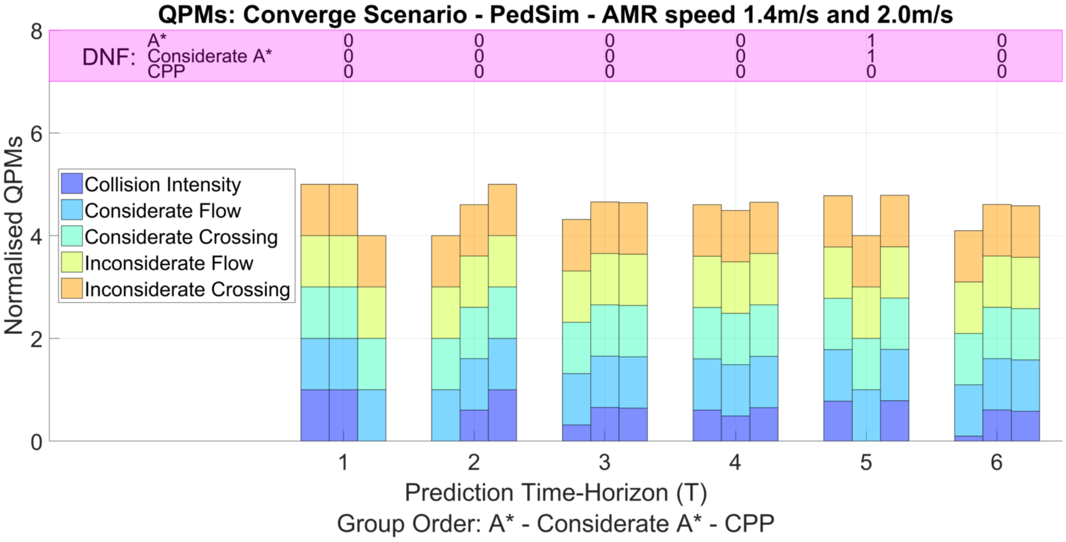
The triangle that forms between all agents, a, and their destinations has an area of $18m^2$. The crowd density, Table 6.2, of the scenario can be deemed as $\frac{10}{18} \approx 0.667am^{-2}$, which would be "medium". When observing the results, the agents move within a continually reducing area which clearly exceeds $0.955am^{-2}$, which becomes "very high", and is "not recommended" This scenario is able to specifically evaluate the CNS at how well the AMR moves with the crowd flow, and avoids adding as much additional congestion as possible.

PedSim Simulator

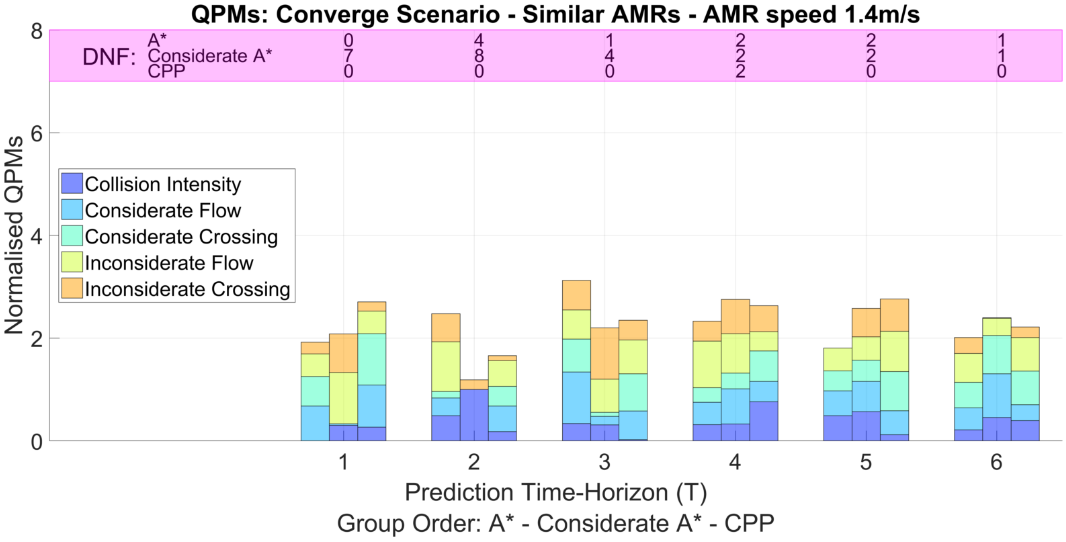
From an evaluation of the QPMs, Fig. 6.14a the CPP has an optimal prediction time-horizon of $T = 2$. When moving with the crowd a smaller T allows the CNS to exploit the gaps in between pedestrians before the predictions overcrowd the environment, which makes collision free paths less likely. This is why the best performance is from the A* and CA* searches whilst $T = 1$, as these planners plot direct paths to the goal, along with the pedestrians, Figs. 6.15a to 6.15f. However, aside from this the CPP outperforms the A* search for the other 6 prediction lengths, $T = 2 - 6$. The CA* also outperforms the A* search for $T = 2, 3, 6$, and marginally outperforms the CPP for $T = 3$ and 6 as well. This is due to the "best-first" heuristic pushing the search into the crowd, rather than exploring around the environment in other directions as well, like the CPP. This experiment shows that search algorithms that utilise consideration are more successful at navigating along with crowd flow than standard search algorithms that prioritize only the distance to the goal.

Similar AMRs

Due to the fact that each AMR moves within the environment for the entire timestep based on the planned FMV, more inconsiderate behaviour is expected. This is because if predictions of pedestrian movement is incorrect (as is possible in crowded environments, Fig. 6.4), the AMR may be set on a collision course with another AMR. Due to this the CPP was not expected to perform consistently until $T \geq 4$, which is when the smoothest paths are formed, as seen in Fig. 6.5, due to an adequate length of prediction. From an evaluation of the QPMs, Fig. 6.14b, the most preferable prediction time-horizon is the longer $T = 5$. When $T \geq 4$ both considerate searches outperform the A*, although the CA* outperforms the CPP in 2 of the 3 T values. This is due to the same reason, when the PedSim experiment was performed.

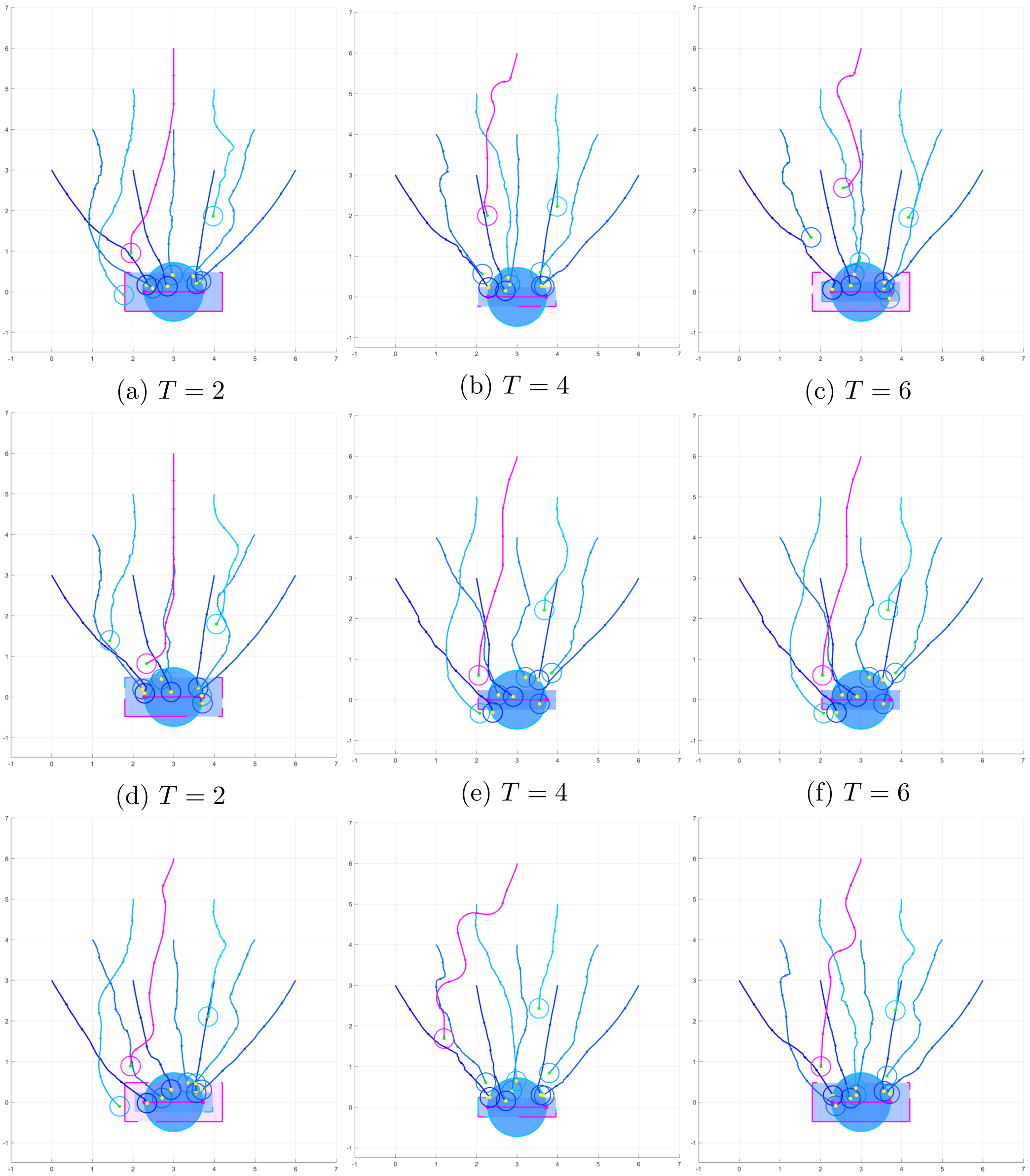


(a)



(b)

Fig. 6.14 The "Converge" scenario, Fig. 6.11a. (a) The CPP has an optimal prediction time-horizon of $T = 2$. The A* search is outperformed by at least one considerate search (4 CA* and 5 CPP) for each prediction length, T. (b) The CPP has an optimal prediction time-horizon of $T = 5$. The A* search is outperformed by 4 considerate searches (4 CA* and 4 CPP) for each prediction length, T.



(a) $T = 2$ (b) $T = 4$ (c) $T = 6$

(d) $T = 2$ (e) $T = 4$ (f) $T = 6$

(g) $T = 2$ (h) $T = 4$ (i) $T = 6$

Fig. 6.15 Random sample of the CNS navigating an AMR (magenta line) through a a simulated pedestrian crowd, using the PedSim simulator. The scenario displayed is of the multiple pedestrians converging on the same goal, Fig. 6.11a, with the AMR set at a maximum of $2.0ms^{-1}$, in order to analyse how it moves through the crowd. Each image is of the 10^{th} AMR moving diagonally-upwards from left to right at $t = 8$. (a-c) The A* search algorithm: This is "best-first" search the AMR attempts to move through the crowd whilst also moving faster than the surrounding pedestrians. It ends up readjusting its path more as T increases, due to the blocked path ahead. (d-f) The *Considerate* A* search algorithm: There is very little change to the AMR's movements as T increases. This is due to the *consideration weights* causing the search to plan paths that move with the crowd flow, regardless of the prediction length. The AMR simply slows down and follows the crowd. (g-i) The CPP algorithm: As the CPP explores more of the surrounding environment, the AMR moves to circumnavigate the crowd as T increases, rather than pushing past the pedestrians.

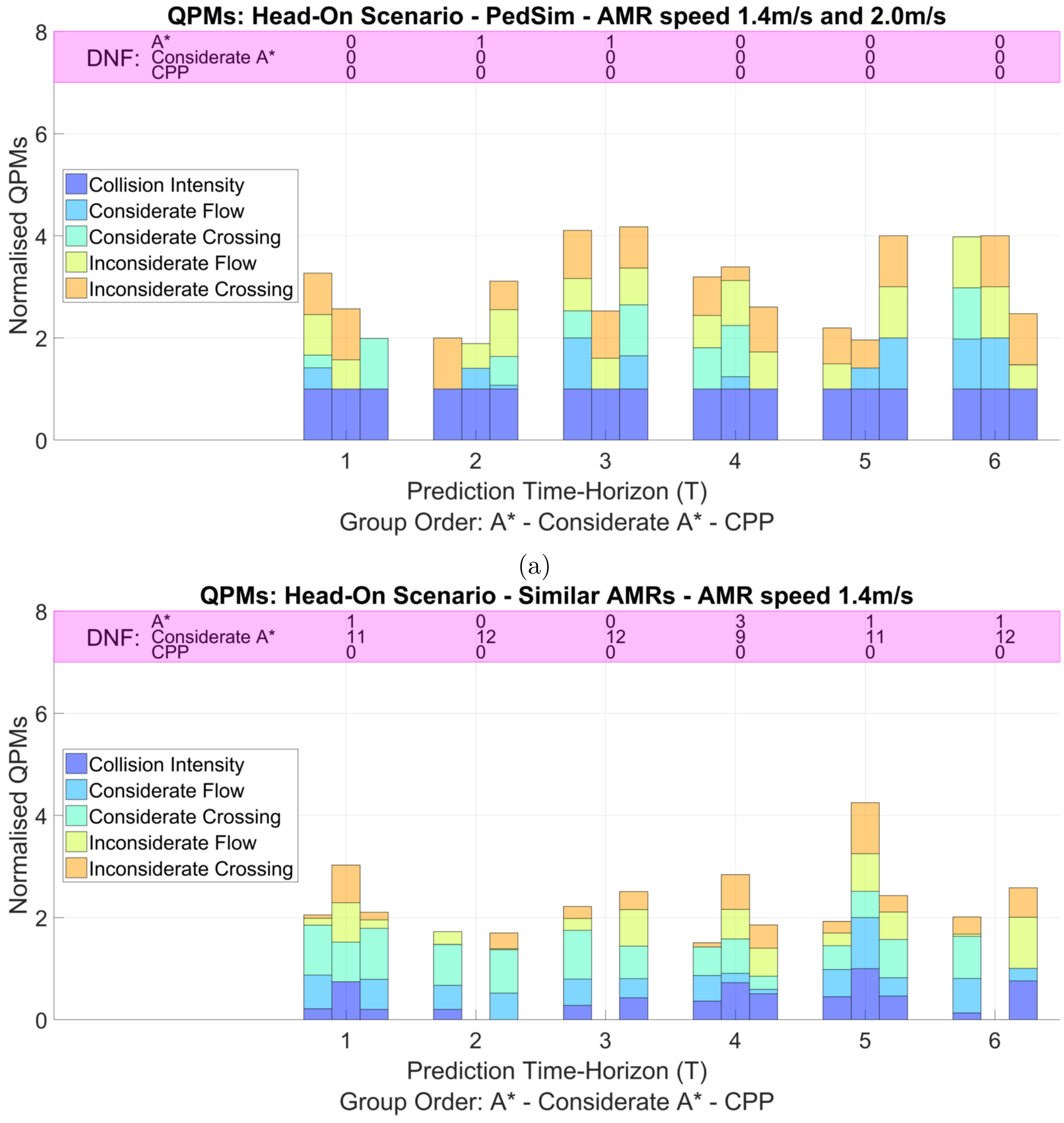
6.4.3 A Shared Hypothetical Collision Point for All Agents

This scenario has been designed to evaluate the potential for the CNS to circumnavigate an area of a high number of potential head-on collisions, Fig. 6.11b. The scenario is an expansion of the empirical study, Fig. 6.3, by increasing the number of crossing angles, whilst all paths are executed simultaneously. The dynamic agents must pass directly past each other, which without any path planning or collision avoidance would see each agent simultaneously collide with one another. The AMR will move across the environment from each position in turn, and travel at the average and maximum pedestrian walking speed, $1.4ms^{-1}$ and $2.0ms^{-1}$, respectively.

The convex polygon (dodecagon) that forms between all agents, a, and their destinations has an area of $75m^2$. The crowd density, Table 6.2, of the scenario can be deemed as $\frac{12}{75} = 0.16am^{-2}$, which would be "low". When observing the results, the agents move around the central point in a circular fashion with a radius of approximately $2m$, which would have an area of approximately $\pi 2^2 \approx 12.6m^2$. The crowd density within this area would be approximately $\frac{12}{12.6} \approx 0.955am^{-2}$, which is "high". This scenario is therefore able specifically evaluate how well the CNS avoids high density areas and avoid crossing other pedestrians' paths, as well as all other objectives, Section 1.3.2.

PedSim Simulator

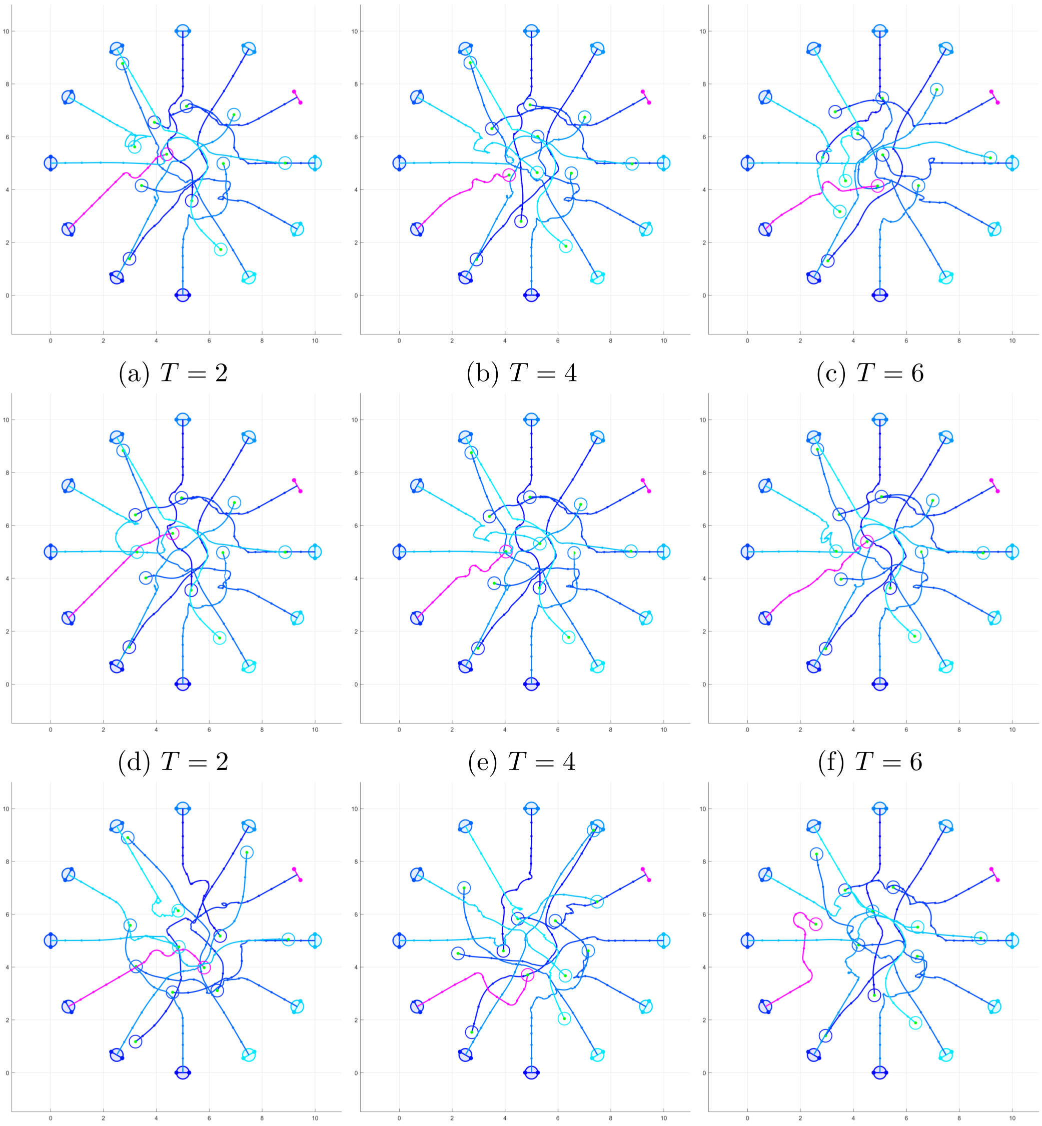
From an evaluation of the QPMs, Fig. 6.16a, the CPP has a marginally optimal prediction time-horizon of $T = 3$. However, the CPP significantly outperforms the other searches when $T = 5$. This scenario was expected to perform with less consistent results, as the PedSim pedestrians are not limited in their turning. As seen in all the visual examples, Fig. 6.17, the pedestrians movement is often stuttered due to the social force model moving them using Newtonian mechanics so they behave similar to particles. Therefore, when the pedestrian suddenly changes direction, the proxemics field used to calculate the considerate QPMs, Fig. 6.2, may register an inconsiderate movement that is not the fault of the CNS, Fig. 6.18. However, both the considerate searches have outperformed the A* search, with the CPP outperforming the A* an additional time. A look at the visual examples, Figs. 6.17g to 6.17i, shows qualitatively the intent of the CPP attempting to manoeuvre round the crowded centre of the environment, much clearer than the alternative searches.



(a)

(b)

Fig. 6.16 For the "Head-On" scenario, Fig. 6.11b. (a) The CPP has an optimal prediction time-horizon of $T = 3$. The A* search only outperforms both considerate searches when $T = 1$, and is outperforms by at least one considerate search (2 CA* and 4 CPP) for each prediction length if $T > 1$. (b) The CPP has an optimal prediction time-horizon of $T = 6$. The CPP outperformed the A* in all prediction lengths, aside form being marginally beaten at $T = 2$. For $T = 2$, 3, and 6 the CA* did not finish any journeys, as well as the majority of journeys for the other T values. Although, for the few successful journeys it outperforms all others.

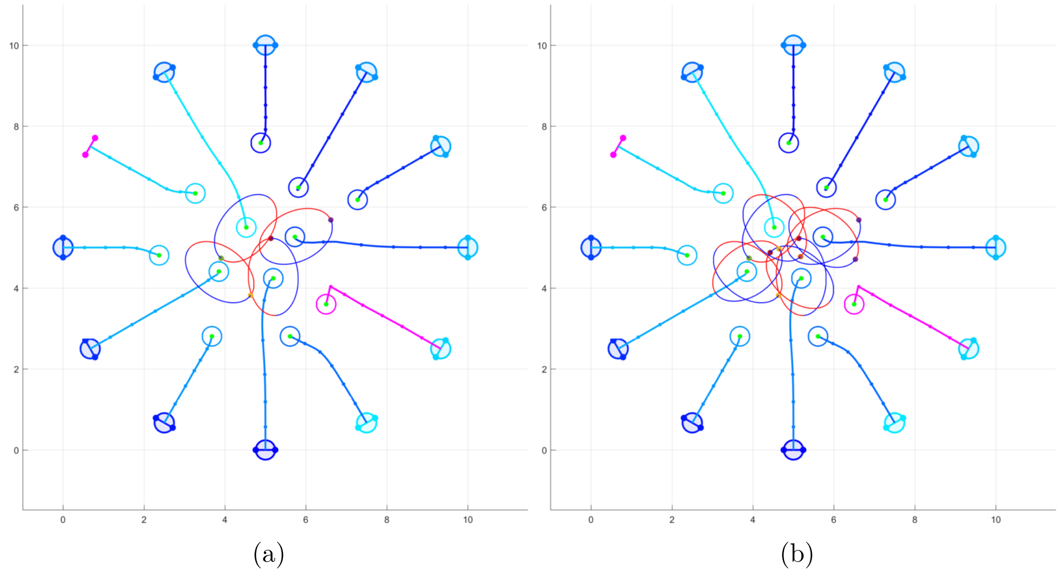


(a) $T = 2$ (b) $T = 4$ (c) $T = 6$

(d) $T = 2$ (e) $T = 4$ (f) $T = 6$

(g) $T = 2$ (h) $T = 4$ (i) $T = 6$

Fig. 6.17 Random sample of the CNS navigating an AMR (magenta line) through a a simulated pedestrian crowd, using the PedSim simulator. The scenario displayed is of the multiple head-on hypothetical collisions, Fig. 6.11b, with the AMR set at a maximum of $1.4ms^{-1}$. Each image is of the 9^{th} AMR moving diagonally-upwards from left to right at $t = 12$. (a-c) The A* search algorithm: There is very little difference as T increases, moving almost straight through the centre. The pedestrians move out of the way of the AMR more often than the AMR moves out of the way of the pedestrians. (d-f) The *Considerate* A* search algorithm: Although the path moves almost straight through the centre as T increases, the slight adjustments its makes do not significantly affect the pedestrians. (g-i) The CPP algorithm: As T increases the AMR becomes much more evasive as it attempts to find a suitable gap to move through. Beyond $T = 4$, it appears that an overprediction of where the pedestrians may be results in the CPP being unable to plan an efficient path.



(a)

(b)

Fig. 6.18 Example of the PedSim simulator changing the direction of 4 pedestrians in the centre of the environment during for the "Head-On" scenario, Fig. 6.11b. The AMR is highlighted magenta, the pedestrians in blue, and the positions of each are updated 25 times during each time-step, which results in an update every $\frac{0.4s}{25} = 0.016s$ $(62.5Hz)$. Because of the particle-like nature of the social force model, the pedestrians can change their direction instantaneously and without turning limitations. This can easily cause the proxemics ellipses that surround each pedestrian, Fig. 6.2, to be moved over the AMR's position. This may artificially inflate any inconsiderate movement metrics being tabulated by effectively having the pedestrian behave inconsiderately to the AMR. (a) The initial proxemics ellipses for each of the 4 pedestrians. (b) The new location of the proxemics ellipses have rotated almost 90^o for 3 of the pedestrians, in only $0.016s$.

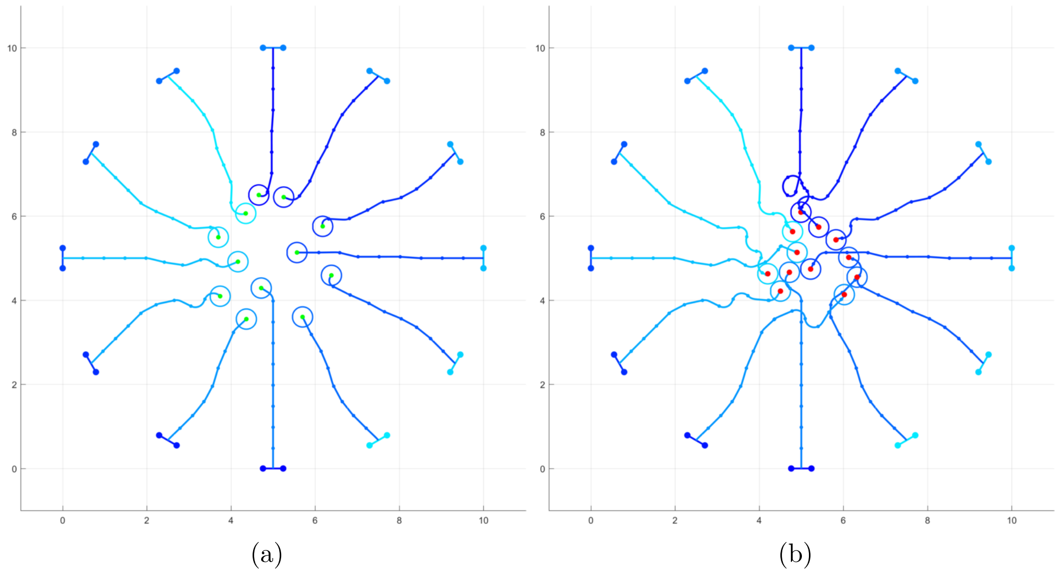


Fig. 6.19 Example of how the maximum turning angle of the differential drive model, Fig. 5.18, can prevent the AMR's from moving. (a) All AMRs begin to show the initial signs of self-organisation in: 7/4 AMRs begin turning right/left in order to avoid the centre of the environment, whilst 1 exploits the now free space in the centre (far right). (b) A few timesteps later the AMRs become stuck as they do not have enough space to turn.

Similar AMRs

From an evaluation of the QPMs, Fig. 6.16b, the CPP has a marginally optimal prediction time-horizon of $T = 6$. This is due to the AMR avoiding the congested central area of the environment (the AMR can also be seen avoiding the centre when navigating amongst simulated pedestrians, Fig. 6.17i). The CPP outperformed the A* search for all prediction lengths, except for the marginal loss at $T = 2$. When navigating an environment with so many hypothetical collision points concentrated in one area (the environment centre, Fig. 6.11b), simply evaluating a path based on distance to the goal (A*) is inefficient. As the paths planned by the CPP factored in both the position of each pedestrian and the direction they are moving, many collisions are avoided. Unfortunately, the CA* failed to complete most of its journeys. However, this is may only be due to the limitations of the differential drive model each AMR operates with, Fig. 5.18, as they would become stuck between one another due to a lack of available turning space, Fig. 6.19.

6.4.4 The Optimal Prediction Time-Horizon

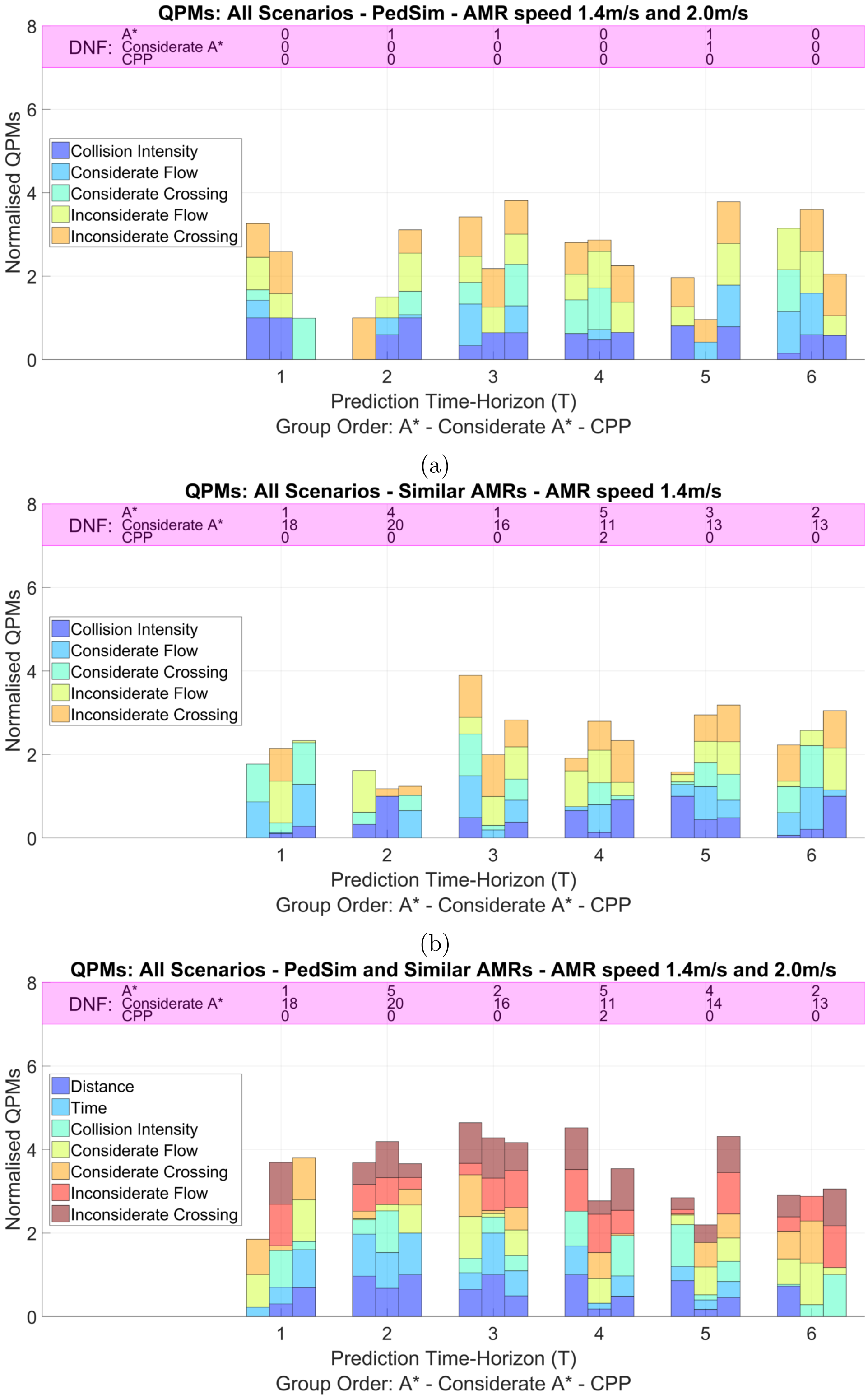
After evaluating all of the test scenarios it can be seen that when a search algorithm uses considerate weights, Section 3.3.4, more often than not it will be able to plan paths that do not negatively affect the movement of other dynamic agents. The CA* was able to plan more considerate paths than the standard A*, showing the simple rules dictating how consideration is calculated can improve an already well established search algorithm.

From an evaluation of the QPMs for all scenarios, Figs. 6.20a and 6.20b, the most preferable prediction time-horizon for the CPP is $T = 5$, when moving amongst both simulated pedestrians and similar AMRs. As stated in Section 6.3.2.1 it was anticipated that for efficient path planning the prediction length must be $T \geq 4$ when navigating amongst larger numbers of dynamic agents. When evaluating all scenarios using all QPMs, Fig. 6.20c, as stated in Section 6.2, the A* becomes more successful due to being able to often reach its destination more directly. However, aside from when the A* does perform better than the considerate searches in $T = 4$, it is only marginal. When it is outperformed by the CPP for $T = 1$ and $T = 5$ its performance is significantly worse. In overall behaviour, the considerate searches outperform the A* algorithm, and more often than not outperform it when being evaluated against all metrics.

6.5 Chapter Summary

This chapter has successfully established that the CNS is capable of qualitatively planning considerate global paths through a model of the real-world environment, be recreating empirical human-human collision avoidance experiments. An AMR is then able to move considerately in the real-world environment, by evaluating its movements using a set of novel QPMs that were designed specifically to evaluate considerate behaviour.

When analysing the QPMs for each test scenario it is clear that for the CNS to move an AMR considerately a global prediction length of $T = 5$ must be used. When observing the movements of an AMR, the CNS is able to implicitly cooperate with other dynamic agents that do not necessarily operate with the same navigation strategy. When using the considerate weights to modify the path selection of an A* algorithm, its performance is also improved. Therefore, the CNS is not only diverse because it does not rely on knowing the specific behaviours of other agents, it can act as an improvement of pre-existing search algorithms. The next chapter will evaluate the CNS using real-life pedestrian data.



(a)

(b)

(c)

Fig. 6.20 Combination of all QPMs for all scenarios and all dynamic agents. (a) The CPP has an optimal prediction time-horizon of $T = 5$. The A* search only outperforms both considerate searches when $T = 1$, and is outperforms by at least one considerate search (3 CA* and 3 CPP) for each prediction length if $T > 1$. (b) The CPP has an optimal prediction time-horizon of $T = 5$. Both considerate searches outperformed the A* for 4 prediction lengths, being beaten in $T = 2$ and 3. (c) The CPP has an optimal prediction time-horizon of $T = 5$. Both considerate searches outperformed the A* for 4 prediction lengths, being beaten in $T = 3$ and 4.

Chapter 7

Evaluating the Considerate Navigation Strategy with Real Pedestrian Data

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

— Alan Turning

THE considerate path planner (CPP) has successfully demonstrated its ability to plan considerate paths, and the considerate navigation strategy (CNS) has successfully shown its ability to execute them in an approximation of reality. However, this chapter will evaluate the CNS' ability to respond to real-life pedestrian data.

Using a dataset of real-life pedestrian movements [14], suitable configurations of naturally moving and free-flowing pedestrians will be obtained, Section 7.1. Each configuration will contain pedestrians moving in the same direction as one another, as well as in alternating directions. The ability for the CNS to direct an autonomous mobile robot (AMR) through each sample of the dataset, at various orientations, will be evaluated, Section 7.2, using the same novel quantitative performance metrics (QMPs) devised in Section 6.2.3. Finally, a qualitative examples will be given of the AMR moving within the crowd. These will reflect the results of the QPM evaluation, demonstrating that the CNS can successfully navigate amongst pedestrians with considerate motion.

7.1 Finding Suitable Pedestrians within the Dataset

To evaluate the performance of the CNS, appropriate pedestrian data must be selected from the dataset, initially presented in Section 3.2.1. The CPP will be made to plan paths that travel in parallel and perpendicular to the pedestrians' movements, up/down and left/right, respectively. This is to observe the thesis' objectives, Section 1.3.2, for the CPP to "plan paths that move in the same pedestrian direction" (Item $b_{2.2}$), and "plan paths that avoid crossing another pedestrians path" (Item $b_{2.3}$).

7.1.1 Suitable Crowd Scenarios

Crowd scenarios must be chosen that challenge the CNS to behave considerately, evoking the same direct (active) and in-direct (passive) qualitative considerate behaviour established in Section 6.2. The variations in the bi-directional traffic in the dataset will allow these behaviours to be challenged:

a_1 **Uni-directional Traffic:** more than 2/3 of the crowd population is moving in the same direction. This will evaluate if the CPP is capable of predicting paths that allow the robot to move with the flow of the crowd. Pedestrians moving in the same direction should greatly reduce the number of conflicting trajectories, resulting in the following behaviour:

  $a_{1.1}$ A lower collision likelihood

  $a_{1.2}$ A more ordered crowd, as pedestrians should flow together

  $a_{1.3}$ More predictable movements, more likely to follow the CVM

a_2 **Uni-directional Traffic:** more than 2/3 of the crowd population is moving in opposing parallel directions. This will evaluate if the CPP is capable of exploiting any gaps that are formed as pedestrians adjust their own trajectories to avoid collisions with one another, resulting in the following behaviour:

  $a_{2.1}$ A higher collision likelihood

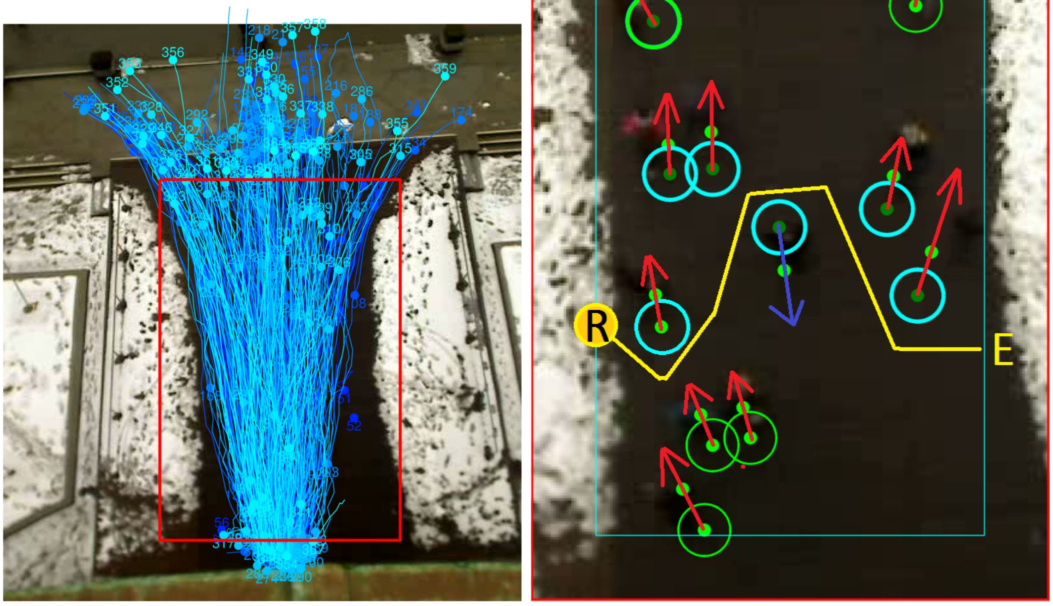
  $a_{2.2}$ A more dispersed crowd, as pedestrians avoid collisions

  $a_{2.3}$ Less predictable movements, less likely to follow the CVM

The CNS should be evaluated using a combination of the two directional possibilities of the crowd, and the AMR moving in parallel or perpendicular to their flow. The dataset is now analysed in order to find scenarios that will suit these tests. The workspace for which the AMR must operate within is also defined, Fig. 7.1, so that it navigates through a populated area.

7.1.2 Finding Appropriate Pedestrian Dataset Frames

To ensure the CPP predicts paths through the crowd from different orientations, the AMR will begin at three evenly spaced intervals along one edge of the workspace. Three destinations lines are then placed at evenly spaced intervals along the opposite side. The ideal paths are formed by connecting each start to each end, creating 9



(a)

(b)

Fig. 7.1 Visualisation of the pedestrian dataset. (a) The trajectories of all pedestrians over the entire dataset. When selecting the pedestrian data to use, it is initially evaluated within the red box. This ensures the CPP predicts a path through the crowd, rather than around it, as the robot's ideal paths will be located over the area of the crowd with the highest continuity. (b) Exaggerated path for the CPP to plan, highlighting that the AMR (yellow circle) must considerately navigate (yellow path) around multiple pedestrians simultaneously, to reach its destination (yellow 'E').

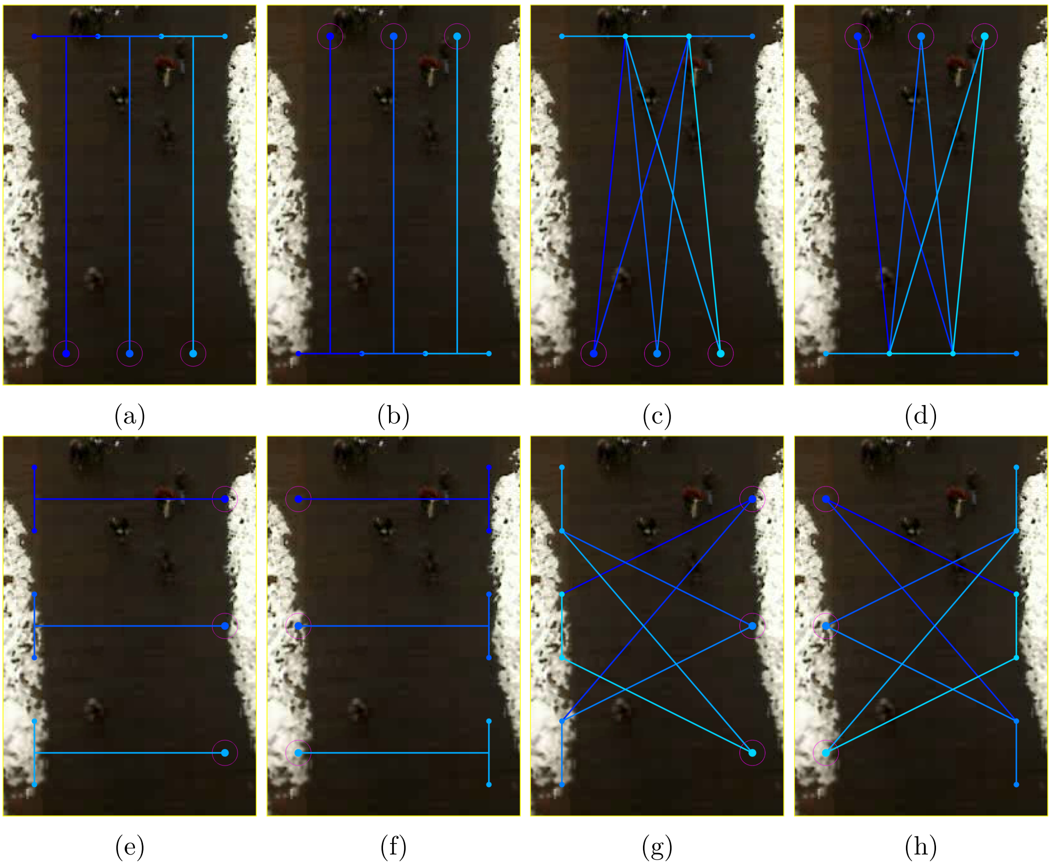


Fig. 7.2 The ideal paths that cross the workspace Up (↑), Down (↓), Left (←), and Right (→), from one workspace edge to the opposite. The CPP will predict paths from three evenly spaced start points, to three evenly spaced destination lines. The CPP will predict paths from each start point to each destination line, creating nine journeys in one general direction. This is repeated for all all 4 sides, generating 36 paths in total. (a-b) Moving directly parallel to crowd. (c-d) Moving diagonally parallel to crowd. (e-f) Moving directly perpendicular to crowd. (g-h) Moving diagonally perpendicular to crowd.

Initial	Crowd		Crowd Analysis		
Demo	Direction	People	Up	Down	Groups
Fig. 7.4a	Same	5	0	5	2
Fig. 7.4b	Direction	9	2	7	4, 2
Fig. 7.4c	Alternating	7	3	4	3, 4
Fig. 7.4d	Direction	8	3	5	0

Table 7.1 The data for each crowd scenario tested.

unique paths that cross the workspace at different angles. This process is repeated for each edge of the workspace, creating a total of 36 unique ideal paths, Fig. 7.2.

To evaluate the CPP using the pedestrian data file, the crowd scenarios must occur within the workspace, Fig. 7.1a. Not only do all pedestrian trajectories consistently pass though this area, the dataset includes their trajectories beyond it. This is beneficial as it aims to prevent the CPP predicting paths that will move the robot around the area where the pedestrians' trajectories are recorded. This will force the CPP to predict paths that cross the crowd, rather than circumnavigating the crowd entirely.

To find suitable pedestrian data, the workspace area must contain pedestrians that meet the directional requirements of Items a_1 and a_2. The area of interest is also extended outward by 1.5 times the robot's radius, $1.5r_r$, to include pedestrians that may be just behind the robot, Fig. 7.3. Therefore, when the robot begins its journey pedestrians can potentially approach the robot from all angles. Finding the instances within the datafile that fulfil the requirements requires qualitative analysis of the video. The four combinations of pedestrian direction and crowd density were found by examining the video and dataset simultaneously, in order to select the appropriate video frames, Table 7.1 and Fig. 7.4.

7.2 Evaluating the Considerate Navigation Strategy

The AMR's movement will be evaluated using each crowd scenario, along all 36 trajectories, with a maximum velocity limit the average pedestrian walking speed, $\max(v_r) = 1.4ms^{-1}$ [246]. The prediction time-horizon will be set to a constant $T = 5$, as this was proved to be to overall optimal value in the previous chapter, Section 6.4.4 (Fig. 6.20).

The evaluation of the AMR's movements will be performed with the same novel QPMs used when testing the CNS in the previous chapter experiments, Section 6.2.2. As the pedestrians cannot interact with the AMR collisions may occur more frequently, due to the pedestrians walking into the AMR. The user defined weight, α, that can be applied to the proximity consideration resistance, Ω_p (Eq. (3.13), Section 3.3.4), could have been increased so that the CPP did not predict paths in

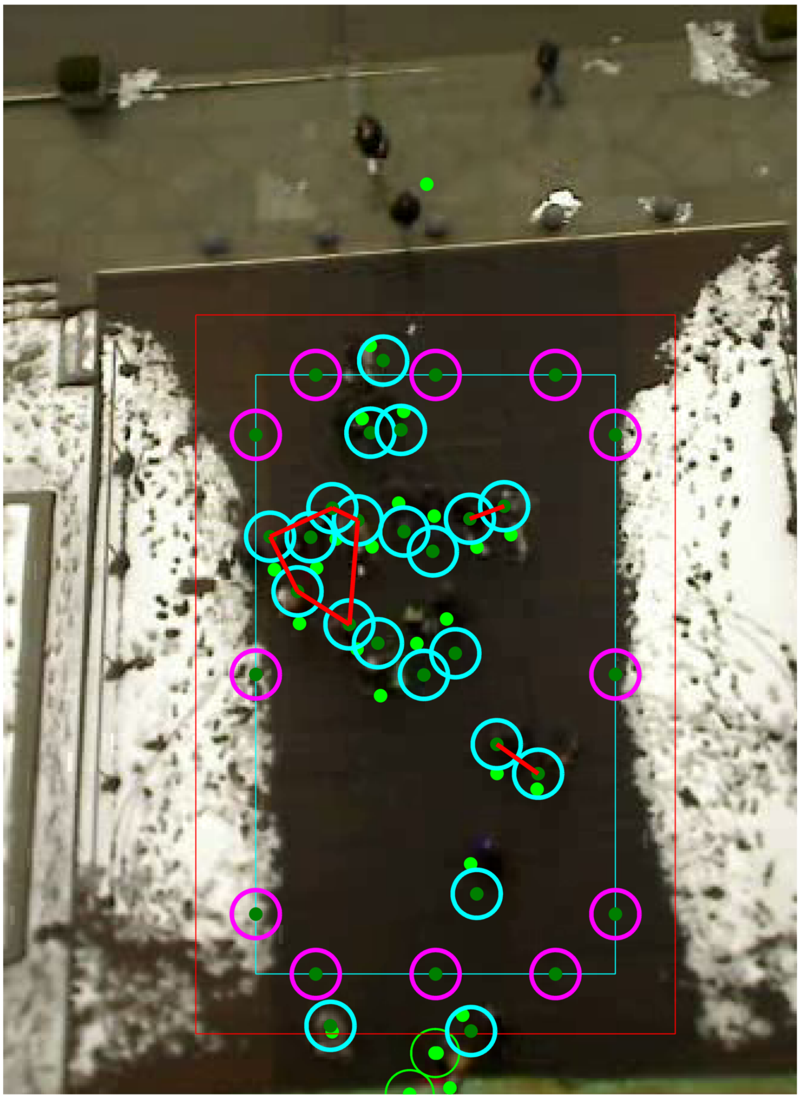
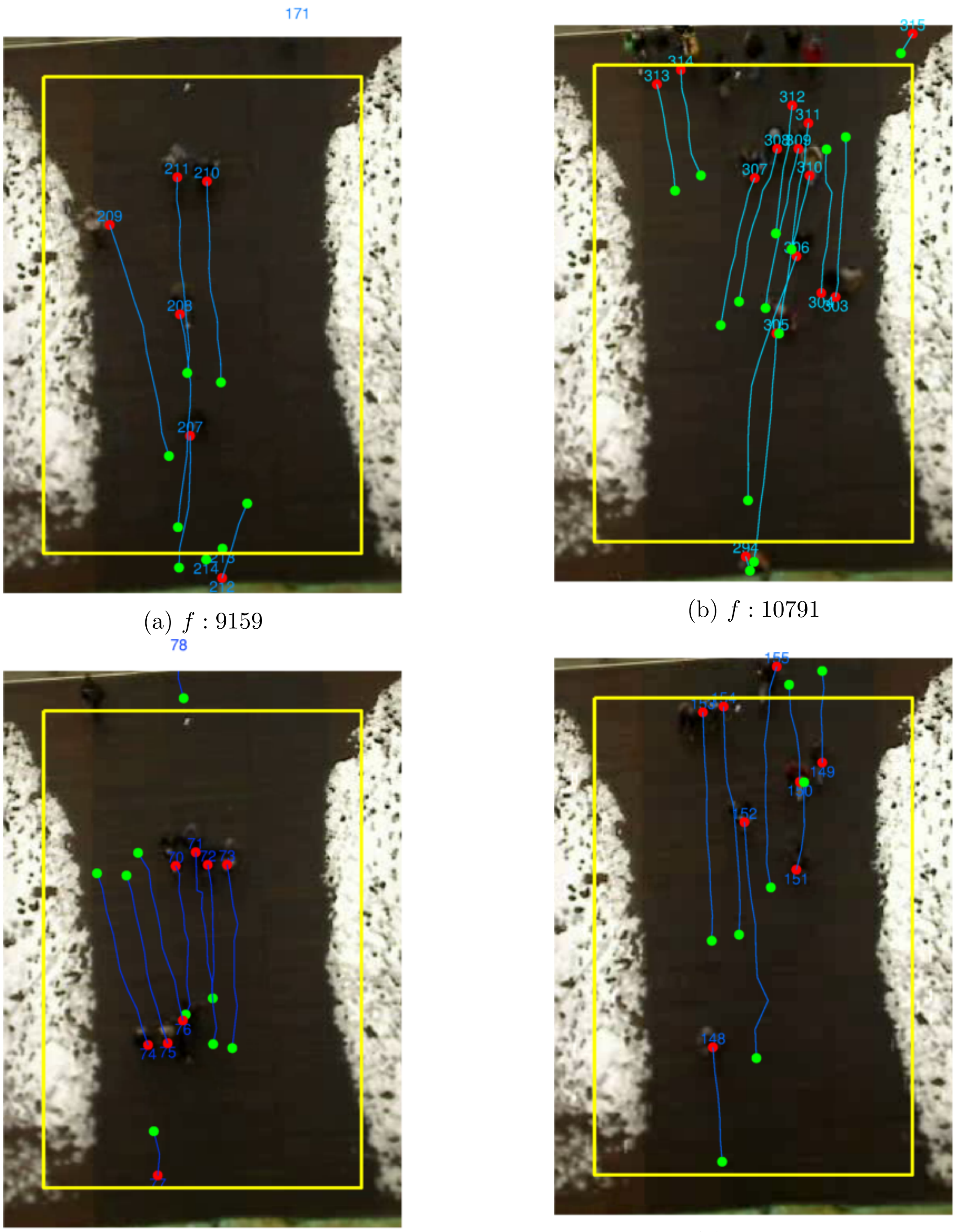


Fig. 7.3 Demonstration of how video frames that fulfil the requirements of the pedestrian scenarios are obtained. Pedestrians are represented by sky blue circles inside, and green circles outside of the red box, respectively. For a frame to be selected, the appropriate pedestrians must be within the red box, and also not intersect with any purple circles that represent any of the robot's start positions. The green dots in front of each pedestrian is its position in the next frame, used to calculated the pedestrians' velocities. The red convex polygon over the pedestrians represents a group of six pedestrians, and the two red lines indicate two couples moving together.



(a) f : 9159

(b) f : 10791

(c) f : 4265

(d) f : 7481

Fig. 7.4 Scenarios to be tested, with either uni-directional pedestrian traffic (a-b) or bi-directional pedestrian traffic (c-d). Pedestrians begin at the chosen frame, f, marked with a red dot, which overlays their position in the video. Their trajectories are then mapped over the next to frames to highlight their behaviour, with the end of their path marked with a green dot. The robot's ideal movements would occur within the yellow box, with the pedestrian data recorded beyond its limits. (a) Pedestrians travel down, with one couple. (b) Pedestrians travel down, with a group of 4. A couple travel up. (c) Two groups of 3 and 4 move past each other. (d) All pedestrians individually move past one another.

Ideal path orientation	Ideal paths' relative direction to the crowd	
	Parallel	Perpendicular
Direct	Move along with the crowd flow, adopting similar movements to the pedestrians	Move behind pedestrians in order to cut directly across the crowd (should provide the biggest challenge for the CPP)
Diagonal	Filter between pedestrian traffic lanes, in order to smoothly move across the crowd	Take advantage of the overall crowd flow when crossing individual pedestrian trajectories

Table 7.2 Subdivision of QPM assessment for crowd scenarios. Each of the four configurations is designed to test a certain aspect of how the AMR should ideally behave, in order to move towards its destination.

such a close proximity to pedestrians. However, the CNS would have resulted in an AMR that is very evasive and would have circumnavigated the crowd instead.

The QPMs will compare how well the CNS moves the AMR through the crowd, along all 36 paths that cross the workspace, Fig. 7.2. All paths will be grouped into the four categories:

b_1 Moving directly parallel to crowd.

b_2 Moving diagonally parallel to crowd.

b_3 Moving directly perpendicular to crowd.

b_4 Moving diagonally perpendicular to crowd.

The purpose for these sub-divisions is highlighted in Table 7.2. These will also be evaluated against uni-directional pedestrian traffic (Item a_1) and bi-directional pedestrian traffic (Item a_2), making a total of 8 classes.

When evaluating the QPMs of the CNS' ability to navigate through a crowd of pedestrians moving in the same direction as the crowd, Fig. 7.5, and to navigate through a crowd of pedestrians moving in the alternating directions, Fig. 7.6, the results were just as expected. The results reflect the behaviour anticipated when subdividing these classes in Table 7.2.

When the AMR moves with the crowd (in parallel to each pedestrians movements), the QPMs show that it took less time and distance to reach its goal. This is due to the fact that the CPP planned paths that moved with the crowd flow. The diagonal variation of this took slightly longer as it must filter across pedestrian trajectories. When the AMR moves across the crowd (perpendicular to each pedestrians movements), the QPMs show that the direct paths took the longest in 3 out of the 4 cases. Also, the total value for all combined metrics show this has the worst performance rating. This is due to the fact that the CPP had to plan paths that moved across pedestrian trajectories. The diagonal variation of this would perform better as it can filter across pedestrian trajectories, rather than cutting directly across them.

7.3 Qualitative Examples of the Considerate Navigation Strategy

Eight examples are given of the AMR behaving considerately, when moving along paths stated in Table 7.2, and are repeated for uni-direction and bi-directional traffic flow. The captions of all figures provide the following information:

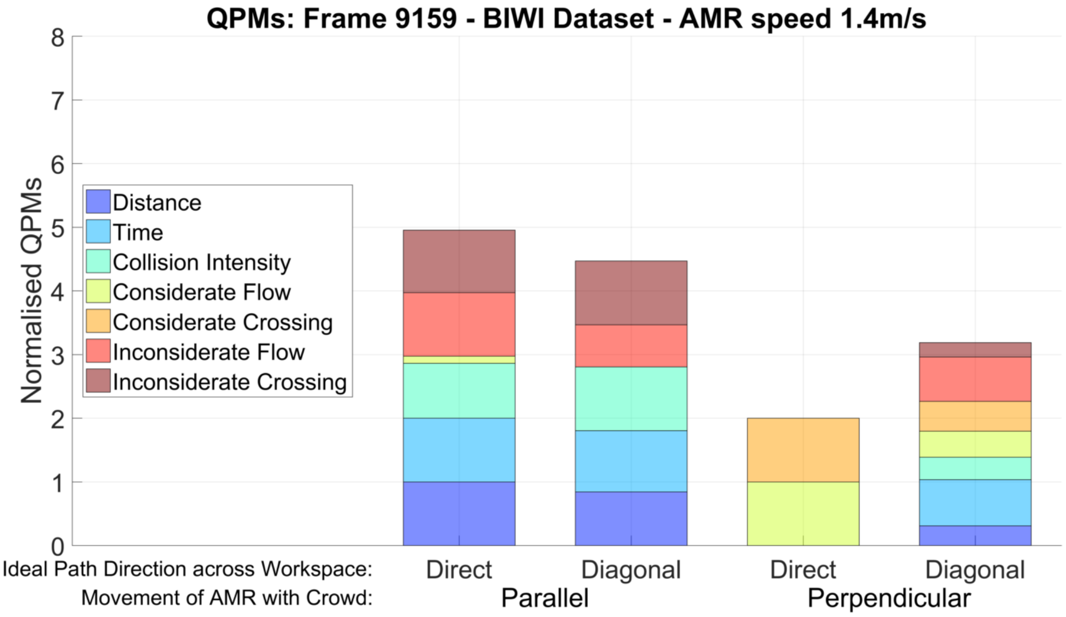
- Ideal path direction: e.g. LEFT/RIGHT/UP/DOWN : Diagonal/Direct
- Dataset frame number, f: Relevant information to be found in Table 7.1
- Start and destination positions, in relation to their order along the workspace edges: e.g. 1 to 3

The uncertainty ellipses of each pedestrian overlay their position in the video frame, and have arrow indication their direction. The FGP predicted by the CPP is the purple line with corresponding dots for each time-step, and a green Bezier curve, starting from the orange robot circle. The pink line behind the robot shows its path taken since the start of the simulation. The destination of the AMR is highlighted with an orange line.

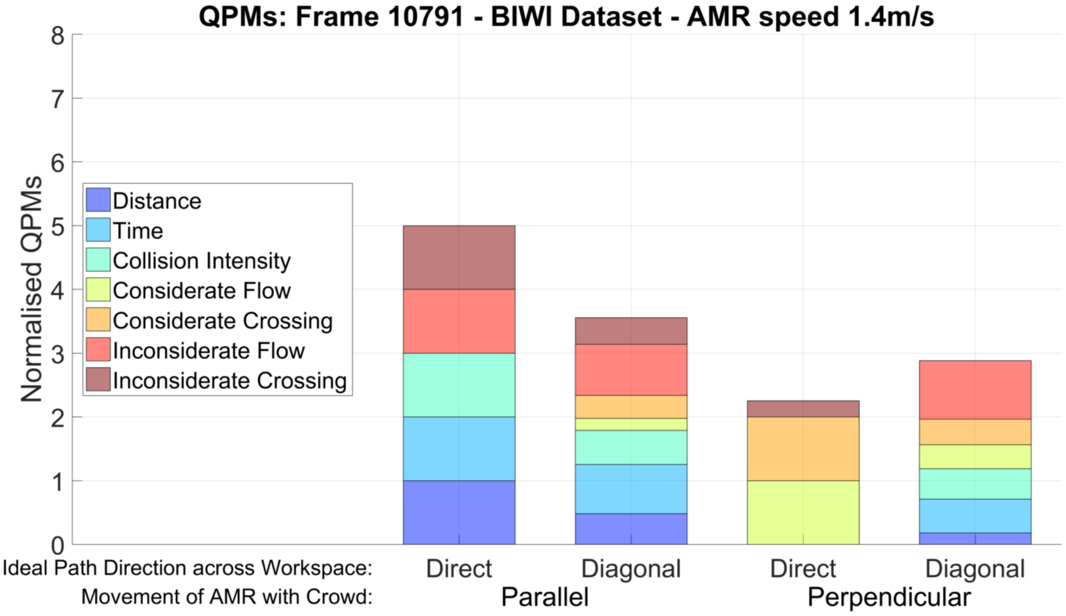
7.4 Chapter Overview

This chapter has provided an initial evaluation of how well the CNS is capable of predicting considerate paths within a dynamic environment of real pedestrians. The BIWI Walking Pedestrians dataset provided crowd data of a number of real-life pedestrian trajectories, for which the CPP was able to predict a number of qualitative considerate paths through.

Despite the fact that the pedestrian data does not respond to the AMR, the QPMs provided evidence that the AMR behaved as indented. Corroborated by the qualitative examples, the CPP was able to plan paths that exploited the gaps made available inbetween pedestrians. Coupled with the considerate weights that cause the CPP to select paths that encourage movement with the crowd, and to move behind a pedestrian when crossing, *truly* considerate paths were planned and executed as the pedestrians did not need to react.

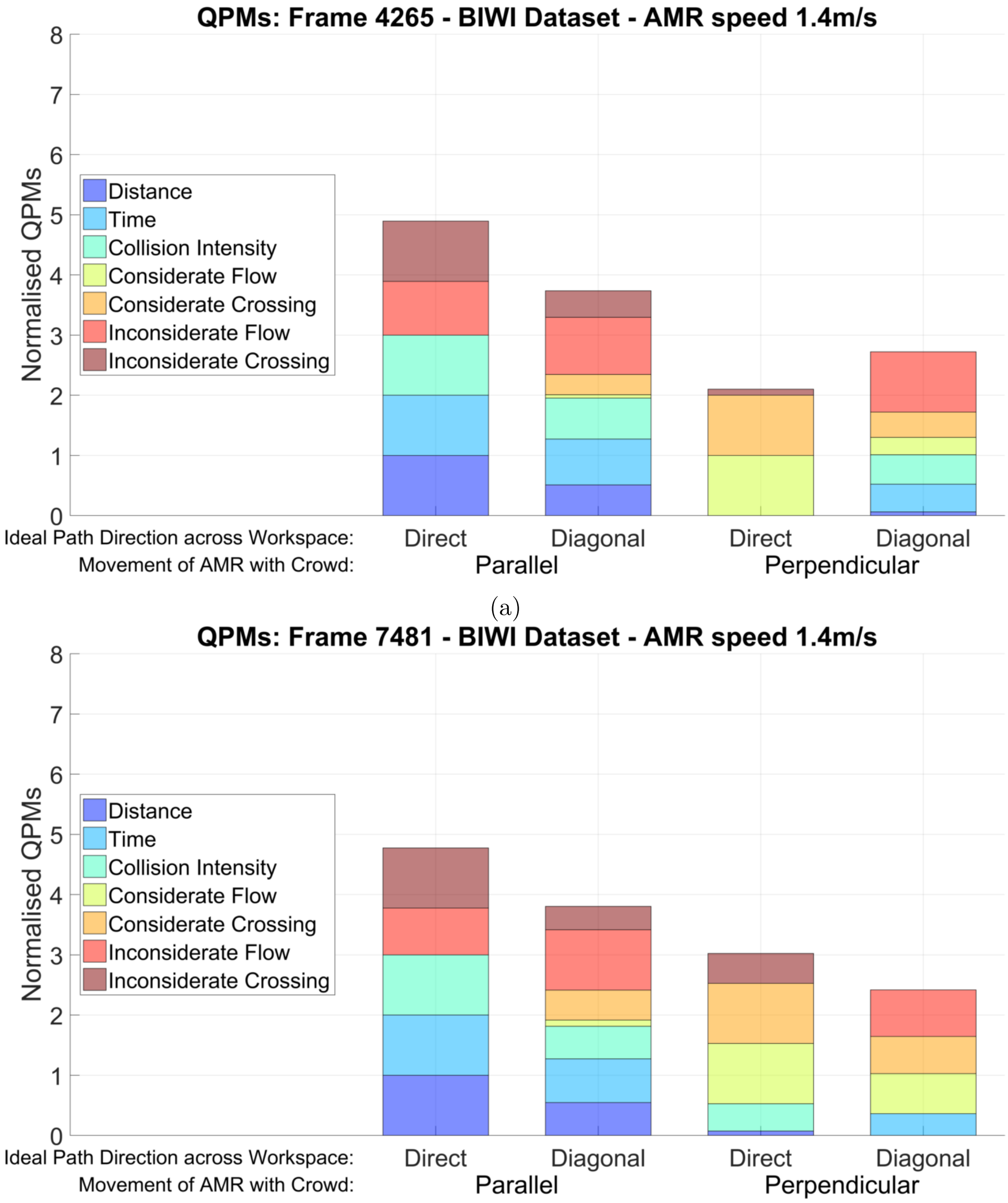


(a)



(b)

Fig. 7.5 QPMs for an AMR moving within a crowd of uni-directional pedestrians. (a) This scenario relates to the pedestrian data configuration seen in Fig. 7.4a (b) This scenario relates to the pedestrian data configuration seen in Fig. 7.4b



(a)

(b)

Fig. 7.6 QPMs for an AMR moving within a crowd of bi-directional pedestrians. (a) This scenario relates to the pedestrian data configuration seen in Fig. 7.4c. (b) This scenario relates to the pedestrian data configuration seen in Fig. 7.4d.

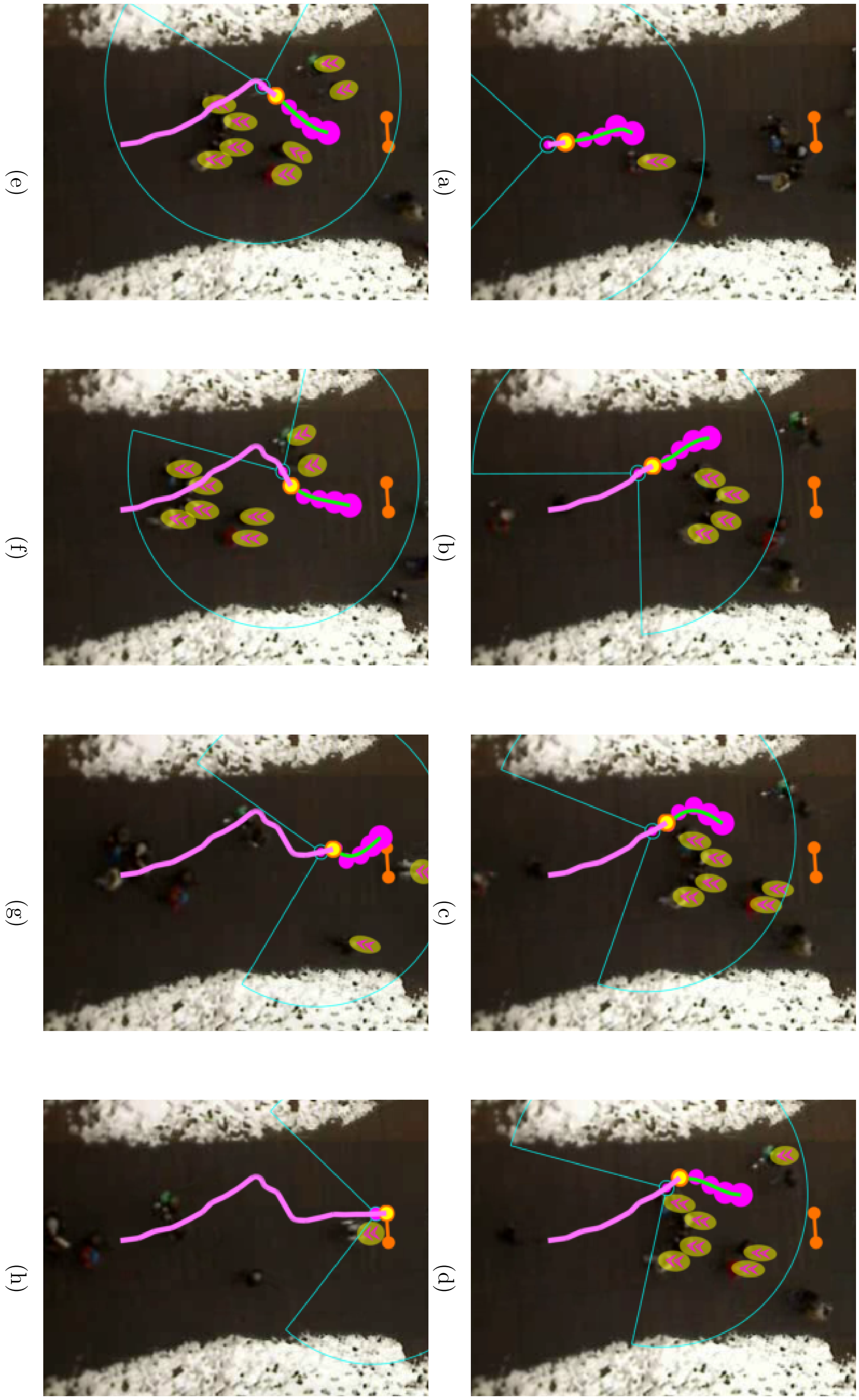
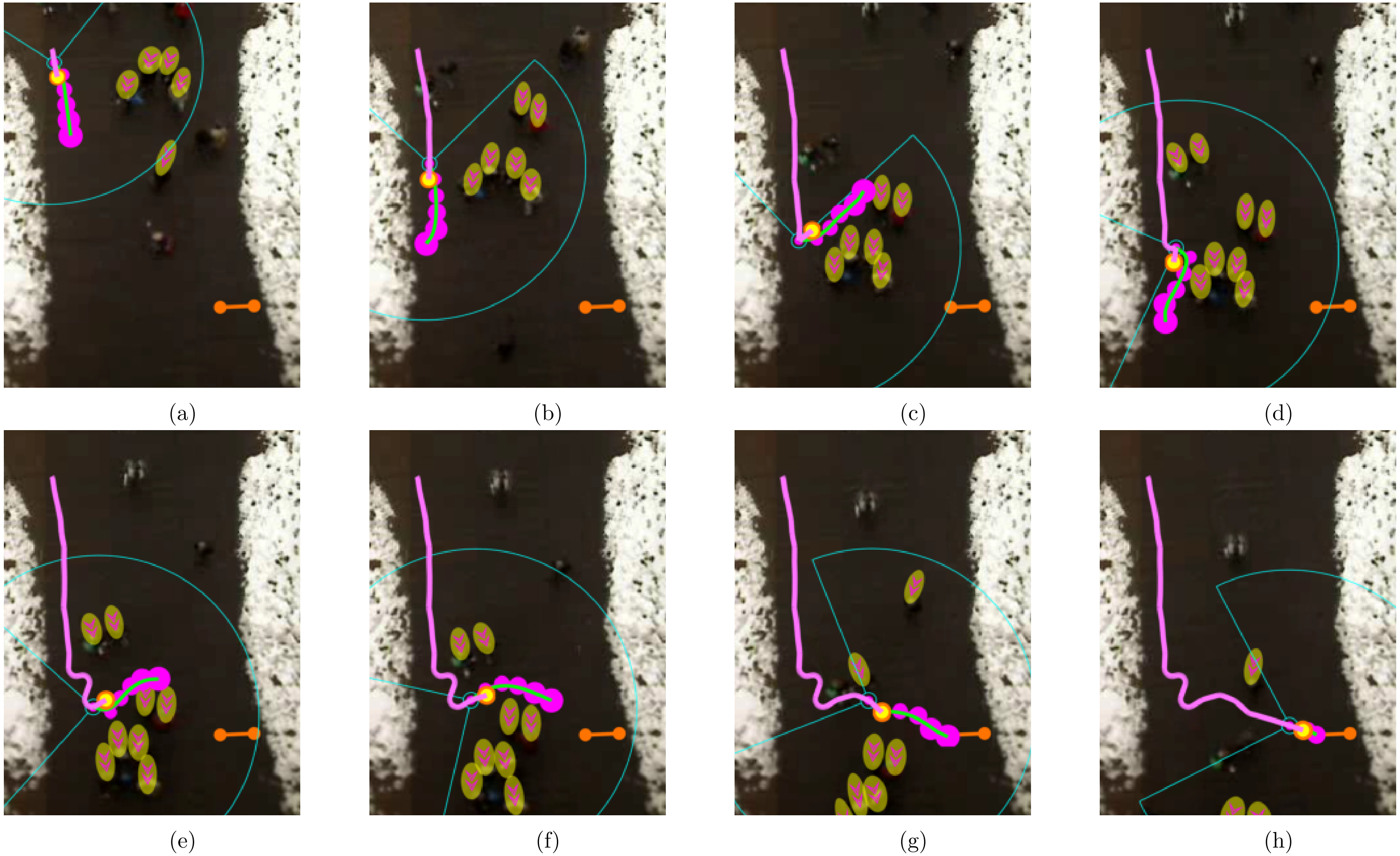


Fig. 7.7 UP: Direct - *f* : 10791 - Start: 2 - Destination: 2.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7.8 DOWN: Diagonal - f : 10791 - Start: 1 - Destination: 3.

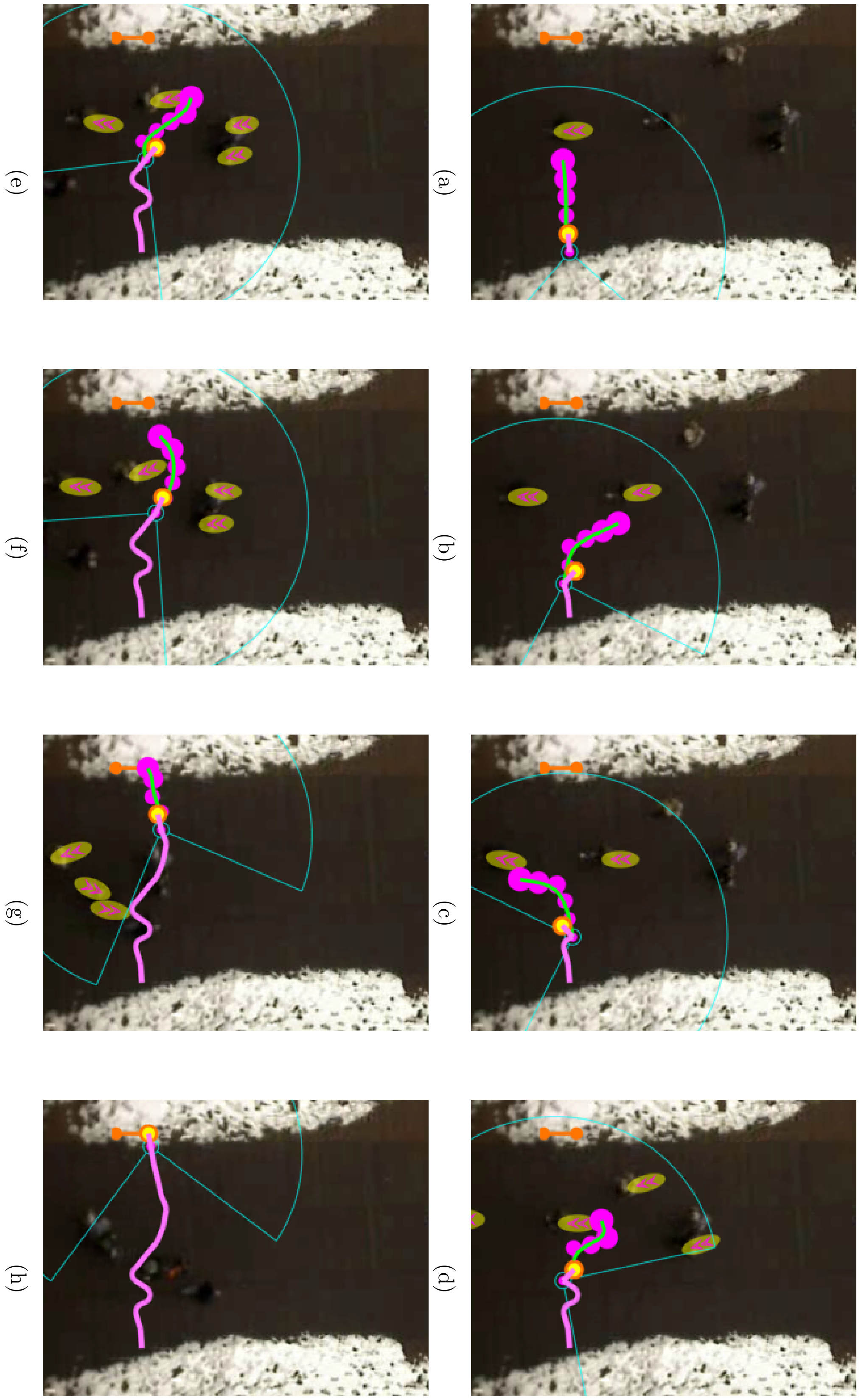

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 7.9 LEFT: Direct - f : 9159 - Start: 3 - Destination: 3.

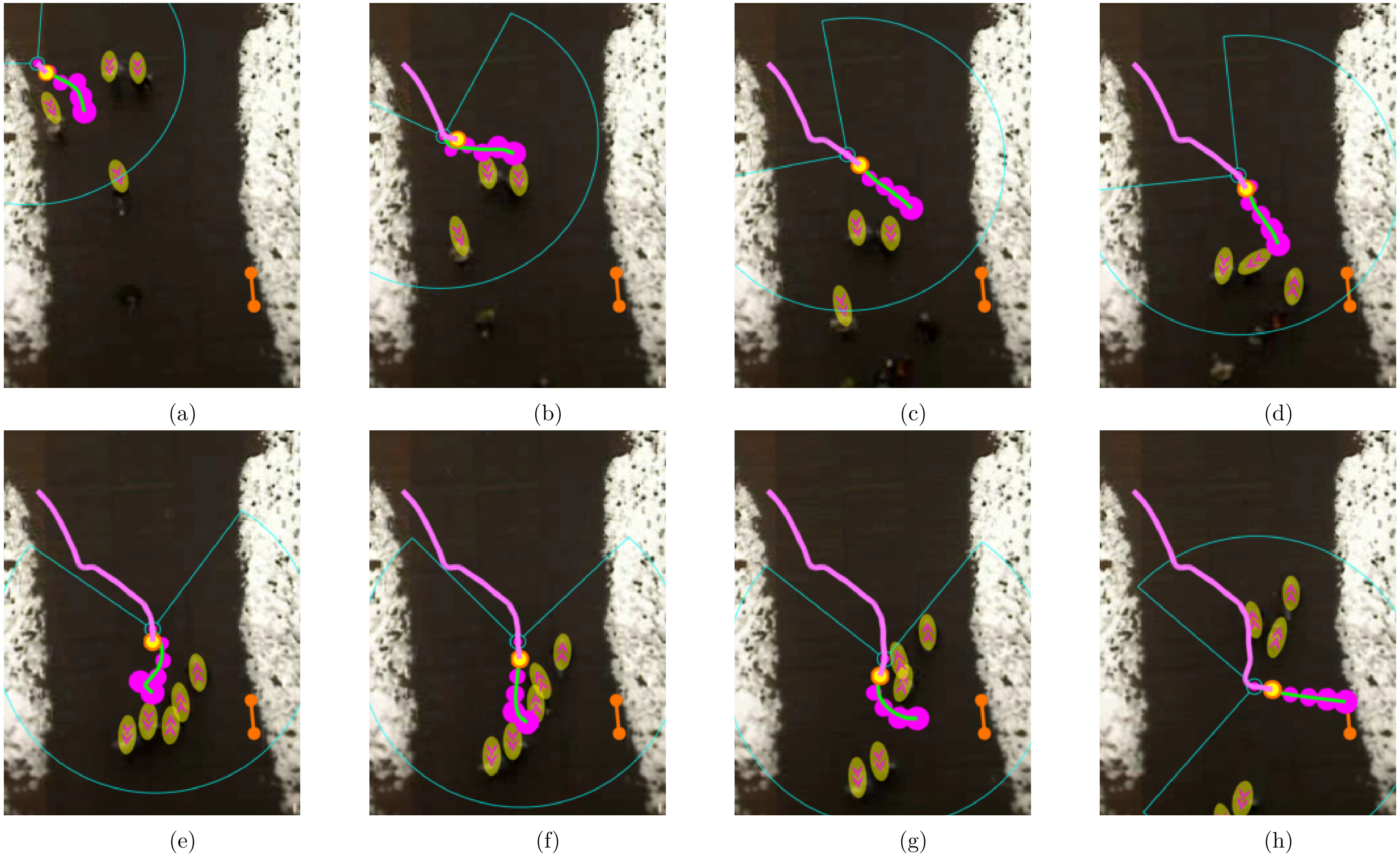
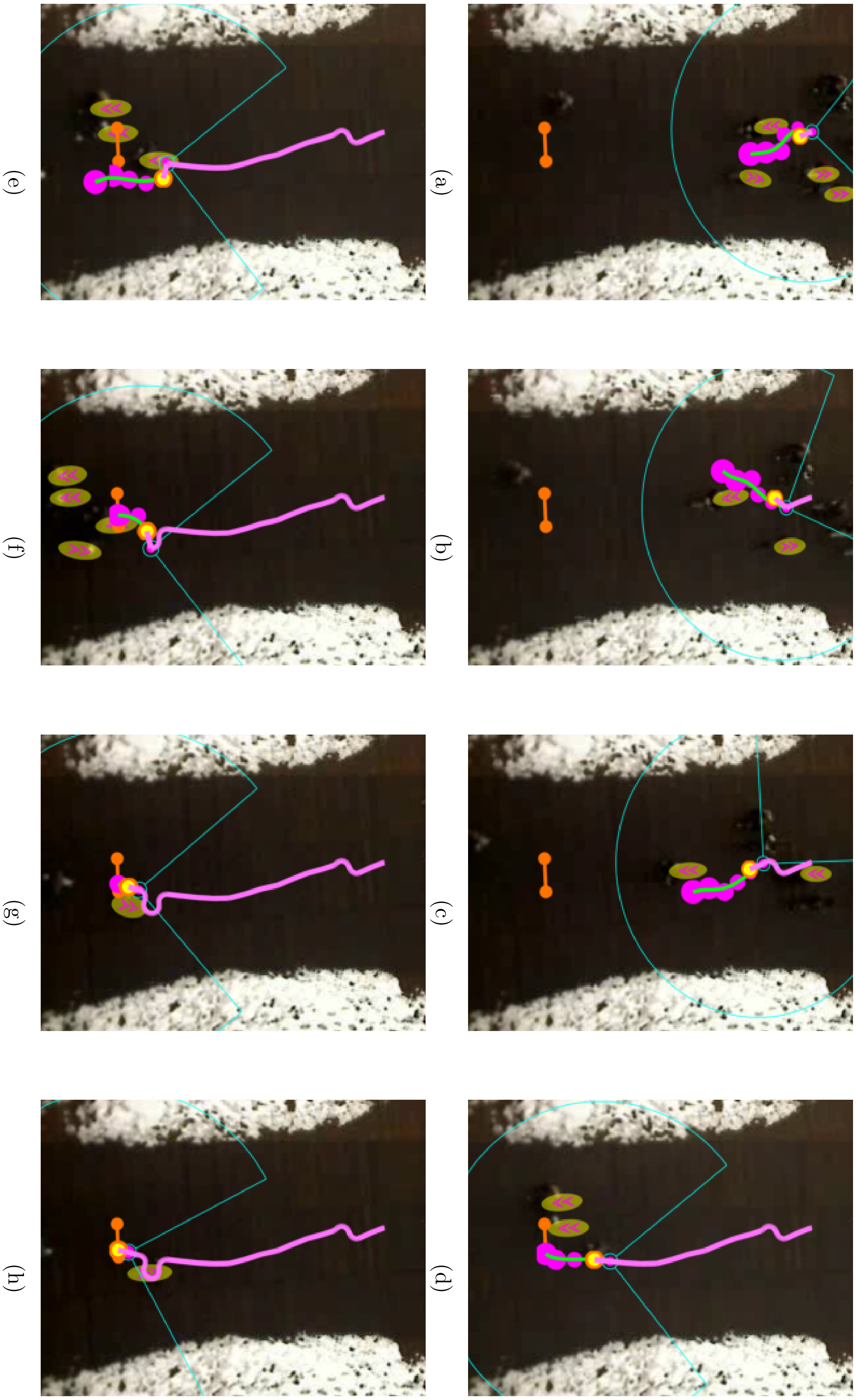
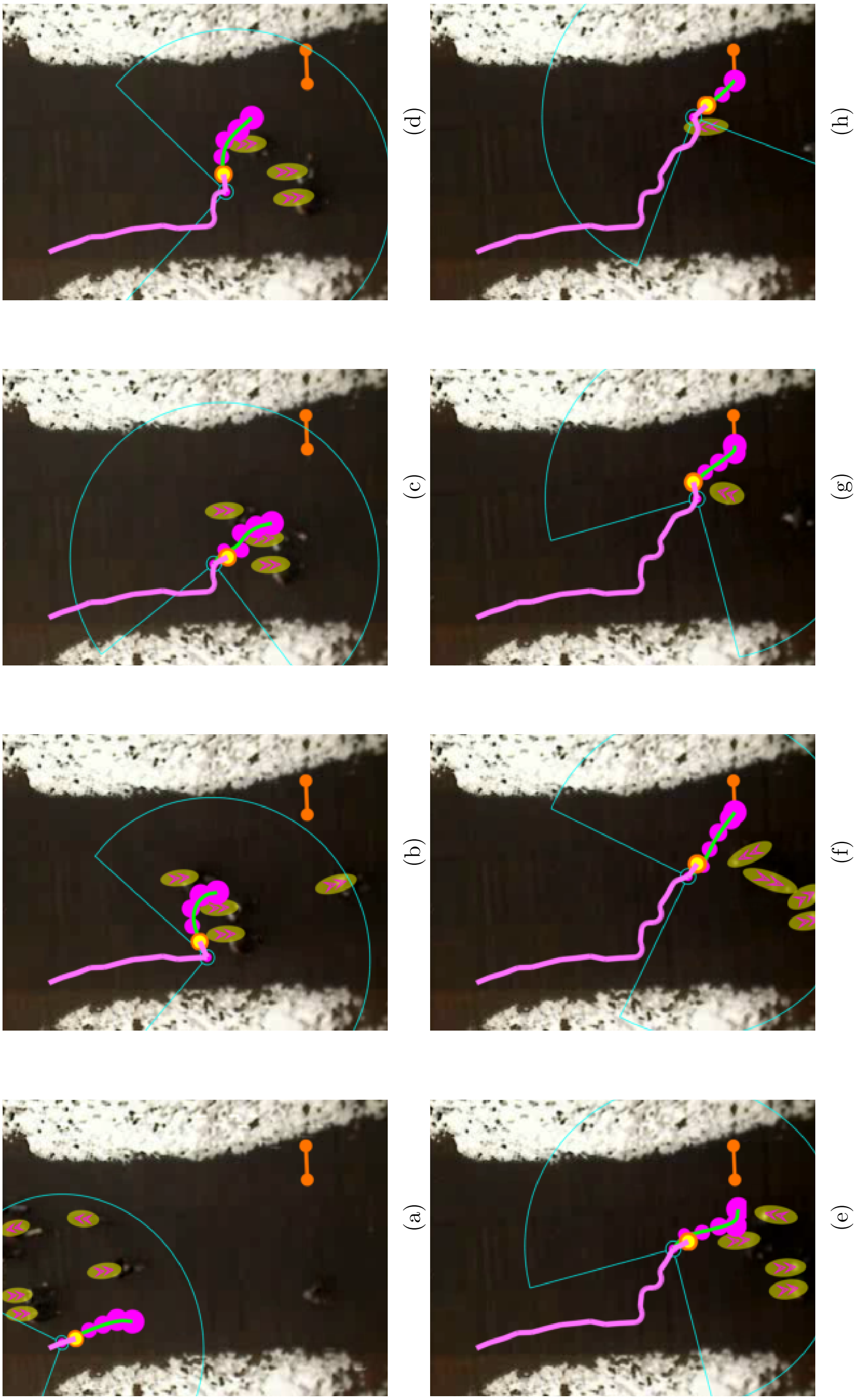


Fig. 7.10 RIGHT: Diagonal - f : 9159 - Start: 1 - Destination: 3.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 7.11 DOWN: Direct - f : 7481 - Start: 2 - Destination: 2.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 7.12 RIGHT: Diagonal - f : 4625 - Start: 1 - Destination: 3.

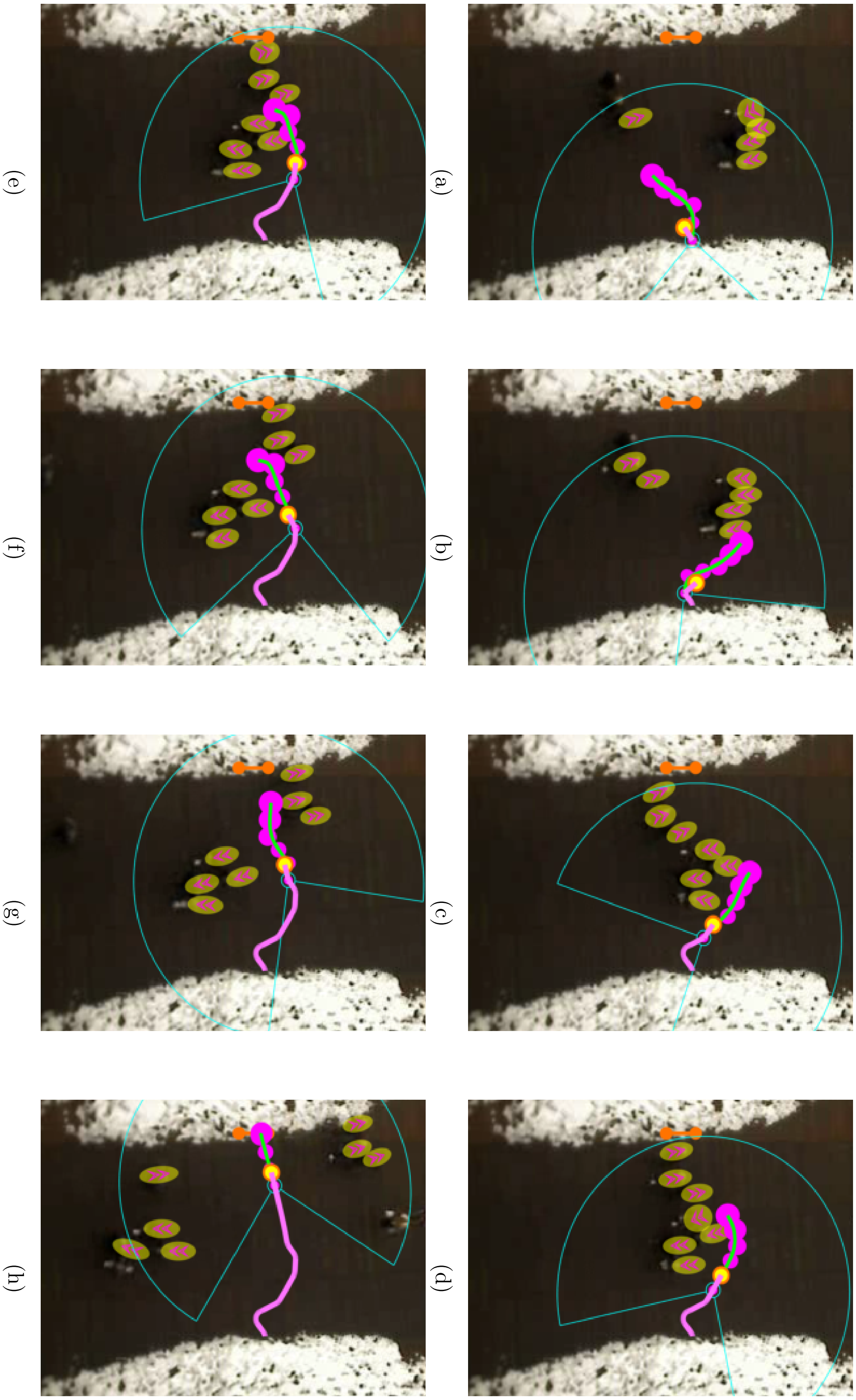


Fig. 7.13 LEFT: Direct - f : 4625 - Start: 2 - Destination: 2.

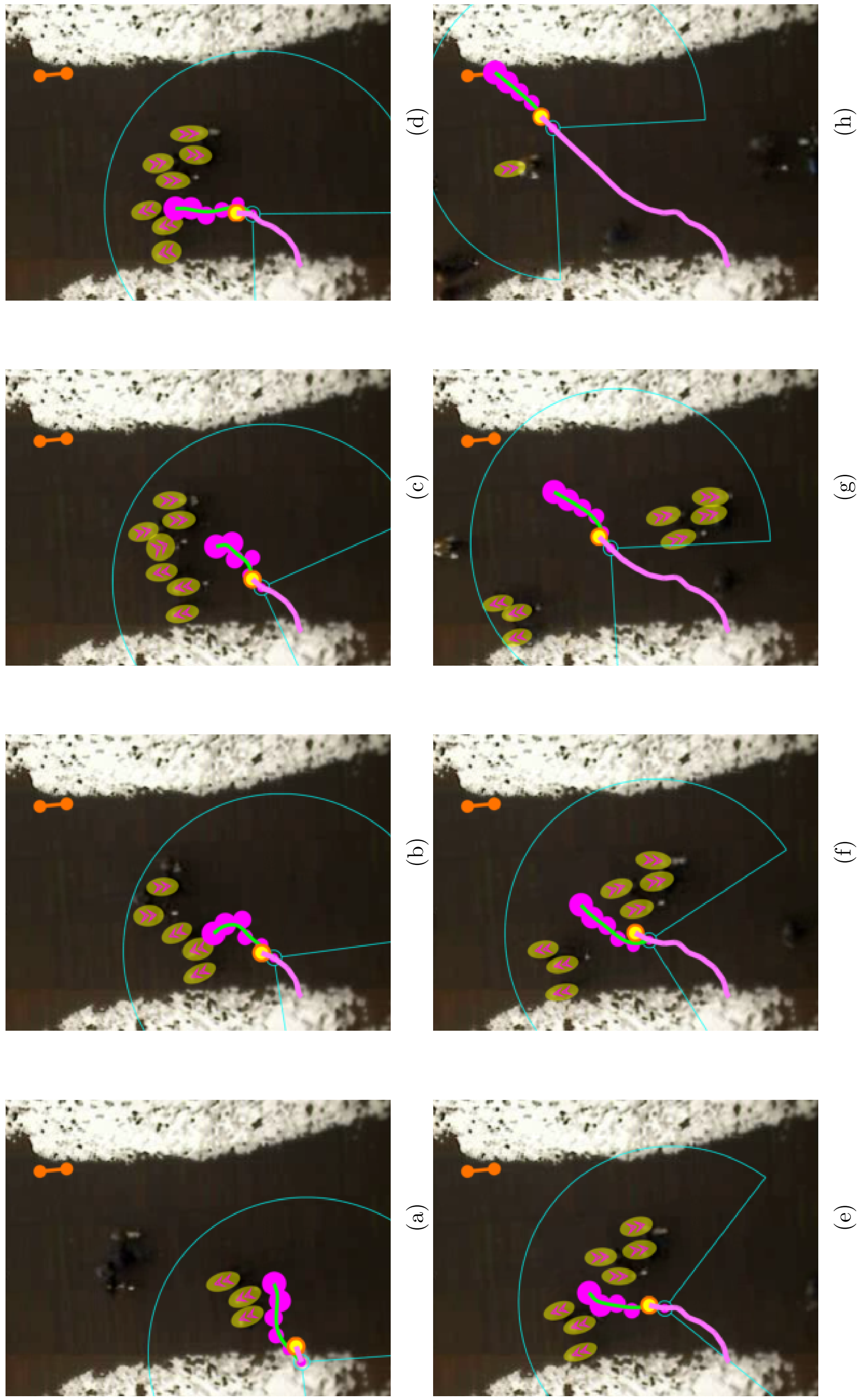


Fig. 7.14 UP: Diagonal - f : 4625 - Start: 1 - Destination: 3.

Chapter 8

Summary & Future Work

People call me a perfectionist, but I'm not. I'm a rightist. I do something until it's right, and then I move on to the next thing.

— James Cameron

This chapter summarises the work completed within this thesis, along with an evaluation of the novel contributions, and the potential for additional work. This thesis has presented a novel robot navigation strategy designed to move an autonomous mobile robot (AMR) within a crowded and dynamic environment whilst exhibiting considerate behaviour. The fundamental element of this considerate navigation strategy (CNS) is the novel considerate path planner (CPP), designed to plan considerate paths through a crowd of free-flowing pedestrians. Using a novel hybrid roadmap that combines a Voronoi diagram (VD) with a visibility graph (VG), the CPP calculates safe and considerate paths. The CNS implements movement of an AMR, the performance of which is analysed using a set of novel quantitative performance metrics (QPMs) designed to evaluate considerate behaviour.

8.1 Aims and Objectives Completion

The aims and objectives defined for the CNS, Section 1.3.2, were all successfully completed. Details of which are now given of how the objectives for the CNS were achieved:

a_1 "Move with consideration towards other pedestrians"

$a_{1.1}$ By recreating the empirical studies designed to evaluate human-human collision avoidance strategies, the CNS was able to avoid a collision with another dynamic agent, Fig. 6.5. The collision avoidance was not only considerate when the oncoming agent was non-reactive, it was implicitly cooperative when the agent was reactive.

a_2 "Arrive successfully at a prescribed destination"

$a_{2.1}$ The AMR was able to arrive successfully at its destination in all but two cases, Section 6.4.2 (Fig. 6.14b. This was due to the the AMR becoming stuck in between other dynamic agents that had moved around it.

a_3 "Avoid collisions with anything"

$a_{3.1}$ The use of a *buffer* zone that surrounded the AMR, Section 3.3.1 (Fig. 3.8), causes the AMR to automatically stop if another agent was too close.

a_4 "Minimize any additional congestion the robot may cause to the crowd as a whole"

$a_{4.1}$ When testing the CNS in a crowd, Section 6.4, the AMR was shown to prefer movement that was in the direction of crowd flow, so as not to act as a blockage that other dynamic agents must pass. The AMR also avoided areas where the crowd became too dense, which did not reduce the amount of available space for other agents any further.

Details are now given of how the objectives for the CPP were achieved:

b_1 "Plan paths that minimise potential collisions"

$b_{1.1}$ When possible, the CPP was shown to successfully exploit available and upcoming gaps within the predicted positions of crowd members, Section 7.3.

b_2 "Plan paths that move in the same pedestrian direction"

$b_{2.1}$ Due to the relative trajectory resistance, Section 3.3.4.2, more desirable paths were ones that moved in the same direction as neighbouring pedestrians. The final global paths planned were a reflection of this calculation, which is made during the path planning stage, Fig. 5.14.

b_3 "Plan paths that avoid crossing another pedestrians path"

$b_{3.1}$ Due to the proximity and relative trajectory resistance, Sections 3.3.4.1 and 3.3.4.2, the CPP was shown to successfully plan paths that avoided crossing in front of the predicted locations of any dynamic agents, Section 7.3, wherever possible. When a clear choice can be seen between moving behind or in-front, the CPP consistently planned a path that moved behind any dynamic agent.

The objectives of the CNS have been achieved by using novel path planning and search techniques. An evaluation of the specific contributions will be presented next.

8.2 Contributions Evaluation

This section provides a summary of all the novel strategic, algorithmic, and analytical components of this thesis.

A New Taxonomy for AMR-human navigation, Table 2.5 No taxonomies exist to evaluate mobile robot autonomy within a pedestrian environment. In the advent of Human-Robot Interaction, autonomy can now no longer be defined by how little direct human controlled input is needed.

The updated taxonomy includes the level of input chosen to be taken by the AMR. Instead of a direct human input that controls of the AMR's movement, the AMR exchanges information between itself and a human in order to act collaboratively with them. Also, "passive" input is introduced, whereby input is taken from neighbouring humans in order for the AMR to make decisions, and so responsibly act around other people.

The Pedestrian Model, Section 3.2 This was designed to "embrace" the uncertainty of pedestrian movement predictions with an *uncertainty ellipse*. By assuming a simple constant velocity model, movements of an agent can be easily predicted. Any likely deviations in speed and trajectory can be confidently contained by increasing the ellipses' axes. Any path planned around the ellipses will therefore be less likely to cause a collision.

When evaluating the uncertainty ellipse's ability to contain pedestrian movements in the future, Section 6.3.1 (Fig. 6.4), it was shown to outperform containment based on purely statistical deviations seen in Fig. 3.3. The results can be confidently confirmed to contain pedestrians movement, as a pedestrian simulator was used to evaluate them.

Approximation of the Voronoi Diagram of Ellipses, Section 4.3 Currently the VD of fully-intersecting ellipses has not yet been found. Therefore, to approximate one, a novel modification was made of the VD of circles by generating *pseudo-ellipses* made from four tangentially aligned circles. The VD of pseudo-ellipses can be implemented with logarithmic time complexity, and produces very accurate predictions of the VD's vertex points. From he roots of the vertex points belonging to the pseudo-ellipses the vertex points of real ellipses, can be converged upon in only a few iterations, Fig. 4.11.

All vertex points external to the ellipses are guaranteed as the pseudo-ellipses are constructed inside the real ellipses. The only potential missing vertex points are inside ellipses over high density areas, Section 4.2 (Fig. 4.8). For the purposes of the CPP the external vertex points are most important, and so this accuracy level is sufficient.

The Voronoi Diagram-Visibility Graph Hybrid Roadmap, Section 4.1 A novel combination of the VD and VG roadmaps was created, to be used as the CPP's roadmap. The VD element was easily created following the novel approximation of the VD of ellipses. However, the visibility graph (VG) traditionally connects polygon nodes together, Section 2.2.1.2 (Fig. 2.5b), and so a modified VG was developed for closed curve conic sections, Section 3.3.2.1 (Fig. 3.10).

The VD-VG roadmap can provide both the safest route, via the VD, as well as more direct and efficient routes, via the VG, Section 6.3.2.3. This allows a search algorithm to find a route along the roadmap that can switch from safest route, to more efficient root, at every node, successfully planing paths that are both a safe and efficient.

Considerate Weights, Section 3.3.4 For the CPP to plan considerate paths, a resistance is applied to the windowed Dijkstra's algorithm in order to slow the search down if it moves into *inconsiderate* areas, Section 5.2.1. This resistance, Eq. (3.13), is based on: proximity, and relative trajectory between the search frontier and any neighbouring uncertainty ellipses.

The CPP to plan considerate paths from an evaluation of the accumulated resistance along each search path. Seen in Chapters 6 and 7, the final global path of the CPP was able to produce paths that allowed the CPP objectives to be fulfilled (i.e. Fig. 6.10 and Section 7.3). This considerate weights were also used to modify the A* search algorithm, Section 6.4.1, which was shown to improve its performance, Sections 6.4.2 and 6.4.3.

Quantitative Performance Metrics, Section 6.2.2 Traditional QPMs focus on easily quantifiable measurements of distance, time, collision number, etc, Section 6.2.1. Novel QPMs were designed to both appraise the intensity of a near-miss collision, Section 6.2.2, as well as to measure considerate movements of an AMR, Section 6.2.3.

For the CNS to have an AMR behave considerately, the paths planned by the CPP must also be considerate. To achieve this paths were evaluated over variable prediction time-horizon, in order to evaluate how the environment may change using the pedestrian model, Section 3.2. When testing the CNS, Section 6.4, a variable prediction time-horizon of $T = 1 - 6$ was evaluated using the QPMs. The optimal prediction range was $T \geq 4$ when recreating empirical studies, Fig. 6.5, and $T = 5$ for more crowded environments, Fig. 6.20. For range of the AMR's field of view,$T \geq 5$ is reflective of a global path planning range, Table 6.1, which is required for considerate navigation to occur.

For crowded scenarios where qualitative analysis may not be possible, the QPMs provided much needed analysis around the CNS' ability to successfully move an AMR considerately. The validity of the *considerate* QPMs is corroborated by comparing the movement of the AMR through the dataset of real pedestrians, Section 7.2. The QPMs, Figs. 7.5 and 7.6, numerically reflect the qualitative observations, Section 7.3.

8.3 Potential Future Work

The CNS presented was able to move an AMR considerately within a crowd of both simulated pedestrians, a real pedestrian dataset, and amongst similar simulated AMRs. Currently the CNS is only a simulated algorithm, and the recursive serial implementation used to plan the paths means that the CPP cannot operate in real-time. However, an immediate improvement would be be implement the CPP using parallel processing.

Each unique P_t path is dependent only upon the end of the previous P_t path the search next search begins at. All path reduction methods to increase search efficiency and remove homotopic paths are also independent. Therefore, all new searches that begin at the end of each P_t path at t can perform the search for all P_T paths at $t+1$ independently. Also, the execution speed of an individual CPP iteration is only $75ms$, approximately. Therefore, if the timestep remains at $0.4s$, theoretically the CPP can plan a final global path to be executed in $375ms$, which is $25ms$ less than the time-step used. A more complex movement model for the AMR to use would also be an immediate improvement. To prevent the AMR getting stuck in between other dynamic agents, updating the differential model so that the AMR can rotate on the spot would allow it to potentially free itself. If the only available space is outside the drives maximum turning angle, this would remedy the problem and allow the AMR to move again.

The results obtained for the CNS were based on only an approximation of reality, using reactive simulated pedestrians or non-reactive pedestrian data. Therefore, the next stage would be to implement the CNS on a real-life robot platform. The AMR model used has already been based on a real-life sensor and platform, therefore if this was coupled with converting the CPP onto a parallel processing architecture, the CNS would be ready for real-world testing.

References

[1] A. Turner and A. Penn, "Encoding natural movement as an agent-based system: an investigation into human pedestrian behaviour in the built environment," *Environ Plann B*, vol. 29, no. 4, pp. 473–490, 2002.

[2] M. Batty, "Predicting where we walk," *Nature*, vol. 388, no. 6637, pp. 19–20, 1997.

[3] S. M. LaValle, *Planning algorithms.* Cambridge university press, 2006.

[4] M. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing*, vol. 9, no. 3, pp. 1102 – 1110, 2009.

[5] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introdutction to Autonomous Mobile Robots.* Intelligent robotics and autonomous agents, MIT Press, 2011.

[6] V. Hlavác, "Motion planning methods." 2015.

[7] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, "Time dependent planning on a layered social cost map for human-aware robot navigation," in *2015 European Conference on Mobile Robots (ECMR)*, pp. 1–6, Sept 2015.

[8] v. W. C. By Lcmp7 (Own work) [CC BY-SA 4.0 (http://creativecommons.org/licenses/by sa/4.0)]. https://commons.wikimedia.org/wiki/File%3AGrafo_de_visibilidad.png.

[9] v. W. C. By Eric O. Scott (Own work) [CC BY-SA 4.0 (http://creativecommons.org/licenses/by sa/4.0)]. https://commons.wikimedia.org/wiki/File%3APRM_with_Ob-maps.gif.

[10] J. Choi, M. Choi, S. Y. Nam, and W. K. Chung, "Autonomous topological modeling of a home environment and topological localization using a sonar grid map," *Autonomous Robots*, vol. 30, no. 4, pp. 351–368, 2011.

[11] N. Marquardt and S. Greenberg, "Proxemic interactions: From theory to practice," *Synthesis Lectures on Human-Centered Informatics*, vol. 8, no. 1, pp. 1–199, 2015.

[12] S. P. Hoogendoorn and P. H. Bovy, "Pedestrian route-choice and activity scheduling theory and models," *Transportation Research Part B: Methodological*, vol. 38, no. 2, pp. 169–190, 2004.

[13] F. Johansson, "Microscopic modeling and simulation of pedestrian traffic," 2013.

[14] S. Pellegrini *et al.*, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 261–268, 2009.

[15] E. W. A. C. Weisstein. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/ApolloniusCircle.html.

[16] C. Gloor, “Pedsim: A microscopic pedestrian crowd simulation system.” http://pedsim.silmaril.org, 2003.

[17] S. Pellegrini *et al.*, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 261 –268, 2009.

[18] M. Huber, Y.-H. Su, M. Krüger, K. Faschian, S. Glasauer, and J. Hermsdörfer, “Adjustments of speed and path when avoiding collisions with another pedestrian,” *PLoS ONE*, vol. 9, no. 2, 2014.

[19] M. R. Endsley, “Level of automation effects on performance, situation awareness and workload in a dynamic control task,” *Ergonomics*, vol. 42, no. 3, pp. 462–492, 1999.

[20] M. Vukobratović, “General about robots,” in *Applied Dynamics of Manipulation Robots*, pp. 1–34, Springer Berlin Heidelberg, 1989.

[21] R. Brooks, “A robust layered control system for a mobile robot,” *Robotics and Automation, IEEE Journal of*, vol. 2, pp. 14–23, Mar 1986.

[22] M. Walters *et al.*, “The influence of subjects’ personality traits on personal spatial zones in a human-robot interaction experiment,” in *Robot and Human Interactive Communication, 2005. IEEE International Workshop on*, pp. 347 – 352, 2005.

[23] D. Lambert, *Body Language 101: The Ultimate Guide to Knowing When People Are Lying, How They Are Feeling, What They Are Thinking, and More.* Skyhorse Publishing Company, Incorporated, 2013.

[24] A. P. J. L. S. A. Ushpiz, “Pedestrian flow and level of service,” *Journal of Transportation Engineering*, vol. 109, no. 1, pp. 46–56, 1983.

[25] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, “The icub humanoid robot: an open platform for research in embodied cognition,” in *Proceedings of the 8th workshop on performance metrics for intelligent systems*, pp. 50–56, ACM, 2008.

[26] Aldebaran Robotics, *NAO Datasheet.*

[27] M. Hirose and K. Ogawa, “Honda humanoid robots development,” *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, 2007.

[28] C. Disco and B. van der Meulen, *Getting New Technologies Together: Studies in Making Sociotechnical Order (De Gruyter Studies in Organization).* Walter de Gruyter & Co, 1998.

[29] P. Harish and P. Narayanan, “Accelerating large graph algorithms on the gpu using cuda,” in *High Performance Computing – HiPC 2007* (S. Aluru, M. Parashar, R. Badrinath, and V. Prasanna, eds.), vol. 4873 of *Lecture Notes in Computer Science*, pp. 197–208, Springer Berlin Heidelberg, 2007.

[30] D. Kumar and M. A. Qadeer, "Fast heterogeneous computing with cuda compatible tesla gpu computing processor (personal supercomputing)," in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, ICWET '10, (New York, NY, USA), pp. 925–930, ACM, 2010.

[31] S. B. Niku, *Introduction to Robotics: Analysis, Control, Applications.* John Wiley & Sons; 2nd Edition edition, 2010.

[32] D. Feil-Seifer and M. Mataric, "Defining socially assistive robotics," in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 465–468, June 2005.

[33] J. M. Beer, A. D. Fisk, and W. A. Rogers, "Toward a framework for levels of robot autonomy in human-robot interaction," *Journal of Human-Robot Interaction*, vol. 3, no. 2, pp. 74–99, 2014.

[34] H. SAWADA, "A talking robot and the expressive speech communication with human," *International Journal of Affective Engineering*, vol. 14, no. 2, pp. 95–102, 2015.

[35] M. Bortole, A. Venkatakrishnan, F. Zhu, J. C. Moreno, G. E. Francisco, J. L. Pons, and J. L. Contreras-Vidal, "The h2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study," *Journal of NeuroEngineering and Rehabilitation*, vol. 12, p. 54, Jun 2015.

[36] W. Burgard *et al.*, "The interactive museum tour-guide robot," in *Proc. of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998.

[37] S. Thrun *et al.*, "Minerva: a second-generation museum tour-guide robot," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 3, pp. 1999–2005 vol.3, 1999.

[38] H. Kretzschmar, M. Kuderer, and W. Burgard, "Learning to predict trajectories of cooperatively navigating agents," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 4015–4020, IEEE, 2014.

[39] M. Moussaïd, E. G. Guillot, M. Moreau, J. Fehrenbach, O. Chabiron, S. Lemercier, J. Pettré, C. Appert-Rolland, P. Degond, and G. Theraulaz, "Traffic instabilities in self-organized pedestrian crowds," *PLoS Comput Biol*, vol. 8, pp. 1–10, 03 2012.

[40] R. Kummerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "A navigation system for robots operating in crowded urban environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 3225–3232, IEEE, 2013.

[41] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[42] M. Shiomi, F. Zanlungo, K. Hayashi, and T. Kanda, "Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model," *International Journal of Social Robotics*, vol. 6, no. 3, pp. 443–455, 2014.

[43] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS one*, vol. 5, no. 4, p. e10047, 2010.

[44] S. Bandini, A. Gorrini, L. Manenti, and G. Vizzari, "Crowd and pedestrian dynamics: Empirical investigation and simulation," in *Proceedings of Measuring Behavior*, Citeseer, 2012.

[45] N. Bellomo and L. Gibelli, "Toward a mathematical theory of behavioral-social dynamics for pedestrian crowds," *Mathematical Models and Methods in Applied Sciences*, vol. 25, no. 13, pp. 2417–2437, 2015.

[46] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, 2006.

[47] J. Krause, G. D. Ruxton, and S. Krause, "Swarm intelligence in animals and humans," *Trends in ecology & evolution*, vol. 25, no. 1, pp. 28–34, 2010.

[48] RIKEN, "Largest neuronal network simulation achieved using k computer." Press Release, August 2013.

[49] N. Katevas, N. Sgouros, S. Tzafestas, G. Papakonstantinou, P. Beattie, J. Bishop, P. Tsanakas, and D. Koutsouris, "The autonomous mobile robot senario: a sensor aided intelligent navigation system for powered wheelchairs," *Robotics Automation Magazine, IEEE*, vol. 4, pp. 60–70, Dec 1997.

[50] I. Werry and K. Dautenhahn, "Applying mobile robot technology to the rehabilitation of autistic children," in *In: Procs SIRS99, 7th Symp on Intelligent Robotic Systems*, 1999.

[51] P. Hoppenot and E. Colle, "Localization and control of a rehabilitation mobile robot by close human-machine cooperation," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 9, no. 2, pp. 181–190, 2001.

[52] B. Hoyle and D. Waters, *Assistive Technology for Visually Impaired and Blind People*, ch. Mobility AT: The Batcane (UltraCane), pp. 209–229. London: Springer London, 2008.

[53] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots.* MIT press, 2011.

[54] E. Guizzo, "Three engineers, hundreds of robots, one warehouse," *Spectrum, IEEE*, vol. 45, pp. 26–34, July 2008.

[55] R. Löhner, "On the modeling of pedestrian motion," *Applied Mathematical Modelling*, vol. 34, no. 2, pp. 366 – 382, 2010.

[56] N. Pelechano, K. O'Brien, B. Silverman, and N. Badler, "Crowd simulation incorporating agent psychological models, roles and communication," tech. rep., DTIC Document, 2005.

[57] S. Ali, K. Nishino, D. Manocha, and M. Shah, *Modeling, Simulation and Visual Analysis of Crowds.* Springer-Verlag New York, 2013.

[58] R. L. Hughes, "The flow of human crowds," *Annual Review of Fluid Mechanics*, vol. 35, no. 1, pp. 169–182, 2003.

[59] A. Kormanová *et al.*, “A review on macroscopic pedestrian flow modelling,” *Acta Informatica Pragensia*, vol. 2, no. 2, pp. 39–50, 2014.

[60] W. Daamen and S. Hoogendoorn, “Experimental research of pedestrian walking behavior,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1828, pp. 20–30, 2003.

[61] M. Moussaïd, D. Helbing, and G. Theraulaz, “How simple rules determine pedestrian behavior and crowd disasters,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6884–6888, 2011.

[62] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, “Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions,” *CoRR*, vol. abs/1405.5581, 2014.

[63] P. Henry *et al.*, “Learning to navigate through crowded environments,” in *Robotics and Automation, 2010 IEEE International Conference on*, pp. 981 –986, 2010.

[64] A. Treuille *et al.*, “Continuum crowds,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 1160–1168, 2006.

[65] T. Bandyopadhyay, C. Z. Jie, D. Hsu, M. H. Ang, D. Rus, and E. Frazzoli, *Intention-Aware Pedestrian Avoidance*, pp. 963–977. Heidelberg: Springer International Publishing, 2013.

[66] T. Bandyopadhyay, K. S. Won, E. Frazzoli, D. Hsu, W. S. Lee, and D. Rus, *Intention-Aware Motion Planning*, pp. 475–491. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[67] T. Kruse, A. Kirsch, E. A. Sisbot, and R. Alami, “Exploiting human cooperation in human-centered robot navigation,” in *19th International Symposium in Robot and Human Interactive Communication*, pp. 192–197, Sept 2010.

[68] D. V. Lu and W. D. Smart, “Towards more efficient navigation for robots and humans,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1707–1713, Nov 2013.

[69] P. Trautman and A. Krause, “Unfreezing the robot: Navigation in dense, interacting crowds,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 797–803, 2010.

[70] N. D. M. Ceballos, J. A. Valencia, and N. L. Ospina, *Quantitative performance metrics for mobile robots navigation.*

[71] A. Jacoff, E. Messina, and J. Evans, “Performance evaluation of autonomous mobile robots,” *Industrial Robot: An International Journal*, vol. 29, no. 3, pp. 259–267, 2002.

[72] L. Jin *et al.*, “A sweepline algorithm for euclidean voronoi diagram of circles,” *Computer-Aided Design*, vol. 38, no. 3, pp. 260 – 272, 2006.

[73] M. Huber *et al.*, “Adjustments of speed and path when avoiding collisions with another pedestrian,” *PLoS ONE*, vol. 9, no. 2, p. e89589, 2014.

[74] R. Walker and T. Dodd, “A novel path planning approach for robotic navigation using consideration within crowds,” in *Towards Autonomous Robotic Systems* (C. Dixon and K. Tuyls, eds.), vol. 9287 of *Lecture Notes in Computer Science*, pp. 270–282, Springer International Publishing, 2015.

[75] F. Iida, "Autonomous robots: From biological inspiration to implementation and control. george a. bekey. (2005, mit press.) hardcover, 577 pages. isbn 0262025787," *Artificial Life*, vol. 13, pp. 419–421, Oct 2007.

[76] A. Cangelosi, M. Schlesinger, and L. Smith, *Developmental Robotics: From Babies to Robots.* MIT Press, 2015.

[77] M. Luck and M. d'Inverno, "A formal framework for agency and autonomy," in *Proceedings of the First International Conference on Multi-Agent Systems*, pp. 254—260, AAAI Press/MIT Press, 1995.

[78] R. R. Murphy and D. Schreckenghost, "Survey of metrics for human-robot interaction," in *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, pp. 197–198, March 2013.

[79] M. Wooldridge, N. R. Jennings, *et al.*, "Intelligent agents: Theory and practice," *Knowledge engineering review*, vol. 10, no. 2, pp. 115–152, 1995.

[80] R. A. Clothier, B. Williams, and T. Perez, "A review of the concept of autonomy in the context of the safety regulation of civil unmanned aircraft systems," in *Australian System Safety Conference (ASSC 2013)* (T. Cant, ed.), vol. 151 of *CRPIT*, (Adelaide, Australia), pp. 15–27, ACS, 2013.

[81] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 3, pp. 286–297, 2000.

[82] H.-M. Huang, K. Pavek, J. Albus, and E. Messina, "Autonomy levels for unmanned systems (alfus) framework: an update," vol. 5804, pp. 439–448, 2005.

[83] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," tech. rep., DTIC Document, 1978.

[84] J. Gunderson and L. Gunderson, "Intelligence= autonomy= capability," *Performance Metrics for Intelligent Systems, PERMIS*, 2004.

[85] G. Beavers and H. Hexmoor, "Types and limits of agent autonomy," in *Agents and Computational Autonomy: Potential, Risks, and Solutions* (M. Nickles, M. Rovatsos, and G. Weiss, eds.), vol. 2969 of *Lecture Notes in Computer Science*, pp. 95–102, Berlin, Germany: Springer, 2004. ID number: ISIP:000223492900008.

[86] J. Albus, "Outline for a theory of intelligence," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 21, pp. 473–509, May 1991.

[87] M. M. Veloso, "Entertainment robotics," *Commun. ACM*, vol. 45, pp. 59–63, Mar. 2002.

[88] J.-M. Yu and S.-B. Cho, *Analysis of an Intention-Response Model Inspired by Brain Nervous System for Cognitive Robot*, pp. 168–176. Cham: Springer International Publishing, 2016.

[89] P. Zeno, "Emulating the functionality of rodents' neurobiological navigation and spatial cognition cells in a mobile robot," *International Journal of Computing*, vol. 14, no. 2, 2015.

[90] D. M. Lyons, R. C. Arkin, P. Nirmal, and S. Jiang, “Designing autonomous robot missions with performance guarantees,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2583–2590, IEEE, 2012.

[91] S. Thrun, “Learning metric-topological maps for indoor mobile robot navigation,” *Artificial Intelligence*, vol. 99, no. 1, pp. 21 – 71, 1998.

[92] F. Bonin-Font, A. Ortiz, and G. Oliver, “Visual navigation for mobile robots: A survey,” *Journal of intelligent and robotic systems*, vol. 53, no. 3, pp. 263–296, 2008.

[93] A. K. Gupta and S. K. Arora, *Industrial Automation and Robotics.* University Science Press, 2009.

[94] J. A. Castellanos and J. D. Tardos, *Mobile robot localization and map building: A multisensor fusion approach.* Springer Science & Business Media, 2012.

[95] E. P. Fotiadis, M. Garzón, and A. Barrientos, “Human detection from a mobile robot using fusion of laser and vision information,” *Sensors*, vol. 13, no. 9, pp. 11603–11635, 2013.

[96] L. C. Wang *et al.*, “Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment,” in *Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on*, pp. 821–826, 2002.

[97] S. Böttcher, “Principles of robot locomotion,”

[98] M. Zohaib, M. Pasha, R. Riaz, N. Javaid, M. Ilahi, and R. Khan, “Control strategies for mobile robot with obstacle avoidance,” *arXiv preprint arXiv:1306.1144*, 2013.

[99] G. Deak, K. Curran, and J. Condell, “A survey of active and passive indoor localisation systems,” *Computer Communications*, vol. 35, no. 16, pp. 1939 – 1954, 2012.

[100] M. A. Goodrich, T. W. McLain, J. D. Anderson, J. Sun, and J. W. Crandall, “Managing autonomy in robot teams: observations from four experiments,” in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 25–32, ACM, 2007.

[101] K. S. Barber, A. Goel, and C. E. Martin, “Dynamic adaptive autonomy in multi-agent systems,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 12, no. 2, pp. 129–147, 2000.

[102] M. Kuderer, H. Kretzschmar, and W. Burgard, “Teaching mobile robots to cooperatively navigate in populated environments,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3138–3143, Nov 2013.

[103] Y. Liu and K. M. Passino, “Swarm intelligence: Literature overview,” *Department of Electrical Engineering, the Ohio State University*, 2000.

[104] G. Beni, *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers*, ch. From Swarm Intelligence to Swarm Robotics, pp. 1–9. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[105] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation: A survey,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726 – 1743, 2013.

[106] C. Granata and P. Bidaud, “A framework for the design of person following behaviors for social mobile robots,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4652–4659, Oct 2012.

[107] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori, “On motion planning of mobile robots which coexist and cooperate with human,” in *Intelligent Robots and Systems 95.’Human Robot Interaction and Cooperative Robots’, Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 2, pp. 518–523, IEEE, 1995.

[108] F. Hoeller, D. Schulz, M. Moors, and F. E. Schneider, “Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1260–1265, Oct 2007.

[109] A. Kushleyev and M. Likhachev, “Time-bounded lattice for efficient planning in dynamic environments,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 1662–1668, May 2009.

[110] D. Althoff, J. J. Kuffner, D. Wollherr, and M. Buss, “Safety assessment of robot trajectories for navigation in uncertain and dynamic environments,” *Autonomous Robots*, vol. 32, pp. 285–302, Apr 2012.

[111] E. A. Martinez-Garcia, O. Akihisa, and S. Yuta, “Crowding and guiding groups of humans by teams of mobile robots,” in *IEEE Workshop on Advanced Robotics and its Social Impacts, 2005.*, pp. 91–96, June 2005.

[112] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita, “How to approach humans?-strategies for social robots to initiate interaction,” in *2009 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 109–116, March 2009.

[113] S. Thompson, T. Horiuchi, and S. Kagami, “A probabilistic model of human motion and navigation intent for mobile robot path planning,” in *2009 4th International Conference on Autonomous Robots and Agents*, pp. 663–668, Feb 2009.

[114] J.-C. Latombe, “Robot motion planning, chapter,” 1996.

[115] N. Buniyamin, W. Wan Ngah, N. Sariff, and Z. Mohamad, “A simple local path planning algorithm for autonomous mobile robots,” *International journal of systems applications, Engineering & development*, vol. 5, no. 2, pp. 151–159, 2011.

[116] N. Sleumer, *Exact cell decomposition of arrangements used for path planning in robotics.* Citeseer, 1999.

[117] R. Finkel and J. Bentley, “Quad trees a data structure for retrieval on composite keys,” *Acta Informatica*, vol. 4, no. 1, pp. 1–9, 1974.

[118] D. V. Lu, D. Hershberger, and W. D. Smart, “Layered costmaps for context-sensitive navigation,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 709–715, Sept 2014.

[119] H. Choset, "Robotic motion planning: Roadmap methods." http://www.cs.cmu.edu/~motionplanning/lecture/Chap5-RoadMap-Methods_howie.pdf.

[120] F. Aurenhammer, "Voronoi diagrams&mdash;a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, pp. 345–405, Sept. 1991.

[121] E. Welzl, "Constructing the visibility graph for n-line segments in o(n2) time," *Information Processing Letters*, vol. 20, no. 4, pp. 167 – 171, 1985.

[122] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Algorithmic Foundations of Robotics V*, pp. 43–57, Springer, 2004.

[123] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and autonomous systems*, vol. 8, no. 1, pp. 47–63, 1991.

[124] P. Foo, W. H. Warren, A. Duchon, and M. J. Tarr, "Do humans integrate routes into a cognitive map? map-versus landmark-based navigation of novel shortcuts.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 31, no. 2, p. 195, 2005.

[125] 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.

[126] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, pp. 1179–1187, 1989.

[127] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278–288, 1991.

[128] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics Automation Magazine, IEEE*, vol. 4, pp. 23–33, Mar 1997.

[129] Y.-K. Na and S.-Y. Oh, "Hybrid control for autonomous mobile robot navigation using neural network based behavior modules and environment classification," *Autonomous Robots*, vol. 15, no. 2, pp. 193–206, 2003.

[130] C. W. Warren, "Global path planning using artificial potential fields," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pp. 316–321 vol.1, May 1989.

[131] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 23–32, Feb 1992.

[132] S. Lindemann and S. LaValle, "Smoothly blending vector fields for global robot navigation," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 3553–3559, Dec 2005.

[133] M. P. Georgeff and A. L. Lansky, "Reactive reasoning and planning," in *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 2*, AAAI'87, pp. 677–682, AAAI Press, 1987.

[134] C. Weinrich, M. Volkhardt, E. Einhorn, and H. M. Gross, “Prediction of human collision avoidance behavior by lifelong learning for socially compliant robot navigation,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 376–381, May 2013.

[135] E. DIJKSTRA, “A note on two problems in connexion with graphs.,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[136] S. Russell, *Artificial intelligence [electronic resource] : a Modern Approach.* Pearson Education, 3rd. edition. ed., 2014.

[137] D. Delling, P. Sanders, D. Schultes, and D. Wagner, “Engineering route planning algorithms,” in *Algorithmics of large and complex networks*, pp. 117–139, Springer, 2009.

[138] P. Hart, N. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, pp. 100–107, July 1968.

[139] T. Kruse, P. Basili, S. Glasauer, and A. Kirsch, “Legible robot navigation in the proximity of moving humans,” in *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp. 83–88, May 2012.

[140] C. Lichtenthäler, T. Lorenzy, and A. Kirsch, “Influence of legibility on perceived safety in a virtual human-robot path crossing task,” in *RO-MAN, 2012 IEEE*, pp. 676–681, IEEE, 2012.

[141] Y. Morales, N. Kallakuri, K. Shinozawa, T. Miyashita, and N. Hagita, “Human-comfortable navigation for an autonomous robotic wheelchair,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 2737–2743, IEEE, 2013.

[142] S. J. Russell, “Efficient memory-bounded search methods.,” in *ECAI*, vol. 92, pp. 1–5, 1992.

[143] R. E. Korf, “Depth-first iterative-deepening,” *Artificial Intelligence*, vol. 27, no. 1, pp. 97 – 109, 1985.

[144] D. Harabor and A. Grastien, “Online graph pruning for pathfinding on grid maps,” in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI’11, pp. 1114–1119, AAAI Press, 2011.

[145] X. Sun, S. Koenig, and W. Yeoh, “Generalized adaptive a*,” in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS ’08, (Richland, SC), pp. 469–476, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[146] S. Koenig, M. Likhachev, and D. Furcy, “Lifelong planning a∗,” *Artificial Intelligence*, vol. 155, no. 1, pp. 93 – 146, 2004.

[147] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” *IEEE Transactions on Robotics*, vol. 21, pp. 354–363, June 2005.

[148] I. Pohl, “The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving,” in *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, IJCAI’73, (San Francisco, CA, USA), pp. 12–17, Morgan Kaufmann Publishers Inc., 1973.

[149] R. Kirby, R. Simmons, and J. Forlizzi, “Variable sized grid cells for rapid replanning in dynamic environments,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4913–4918, Oct 2009.

[150] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, “Evaluation of passing distance for social robots,” in *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 315–320, Sept 2006.

[151] M. Luber, L. Spinello, J. Silva, and K. O. Arras, “Socially-aware robot navigation: A learning approach,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 902–907, Oct 2012.

[152] T. Fraichard and H. Asama, “Inevitable collision states. a step towards safer robots?,” in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 388–393 vol.1, Oct 2003.

[153] K. K. Hauser, “Minimum constraint displacement motion planning.,” in *Robotics: Science and Systems*, 2013.

[154] A. Kimmel and K. Bekris, *Decentralized Multi-agent Path Selection Using Minimal Information*, pp. 341–356. Tokyo: Springer Japan, 2016.

[155] S. Reicher, “The psychology of crowd dynamics,” *Blackwell handbook of social psychology: Group processes*, pp. 182–208.

[156] C. G. Keller and D. M. Gavrila, “Will the pedestrian cross? a study on pedestrian path prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, 2014.

[157] T. Kruse, A. Kirsch, H. Khambhaita, and R. Alami, “Evaluating directional cost models in navigation,” in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI ’14, (New York, NY, USA), pp. 350–357, ACM, 2014.

[158] J. Mainprice, R. Hayne, and D. Berenson, “Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 885–892, IEEE, 2015.

[159] E. Papadimitriou *et al.*, “A critical assessment of pedestrian behaviour models,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 3, pp. 242 – 255, 2009.

[160] “Design implications of walking speed for pedestrian facilities,” *Journal of Transportation Engineering*, vol. 137, no. 10, pp. 687–696, 2011.

[161] M. Moussaïd, D. Helbing, S. Garnier, A. Johansson, M. Combe, and G. Theraulaz, “Experimental study of the behavioural mechanisms underlying self-organization in human crowds,” *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 276, no. 1668, pp. 2755–2762, 2009.

[162] D. Helbing, P. Molnar, I. J. Farkas, and K. Bolay, “Self-organizing pedestrian movement,” *Environment and planning B: planning and design*, vol. 28, no. 3, pp. 361–383, 2001.

[163] K. Kitazawa and T. Fujiyama, “Pedestrian vision and collision avoidance behavior: Investigation of the information process space of pedestrians using an eye tracker,” in *Pedestrian and evacuation dynamics 2008*, pp. 95–108, Springer, 2010.

[164] B. A. Schlake, “Mathematical models for pedestrian motion,” 2008.

[165] W. Shao and D. Terzopoulos, “Autonomous pedestrians,” in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 19–28, ACM, 2005.

[166] G. Gigerenzer, “Why heuristics work,” *Perspectives on Psychological Science*, vol. 3, no. 1, pp. 20–29, 2008.

[167] M. Moussaïd, D. Helbing, and G. Theraulaz, “How simple rules determine pedestrian behavior and crowd disasters,” *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 6884–6888, 2011.

[168] P. M. Todd and G. Gigerenzer, “Précis of simple heuristics that make us smart,” *Behavioral and brain sciences*, vol. 23, no. 05, pp. 727–741, 2000.

[169] I. Karamouzas, P. Heil, P. van Beek, and M. H. Overmars, “A predictive collision avoidance model for pedestrian simulation,” in *Motion in Games*, pp. 41–52, Springer, 2009.

[170] N. Pelechano *et al.*, *Virtual Crowds: Methods, Simulation, and Control.* Morgan & Claypool Publishers, 2008.

[171] X. Shi, Z. Ye, N. Shiwakoti, D. Tang, C. Wang, and W. Wang, “Empirical investigation on safety constraints of merging pedestrian crowd through macroscopic and microscopic analysis,” *Accident Analysis and Prevention*, pp. –, 2015.

[172] S. P. Hoogendoorn and P. H. Bovy, “Continuum modeling of multiclass traffic flow,” *Transportation Research Part B: Methodological*, vol. 34, no. 2, pp. 123 – 146, 2000.

[173] P. Wang *et al.*, “Modeling and optimization of crowd guidance for building emergency evacuation,” in *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pp. 328–334, 2008.

[174] N. Bellomo and C. Dogbé, “On the modelling crowd dynamics from scaling to hyperbolic macroscopic models,” *Mathematical Models and Methods in Applied Sciences*, vol. 18, no. supp01, pp. 1317–1345, 2008.

[175] D. Helbing, L. Buzna, A. Johansson, and T. Werner, “Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions,” *Transportation Science*, vol. 39, no. 1, pp. 1–24, 2005.

[176] H. L. Kluepfel, *A cellular automaton model for crowd movement and egress simulation.* PhD thesis, Universität Duisburg-Essen, Fakultät für Physik, 2003.

[177] S. Camazine, *Self-organization in biological systems.* Princeton University Press, 2003.

[178] M. Moussaïd and J. D. Nelson, *Simple Heuristics and the Modelling of Crowd Behaviours*, pp. 75–90. Cham: Springer International Publishing, 2014.

[179] K. Teknomo, “Application of microscopic pedestrian simulation model,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 9, no. 1, pp. 15 – 27, 2006.

[180] “Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions,” *Transportation Science*, vol. 39, no. 1, pp. 1–24, 2005.

[181] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, “Simulation of pedestrian dynamics using a two-dimensional cellular automaton,” *Physica A: Statistical Mechanics and its Applications*, vol. 295, no. 3–4, pp. 507 – 525, 2001.

[182] A. Pluchino, C. Garofalo, G. Inturri, A. Rapisarda, and M. Ignaccolo, “Agent-based simulation of pedestrian behaviour in closed spaces: A museum case study,” *Journal of Artificial Societies and Social Simulation*, vol. 17, no. 1, p. 16, 2014.

[183] N. Ronald, L. Sterling, M. Kirley, *et al.*, “An agent-based approach to modelling pedestrian behaviour,” *International Journal of Simulation 8.1*, pp. 25–38, 2007.

[184] C. Loscos, D. Marchal, and A. Meyer, “Intuitive crowd behaviour in dense urban environments using local laws,” in *Proceedings of the Theory and Practice of Computer Graphics 2003*, TPCG ’03, (Washington, DC, USA), pp. 122–, IEEE Computer Society, 2003.

[185] D. Helbing and P. Molnár, “Social force model for pedestrian dynamics,” *Phys. Rev. E*, vol. 51, pp. 4282–4286, 1995.

[186] D. Helbing, “Boltzmann-like and boltzmann-fokker-planck equations as a foundation of behavioral models,” *Physica A: Statistical Mechanics and its Applications*, vol. 196, no. 4, pp. 546–573, 1993.

[187] Y. Tamura, T. Fukuzawa, and H. Asama, “Smooth collision avoidance in human-robot coexisting environment,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3887–3892, IEEE, 2010.

[188] J. Van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1928–1935, IEEE, 2008.

[189] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, “Planning-based prediction for pedestrians,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3931–3936, IEEE, 2009.

[190] A. Asahara, K. Maruyama, A. Sato, and K. Seto, “Pedestrian-movement prediction based on mixed markov-chain model,” in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 25–33, ACM, 2011.

[191] H. Kretzschmar, M. Kuderer, and W. Burgard, “Inferring navigation policies for mobile robots from demonstrations,”

[192] T. Hamada, T. Hagiwara, T. Teramoto, S. Morishita, M. Umetsu, and M. Ohgama, *A Three-Dimensional Pedestrian-Flow Simulation for High-Rising Buildings*, pp. 417–424. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[193] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, “Socially compliant mobile robot navigation via inverse reinforcement learning,” *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[194] K. Konolige, E. Marder-Eppstein, and B. Marthi, “Navigation in hybrid metric-topological maps,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 3041–3047, May 2011.

[195] A.-H. Olivier, A. Marin, A. Crétual, A. Berthoz, and J. Pettré, “Collision avoidance between two walkers: Role-dependent strategies,” *Gait & Posture*, vol. 38, no. 4, pp. 751 – 756, 2013.

[196] M. Gérin-Lajoie, C. L. Richards, and B. J. McFadyen, “The negotiation of stationary and moving obstructions during walking: Anticipatory locomotor adaptations and preservation of personal space,” *Motor Control*, vol. 9, no. 3, pp. 242–269, 2005.

[197] P. Basili, M. Sağlam, T. Kruse, M. Huber, A. Kirsch, and S. Glasauer, “Strategies of locomotor collision avoidance,” *Gait & Posture*, vol. 37, no. 3, pp. 385 – 390, 2013.

[198] Q.-C. Pham, H. Hicheur, G. Arechavaleta, J.-P. Laumond, and A. Berthoz, “The formation of trajectories during goal-oriented locomotion in humans. ii. a maximum smoothness model,” *European Journal of Neuroscience*, vol. 26, no. 8, pp. 2391–2403, 2007.

[199] Q.-C. Pham and H. Hicheur, “On the open-loop and feedback processes that underlie the formation of trajectories during visual and nonvisual locomotion in humans,” *Journal of Neurophysiology*, vol. 102, no. 5, pp. 2800–2815, 2009.

[200] 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.

[201] M. A. Goodrich and A. C. Schultz, “Human-robot interaction: A survey,” *Found. Trends Hum.-Comput. Interact.*, vol. 1, pp. 203–275, Jan. 2007.

[202] A. M. Turing, “Computing machinery and intelligence,” *Mind*, pp. 433–460, 1950.

[203] D. Feil-Seifer and M. Mataric, “Defining socially assistive robotics,” in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*, pp. 465–468, June 2005.

[204] W.-L. Chang, S. Sabanovic, and L. Huber, “Observational study of naturalistic interactions with the socially assistive robot paro in a nursing home,” in *Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, pp. 294–299, Aug 2014.

[205] M. J. Matarić, J. Eriksson, D. J. Feil-Seifer, and C. J. Winstein, “Socially assistive robotics for post-stroke rehabilitation,” *Journal of NeuroEngineering and Rehabilitation*, vol. 4, no. 1, p. 1, 2007.

[206] S. Shamsuddin, H. Yussof, L. Ismail, F. A. Hanapiah, S. Mohamed, H. A. Piah, and N. I. Zahari, “Initial response of autistic children in human-robot interaction therapy with humanoid robot nao,” in *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on*, pp. 188–193, IEEE, 2012.

[207] D. Feil-Seifer, K. Skinner, and M. J. Mataric, “Benchmarks for evaluating socially assistive robotics,” *Interaction Studies*, vol. 8, no. 3, pp. 423–439.

[208] T. Fong, I. Nourbakhsh, and K. Dautenhahn, “A survey of socially interactive robots,” *Robotics and autonomous systems*, vol. 42, no. 3, pp. 143–166, 2003.

[209] J. Young, J. Sung, A. Voida, E. Sharlin, T. Igarashi, H. Christensen, and R. Grinter, “Evaluating human-robot interaction,” *International Journal of Social Robotics*, vol. 3, no. 1, pp. 53–67, 2011.

[210] K. Dautenhahn, “Socially intelligent robots: dimensions of human–robot interaction,” *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 362, no. 1480, pp. 679–704, 2007.

[211] B. D. Argall and A. G. Billard, “A survey of tactile human–robot interactions,” *Robotics and Autonomous Systems*, vol. 58, no. 10, pp. 1159 – 1176, 2010.

[212] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, “Common metrics for human-robot interaction,” in *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 33–40, ACM, 2006.

[213] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, G. Lakemeyer, C. Rosenberg, N. Roy, D. Schulz, D. Schulz, and W. Steiner, “Experiences with two deployed interactive tour-guide robots,” in *In Proceedings of the International Conference on Field and Service Robotics*, 1999.

[214] N. Hawes, C. Burbridge, F. Jovan, *et al.*, “The STRANDS project: Long-term autonomy in everyday environments,” *CoRR*, vol. abs/1604.04384, 2016.

[215] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, “Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions,” *CoRR*, vol. abs/1405.5581, 2014.

[216] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, “Intention-aware online pomdp planning for autonomous driving in a crowd,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 454–460, May 2015.

[217] C.-P. Lam, C.-T. Chou, K.-H. Chiang, and L.-C. Fu, “Human-centered robot navigation - towards a harmoniously human-robot coexisting environment,” *Robotics, IEEE Transactions on*, vol. 27, pp. 99–112, Feb 2011.

[218] E. Sisbot and othersl, “A human aware mobile robot motion planner,” *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 874–883, 2007.

[219] C. Lichtenthäler and A. Kirsch, “Towards legible robot navigation-how to increase the intend expressiveness of robot navigation behavior,” in *Int. Conf. Soc. Robot. Embodied Commun. Goals Intentions*, 2013.

[220] M. Beetz, F. Stulp, P. Esden-Tempski, A. Fedrizzi, U. Klank, I. Kresse, A. Maldonado, and F. Ruiz, “Generality and legibility in mobile manipulation,” *Autonomous Robots*, vol. 28, p. 21, Sep 2009.

[221] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. Di Caro, “Human-friendly robot navigation in dynamic environments,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 423–430, IEEE, 2013.

[222] D. Bortot, M. Born, and K. Bengler, "Directly or on detours? how should industrial robots approximate humans?," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 89–90, March 2013.

[223] L. Takayama, D. Dooley, and W. Ju, "Expressing thought: Improving robot readability with animation principles," in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 69–76, March 2011.

[224] A. Kirsch, T. Kruse, E. A. Sisbot, R. Alami, M. Lawitzky, D. Brščić, S. Hirche, P. Basili, and S. Glasauer, "Plan-based control of joint human-robot activities," *KI - Künstliche Intelligenz*, vol. 24, pp. 223–231, Sep 2010.

[225] A. Kirsch, T. Kruse, and L. Mösenlechner, "An integrated planning and learning framework for human-robot interaction," in *4th Workshop on Planning and Plan Execution for Real-World Systems*, 2009.

[226] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Toward human-aware robot task planning.," 2006.

[227] F. Dehais, E. A. Sisbot, R. Alami, and M. Causse, "Physiological and subjective evaluation of a human–robot object hand-over task," *Applied Ergonomics*, vol. 42, no. 6, pp. 785 – 791, 2011.

[228] E. A. Sisbot, A. Clodic, R. Alami, and M. Ransan, "Supervision and motion planning for a mobile manipulator interacting with humans," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 327–334, March 2008.

[229] C. L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami, "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pp. 371–377, Aug 2005.

[230] E. A. Sisbot and R. Alami, "A human-aware manipulation planner," *IEEE Transactions on Robotics*, vol. 28, pp. 1045–1057, Oct 2012.

[231] P. Basili, M. Huber, O. Kourakos, T. Lorenz, T. Brandt, S. Hirche, and S. Glasauer, "Inferring the goal of an approaching agent: A human-robot study," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 527–532, Sept 2012.

[232] R. A. Knepper and M. T. Mason, "Path diversity is only part of the problem," in *2009 IEEE International Conference on Robotics and Automation*, pp. 3224–3229, May 2009.

[233] M. Karg, M. Sachenbacher, and A. Kirsch, "Towards expectation-based failure recognition for human robot interaction," in *Proceedings of 22nd International Workshop on Principles of Diagnosis (DX-2011)*, vol. 15, p. 128, 2011.

[234] M. Heerink, B. Krose, V. Evers, and B. Wielinga, "Assessing acceptance of assistive social agent technology by older adults: the almere model," vol. 2, pp. 361–375, 12 2010.

[235] A. H. Maslow, "A theory of human motivation," *Readings in managerial psychology*, vol. 20, 1989.

[236] C. Bartneck, E. Croft, and D. Kulic, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 71–81, 2009.

[237] H. A. Yanco and J. Drury, "Classifying human-robot interaction: an updated taxonomy," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, vol. 3, pp. 2841–2846 vol.3, Oct 2004.

[238] A. Watanabe, T. Ikeda, Y. Morales, K. Shinozawa, T. Miyashita, and N. Hagita, "Communicating robotic navigational intentions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5763–5769, Sept 2015.

[239] F. J. Diedrich and W. H. Warren Jr, "Why change gaits? dynamics of the walk-run transition.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, no. 1, p. 183, 1995.

[240] D. Helbing, "Models for Pedestrian Behavior," *eprint arXiv:cond-mat/9805089*, 1998.

[241] C. Young, *Precalculus*. John Wiley & Sons, 2010.

[242] "Chi-square distribution table." http://sites.stat.psu.edu/~mga/401/tables/Chi-square-table.pdf. Accessed: 15-01-2014.

[243] D. Althoff, J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Autonomous Robots*, vol. 32, no. 3, pp. 285–302, 2012.

[244] T. Fraichard, "A short paper about motion safety," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1140–1145, April 2007.

[245] T. Ikeda, Y. Chigodo, D. Rea, F. Zanlungo, M. Shiomi, and T. Kanda, "Modeling and prediction of pedestrian behavior based on the sub-goal concept," *Robotics*, p. 137, 2013.

[246] 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, 2013. 2nd Conference of Transportation Research Group of India (2nd CTRG).

[247] R. Kirby, R. Simmons, and J. Forlizzi, "Companion: A constraint-optimizing method for person-acceptable navigation," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pp. 607–612, Sept 2009.

[248] D. Helbing, I. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," vol. 21, pp. 21–58, 01 2002.

[249] D. Silver, "Cooperative pathfinding," in *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'05, pp. 117–122, AAAI Press, 2010.

[250] H. Hüttenrauch, K. S. Eklundh, A. Green, and E. A. Topp, "Investigating spatial relationships in human-robot interaction," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5052–5059, IEEE, 2006.

[251] I. Z. Emiris *et al.*, “Exact voronoi diagram of smooth convex pseudo-circles: General predicates, and implementation for ellipses,” *Computer Aided Geometric Design*, vol. 30, no. 8, pp. 760 – 777, 2013.

[252] M. I. Shamos and D. Hoey, “Geometric intersection problems,” in *Foundations of Computer Science, 1976., 17th Annual Symposium on*, pp. 208–215, Oct 1976.

[253] S. Fortune, “A sweepline algorithm for voronoi diagrams,” in *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG ’86, (New York, NY, USA), pp. 313–322, ACM, 1986.

[254] A. Dobrin, “A review of properties and variations of voronoi diagrams,”

[255] H. S. M. Coxeter, “The problem of apollonius,” *The American Mathematical Monthly*, vol. 75, no. 1, pp. 5–15, 1968.

[256] A. Koestler and T. Bräunl, “Mobile robot simulation with realistic error models,” vol. 1, pp. 46–51, 2004.