



A human behavioral pedestrian simulation model with reinforcement learning approach

a Dissertation Submitted to the
GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF THE
SHIBAURA INSTITUTE OF TECHNOLOGY

by

TRINH THANH TRUNG

Student ID: nb18503

in Partial Fulfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

SEPTEMBER 2021

Acknowledgments

The completion of this dissertation could not be possible without the active supports of Shibaura Institute of Technology. This is even more meaningful throughout the distressing period of a global pandemic. I greatly appreciate all the help and assistance that I have received.

Most importantly, I am sincerely grateful to my supervisor, Professor Masaomi Kimura, for the tremendous assistance and encouragement. Your devotion has truly helped me not only in shaping this dissertation but also in getting through difficult periods in my academic journey. Your aspiring guidance and constructive advice are invaluable to me.

In addition, I would like to express my gratitude to all of my friends in Japan, especially to those who have helped me when I first came here. Your supports have given me a much-needed boost for my life abroad.

I am also thankful for the country and the people of Japan. You have given me a wonderful and unforgettable experience here. There are a lot of things to love about Japan, like the culture, the foods, and the sceneries, aside from occasional natural disasters.

Finally, I am wholeheartedly grateful for my wife and son to be here with me. You are my motivation and the reason for this dissertation to be fully accomplished.

SHIBAURA INSTITUTE OF TECHNOLOGY

Abstract

Graduate School of Engineering and Science
Division of Functional Control System

Doctor of Philosophy

by Trinh Thanh Trung

Pedestrian simulation has a significant role thanks to its contributions in many research fields, including robotics, human safety, and urban planning. However, perfectly simulating pedestrian behavior is difficult because of the complexity of the human cognition system. This complexity causes many problems, such as cognitive bias or human mistakes, which could not be achieved by using an optimization method. Many pedestrian simulation models approach the problem by using an empirical model, often with force-based or rule-based methods. While these approaches could provide believable results in common situations, it does not always resemble natural pedestrian navigation behavior in certain settings. To improve the replicated behavior of the pedestrian, the simulation model needs to consider the ideas in human factors and human cognition.

We proposed a model to simulate pedestrian navigation by adopting several concepts of the human cognitive system in behavioral science combined with reinforcement learning. The proposed model was correspondingly designed consisting of two tasks: a pedestrian path-planning task to simulate the navigation planning process in the pedestrian's mind, and a pedestrian interacting task to replicate the

interaction between the pedestrian and another obstacle while following the planned navigation. For a more realistic human behavior, we also suggested a prediction method based on the predictive process in human cognition.

In addition, risk assessment of the obstacle's danger is another focus in this dissertation. While this process could substantially affect how a pedestrian navigates, this problem is often overlooked in other studies. In our research, we have addressed the risk determination mechanism by humans and its effect on the pedestrian's navigation. Based on that, risk assessment methods were modeled and incorporated extensively in many aspects of our behavioral pedestrian simulation model.

The empirical result demonstrates a highly realistic human behavior of pedestrian interactions, which resembles actual situations in real life. The simulated pedestrian actions share many similarities with a human pedestrian in several aspects such as following common walking conventions and human behaviors.

Contents

Acknowledgments	i
Abstract	iii
List of Figures	vii
List of Tables	xi
List of Abbreviations	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	3
1.3 Human factors and human cognition	4
1.4 Reinforcement learning	7
1.5 Our contributions	8
1.6 Outline	9
2 Related works	11
2.1 In pedestrian simulation	11
2.2 In pedestrian prediction	12
2.3 In navigation behavior	13
2.4 In reinforcement learning	13
3 Background	15
3.1 Reinforcement learning	15
3.2 PPO algorithm	16

4 Behavioral pedestrian simulation model	21
4.1 Cognitive system in navigation	21
4.2 Model overview	26
4.3 Pedestrian decision planner	28
4.4 Obstacle’s danger and risk	32
5 Pedestrian path-planning	34
5.1 Introduction	34
5.2 Model overview	36
5.3 Path-planning navigation training	38
5.3.1 Environment modeling	39
5.3.2 Agent’s observations and actions	41
5.3.3 Rewarding formulation	42
5.4 Point-of-conflict prediction	47
5.4.1 Single diagonal movement obstacle	48
5.4.2 Pedestrian obstacle	50
5.5 Risk assessment	52
5.6 Implementations	54
5.7 Discussion	67
5.8 Summary	69
6 Pedestrian interacting model	70
6.1 Introduction	70
6.2 Model overview	72
6.3 Pedestrian interaction learning	73
6.3.1 Environment modeling	73
6.3.2 Agent’s observations and actions	75
6.3.3 Rewarding behavior	77
6.4 Prediction task	80
6.4.1 Estimation	83
6.4.2 Assessment	84
6.4.3 Prediction	86
6.5 Implementation and discussion	87
6.6 Summary	94
7 Discussion	95

8 Conclusion and Future Work	98
8.1 Summary of the model	98
8.2 Conclusion	99
8.3 Future work	100
8.3.1 Considering the development of humans	100
8.3.2 Approaching reinforcement learning using concepts in neu- rosience	101
8.3.3 Designing a cognitive decision planner	101
8.3.4 Increasing the number of obstacles	102
Bibliography	111

List of Figures

1.1	Outline of the chapter structure.	10
3.1	Overview of a reinforcement learning model. [21]	16
3.2	Clipping method in PPO's loss function calculation. [22]	18
3.3	The neural network structure in path-planning model.	19
4.1	Overview of the human cognitive system in navigation.	22
4.2	The hippocampus and the related regions inside the human brain.	24
4.3	The anatomy of the reinforcement learning process with the basal ganglia, based on the structure suggested by Ludvig et al. [56] . .	26
4.4	Flow chart of the decision planner task.	30
4.5	Example of the timing when the decision planner is called.	31
4.6	Path planned by an agent with different obstacle's danger.	33
5.1	Overview of the pedestrian path-planning model.	37
5.2	Path-planning environment modeling	39
5.3	Determining the advantage value \hat{A}_t	46
5.4	Obstacle avoidance with point-of-conflict.	48
5.5	Point-of-conflict of a single diagonal movement obstacle.	49
5.6	Point-of-conflict of a pedestrian obstacle.	51
5.7	Path-planning task implementation screenshot.	54
5.8	Cumulative reward statistics.	55
5.9	Screenshot from path-planning model experimental dataset.	56

5.10	Agent’s planned path in different situations: (a) no obstacle; (b) with a static obstacle with a low danger level; (c) with a static obstacle with a high danger level; (d) with an obstacle moving straight in one direction away from the agent (e) with an obstacle moving straight in one direction toward the agent; (f) with a pedestrian obstacle.	57
5.11	Comparison with SFM in different situations: (a) no obstacle; (b) with a static obstacle; (c) with a moving obstacle; (d) with a pedestrian obstacle	58
5.12	Questionnaire used to assess the human likeness of the implemented models.	60
5.13	Implementations of different coefficient sets: (a) default; (b) high priority on <i>shortest path</i> ; (c) low priority on <i>obstacle avoidance</i>	63
5.14	Component reward values during training.	65
5.15	Adjusted component reward values during training.	66
6.1	Pedestrian interacting environment setting.	73
6.2	Learning task training environment.	73
6.3	Agent’s destination as a sub-goal.	74
6.4	Training statistics for radial and Euclidean coordinate methods.	76
6.5	The problems with the position forwarding prediction model.	81
6.6	Prediction task model.	82
6.7	Prediction process flowchart.	83
6.8	Plot of the function $\varepsilon = f\left(\frac{\delta_e}{D}\right)$	85
6.9	Resulted prediction with different θ	86
6.10	A screenshot from the implementation application.	88
6.11	Learning task training statistics.	88
6.12	Screenshot from interacting model experimental dataset.	89
6.13	Example interacting situations between agent and obstacle in comparison with Social Force Model and Unity NavMesh. The green circle represents the agent and the red circle represents the obstacle.	91

List of Tables

5.1	Number of people favoring each model's implementation.	62
5.2	Total scores awarded to each model's implementation.	62
5.3	Coefficient parameter value.	63
6.1	Comparisons with Social Force Model and Unity NavMesh in average length, time and collisions.	93

List of Abbreviations

RL	Reinforcement Learning
PPO	Proximal Policy Optimization
TRPO	Trust Region Policy Optimization
AI	Artificial Intelligence
SFM	Social Force Model
CNN	Convolutional Neural Network
GDPM	Gaussian Process Dynamical Models
MDP	Markov Decision Process
POC	Point-of-conflict
GO	Goal Optimization
NB	Natural Behavior

Chapter 1

Introduction

In this chapter, the motivation for this dissertation is presented. Subsequently, the related concepts in this study, including human factors, human cognition and reinforcement learning, are introduced. Lastly, the contribution of the study is described, followed by the outline of the dissertation.

1.1 Motivation

Simulation of pedestrian movement is a topic of great interest to many researchers thanks to a large number of its application domains. An example of this is the applications in the robotic field, which aim to replicate human navigation behaviors. This could be remarkably beneficial in the future, where robots could navigate among humans and actively assist people in many different tasks. Pedestrian safety is also another critical aim in pedestrian movement simulation. These studies address the assurance of safety for pedestrians by, for example, simulation of multiple pedestrian movements to ensure no harm could be induced. For instance, many studies of pedestrian simulation for evacuation activities have been beneficial to the design in safety features of construction projects [1]. Another example is the studies in pedestrian behavior, which are crucial for urban planning and landscape design [2, 3]. Recently, along with the rising trend of autonomous vehicles, pedestrian simulation studies have attracted increasing interest, especially in the situation of crossing with vehicles, to avoid possible fatal accidents [4, 5].

However, simulating the navigation of pedestrians is a highly complicated task. While these studies could construct a sufficient reproduction of the pedestrian navigation behavior in certain applications, for example, the robot movement in pedestrian roads, their approaches might not be able to provide a realistic behavior needed for some research, in risk and safety problems for instance. This is because the goal of a navigation model in robotics is to create a robust and efficient movement that is deemed safe and comfortable by humans, which does not require an accurate replication of human navigation. For a human-like behavior, there are many problems that need to be addressed whilst modeling the navigation. One of the most challenging problems is the unpredictable nature of human behavior. Upon different circumstances or states, subjectively or objectively, a person could behave in totally different ways. In the case of pedestrian navigation, for instance, the route chosen by a pedestrian could significantly be altered by a subtle gesture or signal from another pedestrian. Humans also tend to behave differently when they are in different social situations, such as going along with a friend or in a group [24]. Another problem is that differences in regulations and cultures also contribute to the way pedestrians navigating in the environment. There are certain behaviors considered to be normal in one region that could be recognized as inappropriate in another. Such behaviors are usually insufficiently researched, thus formulating these behaviors in a navigation model could be greatly demanding, as a consequence.

Because of this, research in pedestrian navigation simulation has been greatly active, addressing different problems of the simulation. A great number of approaches have been used for the problems. Many studies tried to replicate the abstract navigation of pedestrians by implementing various empirical models, which often employ various physics-based methods such as force-based and fluid dynamics [8] to realize the pedestrian's movement. The basic idea of these models is that pedestrian agents are attracted to a specific point-of-interest (e.g. pedestrian's destination) and repulsed from possible collisions (e.g. walls, obstacles, and other agents). The representation of the force-based models is similar to the interactions between magnetic objects with some certain improvements. These methods have some certain resemblance in basic movement and collision avoidance in some applications. However, in many circumstances, their movement implementations might be too generic and do not depict actual human-like interactions. For instance, when an agent plans a path to go from its current position

to a destination, a force-based agent often chooses the shortest path without colliding with other obstacles most of the time. In real life, a human pedestrian has many other aspects affecting his decision such as social comfort, law compliance, or his personal emotion.

1.2 Objectives

In this study, we concentrate on simulating pedestrian behaviors in a *microscopic* setting. Regarding the scale of the research in pedestrian simulation, most studies in pedestrian simulation are often categorized into three levels of interaction: *macroscopic*, *mesoscopic*, and *microscopic* [32]. The macroscopic simulation models often use the concept of fluid and particles originated from physics to construct pedestrian navigations while ignoring the interactions between pedestrians as well as individual characteristics of each pedestrian. For an excessively high-density crowd, a macroscopic model could be sufficient; however, for a smaller size of pedestrians where social interactions are essential, a mesoscopic or microscopic model would be more suitable. A mesoscopic model sits between macroscopic and microscopic, which is still able to simulate a relatively large-sized environment but with the cost of the agent's movements and interactions. Compared to mesoscopic, a microscopic model is more realistic as each pedestrian is considered as an independent object or a computer agent whose behaviors and thinking processes could be modeled upon.

Specifically, we try to replicate the navigation mechanism of pedestrians when planning a path to their destination while taking avoiding obstacles and interacting with other pedestrians into account. Many real-life situations require this problem to be resolved, for example, preparing the necessary safety precautions for an infrastructure project, like an apartment complex or a shopping mall, accurate pedestrian behavior needs to be precisely simulated. By reflecting the human behavior in the navigation around an area, which areas the pedestrian could look at and where the pedestrian would most likely to reach could be observed, therefore potential risks could be early detected and eliminated. This could also benefit other related activities, such as placing notices for citizens or advertisement placements.

Addressing the obstacle’s risk and danger is also a factor that is often overseen by many studies. Although the majority of research in pedestrian simulation considers the obstacle in collision avoidance, to our best knowledge, not many studies have addressed how its danger affects the pedestrian’s choice. For the papers that discuss this problem [39], the models proposed are quite limited in using the empirical approach without considering the human cognitive factors. The results of these models could be consequently insufficient, especially in the case the danger of the obstacle greatly alters the path choice of the pedestrian. For safety-focused applications, this problem could produce undesirable results, possibly causing significant consequences as a result.

For that reason, our objective is to design a pedestrian model that is able to construct a natural navigation behavior. The natural behavior in our model is defined as realistic and human-like navigation when assessed by human observers. Because there are a great variety of navigating traits that could be considered natural by humans, the navigation behavior in our pedestrian model does not have to replicate the exact characteristics of a specific human behavior. Additionally, unlike the approaches from other studies, the behavior is not formulated by optimizing certain factors, because we believe that optimization may lead to a less human-like pedestrian demeanor.

Consequently, we proposed a novel cognitive pedestrian simulation model considering the obstacle’s danger and risk assessment while taking account of human cognitive factors. Our model adopts the concept of deep reinforcement learning, a neural network-based machine learning technique, for the training of the pedestrian agent. The approach has many similarities with the mechanism of human cognition. Deep reinforcement learning approaches also employ artificial neural networks, which were inspired by the mechanisms of the biological neural network in the human brain. Thanks to that, the aspects in obstacle’s danger and risk assessment are further explored in a similar mean as to how humans address dangers in real life.

1.3 Human factors and human cognition

To have a better understanding of human behavior, particularly in pedestrian navigation, the aspects of human factors and human cognition need to be con-

sidered. The study of human factors is a research domain that focuses on the psychological, social, physical, and biological characteristics of humans in interacting with others. In human navigation, in particular, many factors could affect walking behavior. For instance, age and gender play a significant role in how a person navigates [77]. For instance, old people are often less confident than younger people, thus often focusing on safety when navigating. On the other hand, young people are usually more confident and focus more on the efficiency of the navigation, although this could lead to more accidents. Male pedestrians also have been proven to have higher confidence in navigating compared to female pedestrians. Children, while generally less confident than adults, they also lack the ability to accurately identify danger, which may lead to a higher chance of an accident. On the other hand, there is no difference between male and female children. These concepts like confidence and different priorities in navigation are essential for a behavioral pedestrian model.

Regarding human cognition, this is the concept of information processing in humans. The process is carried out by the cognitive system, the thinking process inside the human brain, which is responsible for the decision-making process of everyday tasks. The human thinking process is remarkably complex and difficult to be analyzed. There have been many studies in different research domains, including behavioral psychology and cognitive science, conducted to have a better understanding of the cognitive system. Every moment, an enormous amount of information surges into the human brain, and the decision needs to be promptly made. Most of the time, many of the choices are made by the human forming a “heuristic shortcut” to quickly form the decision. As a result, occasionally the decisions are non-optimized, inaccurate, and even completely wrong. These types of faults are called cognitive bias, which is a concept coined by Tversky and Kahneman [72]. The number of cognitive bias types is highly diverse, however, they could be sorted into three main categories: attention bias, memory bias, and judgment bias [41]. By understanding the ideas behind the human cognitive system, such as how a certain bias is made, to replicate in a human behavior model, the model’s performance could be improved. The approach of considering concepts in cognitive science has been addressed in other studies [29, 42, 43], which have achieved good results. However, this approach has not been properly considered in studies of pedestrian models.

As a result, we also need to consider the mechanisms of the cognitive system in the pedestrian navigation problem. More specifically, the concepts such as cognitive maps, spatial knowledge, and goal-oriented planning are the contributing factors in the decision process of humans in navigation. An adequate navigation model should consider these factors and is able to incorporate these concepts in its realization. The realization of these concepts could be different between models as each model would have a different approach to streamline the complicated human decision-making process. This could be even more demanding if the model does not have an appropriate thinking mechanism, such as the force-based or fluid dynamic models.

Among the elements within the cognitive system, cognitive prediction is one of the most significant factors in human navigation. This is also the main cause of many human biases [18]. To reduce the cognitive load, many processes inside the human brain often employ a prediction method to help with decision making. For example, when a pedestrian is navigating, most future states of the environment would be forecasted, such as where the other pedestrians are going or will the traffic light turn red. Another example is when a person wants to get to a specific destination, he would always try to address the concerns like is it going to be more inconvenient if a certain path is chosen. For various reasons, making incorrect predictions is common in practical situations. As a result, we explored how the agent could incorporate the cognitive prediction into its navigation behavior, which we believe could improve the resemblance of the pedestrian interaction in real life. The difference between this and the prediction in many studies is that, while these studies aim at the accuracy of the prediction, the focus of our research is to imitate the prediction in the human cognitive process. This may lead to a less optimized navigation route or accurate prediction, but this would be closer to real-life human behaviors.

Not every decision that resulted from heuristic shortcuts leads to error or suffers from cognitive bias. Humans, similar to many other animals, have a “trial-and-error” learning mechanism based on the feedback they receive [41]. In the cases that the decisions result in getting good feedbacks, people would embrace that decision and would make the same decision, which means using the same heuristic shortcut in the future and vice versa. This is also the cause of many types of human bias, as in real life, the feedbacks are often greatly lacking to reinforce the correct behavior. For that reason, we expect that by using reinforcement

learning, with similar feedback as in actual situations, a more realistic navigation behavior would be modeled.

1.4 Reinforcement learning

Similar to a concept of the same name in behavioral psychology, reinforcement learning is a machine learning paradigm in which the agent gradually learns to interact with the environment via trial-and-error progression, in which the learner needs to find the appropriate actions in the current state for an optimum reward. For every action taken by a person, he will get some feedback called reward from the environment. Depending on the reward, which can be either positive or negative, the actions leading to that reward would be encouraged or avoided, respectively.

Recently, several reinforcement learning techniques that incorporate neural network utilization have been proposed. Neural network is a concept in artificial intelligence, which is considerably adopted in many deep learning techniques. Like biological neural networks, an artificial neural network consists of multiple nodes or neurons. In human cognition, these neurons help to analyze and categorize all sorts of problems, which in turn support the person in most decisions. Correspondingly, the neurons in an artificial neural network are also organized into layers, which also helps to solve many machine learning technique problems.

This is particularly similar to how children learn to navigate. Other than learning to reach a destination, they also need to learn to walk in the right way and avoid other obstacles. The instructions come from encouragements as well as punishments from different people, which resemble the reward signals in reinforcement learning. As an example, in the path-planning task, the child needs to plan a path to the destination. If he feels uncomfortable with his decision, because of taking a longer path or colliding with obstacles, for instance, he will then receive a negative reward and will try to improve his behavior. As a result, once an environment is observed, he will be able to come up with a path using his current optimum policy without the need for various calculations such as “forces” realized in many microscopic pedestrian models. Although the neural network used in a machine learning program is much less developed compared to even a child’s brain, a reinforcement learning technique could benefit from

much higher training scenarios compared to actual human beings. For example, a child could learn to reach the correct destination after several tries, it could take a reinforcement learning agent a few minutes to learn through millions of states of the environment. For that reason, the neural network could still learn to accomplish the equivalent task despite the limitation in its network structure.

1.5 Our contributions

The main contributions of this study are the design and formulation of a novel pedestrian simulation model. The model concentrates on replicating the mechanism a pedestrian decides how to navigate inside the environment, considering the risk from the nearby obstacle while conforming to the natural human behavior. To do this, the agent needs to appropriately choose between planning a task to its destination and interact with the closing pedestrians or obstacles.

As a result, our proposed pedestrian simulation model consists of:

- A novel pedestrian path-planning model using reinforcement learning. This model replicates the mechanism an agent observes the environment and plans a path to the destination. The agent needs to do this under the consideration of the risk from the environment, such as an obstacle or other pedestrians. Reinforcement learning is employed to train the agent's navigation planning using rewarding based on the concept of human comfort.
- A model to simulate the interaction between a pedestrian agent with another obstacle or pedestrian. This interaction occurs when the pedestrian is particularly close to another pedestrian or an obstacle, and the pedestrian must react appropriately based on the intermediate actions of the others. Similarly, reinforcement learning is also used for the training of the agent for interaction behavior. In addition to that, a cognitive prediction model was proposed using a continuous interpolation method combined with the concept of prediction in the human cognition system.
- A perpetual task controlling model. This model assesses the current situation to decide when to use which task, the path-planning or the interacting one. This is realized by a modest rule-based system, which is continuously carried out as the pedestrian navigating in the environment.

The implementation of the model is capable of replicating real-life situations, in which the pedestrian agents could perform natural behaviors in path-planning and interacting with other pedestrians. The resulted behavior of the agent shares many similarities with a human pedestrian, conforming to social rules and regulations.

1.6 Outline

The remainder of this dissertation is organized as follows:

Chapter 2 comprises the literature review of other related studies in different areas. Firstly, the prevalent or recent significant approaches in pedestrian simulation scope are explored. In addition, studies in pedestrian prediction are reviewed. The chapter also covers the literature review of several studies on human behavior and human cognition. The survey of other research in pedestrian simulation using reinforcement learning is also presented in the chapter.

Chapter 3 presents the main background concepts of this dissertation. This includes reinforcement learning and the PPO algorithm.

Chapter 4 introduces various concepts of the cognitive system in navigation, particularly those inspire the design of our model. Subsequently, the chapter demonstrates the overview of the model, followed by our realization of the decision planner in our behavioral pedestrian model. The concept of risk and danger is also covered in this chapter.

Subsequently, Chapter 5: Path-planning model and Chapter 6: Pedestrian interacting model are presented to demonstrate the detailed model of each task in the pedestrian agent's navigation. In each chapter, the corresponding methodology is described, followed by its implementation and evaluation.

The discussion of the overall model is given in Chapter 7. Finally, the dissertation is concluded in Chapter 8.

The outline of the dissertation's chapter structure is presented in Figure 1.1

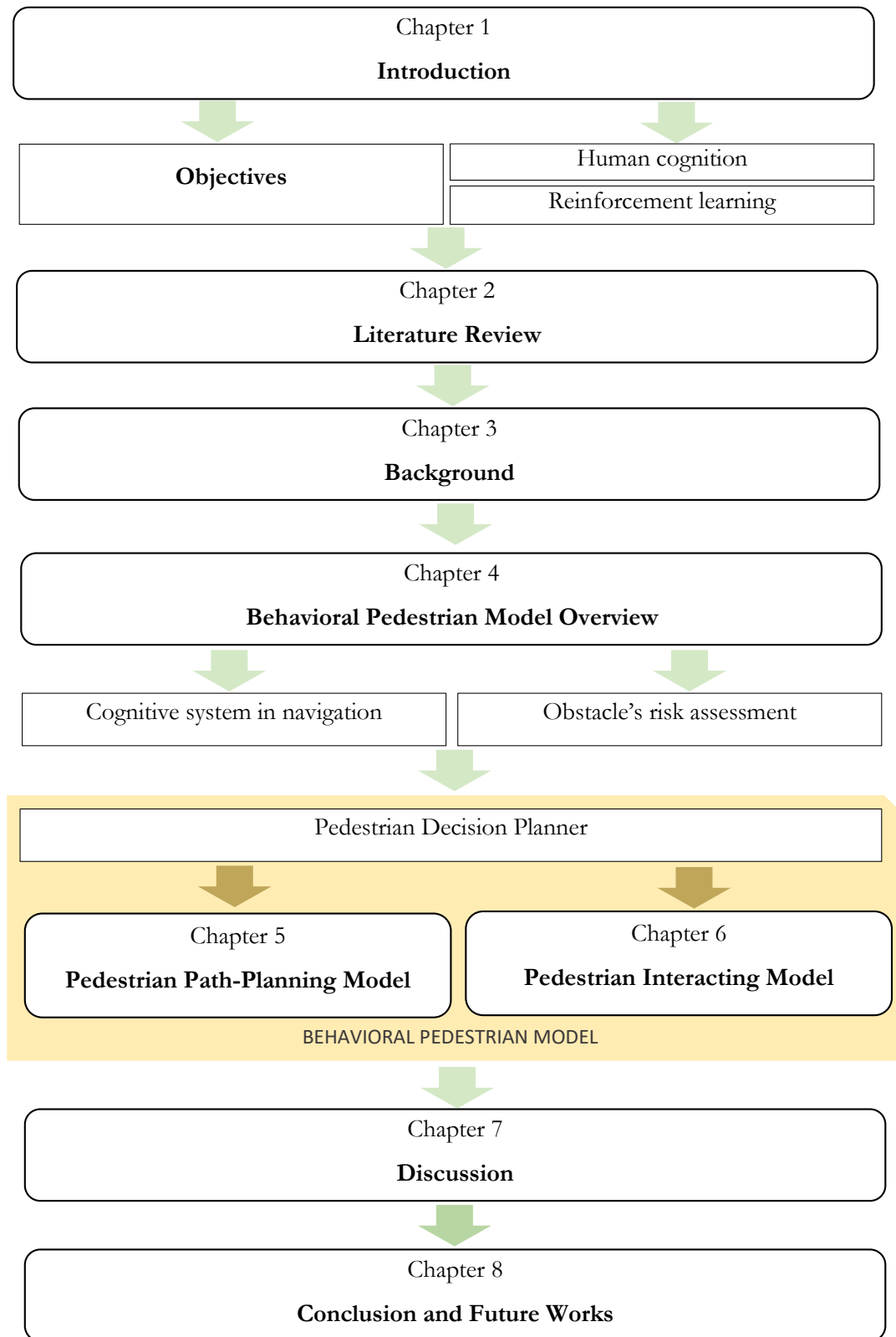


Figure 1.1: Outline of the chapter structure.

Chapter 2

Related works

In this chapter, the literature review of related studies is introduced. The literature reviews are categorized into four domains: the substantial or recent work in pedestrian simulation; the studies in pedestrian prediction; the research around human behavior and the studies in pedestrian simulation using reinforcement learning.

2.1 In pedestrian simulation

Early models in pedestrian interacting simulation often treat pedestrians as *force-based* objects, using Newtonian mechanics to form the forces or accelerations applied to the pedestrians. Social Force Model, introduced by Helbing and Molnar [7], is a notable model that many subsequent models are built upon. The core idea of Social Force Model is that the acceleration applied to the pedestrian agent will be driven by the sum of driving forces, agent interact forces, and wall interact forces. The driving force attracts the agent toward the destination, the agent interact force repulses the agent from other agents, and the wall interact force repulses the agent from walls or boundaries. Generally, the agents are similar to magnetic objects which can attract to or repel from each other and obstacles. The Social Force Model is simple to implement and could be sufficient for modeling a large crowd in straightforward situations. Many studies later have tried to improve the Social Force Model, for example by introducing heading direction [10] or proposing relations between velocity and density [11]. However, in

specific situations which involve human cognition tasks, these models are usually not able to demonstrate a natural interaction behavior between pedestrians.

Many studies were conducted to improve the interactions between pedestrians, considering human behavior factors. Instead of force-based, these models are usually *agent-based*. Compared to force-based models, adopting human thinking is more accessible in agent-based ones. Several studies, mostly in the robotic domain, have tried to simulate human behaviors in their models by proposing various concepts. As an example, the paper by Bonneaud and Warren [12] proposed an approach for a pedestrian simulation model, taking account of speed control behaviors and wall following, meaning the agent would navigate along the walls in the corridor. Another example is a study focusing on the dynamic nature of the environment by Tekmono and Millonig [13], in which the agent imitates the method humans find a path when being uncertain about which doors are open. The agents in these models are rule-based, which means the behaviors are constructed using a finite set of rules. As a result, it often lacks flexibility in the choice of actions, as it could be impossible to build these rules based on the understanding of behavioral psychology in its entirety.

2.2 In pedestrian prediction

In terms of studies in pedestrian prediction, there has been an extensive amount of research, ranges from simple collision detection to body language analysis. Some studies have proposed solutions to present the prediction as a “map” of probability, for example in the papers by Karasev et al. [63] and Ziebart et al. [64]. Those approaches could be difficult to be applied in a reinforcement learning problem as using these data for training could be challenging and highly unstable. Many other studies introduce pedestrian navigation prediction based on image or video processing. For instance, Møgelmoose et al. [65], Goto et al. [66], and Dominguez-Sanchez et al. [67] have proposed different approaches for recognizing pedestrian movement based on photo and video inputs.

While the studies on highly accurate prediction are extensive, especially in the robotic domain, there is not much research in the prediction by the human cognitive system. In a study in human neuroscience, Bubic et al. [18] discussed the mechanism of the prediction in the human brain, which could also be practical in

walking situations. Ikeda et al. [19] proposed an approach to the prediction of the pedestrian's navigation employing the sub-goal concept, meaning the navigation path would be segmented into multiple polygonal lines.

2.3 In navigation behavior

Regarding research in human behavior, many studies can be found in the field of robotics research. Many researchers have tried to solve the problems in *human comfort* and constructing naturalness [25]. For an agent to navigate naturally, not conflicting with other pedestrians or obstacles is not enough; but the agent also needs to replicate different behaviors from humans. Another concept proposed in human behavior research is *human bias* or *cognitive bias*, which causes the anomaly in the human decision process. For example, Golledge [34] has shown that pedestrians do not always choose the most optimized decision while selecting a path. Another study by Cohen et al. [35] also discussed how the human brain makes decisions between exploitation and exploration. These aspects were supportive for forming the agent behavior in our research.

There are also several studies focusing on pedestrian prediction based on human behaviors. An example is a study by Yi et al. [68], as the pedestrian walking behavior is encoded from video data using a convolutional neural network (CNN). Another example is the prediction model by Schneider and Gavrilu [70], which focuses on the pedestrian motion types extracted from the automated vehicles' camera inputs. Body language also contributes to the research in pedestrian prediction. For example, Quintero et al. [69] proposed a pedestrian path prediction based on the human pose data, utilizing Gaussian Process Dynamical Models (GPDM) method.

2.4 In reinforcement learning

The use of reinforcement learning in the agent-based model has recently become more prevalent. Prescott et al. [14] proposed a reinforcement learning method to train the agent's basic collision avoidance behavior. Recently, Everett et al. [15] introduced a novel method for the agent to avoid collisions, using reinforcement learning with deep learning. The resulted behaviors of these models are very

competent, however, the effect of human cognition is still lacking. Other studies have been trying to resolve this problem. For instance, Chen et al. [16] proposed a deep reinforcement learning model with the agent respecting social norms in situations like passing, crossing, and overtaking.

In a study by Martinez-Gil et al. [36], an experiment in using reinforcement learning for a multi-agent navigation system has been implemented; however, the algorithm used was q-learning which is too simple and does not suit well to a dynamic environment. Another approach is learning from observing examples from human behavior. In their paper by Kretzschmar et al. [37], a navigation model was proposed using inverse reinforcement learning. One difficulty in such approaches is the example or the dataset from human behavior is not easy to be extracted or readily available.

Chapter 3

Background

The background concepts related to the study are presented in this chapter. The two concepts presented are reinforcement learning and the PPO algorithm.

3.1 Reinforcement learning

The concept of reinforcement learning was first coined by Sutton and Barto [21]. In reinforcement learning, the agent needs to optimize the *policy*, which specifies the *actions* that will be taken under each *state* of the observed environment. For each action taken, a *reward signal* will be given. Depending on the reward, which can be either positive or negative, this could encourage or discourage the action, respectively. The aim of the agent is to maximize the *cumulative reward* in the long term. Figure 3.1 illustrates the overview of a generic reinforcement learning model.

The formulation for a reinforcement learning problem is often modeled as a *Markov Decision Process* (MDP). An MDP is a tuple $(\mathbb{S}, \mathbb{A}, P, R, \gamma)$ where \mathbb{S} is a finite set of states; \mathbb{A} is the set of the agent's actions; P is the probability function which describes the state transitions from s to s' when action a is taken, R is the reward function immediately given to the agent; $\gamma \in [0, 1]$ is the discount factor.

The probability P is calculated by

$$P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a) , \quad (3.1)$$

where a is the taken action, s is the previous state and s' is the current state.

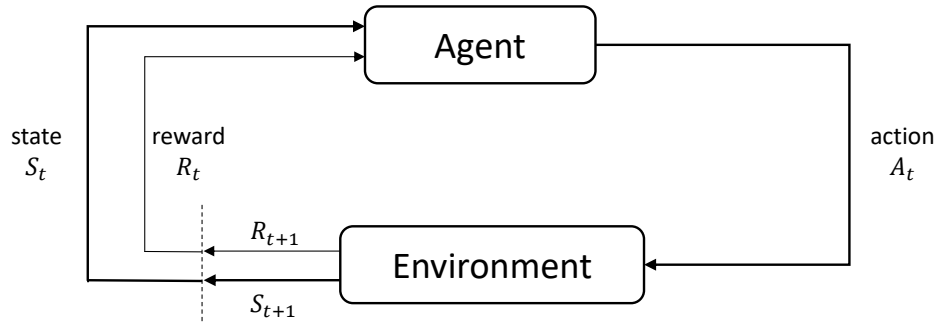


Figure 3.1: Overview of a reinforcement learning model. [21]

The reward function R is formulated as

$$R_a(s) = (R_{t+1} | s_t = s, a_t = a) . \quad (3.2)$$

To solve a reinforcement learning problem is to find the optimal policy that maximizes long-term cumulative reward. Because certain actions could receive an intermediate negative reward but may achieve the highest conclusive reward, a *value function* is necessary to estimate the present state of the agent. The value function for the state s would be presented as:

$$V(s) = \max E \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \right] , \quad (3.3)$$

where $\pi : \mathbb{S} \rightarrow \mathbb{A}$ is the policy for the action \mathbb{A} in the state \mathbb{S} .

3.2 PPO algorithm

Reinforcement learning algorithms are categorized into 2 categories: *model-based* and *model-free* algorithms. A model of the environment could be interpreted as the understanding of the agent about the environment. A model-based algorithm uses the model of the environment for planning by estimating future states before

taking action. On the other hand, a model-free algorithm learns mostly by trial-and-error without any planning.

Proximal Policy Optimization (PPO) algorithm, proposed by Schulman et al. [22], is a model-free reinforcement learning algorithm using a neural network approach to optimize the agent’s policy via a training process. The idea of PPO algorithm was primarily based on the Policy Gradient algorithm by Mnih [73] and improved from their previous Trust Region Policy Optimization (TRPO) algorithm [74]. Similar to the vanilla Policy Gradient method, the loss function of the neural network is constructed using an advantage value \hat{A}_t , the deviation of the expected reward compared to the current state’s average reward. This advantage value is calculated by running the policy for T timesteps and compared with the baseline estimation.

$$\hat{A}_t = -V(s_t) + R_t + \gamma R_{t+1} + \dots + \gamma^{T-t-1} R_{T-1} + \gamma^{T-t} V(s_T) , \quad (3.4)$$

where $V(s_t)$ is the state value function; t is the time index in $[0, T]$ and $gamma$ is the discount factor of the future states.

For algorithms like Policy Gradient, a policy $\pi_\theta(a_t|s_t)$ will be updated after every training step. With a noisy environment, the old policy $\pi_{\theta_{old}}(a_t|s_t)$, which might actually be better than the new one, will be overwritten; causing the training process to be less efficient. To avoid the problem, the PPO algorithm proposed a method to avoid staying away too far from a good policy by keeping the old good policy and compare it with an updated one using a clip surrogate objective.

The clip surrogate objective is formulated as

$$L^{clip}(\theta) = \hat{\mathbb{E}} \left[\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right] , \quad (3.5)$$

where $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ and ϵ is a clipping hyper-parameter; θ is the policy parameter and $\hat{\mathbb{E}}$ indicates the empirical expectation over predefined timesteps.

The clipping helps the training become more stable, as the previous policy will not be overwritten by a worse newer policy in a noisy environment. Figure 3.2 demonstrated the clipping method in the calculation of the loss function.

With the inclusion of policy surrogate and value function error term, the loss function in the PPO algorithm is formulated as below

$$L^{clip+VF+S}(\theta) = \hat{\mathbb{E}}[L^{clip}(\theta) - c_1 L^{VF}(\theta) + c_2 S[\pi_\theta](s_t)] , \quad (3.6)$$

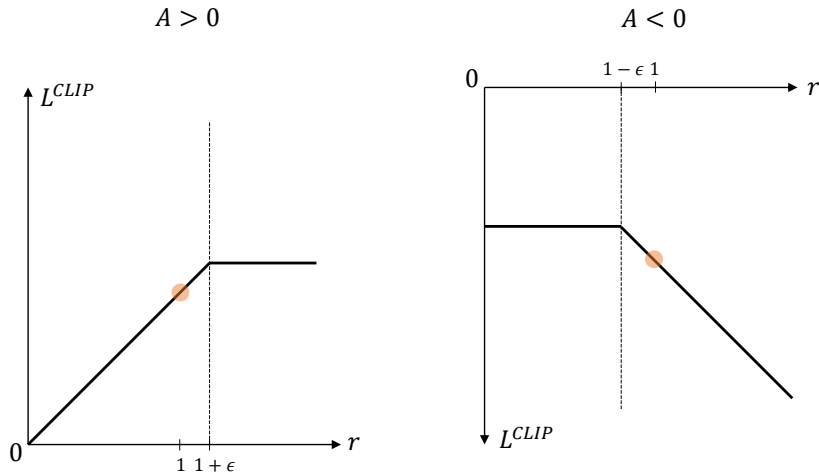


Figure 3.2: Clipping method in PPO’s loss function calculation. [22]

where c_1 and c_2 are coefficients, S represents entropy bonus and L^{VF} is the squared-error loss $L^{VF}(\theta) = (V_{\theta}(s_t) - V_t^{targ})^2$. The use of policy surrogate and value function error term is not compulsory and could be omitted when performance is in high priority. However, this is required when using a neural network structure that parameters between the policy and value function are shared.

The more detailed network structure used in the PPO algorithm is illustrated in Figure 3.3. Similar to Policy Gradient, the neural network used in the PPO algorithm is an Actor-Critic network. The network has the same input layer but has two heads in the output layer. The first head is the Policy head (Actor), consisting of the agent’s action probability distributions. The structure of the action probability distribution in the output layer of the Policy head depends on the type of the agent’s actions. In the case of discrete action outputs, a categorical probability distribution is used, which consists of the probability for each action. In the case of continuous action outputs, the neural network will output the Gaussian distributions of the actions, represented in their parameters: mean and standard deviation. The second head is the Value head (Critic), which outputs the state value $V(s_t)$, which indicates the current estimation of the environment.

Algorithm 1 expresses the PPO algorithm in detail.

There are two loops involved with the algorithm. The first loop is to calculate the advantage estimate $\hat{A}_1, \dots, \hat{A}_T$ by using the policy $\pi_{\theta_{old}}$ for T timesteps. This is done by letting the agent interacts with the actual environment through online learning. Instead of estimating the expected reward, the advantage value could

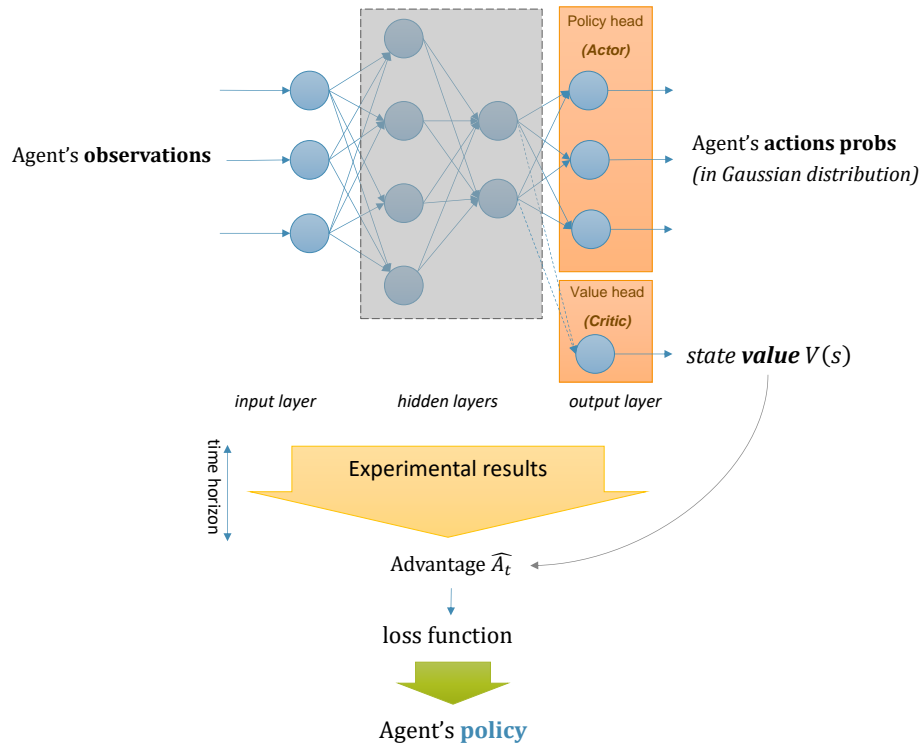


Figure 3.3: The neural network structure in path-planning model.

Algorithm 1: PPO Algorithm, Actor-Critic style [22]

```

for  $iteration = 1, 2, \dots$  do
  for  $actor = 1, 2, \dots, N$  do
    Run policy  $\pi_{\theta_{old}}$  in environment for  $T$  timesteps
    Compute advantage estimates  $\hat{A}_1, \dots, \hat{A}_T$ 
  end for
  Optimize surrogate  $L$  wrt  $\theta$ , with  $K$  epochs and minibatch size  $M \leq NT$ 
   $\theta_{old} \leftarrow \theta$ 
end for

```

be precisely calculated using the formula 3.4, as the cumulative reward is given through the agent's actual interactions with the environment. This is contrary to an off-policy reinforcement learning method, such as Q-learning or DQN, in which the agent learns from the existing experience and the action-state value (i.e. Q value) must be estimated. Because of this reason, using a simulation tool for the agent is beneficial as it could speed up the learning process and allow the agent to explore the environment. The training process starts with the agent mostly takes stochastic actions. After getting some feedback, the agent would be better with its action and less dependent on exploration. On the second loop, the algorithm collects all of the advantages values, then uses gradient descent on the policy network using the clip objective 3.5. Usually, the PPO algorithm maintains two policy networks, one for old policy and the other for updated policy. After every K epochs, the algorithm will synchronize the updated policy to the old policy, using optimization on the surrogate loss with M -sized minibatch.

Chapter 4

Behavioral pedestrian simulation model

In this chapter, we introduced various concepts of the human cognitive system in pedestrian navigation. Subsequently, the behavioral pedestrian model's overview is presented. The consisted pedestrian decision planner task of the model is also demonstrated in detail. In addition, this chapter introduced the concept of risk and how it could affect the pedestrian's decision in navigation.

4.1 Cognitive system in navigation

Many animals also use a cognitive system for their navigation. To comprehend the cognitive system in navigation for human pedestrians, researchers have been studying the navigation behavior of animals. The studies have suggested that the navigational cognitive system in animals shares many similarities to that in humans. Furthermore, many errors that happened in human navigation also persist in the navigation behavior in animals. For instance, the mechanism of memorizing and choosing routes by humans is similar to the routing mechanism in bees [44]. Another example is the navigation behavior of grizzly bears, which is impacted by cognitive bias and also is capable of learning from past experience [41]. Additionally, cognitive maps and hippocampus are also used in mammals and other animals to help them to make decisions in navigation [45].

The overview of the cognitive system in navigation could be illustrated in Figure 4.1. To make decisions in navigation, firstly, the pedestrian would need

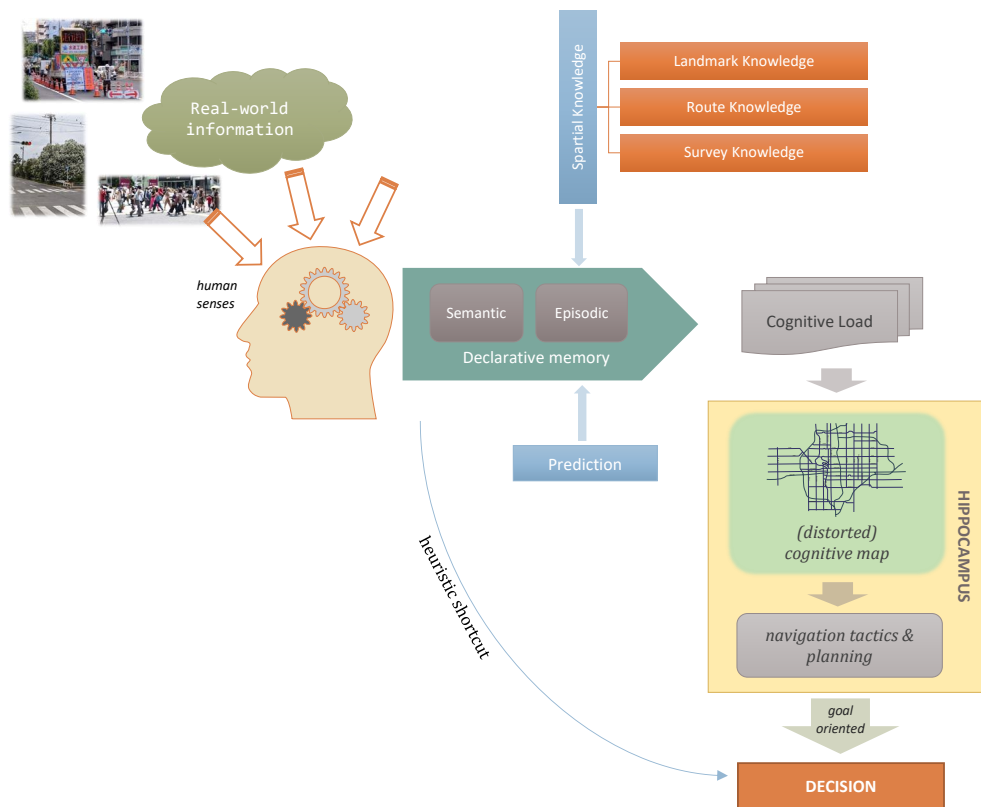


Figure 4.1: Overview of the human cognitive system in navigation.

to perceive the environment. The information is then consciously and subconsciously stored in the memory. At the same time, this information, combined with the pedestrian's experience, forms the cognitive map. Finally, the hippocampus is responsible for making the decision depending on the current goal of the pedestrian.

The first step in the pedestrian's decision-making process is perception. There is an immense amount of information constantly feeding into the human brain every moment. This information includes all types of data including visual, audio, and haptic perception, which is brought to the brain via different human senses. To make decisions from this, the human brain consciously and subconsciously chooses which information to be passed into the cognitive system and which information to be discarded. These data, as they need to be transferred through the senses, are sometimes incorrect and do not reflect the actual real-world information. For that reason, we have carefully considered which information is perceived by a human pedestrian in modeling the environment and also in the designing of the pedestrian's observations.

Among the selective information, a portion of that will be stored within the human memory [45]. There are two types of memory in humans: declarative and procedural memory. While procedural memory is used for various types of automatic processing and controlling human locomotion, declarative memory is used to store and process past experiences and events. For the cognitive system to make decisions from the received information, only declarative memory is used. Declarative memory can also be further categorized into semantic memory, which is used for storing information such as names, facts and concepts; and episodic memory, which is for experienced events. The environment information is stored in the memory in the form of spatial knowledge. There are more than three levels of spatial knowledge [46]. The lowest level of spatial knowledge is landmark knowledge, which represents the memory of the pedestrian for the objects within the environment. The next level is called route knowledge, which could be interpreted as the memory of a series of routes to navigate to the destination. A higher level is survey knowledge, which is stored as a mental representation of the spatial environment, like a bird-eye view of the environment for instance.

Not only the experiences of the past navigation are used in the human cognitive system. In most situations, humans also make predictions of future environment states over a certain planning horizon for more efficient navigation [18]. Generally, the mechanism of cognitive prediction involves the following steps. Firstly, the person anticipates the current state by comparing the short-term expectation with the perceived data from human sensors. From the anticipation, the prediction of the future state will be made under consideration of the prospection, potential distant future occurrences. The mechanism of prediction in human navigation is greatly acknowledged in our study. Different pedestrian prediction methods, depending on the current task of the pedestrian agent, are accordingly realized. These will be presented in more detail in Section 5.4 and Section 6.3.

The perceived information, combined with the current spatial knowledge and planned prediction, will be processed within the cognitive system. All of these data put heavy stress on the cognitive load. To reduce the cognitive load, various tasks will be carried out in the human brain as a result. These tasks are mostly executed by an organism placed under the temporal cortex inside the human brain, called the hippocampus [50]. The hippocampus is accountable for the decision-making process, particularly in navigation [45]. Figure 4.2 illustrates the shape of the hippocampus (yellow) inside the human brain.

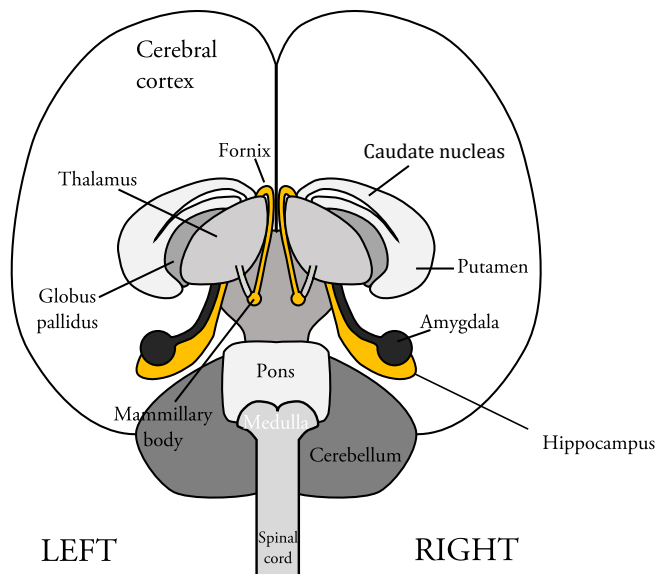


Figure 4.2: The hippocampus and the related regions inside the human brain.

One of the tasks performed by the hippocampus is to model the aforementioned information into a cognitive map, a visual representation of the navigation inside an environment. The concept of cognitive map was first coined by Tolman [55], presented in both human and animal brains. This cognitive map is often a distorted representation of the actual environment due to the complex processes involved with the construction of the cognitive map as discussed. Decisions on how to navigate the cognitive map could be made using some navigating mechanisms, such as path integration, piloting, and guidance. More specifically, path integration or dead reckoning is used by the pedestrian to expect a general orientation, such as heading toward a cardinal direction. Piloting is used when the pedestrian reaches a matching view or snapshot of a location stored within memory. Lastly, guidance is applied when the pedestrian is following its usual routine to navigate in a familiar setting or environment. Another common mechanism to navigate the cognitive map is to perform the planning of the path from the pedestrian's position to the destination before actual navigation [47]. Planning helps the pedestrian maneuver more strategically, thus more efficient navigating compared to other animals. The planning process often involved different levels of spatial knowledge as mentioned above. That means, depending on the situation, the human pedestrian could plan the path using his survey knowledge or try to navigate using the route knowledge. There are other tactics the pedestrian could use in navigation, like wall-following and turn-into-door for example. The

cognitive system often combines different tactics to generate the decision based on the pedestrian’s current goal. The mechanism of the hippocampus is known to be goal-oriented, which means that it will consider the current goal of the pedestrian to provide the appropriate decision.

Inspired by this mechanism of the cognitive system, we designed our path-planning model that replicates the pedestrian’s process of planning the path from his position to the destination. The model will be represented in Chapter 5. We also modeled a decision planner, presented in Section 4.3, which takes hints from the goal-oriented decision-making process of the hippocampus inside the human brain.

As previously presented in Section 1.2, in many cases, the human brain could skip part or all of the thinking process to make the decision, and sometimes, this could lead to an incorrect or unoptimized choice. This is called human bias or cognitive bias. The human nervous system has a mechanism of learning from these mistakes in a trial-and-error approach called reinforcement learning. In the human brain, an organ called *basal ganglia* is responsible for the human reinforcement learning process. In principle, the mechanism of the basal ganglia is similar to the structure of an actor-critic model, in which the actions and the movement of the human are matched with the states and rewards from the environment to help shape the human’s actions. In the human brain, the sense of rewarding is produced by the neurotransmitter called dopamine. The dopaminergic neurons send the dopamine to striatal neurons to signal the reward expectation. This, combined with the sensory prediction from the cerebellum’s forward model to output the desired actions to the thalamus. These actions are similar to the action outputs in the reinforcement learning paradigm in machine learning. Figure 4.3 illustrates the basal ganglia inside the human brain, together with the anatomy of the related brain regions involved in the reinforcement learning process, as suggested by Ludvig et al. [56].

By giving our model an analogous method in making decisions in navigation, we expect our pedestrian agent to produce a human-like behavior with similar choices as well as common human navigational errors.

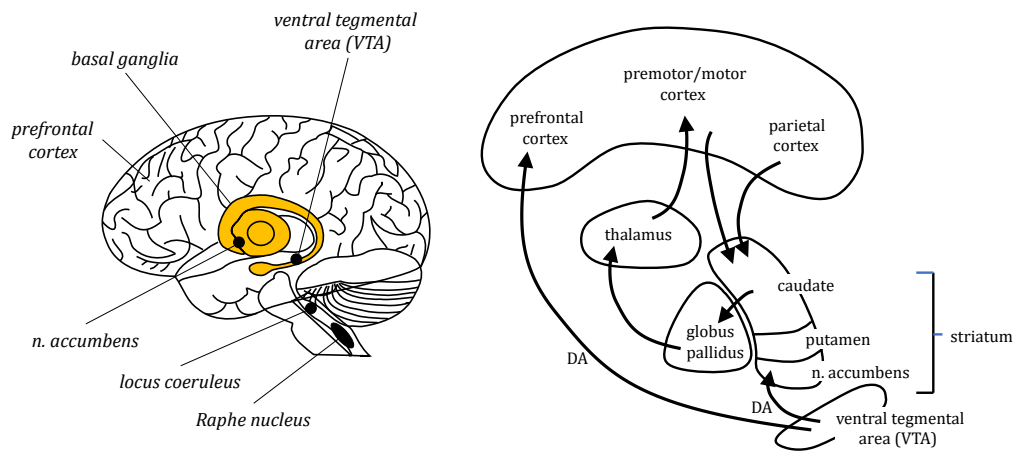


Figure 4.3: The anatomy of the reinforcement learning process with the basal ganglia, based on the structure suggested by Ludvig et al. [56]

4.2 Model overview

There are three levels involved with the procedure of navigating in the environment of a human pedestrian [23]. In the first level, the *strategic level*, the pedestrian needs to initiate the planning, such as determining the destination and planning the means to get there. For instance, if a pedestrian planning to navigate from his house to the supermarket, the purpose of the strategic level for his decision-making process is to choose which way to go. For example, he could choose the shortest route, or he could choose the route with the least detours. In practical situations, humans often choose the most familiar route, meaning that the route chosen should have the least difference from the highly used options in the past. The second level is the *tactical level*, the pedestrian needs to plan the navigation path to achieve the intermediate desired goal, such as reaching the local destination. More specifically, if the choice at the strategic level is a set of paths or roads from the starting position to the destination, the navigation in each path is what needs to be done at the tactical level. At the tactical level, the pedestrian also needs to consider possible obstructions that may hinder the navigation. For instance, if there are obstructions like physical obstacles or other pedestrians, the agent also needs to plan forward so that the path will not conflict with their navigation. The third level, which is the *operational level*, will

handle the agent's operational dynamics such as movement or gesture controls. An example of interaction is when the pedestrian is getting close to another person, but that person suddenly changes the movement in an unpredictable manner that could intervene in the planned path. In this situation, the agent needs to continuously observe the other's every action, and accordingly decide which interaction or movement to make. For example, if that person moves to the left of the pedestrian, he could go to the right or slow down to observe more responses from the other person.

In our study, the choice made at the strategic level is disregarded, as we want to concentrate on the behavioral interaction of the pedestrian at lower levels. There has been a great amount of research regarding pedestrian route choice at the strategic level. The most common method is using a graph node structure for traversal [19]. The needs or interests of the pedestrian also take an important role in how the pedestrian chooses the route, as indicated by Koh and Wong [75]. Jaros et al. [76] took a different approach by observing the activity pattern of the pedestrians and subsequently replicating its behavior.

For this reason, our behavioral pedestrian simulation model consists of two main parts: A pedestrian path-planning task, which simulates the pedestrian's path-choosing process at the tactical level; and a pedestrian interacting task, which simulates the pedestrian's interaction behavior at the operational level. The model also includes a pedestrian decision planner, of which the primary function is to determine when the path-planning task is performed and when the interacting task is carried out instead.

The model consists of:

1. **Path-planning task:** In this task, the pedestrian agent observes the state of the environment, then plans a draft path to the next destination. Instead of executing consecutively, the task will be carried out gradually. A particular example of this is a pedestrian using a mobilephone while walking. Each time the pedestrian is not looking at the mobilephone to observe the surrounding, this planning task is executed. If there is no remarkable event that requires special attention, the pedestrian could navigate following his planned path without the need of observing the environment constantly. The details of this task are presented in Chapter 5 of this dissertation.

- 2. Interacting task:** This task is usually inactive; however, when the pedestrian is following the path that was planned in the planning task but there is an unexpected event or anomaly that occurred, this task will be executed. For example, an unexpected vehicle or pedestrian emerges and may conflict with the pedestrian's path, or an existing obstacle does not behave like the prediction of the pedestrian. In such cases, the pedestrian needs to carefully perceive the obstacle's actions to interact properly. Unlike the planning task, this task will be carried out consecutively until these interactions are no longer required. This task is presented comprehensively in Chapter 6.
- 3. Decision planner:** This component decides which task is performed under the current environment's states. In the example above, when the pedestrian is navigating without any obstructions, the path-planning task is periodically called. However, when the pedestrian is significantly close to an obstacle and has a high chance of colliding with it, the interacting task will be consecutively executed. This will be further explained in Section 4.3.

4.3 Pedestrian decision planner

Humans are consistently required to choose between multiple objectives in their lives. For example, a researcher may need to consider when to research in a new direction and when to continue improving the current hypothesis. The decision results from a number of factors, such as the positivity of the current finding or the available resources for testing out the new hypothesis. The same process happens in many other activities in our lives, including minor tasks like shopping, cooking, studying, and significant responsibilities like getting married or finding a job. The human brain always needs to carefully select the appropriate task to handle the situation before the detailed processes of that task are carried out.

This decision-making process is accomplished by the hippocampus inside the human brain. Many studies have indicated that the mechanism of the hippocampus is goal-oriented [47, 48], meaning that depending on the current goal of the pedestrian, the hippocampus will choose the appropriate task to be carried out. Regarding pedestrian navigation, in the scope of navigating from one position to

another, there are at least two main tasks that need to be addressed in order to form the cognitive map [49]. The first one is planning a path to navigate and the second one is interacting with other obstructions. When to choose each task mostly depends on how close the pedestrian is to the obstructions and how the pedestrian is confident with his choice (i.e. his prediction is highly correct). More specifically, if there is an obstacle such as another person approaching, and the obstacle is still far, only the path-planning task is necessary. On the other hand, if the obstacle is moving unpredictably, planning a path should be largely inefficient. This means the pedestrian's brain would use the interacting task to give the instructions for his action. By doing this, the pedestrian needs to continuously observe what is happening and act correspondingly. Naturally, if no obstacle is present, the interacting task is redundant.

As a result, we created a rule-based system for the pedestrian decision planner. Using a rule-based system has several advantages. Firstly, a rule-based system should be similar to goal-oriented approach of the hippocampus to some extent. In addition, if a rule-based model is used, the path-planning task and the interacting task could be separated. This means we could research and evaluate each task more profoundly without affecting the other's results. Lastly, implementing the rule-based system is more straightforward than other implementations.

For the implementation of the rules, we consider the following requirements for each related task:

1. Path-planning task

- Is usually called on a regular basis.
- Is often ignored in case of being close to a moving obstacle.
- Does not need to be called if the pedestrian is still following an existing planned path and there is no significant change in the environment states.

2. Interacting task

- Uses the planned path as a guide.
- Is called when there are the needs to change in the initially planned path (e.g. change in environment states, obstacle with unpredictable behavior, complex environment).

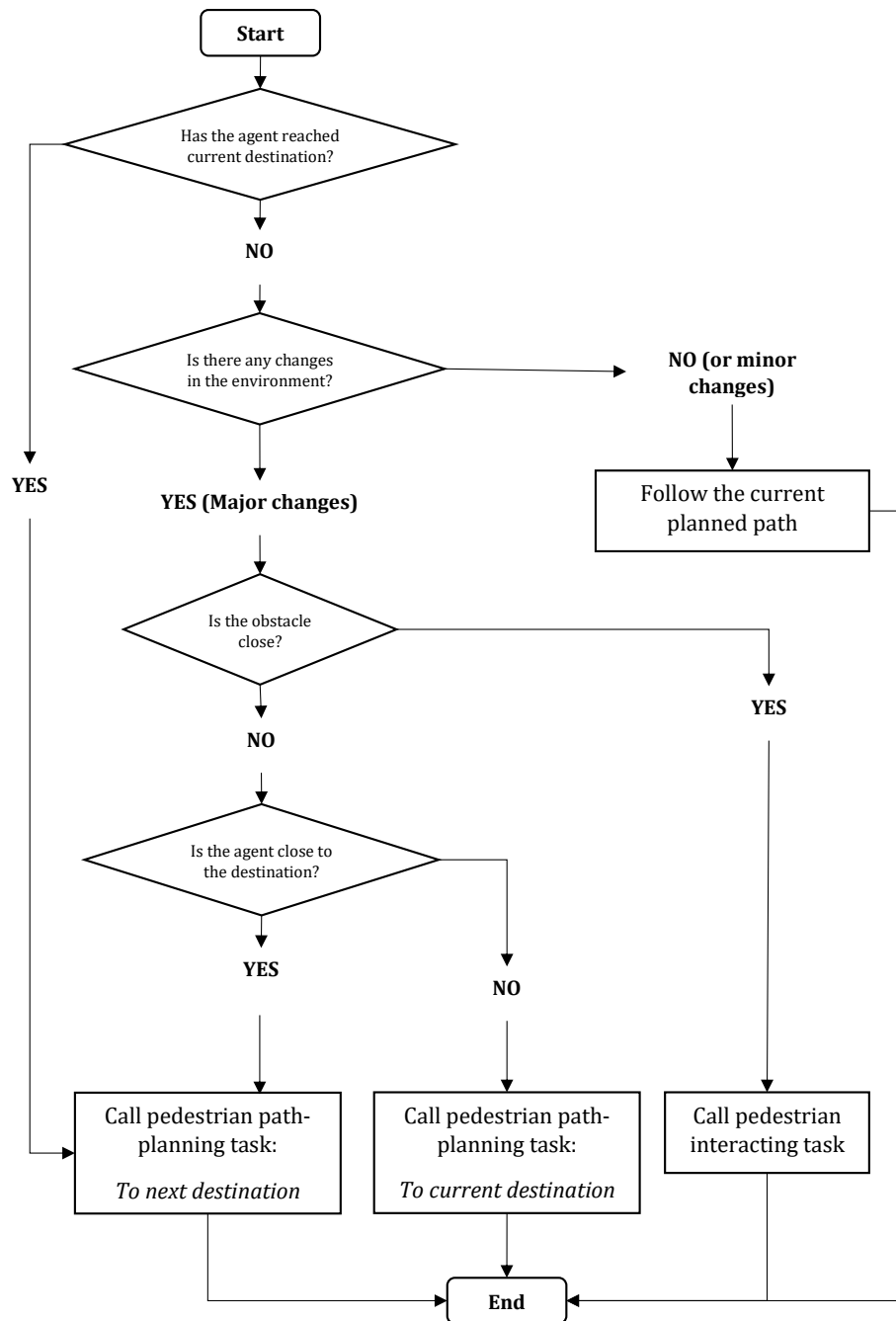


Figure 4.4: Flow chart of the decision planner task.

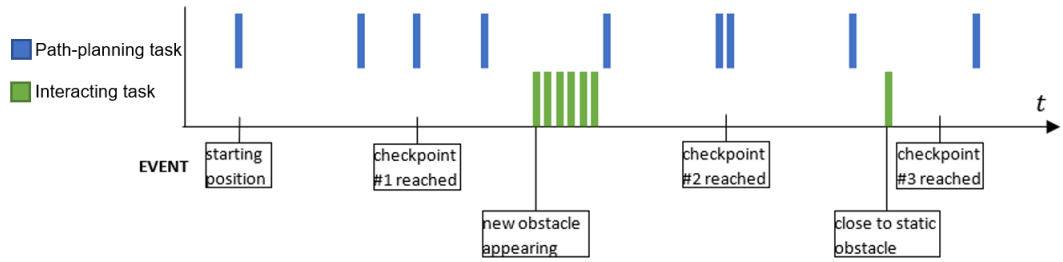


Figure 4.5: Example of the timing when the decision planner is called.

Based on the requirements, The rule sets for our pedestrian decision planner are designed as a flow chart in Figure 4.4.

The task is carried out periodically, sometimes in a shorter interval than others. As an example, when there is no visible obstruction in the environment, the decision planner could be executed at a lower frequency. In contrast, when the pedestrian is interacting with the obstacle, the interval is much shorter for the interacting task to be called. The interval period also varies between genders and age groups. For example, older people often need to frequently observe and make decisions, while young pedestrians tend to trust their environment analysis and make fewer navigation choices. The interval period choice of children is much less inconsistent because of their inexperience in understanding the situations. They might continuously observe and take decisions even when there is no threaten obstacle, but in a more dangerous situation, their decision planner may be insufficiently performed. Cultural differences also contribute to how the decision planner task performs. Upon observation, we perceived that the pedestrians in the countries with a more ordered navigating culture, including Japan, require less attention in decision making than in other countries. That means their decision planner does not need to be carried out in a short interval, as opposed to the others. Figure 4.5 represents an example of when the decision planner is called in correspondence with the interaction task. In this figure, the period when the interaction task and the decision planner are called is presented in blue and green, respectively.

Undoubtedly, the rule-based implementation of the pedestrian decision planner is much more simple compared to the one in the human cognition system. The decision planner task by the hippocampus is much more sophisticated for several reasons. One of the reasons is that all parts of the brain are intertwined, meaning any decisions or results will also affect the choices in other parts of the

brain. Another reason is that the data interpreted in the human brain is not in concrete form, but is rather more like fuzzy values. As a consequence, all operations on the data would be performed using fuzzy logic instead. Humans also often make decisions based on their instinct and experience that could lead to certain unexplainable behaviors, which is also another reason that makes the human decision planner task much more complex.

4.4 Obstacle's danger and risk

Obstacle is a substantial concept in our study, as its presence apparently has a great impact on how the pedestrian navigates in the environment. Different from a physical obstacle in the real world (e.g. a rock, a wall, or a construction site), the obstacle in our model is any person, animal, or object that would be considered as an obstruction in the pedestrian's thinking. Occasionally, the obstacle could be physical or abstract, such as a restricted area defined by traffic laws, for instance. The observed obstacle is defined by *spatial effect*, a term introduced by Chung et al. [38]. An example of this is a group of other pedestrians walking together. Theoretically, these are considered multiple obstacles, but because planning a path through these obstacles is viewed as unnatural and even impolite, such practice is not encouraged. In our model, these obstacles would be considered as a single obstacle. In addition, because of the spatial effect, the obstacle may dynamically change its properties. An example of this is the crossroad. If the light is red, the entire crossroad would be treated as an obstacle, but if the light is green, it is no longer viewed as an obstacle from the pedestrian agent's perspective.

One of the most critical properties of an obstacle would be its *risk* perceived by the agent. The difference in the perceived risk of the obstacle could greatly change how the agent plan the path. For example, if the obstacle is a highly dangerous one (e.g. a deep hole on the street), the pedestrian would very likely stay further away from it, as represented in Figure 4.6.b. On the other hand, if the obstacle is safer (e.g. a shallow water puddle), the pedestrian is less likely to avoid it too much. In certain situations, such as when the pedestrian is in a hurry, he may choose to walk over the water puddle obstacle, as presented in Figure 4.6.a.

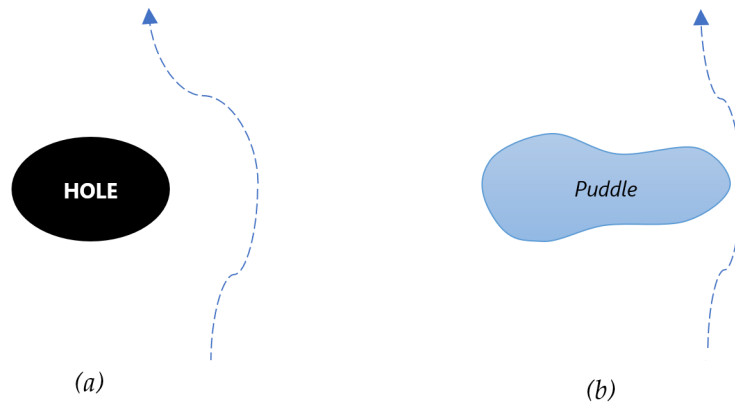


Figure 4.6: Path planned by an agent with different obstacle's danger.

The risk perceived from the obstacle could depend on many factors. As in the ISO/IEC Guide 51 in Safety Aspect [79], risk is defined as the “combination of the probability of occurrence of harm and the severity of that harm”. For instance, the danger of a lion should be remarkably high, but if that lion is kept inside a cage, its risk should be close to 0 as the chance of the lion interacting with others is low. In pedestrian navigation, the danger from a human should be lower than a construction machine, for example. However, the risk coming from a pedestrian running at high speed, toward the pedestrian agent, should have a greater risk compared to the construction machine moving slowly on the side.

Accordingly, we model our obstacle consisting of the following properties: *danger*, *size*, *direction*, *speed*, and *type* of obstacle. Similar to ISO/IEC Guide 51, the risk from the obstacle is formulated by the obstacle's harm and its probability of collision perceived by the agent. The size of the obstacle should cover the concept of spatial effect mentioned above, not just the size of the physical obstacle. For example, a damaged or unstable power pole would have a much larger “size” compared to a steady or stable one due to the fear of the pole falling. For simplicity, we assume our obstacle has a round shape; thus, the size of an obstacle will be expressed by a radius value.

All risk, danger level, and other obstacle's properties used in our study are perceived by only the pedestrian's cognitive system, which could be different from the actual information of the obstacle.

Chapter 5

Pedestrian path-planning

This chapter demonstrates the model for the path-planning task of the pedestrian agent. The path-planning task simulates how the pedestrian plan the path from the current position to the destination in the decision making process.

5.1 Introduction

The path-planning process is carried out within the human cognitive system before the pedestrian's actual navigation. In this step, the two following tasks are carried out sequentially. In the global path-planning task, the pedestrian uses his experience and knowledge to specify his destination and plan the route to get there. In the local path-planning task, the surrounding environment is often observed via human vision and transformed into a topological map. Subsequently, the pedestrian estimates the path would be taken before carrying out the actual movements [51]. While there is a great deal of research that addresses the global path-planning, the route selection process to the destination [52, 53], the studies of the local path-planning problem are generally scarce. For the few studies that focus on this problem, their models often try to optimize certain objectives, such as next state optimizing [13] or way finding [54]. In real life, people tend do not usually choose the most optimized solution [34]; therefore, these models may yield inaccurate navigation behavior in certain situations.

The path-planning process is crucial because a tolerable plan could help the pedestrian avoid a foreseeable accident. This process is usually carried out unconsciously in most navigating situations. To accurately simulate the path-planning

process is a challenging task. As this process happens only inside the human mind, the pedestrian’s observable behavior and the planned path could be not entirely alike. To properly address this problem, different aspects in behavioral psychology and cognitive science should be considered. Many studies only focus on replicating the navigating behavior by optimizing the path taken, however, it has been shown that humans do not always take the most optimized action. This is due to the reason that for every task, humans usually consider cognitive biases, the systematic flaws developed when humans trying to make decisions based on their previous limited experiences. Other studies try to approach the problem by creating empirical models based on the observable behavior of pedestrians. These models usually concentrate on only the most essential behaviors while certain factors might be ignored.

To overcome these problems, we adopted using reinforcement learning for the pedestrian agent’s local path-planning process, as discussed in Section 1.2, reinforcement learning techniques share many similarities with the operation of the human cognitive system. Moreover, reinforcement learning techniques using neural networks, such as the PPO algorithm, even use a resembling structure as the human’s neural system. Consequently, we need to determine how the human brain works in doing that task. More specifically, we need to address the mechanism of planning a navigation path by the human pedestrian.

This process of the agent learning to navigate is similar to a child learning how to get to the destination and avoid colliding with any obstacle. Once the behavior is learned, he can naturally do the task simply from experience without the need of learning again. However, just learning the navigation task through a trial-and-error approach might not be enough for efficient path planning. For a grown-up human to carry out the path-planning task, further thinking processes are utilized. In particular, the cognitive predictive process is essential in the way the human brain processes many tasks, including navigation. This helps the adult pedestrians navigate more competently with fewer collisions with surrounding obstacles.

Another important process which humans gradually learn through their lives is the risk assessment of obstacle’s danger. In a study by Ampofo-Boateng [57], it is indicated that children at different ages perceive danger differently. The older children could identify the danger more correctly, while younger children usually could not specify the danger apart from moving vehicles.

Because of these reasons, we need to address the risk assessment process and the prediction in the path-planning task for the model to replicate the planned path more accurately. More specifically, the risk assessment process in our model aims to replicate the observation of risk for our pedestrian agent, to be subsequently employed by the reinforcement learning model.

As a result, we design our model focusing on two tasks: learning task and prediction task. The learning task helps the agent learn the natural behavior of navigating. The prediction task simulates the human prediction of the obstacle's upcoming position, which subsequently the pedestrian will avoid instead of the current position of the obstacle.

5.2 Model overview

Figure 5.1 demonstrates the overview of our pedestrian path-planning model. The model consists of two components:

1. Path-planning training. This component instructs the agent to learn the basic navigation and collision avoidance within the environment using reinforcement learning. The details of the component are presented in Section 5.3.
2. Point-of-conflict (POC) prediction. This component simulates the agent's prediction of the collision with the obstacle. The process of prediction updates the input of the path-planning components and is handled before the planning process. Section 5.4 explains the prediction model in more detail.

For a reinforcement learning model, the design of the environment plays an important role. An environment that is similar to the real-world environment is usually unsuitable, as its complexity often leads to multiple problems. First of all, generally, the agent is not able to learn efficiently in a complex environment. For example, if there are many obstacles within the environments, they would create an extensive number of different states, leading to a considerably noisy training environment. To learn in an environment like that could be difficult for the agent, as the training would be quite unstable. Another problem of that is

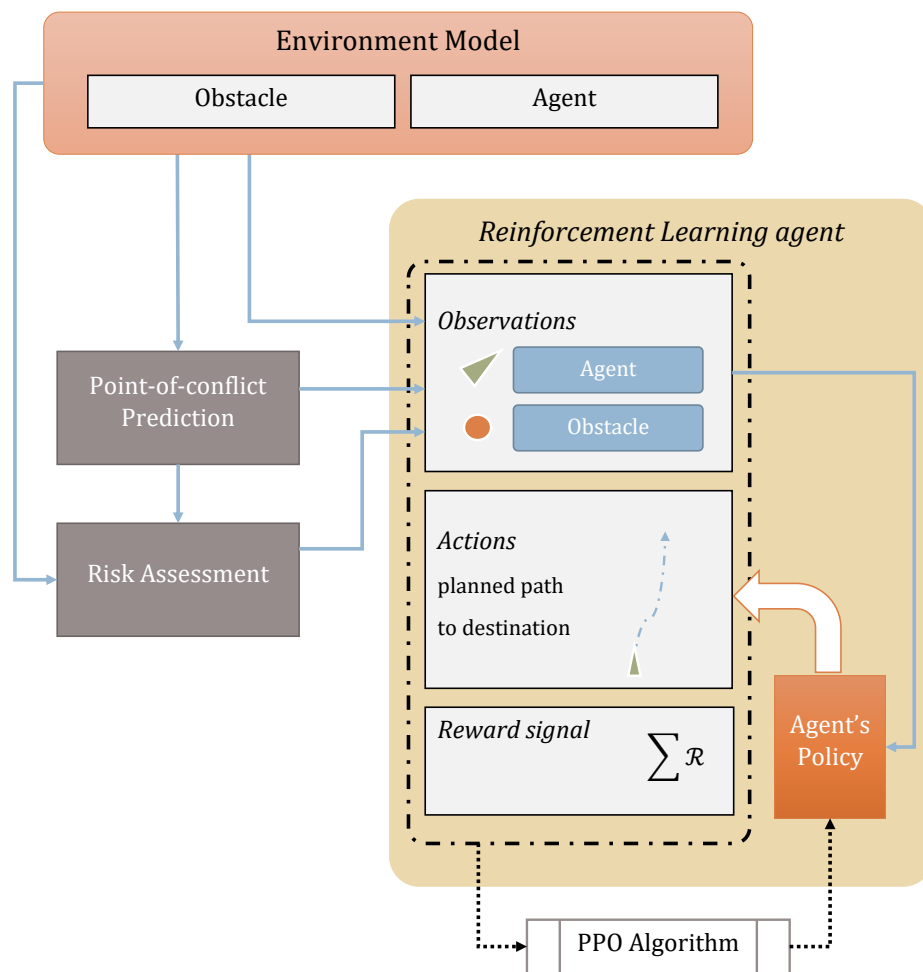


Figure 5.1: Overview of the pedestrian path-planning model.

the overfitting problem. This means the agent could learn to navigate in the training environment, but its knowledge could not be transferred into unfamiliar environments.

For that reason, our environment is designed to have a fixed area size, and also there is only one obstacle that may exist inside. A complex environment would be scaled down or divided into multiple parts, depending on the situation. There are several methods to realize this. For instance, in a study by Ikeda et al. [19], the agent would treat each component navigation part as its sub-goal when planning the route to a certain location. This would greatly help stabilize the training process while still is able to expand its applicability to new environments.

Consequently, our environment is modeled as illustrated in Figure 5.2. The area of the environment is 22 meters by 10 meters. The position of the agent is randomized between the coordinates $(-5, -12)$ and $(5, 12)$. The agent’s current destination is randomized between the coordinates $(-5, 10)$ and $(5, 10)$.

The navigation path from the agent’s position to its current destination consists of 10 component nodes whose coordinates’ y values are predefined. The x coordinates of these nodes correspond to 10 outputs of the neural network. This will be presented in more detail in Section 5.2.

This modeling of the environment is similar to the concepts of spatial knowledge in the human cognitive system. The environment modeling conforms to the representation of survey knowledge, and the planned path of the agent conforms to the concept of route knowledge. These are used by the hippocampus to form the cognitive map for planning and making decisions.

5.3 Path-planning navigation training

The path-planning training utilizes reinforcement learning for the pedestrian agent to learn the navigation behavior. In reinforcement learning, the agent needs to continuously observe the states (usually partially) of the environment and subsequently take appropriate actions. These actions would be rewarded using the rewarding functions to let the agent know how good these actions are.

For the training task, the model utilizes the PPO reinforcement learning algorithm. The training model uses the observations of the agents, the agent’s actions

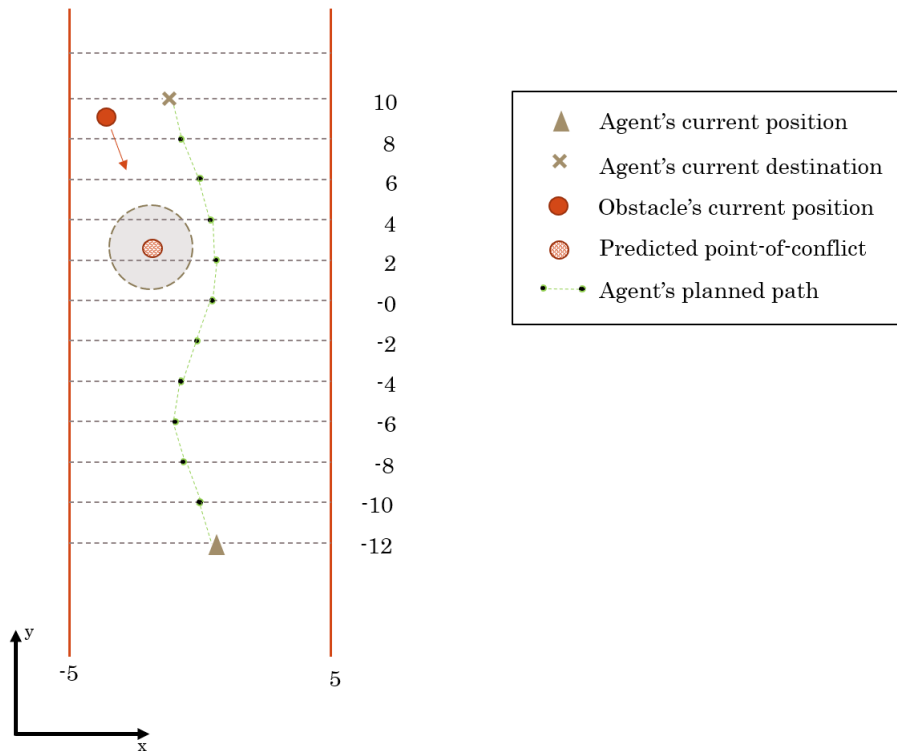


Figure 5.2: Path-planning environment modeling

based on the current policy, and the resulted reward of the actions to train in a neural network to output an optimized policy. In the inference phase, the agent could use the policy to decide which actions to take based on the current observations. In our model, the POC prediction and the risk assessment tasks affect the observations of the agent in the inference phase, but they are not implemented in the training phase.

Consequently, the following issues need to be addressed: modeling a learning environment, specifying the agent's observation of the environment and actions taken, and rewarding for the agent's actions.

5.3.1 Environment modeling

The environment is modeled as presented in Figure 5.2. For the learning task, the chance of an obstacle appearing in the environment is randomized in each training episode. In the case of the obstacle's appearance, its size is randomized between 0.5 and 2, and its danger level is randomized between 0 and 1. The entire environment might be scaled along its length (the y axis as in Figure 5.2)

so that the agent could adapt its actions better to different real-life environments. Accordingly, in each training episode, the environment’s scale will be randomized between 0.2 and 1.

The training episode is finished immediately when the path is planned, conforming to the real-life path-planning process. Also correspond to the path-planning process in real life, the agent’s action is the entire planned path to the destination. This also means each training episode has exactly one step, and the environment’s states will be randomized in the next step. In addition, the agent’s actions, in this case, cannot affect the states of the environment, also similar to the human planning process. In real life, apart from planning the path, the agent could not take any further actions until he needs to actually carry out the navigation, which will be later discussed in Chapter 6. As a result, the agent will need to collect all necessary information from the environment and quickly finish the task by constructing a planned path. Alternatively, the path-planning process could be realized by simulating the navigation path and perform an optimization method on the path. However, this method is not appropriate because, in real life, the human pedestrian carries out the planning process by following their intuition based on their previous experience. For this reason, training the policy using reinforcement learning for the path-planning task is a more proper method. In particular, when using an algorithm like PPO, we could construct a policy for the agent to form the planned path in one step similar to human pedestrians via learning through multiple experiences with various states of the environment.

A problem with this resetting mechanism is that this could cause the environment to be much noisier, which could lead to subsequent problems with the training of the neural network. An example of this is when the training environment has an obstacle in an episode, but no obstacle in the next one. In this case, even if the agent could not plan a path that successfully avoids the obstacle in the first episode, it is easy for the agent to do that in the second one and achieve a more favorable reward. This makes the agent accommodate the newer policy despite it may achieve worse results than the previous one. With a noisy environment like this, it would take much longer for the neural network to successfully converge the cumulative reward, and occasionally the policy could not be improved any further due to its inability to notice a better policy over the timesteps. To prevent this, we add another resetting mechanism for our environment. Instead of resetting immediately, we only reset the environment if the

agent could plan a path without conflicting with the obstacle. Otherwise, the current states are kept so that the agent could try planning again. If the agent takes over a predefined number of steps without being able to plan a successful path, we also need to reset the environment, or the agent could be stuck in finding the appropriate policy.

Regardless of whether the environment is reset or not, the training episode is terminated every step, conforming to the path-planning process of human pedestrians. The PPO algorithm, however, will always collect a predefined number of steps M to put into a minibatch to optimize the training. That means each minibatch contains the data from M episodes or steps. By randomizing the environment using the aforementioned mechanism, the neural network would be provided with a sufficient amount of data for policy optimization.

For the advantage values \hat{A}_t to be specified, a total of T episodes will be performed, with T is the time horizon value. Subsequently, the algorithm specifies the advantage values using formula 3.5 and uses that to form the loss function. Because each episode has only one step, instead of considering the future states, the algorithm considers the different states of the environment through stochastic initialization after reset. As a result, the agent could update the policy while still considering the other states of the environment. For example, in the case that the obstacle is not on the direct path to the destination, the agent would still consider the existent of the obstacle instead of updating the policy so that the navigation always heads straight to the destination when updating the policy. However, in the case that the agent fails to plan the path without colliding with the obstacle, the agent would be given more chances to optimize the policy to avoid the obstacle as we do not reset the environment in that case. While this could be less sample efficient than other off-policy methods, this is not a problem as the simulation tool allows running millions of episodes in a few hours. An advantage of this method is that the advantage value could be precisely calculated and no value estimation needs to be formulated.

5.3.2 Agent's observations and actions

In each step, the agent will observe the following states:

- x position of the agent's position;

- x position of the agent’s destination;
- Whether the obstacle appears in the environment or not;
- (if the obstacle is present) The obstacle’s position, size and risk;
- The scale of the environment.

The y positions of the agent’s position and destination do not need to be observed, as they are constantly determined in the modeling of the environment, as presented in Section 4.2.

For the learning task, the risk of the obstacle has the same value as its danger level. The purpose of this is to let the agent learn how to act differently with diverse values of risk. This does not teach the agent how to assess the risk from the obstacle’s danger, however, as this would be carried out in the prediction task of the agent.

We need to specify the actions that the agent takes following its observations. In our model, these are a set of 10 values corresponding to the x coordinates of the navigation path. Each output is mapped to the x coordinate of the navigation nodes. Specifically, assuming the outputs of the network are $x_1, x_2, x_3 \dots x_{10}$, the navigation of the agents would be the path through the following nodes: $(x_1, -10)$, $(x_2, -8)$, $(x_3, -6) \dots (x_{10}, 8)$, and finally, the agent’s destination.

5.3.3 Rewarding formulation

The rewards are used to tell the agent how good its taken actions are, which in this case are the planned path to the destination. Rewarding is an essential task in any reinforcement learning model. Different from rule-based models, rewarding is usually based on the results of the agent’s actions or the effect of the agent’s actions on the states. For a natural behavior to be conducted, the rewards should correspond to how humans view the navigation behavior as natural or not, or how comfortable the humans would feel when observing the movement. In this regard, Kruse et al. [25] proposed the idea of human comfort. This idea introduces a number of factors in movement that could help the observing humans to feel comfortable, consequently perceiving the movement to be more human-like.

Within the scope of our study, we choose to adopt the following factors for our rewarding mechanism:

- Choosing the shortest path to the destination;
- Avoiding frequently changing direction;
- Following basic navigation rules and common-sense standards;
- Colliding with obstacles.

The first factor, which is also considered a decisive factor in many studies, is to plan the shortest path to the destination. While in real life navigation, human pedestrians may subconsciously aim at the shortest navigation time, they still consider shortest path to be the highest-ranking factor, as in a study conducted by Golledge [34]. As each rewarding factor correlates with the aspect that the human pedestrian is aiming at or wants to achieve, planning the shortest path would be formulized.

Consequently, we calculate the rewarding for this behavior by placing a negative reward corresponding to the sum of the squared length of each component path. This means if the path is longer, the agent would receive a larger penalty. This rewarding is formulated as follows:

$$\mathcal{R}_1 = -\lambda \sum_{i=0}^{11} \|p_i\|^2, \quad (5.1)$$

where λ is the environment's scale, and p_i is the vector of each component path.

The following factors are the essential behaviors to ensure safety in interactions with others. More specifically, accidents could happen when a person abruptly changes direction or does not follow the flow of the navigation within the environment. If a person navigates in that way, others would view him as a possible risk, therefore that behavior could be considered unnatural.

Regarding the rewarding for changing direction, we only consider the changes in angles which are larger than 30° . Any changes in angles which are smaller than this could be acceptable and are still considered natural. For this reason, we formulate the rewarding for this behavior by placing a penalty each time there is a large change of direction in the planned path as follows:

$$\mathcal{R}_2 = - \sum_{i=0}^{10} \theta(\text{angle}(p_i, p_{i+1})) , \quad (5.2)$$

where $\text{angle}(p_i, p_j)$ is the angle value between the vectors p_i and p_j ; $\theta(x)$ is the Heaviside step function, specified by

$$\theta(x) = \begin{cases} 0, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0. \end{cases} \quad (5.3)$$

As for the rewarding based on following basic navigation rules and common-sense standards, the rules may vary between different regions and cultures. From our observation, the following rules are applied in our study:

1. Following the flow of navigation by walking parallel to the sides.
2. Walking on the left side of the road. While pedestrians are not required to strictly follow this, in real life, people still choose to follow this as a general guideline to avoid accidents. Similarly, in right-side walking countries, pedestrians would choose to walk on the right side of the road.
3. Avoiding getting close to the sides.

To define the appropriate rewarding formulations, the planned path of the agent is sampled into N values s_i with i ranges from 0 to N . The respective rewarding functions are calculated as follows:

$$\mathcal{R}_3 = -\lambda \sum_{i=0}^N \theta(\|x_{pos}(s_{i+1}) - x_{pos}(s_i)\| - H_1) , \quad (5.4)$$

$$\mathcal{R}_4 = -\lambda \sum_{i=0}^N \theta(-x_{pos}(s_i)) , \quad (5.5)$$

$$\mathcal{R}_5 = -\sum_{i=0}^N \theta(\|x_{pos}(s_i)\| - H_2) , \quad (5.6)$$

where $x_{pos}(s_i)$ function returns the x coordinate of the point s_i .

The value H_1 in equation 5.4 is the threshold value for the difference in x coordinates that the agent could make in each sample navigation part. The smaller difference in x coordinates of the navigation produces the path that is more parallel to the sides. In our model, with $N = 200$, H_1 is given a value of 0.4. In addition, our model will put a negative reward on the agent whenever its

x coordinate is less than 0 as in equation 5.5, meaning the agent is at the left side of the road. Regarding equation 5.6, as suggested in other studies [7, 12], the agent would stay approximately 0.5 meters from the walls to avoid possible accidents. In our model, the navigation path has a width of 10 meters, therefore the value H_2 is set to 4.5 so that when the agent’s position has an x coordinate higher than 4.5 or less than -4.5 , it would receive a negative reward.

Lastly, with respect to collision avoidance, the agent needs to keep a certain distance from the obstacle. The highest risk would seemingly be at the center of the obstacle, and the risk gradually decreased with longer distance. However, once the agent has reached a certain distance with the obstacle, any further than this would be unnecessary. For example, if the pedestrian in real life would like to avoid stepping on a puddle, as long as the navigation path does not conflict with the puddle, it does not matter if the path needs to be much further away from it. Because of this reason, we formulate our rewarding for the collision avoidance behavior as follows:

$$\mathcal{R}_6 = \sum_{i=0}^N \begin{cases} \frac{\delta(s_i, obs)}{R_{obs}^2} r^2, & \text{if } \delta(s_i, obs) \leq 0, \\ 0.01 r^2, & \text{if } \delta(s_i, obs) > 0, \end{cases} \quad (5.7)$$

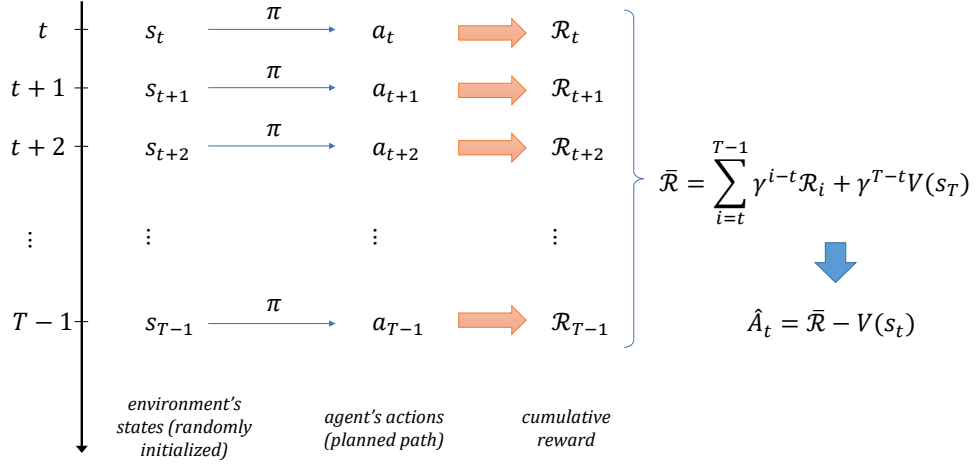
with $\delta(s_i, obs) = d(s_i, obs)^2 - R_{obs}^2$, where $d(s_i, obs)$ is the distance from the sampled position s_i and the obstacle; R_{obs} is the radius of the obstacle’s area; and r is the risk from the obstacle. In the training task, r has the value of obstacle’s danger, as presented in Section 5.2.

The resulted cumulative reward \mathcal{R} that is given to the agent each episode is the sum of all components rewards multiplied by the corresponding coefficients:

$$\mathcal{R} = \sum_{i=1}^6 \mathcal{R}_i \kappa_i, \quad (5.8)$$

where κ_i is the coefficient of the appropriate reward.

Each variation of a set of κ_i results in a different personality in the agent’s path planning process. In real life, different people have different priorities in how the navigation path is formed. For example, to simulate the pedestrian who prioritizes following the regulations, the coefficient for \mathcal{R}_4 , walking on the left side, should be higher. Similarly, to replicate the behavior of a cautious pedestrian, the model should use a higher value for \mathcal{R}_6 , obstacle avoidance rewarding.


 Figure 5.3: Determining the advantage value \hat{A}_t .

In PPO algorithm, to calculate the advantage value, a total of T steps must be carried out using the current policy, where T is the time horizon value. In our path-planning task's environment modeling, this corresponds to T episodes as each episode has only one step. This means the agent will output a series of actions a_t , each action is an array of 10 values to form the path to its destination, from a series of state s_t with $t \in [t, T]$. Noted that the same policy will be utilized for the agent to generate the action a_t from the state s_t .

Consequently, with $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_T$ determined, the advantage values $\hat{A}_1, \hat{A}_2, \dots, \hat{A}_T$ are determined by following the formula 3.4:

$$\hat{A}_t = -V(s_t) + \mathcal{R}_t + \gamma \mathcal{R}_{t+1} + \dots + \gamma^{T-t-1} \mathcal{R}_{T-1} + \gamma^{T-t} V(s_T), \quad (5.9)$$

with $t \in [1, T]$. This process is illustrated in Figure 5.3.

The objective of the algorithm is to maximize the L^{CLIP} value in the formula 3.6. That means we need to optimize the policy so that the advantage value is maximized.

$$\hat{A}_t = \bar{\mathcal{R}} - V(s_t) \quad (5.10)$$

where $\bar{\mathcal{R}} = \sum_{i=t}^{T-1} \gamma^{i-t} \mathcal{R}_i + \gamma^{T-t} V(s_T)$ with \mathcal{R}_i is the cumulative reward from the agent's action (the planned path) at the state s_i . \mathcal{R}_i measures how good the policy is under the state s_i .

Consequently, the value $\bar{\mathcal{R}}$ indicates how good the current policy is under multiple states s_t, \dots, s_{T-1} . Accordingly, by initializing the environment’s state, the neural network is able to optimize the current policy under consideration of other states of the environment, which means maximizing the weighted sum $\sum_{i=t}^{T-1} \gamma^{i-t} \mathcal{R}_i$.

For that reason, the agent’s action a_t cannot affect the state of the environment, therefore s_{t+1} does not depend on the action a_t . This conforms to the real-life human planning process, in which the planning is unable to alter the environment’s conditions. As a result, with the environment being initialized with random states every step, the neural network could train the agent’s policy to be able to plan the appropriate path that maximizes the cumulative reward in any environment.

The discount value γ , instead of adjusting the effect of the agent’s action on the future state, is responsible for how the neural network considers the variety of environment’s states while training the agent’s current policy. In our study, we employed $\gamma = 0.99$.

5.4 Point-of-conflict prediction

To accurately simulate the navigation of a pedestrian, the incorporation of the prediction is necessary. This prediction might not be accurate, as humans in real life usually make inaccurate predictions. As a result, the prediction process in our model also focuses on replicating a similar prediction mechanism.

We proposed a concept called *point-of-conflict* (POC), a location within the environment that the agent thinks could collide with the obstacle or at the predicted position of the obstacle when it is closest to the agent [20]. Even in the case of a low chance of collision (e.g. when the agent and the obstacle are navigating on two sides of the road), a POC is still predicted. The motivation is that, when the human has already learned the appropriate prediction method, the prediction process would occur in most cases. This would happen naturally inside human cognition without much reasoning.

When the prediction task is handled, the agent would use the information from the POC instead of the actual obstacle in the path-planning training task as introduced in Section 4. The location of the POC will be predicted by the agent

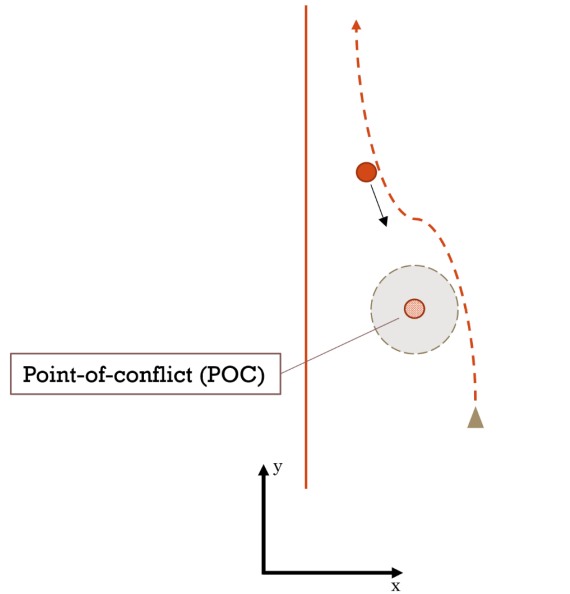


Figure 5.4: Obstacle avoidance with point-of-conflict.

depending on the obstacle’s type, which will be demonstrated in more details subsequently. Figure 5.4 illustrates the path-planning process of the agent after the prediction task is utilized.

The position of the POC depends on the type of obstacle. For example, if the obstacle is *stationary*, the POC’s position should be the same as the position of the obstacle. Apart from stationary obstacle, we define two other obstacle’s types: *single diagonal movement* obstacle, and *pedestrian* obstacle. Each type of obstacle has a different method of calculating the POC’s position. To simplify the prediction of the POC, we assume the agent has the information of the obstacle’s speed and heading direction. It is worth noting that the heading direction is the direction toward the obstacle’s destination instead of its current orientation. This is because when moving, the pedestrian may not always heading toward his destination, but could turn in another direction for various reasons (e.g. steering to the left-hand side). There have been several studies addressing the problem [66, 67, 71], which could be applicable to our study.

5.4.1 Single diagonal movement obstacle

A single diagonal movement obstacle is an obstacle that is mostly moving in one direction and with a uniform speed. Some examples of this obstacle’s type are

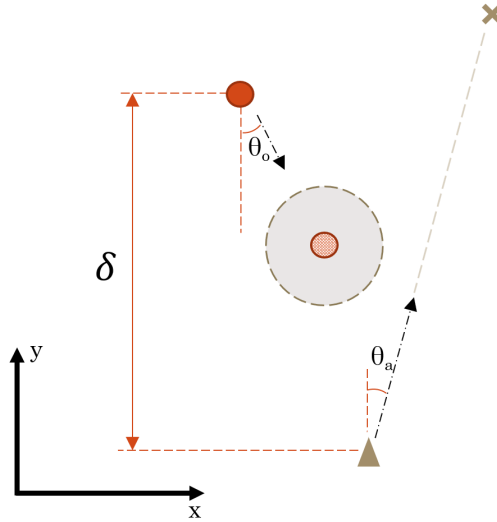


Figure 5.5: Point-of-conflict of a single diagonal movement obstacle.

a pedestrian crossing the environment or a road construction machine moving slowly on the sidewalk. This type of obstacle does not include a vehicle moving at normal speed. In that case, the pedestrian agent should exclude its navigation area from the model's environment, as it would be too dangerous to navigate inside that area.

Figure 5.5 illustrates the POC prediction process in the case of a single diagonal movement obstacle. In order to specify the area of the POC, we need to figure the approximate time until the obstacle is getting close. As the prediction process is carried out before the path-planning task, we could only estimate this using the agent's general direction toward its destination. The calculation for this approximate time is formulated as:

$$t = \delta \frac{v_{obs}}{v_{agent} \cos \theta_a + v_{obs} \cos \theta_o} \quad \text{if } (v_{agent} \cos \theta_a + v_{obs} \cos \theta_o) > 0, \quad (5.11)$$

where δ is the distance in y coordinate between the agent and the obstacle, v_{agent} and v_{obs} are the velocity of the agent and the obstacle; θ_a is the agent's direction angle relative to the upward vertical axis, and θ_o is the obstacle's direction angle relative to the downward vertical axis.

As a result, the POC's position (x_{POC}, y_{POC}) is specified as follows:

$$(x_{POC}, y_{POC}) = (x_{obs}, y_{obs}) + t \lambda v_{obs} \hat{e}_{obs}, \quad (5.12)$$

where (x_{obs}, y_{obs}) is the position of the obstacle, $\hat{\mathbf{e}}_{obs}$ is the unit vector having the direction of the obstacle, and λ is the environment's scale as presented in Section 5.1.

If the $(v_{agent} \cos \theta_a + v_{obs} \cos \theta_o) \leq 0$, it is unlikely for the agent to collide with the obstacle. In this case, the POC is omitted in the planning task of the model. In addition, if the calculated POC's position is outside the range of the agent's environment, the POC is also ignored in our model.

5.4.2 Pedestrian obstacle

Pedestrian obstacles are usually the most common type of obstacle that could interact with the pedestrian agent. However, the definition of pedestrian obstacle in our study does not include a pedestrian crossing the environment, as it is considered as a single diagonal movement obstacle discussed above. To predict the position of a POC, the agent needs to specify the navigation path that the obstacle might take. While the model for single diagonal movement obstacle could also be adopted in this case, its result would be fairly inaccurate, and more importantly, does not conform to the human predictive system.

For that reason, we have proposed a unique method of predicting the POC for a pedestrian obstacle. Firstly, to define the predicted navigation path of the obstacle, we utilized our existing reinforcement learning path-planning model. By doing this, the predicted navigation path would have the same advantage as our reinforcement learning model and therefore could replicate a realistic navigation path. Subsequently, the POC will be specified on that navigation path, using the velocity of the agent and the obstacle. Figure 5.6 represents the POC's prediction in the case of a pedestrian obstacle.

Before the obstacle's navigation path could be constructed, its estimated destination needs to be determined. This could be achieved by projecting the obstacle's orientation to the end of its navigation environment (separate from the agent's environment). The projected destination (x_{Dobs}, y_{Dobs}) could be formulated as follows:

$$(x_{Dobs}, y_{Dobs}) = \left(x_{obs} - \frac{\lambda L v_x}{v_y}, y_{obs} - \lambda L \right), \quad (5.13)$$

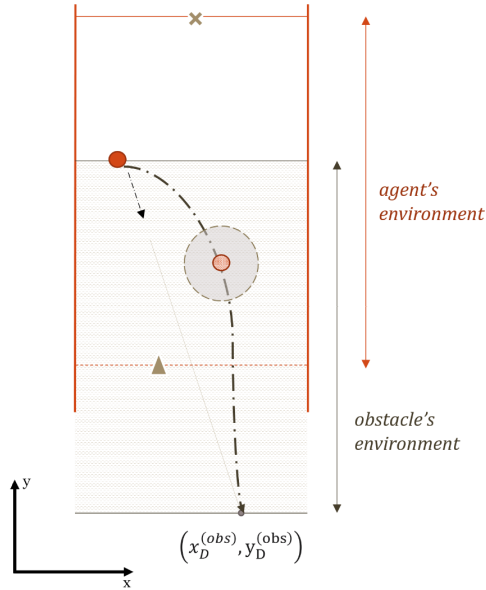


Figure 5.6: Point-of-conflict of a pedestrian obstacle.

where (x_{obs}, y_{obs}) is the obstacle's position, (v_x, v_y) is the orientation vector of the obstacle and L is the length of the obstacle's environment. In our proposed model's environment, L has a length of 22 meters.

The observations of the obstacle consist of the obstacle's position and its projected destination. The observations do not include the observation of an obstacle (i.e. the pedestrian agent in the obstacle's environment) for two reasons. The first reason is that the POC prediction happens before the path-planning process, therefore the obstacle can't specify the agent's path. Trying to specify the paths of the agent and the obstacle at the same time would certainly cause conflict. Another reason is related to the process of human thinking in real life. When a pedestrian is predicting the navigation path of the obstacle, he would not consider himself as an obstacle, but rather trying to navigate in a way that could avoid a collision.

The RL model used in our obstacle's path-planning process is the same one used by the pedestrian agent. The reason is that usually a person often thinks other people would act the same way, for example, navigating the same way as he would do. Alternatively, the obstacle could use the mean RL model from multiple training.

The predicted position of the POC could be subsequently determined using

the scale between the velocities of the agent and the obstacle. The calculation of the POC's y coordinate is formulated as follows:

$$y_{POC} = y_{obs} - \delta \frac{v_{obs}}{v_{agent} + v_{obs}} , \quad (5.14)$$

where v_{agent} and v_{obs} are the velocity of the agent and the obstacle, respectively; δ is the difference in the y axis between the agent and the obstacle.

Finally, the location of the predicted POC is specified by the point on the pedestrian's navigation path at the y_{POC} value in the y axis.

5.5 Risk assessment

With the point-of-conflict prediction, the pedestrian agent would observe the POC's position and assess its risk instead of using the obstacle's position and danger level. As previously discussed in Section 4.1, risk is calculated based on the obstacle's harm and the probability of collision, which is formulated as

$$r = harm \cdot P , \quad (5.15)$$

where r is the risk, $harm$ is the possible harm caused by the obstacle and P is the probability of collision with the obstacle. To conform to the agent's observations in our reinforcement learning model, all values r , $harm$ and P have a range from 0 to 1.

To estimate the probability of collision, we need to specify the proximity of the POC's position to the navigation path. Because the risk assessment is carried out before the path-planning task, the navigation path could be approximated as a straight line from the agent's position to its current destination. Its line formula could be represented by

$$\frac{x - x_a}{x_D - x_a} = \frac{y - y_a}{y_D - y_a} , \quad (5.16)$$

which equals to the following general linear equation

$$(y_D - y_a)x + (x_a - x_D)y + (x_D - x_a)y_a - (y_D - y_a)x_a = 0 . \quad (5.17)$$

Considering $(y_D - y_a) = A$, $(x_a - x_D) = B$, and $(x_D - x_a)y_a - (y_D - y_a)x_a = C$; the distance from the POC's position (x_{POC}, y_{POC}) and the line above is calculated as follows:

$$\delta_{POC} = \frac{|Ax_{POC} + By_{POC} + C|}{\sqrt{A^2 + B^2}}. \quad (5.18)$$

The collision probability P is highest when $\delta_{POC} = 0$, and gradually decline with higher δ_{POC} . P is formulated in our model as follows:

$$P = 1 - \frac{\delta_{POC}}{\delta_{POC} + M}, \quad (5.19)$$

where M is a distance constant. When $\delta_{POC} = M$, the collision probability P would be at 0.5. For that reason, we adopted using $M = 3$ in our implementation.

To estimate the harm from the obstacle, we use the obstacle's danger level and also its speed, as the speed could also impact the harm caused by the obstacle [58]. As an example, the risk observed from a person running at a high speed should be higher than the risk observed from a person walking at a normal speed toward the agent, even when the perceived danger from the two persons is the same.

Arguably, the obstacle's speed could also contribute to the probability of the agent's avoidance. However, because the agent's navigation path was not formed at the current process, the avoidance probability is unspecified. Assuming the capability of avoidance of the pedestrian agent is constant, the obstacle's speed should not affect the probability of collision P .

In the case that the obstacle's speed is irrelevant, such as a static obstacle, the harm of the obstacle is equivalent to its danger level.

Otherwise, we adopt using the concept of kinetic energy to estimate the harm of the obstacle, similar to how humans feel the impact of a moving object when it hits. As a result, harm is formulated as

$$harm = \max \left(1, danger \left(1 + \gamma \frac{K_{obs}}{K_{normal}} \right) \right), \quad (5.20)$$

where K_{obs} is the kinetic energy of the moving obstacle, K_{normal} is the kinetic energy of an object moving at a normal speed, and γ is the discount value.

Considering $K = \frac{1}{2}mv^2$, the harm of a moving obstacle could be formulated as follows:

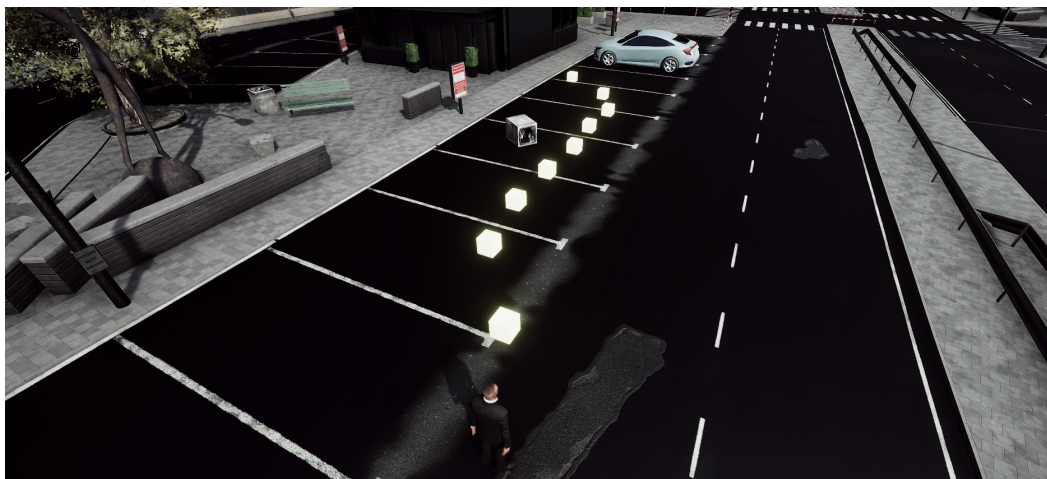


Figure 5.7: Path-planning task implementation screenshot.

$$harm = \max \left(1, \text{danger} \left(1 + \gamma \left(\frac{v_{obs}}{v_{normal}} \right)^2 \right) \right), \quad (5.21)$$

where v_{normal} is the average speed of a moving object which could be perceived as normal. In several studies [59, 60, 61], v_{normal} is specified to be approximately $1.31m/s$.

Finally, with the harm value calculated, the risk of the obstacle is formulated by equation 5.15. This risk value together with the POC’s position specified in Section 6, is used in the agent’s observations in the pedestrian reinforcement learning model. More specifically, regarding the obstacle’s properties, the pedestrian agent will observe the POC’s relative coordinates, the obstacle’s size and the risk formulated in this section. Consequently, the formulated reward in equation 5.7 is updated (which, in the training process, uses the same value of the obstacle’s danger for its risk). This could result in a more precise navigation path in a similar way the path is planned by a human pedestrian.

5.6 Implementations

The model of our study was implemented using the real-time development platform Unity. The source code is available at <https://github.com/trinhthanhtrung/unity-pedestrian-rl>, by opening the scene *PathPlanningTask* within the *Scene* folders. Figure 5.7 presents a screenshot of our implemented application.

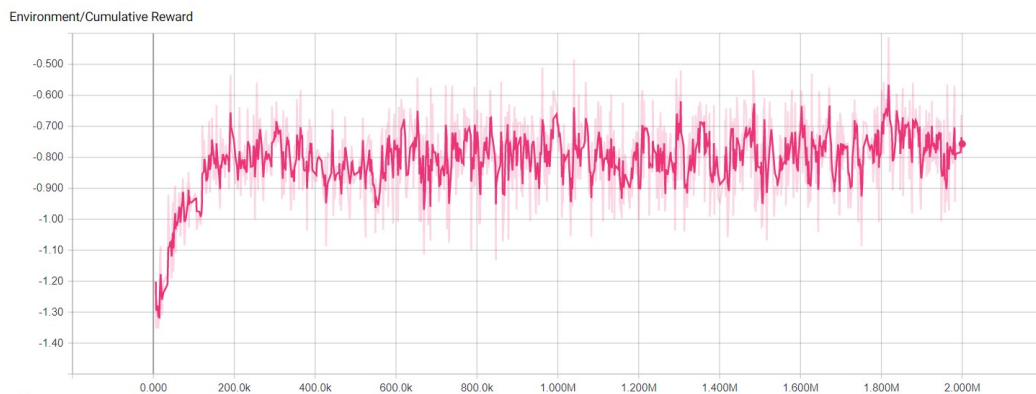


Figure 5.8: Cumulative reward statistics.

For the training task of the pedestrian agent, we adopted the reinforcement learning library ML-Agents [26], which acts as a communicator between Unity and Python machine learning code. In each training episode, the information of the model, consisting of the agent’s observations and actions, and the cumulative reward value, is sent to Python. The information is subsequently used for training the agent’s policy in a neural network using the PPO algorithm, then the updated policy will be sent back to the pedestrian agent.

Because the environment’s states are moderately noisy, it is recommended to train the agent with a large batch size. We utilized 2 hidden layers, each consisting of 128 hidden nodes. Furthermore, multiple instances of the same training environment are created to speed up the training process. We have been able to successfully get the cumulative reward to converge after two million steps with a learning rate of 2.3×10^{-4} and time horizon of 512. For a smoother navigation path, we used the mean of the agent’s actions in multiple episodes of the same environment’s state. Figure 5.8 shows the statistics of the training process in TensorBoard.

The coefficient parameters for the rewarding components are adjusted so that the resulted navigation behavior closely matches the experiments conducted in our laboratory. Figure 5.9 shows a screenshot from our experimental video datasets. In our experiment, we have two pedestrians. One pedestrian acts as a pedestrian obstacle, navigating with a predefined script; and the other acts as the pedestrian agent, walking to the destination while avoiding the collision.

Figure 5.10 shows the planned path of the agent in different situations in our implementations. In these figures, the actor model at the bottom is the

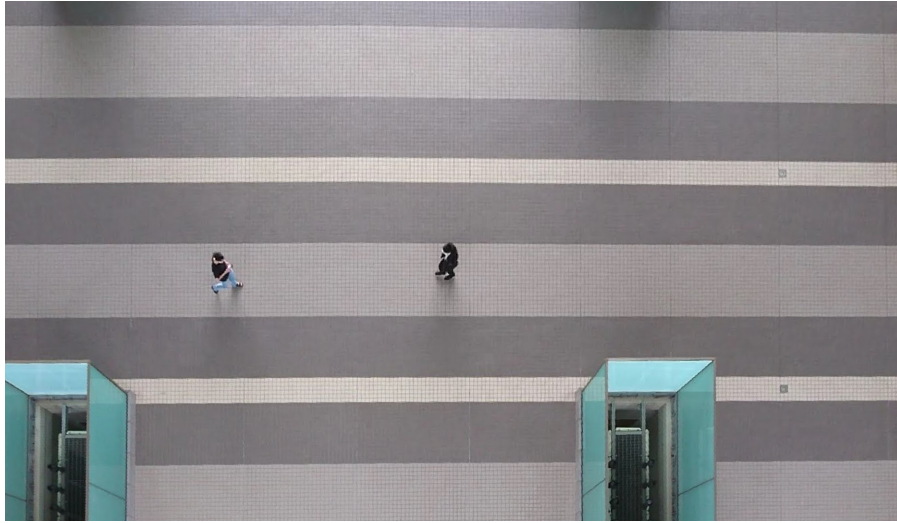


Figure 5.9: Screenshot from path-planning model experimental dataset.

pedestrian agent, the red point on the top is the agent’s destination, the black circle represents the predicted POC by the agent, and the red circle (covered by the POC in (b) and (c)) is the current obstacle.

We have implemented an SFM model to evaluate our model. The parameters of the SFM agent are calibrated so that the resulted behavior matches the conducted experiment as previously presented. The repulsion parameters are adjusted so that the distance between two persons is often higher than a comfort distance (around 1.5m in our experiment). Figure 5.11 shows how the implementation of our model compares to SFM. In each figure, our implementation is on the left and the SFM implementation is on the right (darker environment).

From observation in Figure 5.10, the agent could be able to plan a sufficiently realistic path to the destination, while also considering the rules and following common conventions like walking to the left side and naturally changing direction. This can be seen in situation (a), where the agent has chosen to walk on the left side of the road and gradually move toward the destination when needed, instead of walking straight to the destination. Although the planned path was constructed from the outputs of the neural networks, it is shown to be remarkably stable. The agent has also shown its capability to avoid the obstacle, as the planned path does not collide with the obstacle or its prediction in most situations.

Furthermore, its planned path also seems to be adapted to the risk from the obstacle. This is observable by comparing the paths planned by the agent in

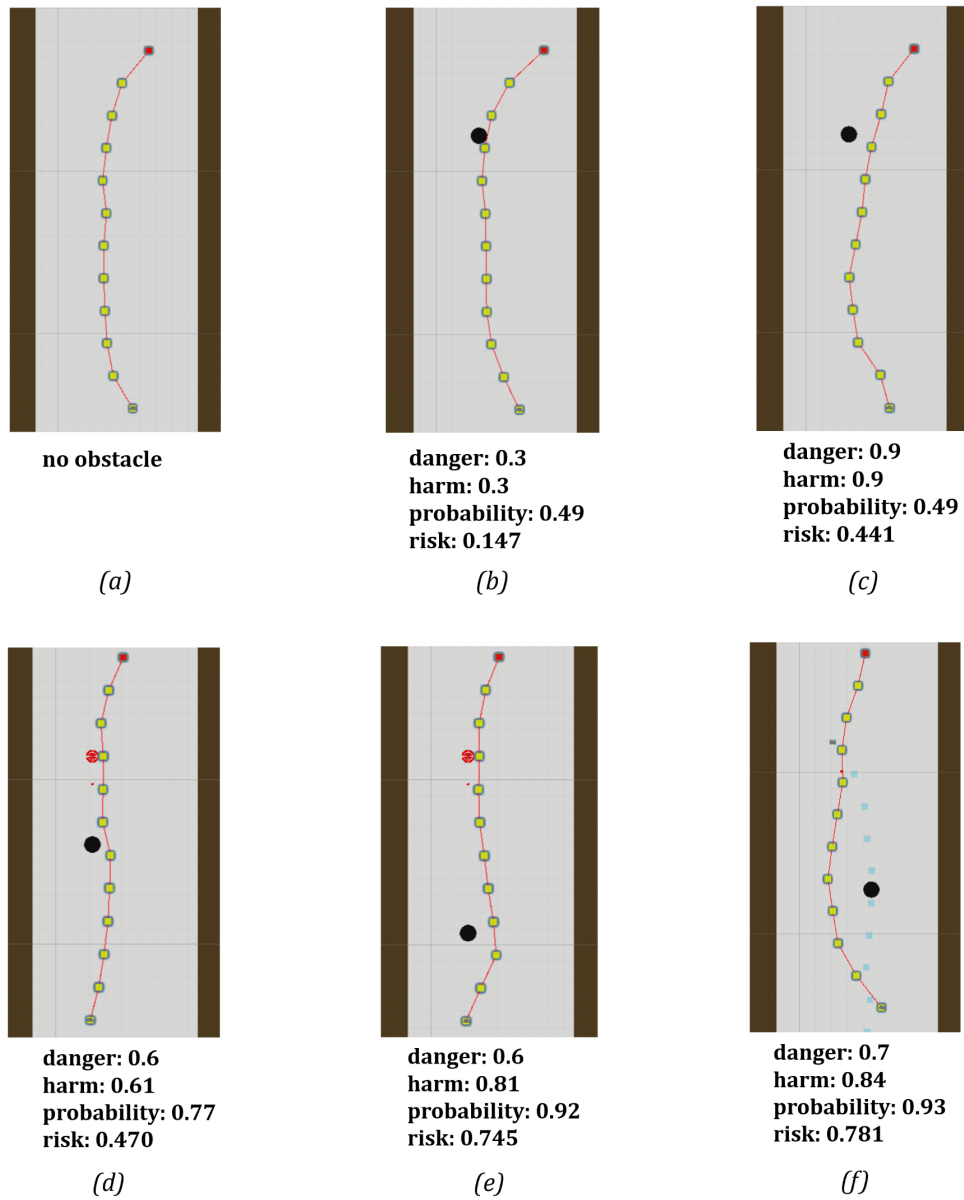


Figure 5.10: Agent's planned path in different situations: (a) no obstacle; (b) with a static obstacle with a low danger level; (c) with a static obstacle with a high danger level; (d) with an obstacle moving straight in one direction away from the agent (e) with an obstacle moving straight in one direction toward the agent; (f) with a pedestrian obstacle.

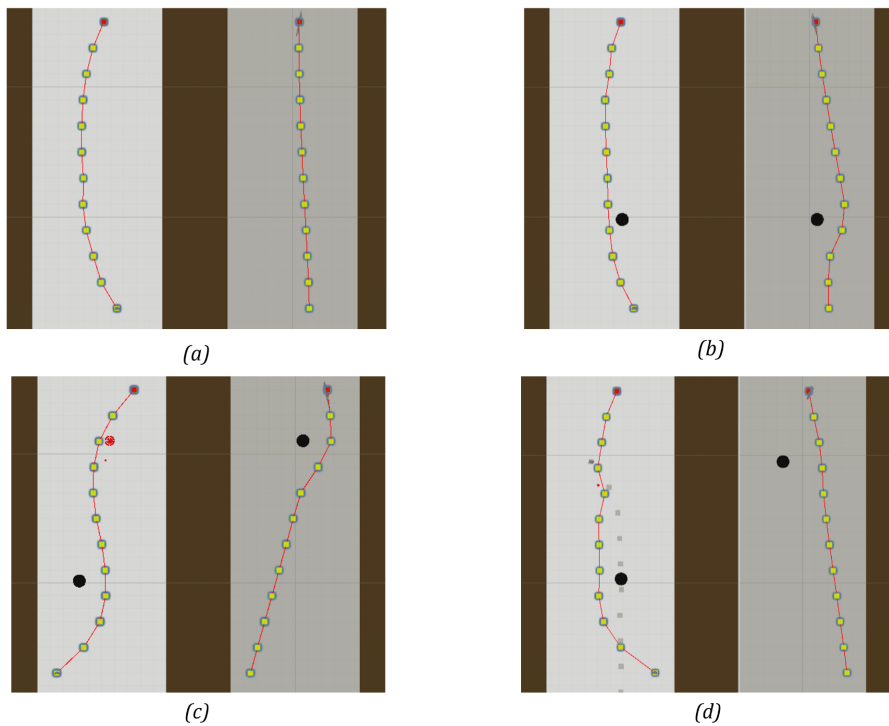


Figure 5.11: Comparison with SFM in different situations: **(a)** no obstacle; **(b)** with a static obstacle; **(c)** with a moving obstacle; **(d)** with a pedestrian obstacle

(b) and (c). We implemented the obstacles to have the same properties in both situations; however, the obstacle in (c) has a much higher danger level than the obstacle in (b). The result is that in Figure 5.10.b, the agent only almost avoided colliding with the obstacle, while in (c), the agent chose a path that steers much further away from the obstacle than in (b). This resembles actual human thinking when planning a navigation path where there is a dangerous obstruction on the road.

Figures 5.10.d and 5.10.e demonstrate how the agent adopts the prediction process into path planning. In both situations, the agent planned the path to avoid the possible point-of-conflict instead of the actual current position of the obstacle, which is similar to how an adult person plans the navigation path. However, the obstacle in (e) was moving at a higher speed, therefore the risk perceived from the obstacle is higher. As indicated in the figure, the *harm* value calculated in (d) was 0.61, compared to the higher *harm* value calculated in (e) at 0.81. Additionally, the difference in the POC's position causes the change in their probability estimations, which are 0.77 in (d) and 0.92 in (e). The increased probability also contributes to a higher resulted risk specified in (e) situation

(0.745 in *(e)*, compared to 0.470 in *(d)*). This was reflected in the navigation path by the pedestrian agent, as it is shown that the agent could plan the path to quickly avoid the possible collision. The risk formulation in *(d)* and *(e)* has shown that it could be greater to or less than the obstacle's danger level (0.6 in both situations) depending on the speed of the obstacle, consistent with human thinking in real life.

Figure 5.10.f presents the planned path of the pedestrian agent in the case of another pedestrian obstacle. In this case, the obstacle's path was formed to predict the possible collision, which also resembles the human thinking process when a pedestrian trying to avoid another person while walking.

Additionally, a survey was provided to further assess the human likeness of our model, compared with the SFM implementation. The objective of the questionnaire is to specify how real humans evaluate the experimental results. Therefore, we have prepared a questionnaire in Google Form format and send it to several people. To let people determine the level of human likeness in our model compared to the SFM implementation, we provided the 4 situations in the implementation results presented in Figure 5.11. Our implementation results were placed on the left and the SFM implementation results were place on the right. However, the participants were not informed of which one is ours and which one is SFM's. In each situation, people could choose a number between 1 to 5, with the following denotations:

1. The left one is much more natural
2. The left one is slightly more natural
3. They are similarly natural
4. The right one is slightly more natural
5. The right one is much more natural

A screenshot of the questionnaire is presented in Figure 5.12.

Additionally, we also collect the participants' genders and their age groups to observe how different human factors could contribute to the evaluation. As most of the participants are not aware of the study, the questionnaire came with a short basic explanation of how the planned path is formed in each situation.

(There are 4 questions in the questionnaire)
Please look at the following pictures and choose which one is more human-like.


Assuming:
- You are at the starting position at the bottom
- You are trying to walk to your destination (red box on the top). The destination is about 20 meters away from you.
- People walk on the left-hand side
- There could be a visible obstacle on the road

Which one between these two planned paths (red line) is more human-like

- 1: The left one is much more natural
- 2: The left one is slightly more natural
- 3: They are similarly natural
- 4: The right one is slightly more natural
- 5: The right one is much more natural

*必須

1. No obstacle / 障害物なし



Which path in (1) is more human-like *

1 2 3 4 5

The left one is much more natural The right one is much more natural

2. A non-moving obstacle on the road / 道路上の動かない障害物

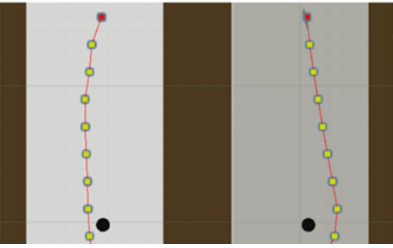


Figure 5.12: Questionnaire used to assess the human likeness of the implemented models.

Specifically, the planned path is presented as a path to a predefined destination formed within the human pedestrian’s mind before navigation. In the case of an obstacle on the road, the pedestrian also needs to plan the path so that the collision with the obstacle could be avoided. Because the participants come from different cultural backgrounds, we also specify a constraint of the pedestrian navigation, which is that the pedestrian navigates on the left side of the road. For people from right-side walking countries, they were instructed to invert the presented navigation in each model. All participants were not given any additional details, such as the approach of our model or SFM. The additional message sent to participants is provided as follows:

- For Japanese participants: *“The assumption is that pedestrians are required to navigate on the left side of the road. In both cases, you (the pedestrian) start from the green circle and walk to reach the red circle. Please respond assuming that you avoid (do not collide with) the obstacles moving from the black circle to the white circle.”* [Translated from Japanese]
- For Vietnamese participants: *“Assuming you need to plan the path to navigate from the bottom to the red dot on the top. Which one do you think is more human-like, the path on the left or the right? Noted that you need to plan the path so that you will not collide with the obstacle. In the case of (3) and (4), the obstacle could move along the dotted line. You may want to invert the presented navigation in each model, as people in Japan walk on the left-hand side.”* [Translated from Vietnamese]
- For participants from other countries: *“Assuming you need to move from the bottom to the top (red dot) and you must plan the path before navigation. In the case that there is an obstacle (black), you need to plan the path so you will not collide with the obstacle. The obstacle could move in the dotted line in the figures. Noted that in Japan, people need to navigate on the left side of the road.”*

There have been 18 people participating in the survey. A minority of the participants are Japanese students in the same laboratory as ours and only a few people are aware of our study. The rest of the participants are from different countries, including Vietnam, Egypt, Singapore and also Japan and are not aware of our study. Among the participants, the majority are male, accounting

	PPRL	Similar	SFM
No obstacle	10	1	7
Static obstacle	12	2	4
Moving obstacle	10	2	6
Pedestrian obstacle	10	2	6

Table 5.1: Number of people favoring each model’s implementation.

Total points	PPRL	SFM
ALL	60	32
No obstacle	17	12
Static obstacle	15	6
Moving obstacle	15	7
Pedestrian obstacle	13	7

Table 5.2: Total scores awarded to each model’s implementation.

for 72.2% of total responses, with female participants contributing 27.8% to the total responses. Regarding the age groups of the participants, there has been a percentage of 55.6% of responses were given by people from 30 to 44 years old. The remaining 44.4% of participants are younger people from 18 to 29 years old.

The number of people favoring the implementation of each model is presented in Table 5.1.

To evaluate our model in more detail, we also introduced a scoring system as follows: Each time the participant chooses option (1), 2 points are added to our model’s score; If the participant chooses (2), 1 point is added. Similarly, if the participant chooses (4) or (5), 1 point or 2 points are added to SFM’s score, respectively. No point is awarded if the participant chooses (3). The resulted score of the two implementations are presented in Table 5.2.

From the evaluation, our model has demonstrated better results in all situations, compared to the implementation of SFM. This result proves that our model is more human-like or natural to human pedestrians in all situations. However, we noticed that in the situation of “No obstacle”, the difference between the results of our model and SFM is not significant. Furthermore, most participants either chose option (1) - our model is much more natural, or option (5) – SFM model is much more natural. Our assumption is that in the case when no other people

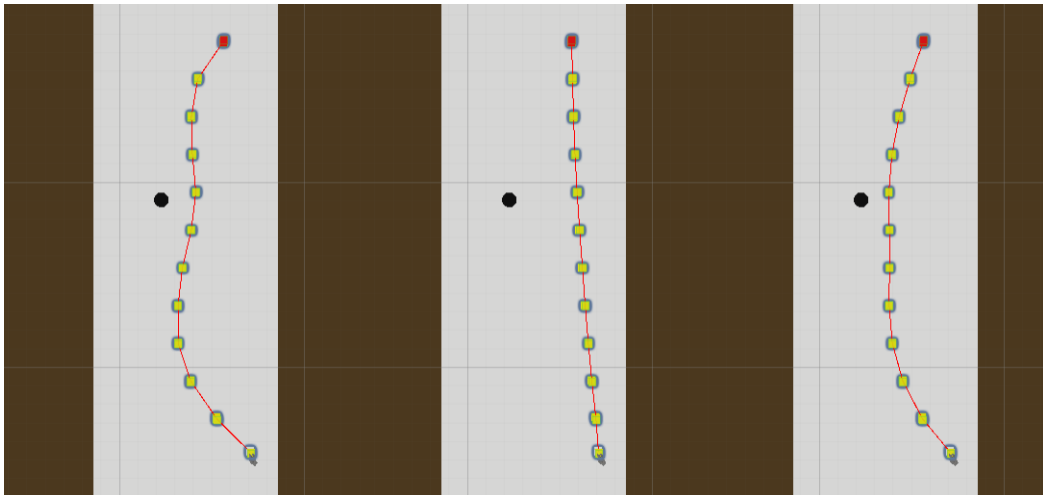


Figure 5.13: Implementations of different coefficient sets: **(a)** default; **(b)** high priority on *shortest path*; **(c)** low priority on *obstacle avoidance*

Implementation	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6
(a)	0.03	0.006	0.001	0.001	0.0005	0.03
(b)	0.06	0.006	0.001	0.0002	0.00025	0.03
(c)	0.03	0.006	0.002	0.001	0.00025	0.015

Table 5.3: Coefficient parameter value.

are around, many pedestrians may ignore certain navigation rules or behaviors such as those presented in our model.

There is not much difference between the choices given by people from the two age groups participating in our survey. Regarding the choices given by different genders, female participants are more in favor of our implementation's result. However, because the number of female participants is fairly small, this may not guarantee that the survey results accurately reflect navigation behavior by female pedestrians.

We have realized several path-planning models by implementing with different sets of coefficients. Figure 5.13 presents the implementation results with the following parameters: (a) default set; (b) high priority on shortest path; (c) low priority on obstacle avoidance. This process is done by altering the set of coefficients and matching the agent's behavior with several observed pedestrian behavior types in Japan. The coefficient parameters in each implementation are presented in Table 5.3.

To measure the contribution of the rewarding components to each model, we implemented a custom reward logging mechanism. During the training process, all component reward values are stored in a text file every fixed time duration. Figure 5.14 illustrates the recorded values in the training of each model, using the moving averages of 300 records for more accessible observation.

In order to facilitate the observation of the changes in the rewarding statistics, we offset all initial rewarding values to 0, so we easily assess how each component reward is optimized during the training process. The adjusted rewarding values are presented in Figure 5.15.

Looking at the reward value statistics in implementation (a) and (c) in Figure 5.15, it can be seen that the reward values for shortest path in (a) are notably higher than in (c) while their coefficients for shortest path γ_1 are the same. To measure the differences, we compared the maximum and the average of the corresponding rewards. We omitted the first 100 records of each implementation in the mean calculation because in the earlier stage of the training process, the agent mostly takes random action, which may consequently lead to inaccurate reward values. Comparing the max reward values, the shortest path reward in (a) reaches 0.216 while in (c), the respective reward value reaches 0.527. Regarding the average reward value for shortest path, the mean values are 0.081 in (c) and 0.201 in (a). This is because in (c), the agent does not have to put as much attention on obstacle avoidance, therefore the planned path could be shorter.

Walking on the left could also affect the shortest path reward, as could be seen in the implementation (b). Compared to (a) and (c), the coefficient parameter for shortest path, at 0.06, is twice the respective parameter in (c), at 0.03. However, as the left-side walking coefficient parameter in (b) is much lower (0.0002 in (b) compared to 0.001 in (a) and (c)), the reward values for shortest path in (b) while training is much higher than in other implementations. More specifically, the max value for shortest path reward in (b) is 1.566 and the average value is 0.720, compared to 0.216 and 0.081 in (b); and 0.527 and 0.201 in (c).

By observing the distribution of the coefficients in all implementations, we find that the coefficient for shortest path has the highest priority in all of the model implementations, followed by obstacle avoidance. This finding agrees with many other studies in pedestrian navigation, which also suggests human pedestrians subconsciously choose the shortest path as the highest-ranking priority in



Figure 5.14: Component reward values during training.



Figure 5.15: Adjusted component reward values during training.

navigation [34]. The results also suggest that changing the priority of one coefficient parameter could affect the received reward values of other factors. On the other hand, upon observation of the rewarding statistics, the reward value for changing direction does not seem to be affected by other factors.

5.7 Discussion

The implementations have shown that the agent in our model could develop a relatively natural path compared to how humans plan the path right before navigation. As each individual thinks and plans differently, the planned path by the agent may not be identical to a specific person's thinking. However, this planned path could still be seen as natural or human-like thanks to several similar traits found in the result, such as smooth navigation and following common regulations. This also indicates that by providing the appropriate rewarding formulations, the reinforcement learning agent could develop a behavior similar to the human decision-making process, thus partly confirming the hypothesis raised by other studies [62]. By supplementing and refining the rewarding formulation, a more realistic and natural navigation could be replicated.

Nonetheless, admittedly with enough complex rule sets, a rule-based model could achieve a similar result as our model. However, it could be difficult to develop the rule sets for extended states of the environment, while with reinforcement learning, the agent could adapt well to an unfamiliar environment. Another advantage of utilizing reinforcement learning in the path-planning model is that a reinforcement learning model always retains a slight unpredictability, providing some sense of the same unpredictability in human nature, which makes the navigation path more believable. On the other hand, that could also result in unknown outcomes in unforeseeable situations.

When comparing to the Social Force Model's implementation, it is apparent that the two implementations take distinctive approaches, as this could be seen in Figure 5.11. For the human's local path-planning task, our model has shown a better result, mostly because humans rarely generalize the idea of "force" when planning the path in real life. The inaccuracy of the SFM's implementation is more noticeable in the case in which the pedestrian needs to navigate from one side to the other, as the SFM agent tends to disrupt the flow of the navigation

path. The lack of a prediction method also makes SFM less ideal to realize the human's path-planning process. As a result, the path planned by the SFM agent heads straight to the destination without considering obvious possible collisions within the navigation. The path only avoids the walls and the obstacles when it is at a certain distance from those obstructions. Nonetheless, in the case when planning is difficult, like in a crowded environment for example, or for people who rarely plan before navigating, the SFM model could be sufficient.

Despite having shown a relatively natural path, assessing the model's resemblance to the human solutions is a challenging task since the path-planning process only happens in the thoughts of pedestrians. This makes evaluating the human-likeness of the result difficult, which is the major limitation of our study. We have considered several mechanisms in human cognition of assessing human likeness in pedestrian behavior. The problem is, when observing the movement, humans do not have the exact criteria to determine specific behavior is human-like or not. Instead, the human conscious and subconscious recognition processes will subjectively evaluate the movement by matching it with existing sensory data. Occasionally, even a more realistic behavior may trigger the uncanny effect, consequently leading humans to negate the human likeness of that behavior. As a result, to overcome this limitation, more insight into the human cognitive system needs to be carefully addressed.

The risk assessment seems to have contributed to the model's reasonable result. This corresponds to actual human pedestrians when perceiving different properties from an obstruction. The observable result seems to resemble how humans would perceive risks from the obstructions; however, as aforementioned, to estimate its resemblance to the task performed by humans could be demanding.

It should be noted that, in this task, only the path-planning task happening inside a human pedestrian's thinking before navigating is replicated. This path could be different from the actual path taken by the pedestrian. When following the planned path, the agent should be able to interact with the surrounding obstructions, especially when the obstructions are not navigating as predicted. In our future work, the pedestrian interacting problem for those situations will be addressed to further improve the movement of the pedestrian agent.

5.8 Summary

We have developed a novel pedestrian path-planning model using reinforcement learning while considering the prediction of the obstacle's movement and the risk from the obstacle. The model consists of two main components: a reinforcement learning model to train the agent the behavior to navigate in an environment and interact with the obstacle, and a point-of-conflict prediction model to form the estimated interacting position of the agent with the obstacle. Both components of the model acknowledge the risk assessment of the obstacle to provide corresponding results. The implementation results of our model have demonstrated a sufficiently realistic navigation behavior in many situations, resembling the path-planning process of a human pedestrian.

Chapter 6

Pedestrian interacting model

In this chapter, the model for our pedestrian interacting task is presented. In the interacting task, the pedestrian needs to carefully observe the movement of the obstacle and act accordingly.

6.1 Introduction

Recent studies in pedestrian simulation have been able to sufficiently construct a realistic navigation behavior in many circumstances. However, when replicating the close interactions between pedestrians, for example, when the pedestrian needs to avoid another person who suddenly changes his direction, the replicated behavior is often unnatural and lacks human likeness. There are two possible reasons for that. Firstly, these models often ignore the cognitive factor in the human pedestrian in the interactions. The majority of the current studies are physics-based, such as using forces [7] or fluid dynamics [8] to realize the pedestrian's movement. In real life, human pedestrians do not interact with others using force. When moving, humans do not feel the forces of repulsion from surrounding objects, but instead, the cognitive system is used to process the information and make decisions. The human cognitive system is remarkably complex and is an important research object in many different scientific fields, such as cognitive science and behavioral psychology. Several studies have adopted the ideas in cognitive science into their applications, such as autonomous robots [29], and achieved favorable results. However, to our best knowledge, no pedestrian model has considered these ideas. Another reason is that humans do not always make

optimized decisions [6]. Many approaches replicate the pedestrian behavior by using rule-based models [12] or more recently, using neural networks [30]. They usually aim at optimizing certain objectives, such as shortest path or minimizing the number of collisions. Although people usually aim at the best solution, the choices are often affected by different determinants such as personal instinct and human biases. By optimizing certain factors like shortest path or minimize the number of collisions, the resulted behavior might be unnatural or unrealistic to real-life pedestrians.

As a result, we tried to address the problem of simulating the pedestrian's interacting process using reinforcement learning, similar to the approach we have presented in Chapter 5. Correspondingly, we also explored various concepts in cognitive science to incorporate into our pedestrian interaction model. In particular, we propose a cognitive prediction model which is inspired by the predictive system in the human brain. The difference between our cognitive prediction and the prediction in many studies is that, while these studies aim at the accuracy of the prediction, the focus of our research is to imitate the prediction in the human cognitive process. By integrating the prediction with the reinforcement learning model, the navigation behavior in pedestrian interaction scenarios would be improved.

An example of this is the circumstance when the obstacle is navigating unpredictably and the pedestrian has already been close to the obstacle. In this case, the pedestrian needs to look at how the obstacle is moving and try not to collide with it. When this happens, a pedestrian interacting model is necessary to replicate the interactions between the pedestrian agent and other objects. Otherwise, if there is no obstacle, or the obstacle is moving predictably, the agent only needs to navigate along the planned path, which is the result of the path-planning process as presented in Chapter 5. By incorporating the pedestrian interacting process with the path-planning process, we could replicate a more natural and realistic navigation behavior of a real-life pedestrian.

Without integrating with the pedestrian path-planning model, the interaction model could still contribute to many studies in several application domains. An example of its applications could be the research and development of an automated vehicle model. Understanding the pedestrian's behavior could improve the model in the case of possible interactions with other people crossing the vehicle's path. For example, in the mixed traffic roads where the navigations of vehicles

and pedestrians are not separated, the pedestrian may accidentally walk into the car moving area while trying to avoid an obstruction. By appropriately assessing the situation, the automated car could avoid possible collisions. Computer games could also benefit from the research, as a more realistic human behavior would greatly enhance the user immersion.

Many studies in pedestrian simulation often approach the interaction problem using an empirical model while ignoring the concepts of the human cognition system. For example, for collision avoidance, many models adopted a repulsive mechanism to simulate the interaction between two pedestrians. In real life, on the other hand, there are many actions that the pedestrians could take, like slowing down [9] or predicting where the other would advance. Sometimes, the pedestrian may still fail to successfully avoid, thus subsequently collide with the other. This could cause problems if the simulation needs the preciseness of pedestrian behavior, for instance, a traffic simulation system for automated vehicles.

To avoid this, we analyzed the problem with the consideration of human cognition incorporated with our pedestrian interacting model. Specifically, we proposed a reinforcement learning model for pedestrian interaction simulation and a cognitive prediction model motivated by the human predictive system.

6.2 Model overview

Similar to the path-planning task, we also employed reinforcement learning for the agent's interaction learning and a prediction model for a more natural interacting behavior, especially observable from adult humans.

Generally, when the navigation needs to proceed to this process, the pedestrian is already close to the obstacle, within a distance of a few meters. Accordingly, the environment modeled in this task does not need to be too extensive. In addition, the model does not need to include too many obstacles. In our study, the environment is designed so that there is a chance of one obstacle possibly conflicting with the navigation of the pedestrian.

The model for our setting is illustrated in Figure 6.1. The pedestrian agent A has to try to get the destination D and also avoid the obstacle O (if exists). In usual circumstances, the agent does not always avoid the obstacle in its current

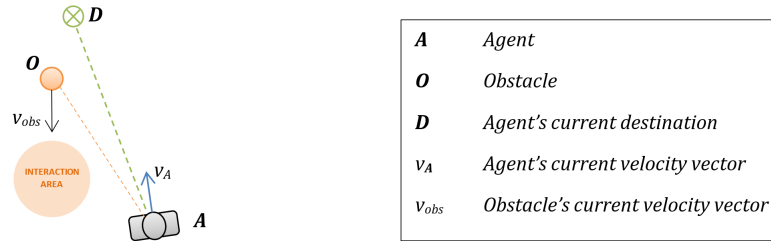


Figure 6.1: Pedestrian interacting environment setting.

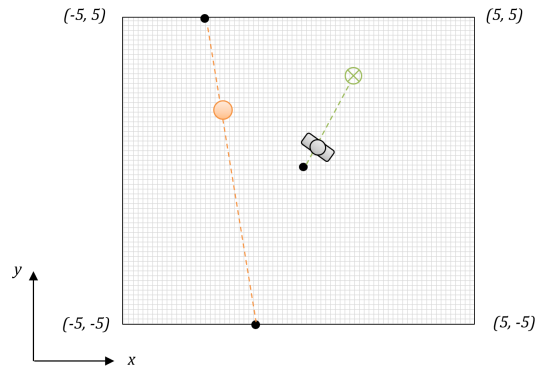


Figure 6.2: Learning task training environment.

position. Instead, the agent will form a prediction of the obstacle's movement and avoid the future *interaction area*.

6.3 Pedestrian interaction learning

Our model also uses reinforcement learning for the learning task, similar to the learning task in the agent's path-planning process, presented in Section 5.3. Similarly, to realize the reinforcement learning model for the pedestrian interaction learning task, we need to address the following problems: designing the agent's learning environment and proper rewarding approach for the agent's actions.

6.3.1 Environment modeling

Figure 6.2 presents the design of the learning environment for our model. Our training environment is an area of 10 by 10 meters. In each training episode, the pedestrian agent starts at $(0, 0)$, which is the center of the environment. The agent will be heading to an intermediate destination, placed at a distance

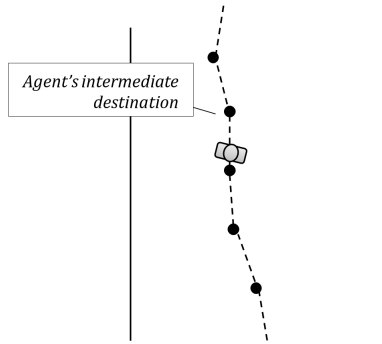


Figure 6.3: Agent’s destination as a sub-goal.

randomized between 2 to 4.5 meters and could be in any direction from the agent. This could be considered as a sub-goal [19] of the agent for the long-term planned navigation path. For example, with the agent’s planned-path to the goal presented in our previous paper [31] consisting of 10 component path nodes, the intermediate destination would be the closest component node to which the agent is heading, as demonstrated in Figure 6.3.

An obstacle could be randomly generated inside the environment. The obstacle is defined as another pedestrian that could walk into the pedestrian agent’s walking area or a slow-moving physical obstacle such as a road marking machine. We chose not to include a fast-moving object like a car in our definition of obstacle. In that case, the entire area exclusive for its movement will be too dangerous for a pedestrian and will be excluded from the agent’s navigation area. Regarding static obstacles, like an electric pole or a water puddle, these could have been addressed in the planning process and could not interfere with the pedestrian agent’s path. From the definition, the obstacle will be randomly initialized between $(-5, 5)$ and $(5, 5)$ in each training episode. After that, it will move at a fixed speed to its destination, randomly positioned between $(-5, -5)$ and $(5, -5)$. With this modeling, the pedestrian agent’s path might collide with the obstacle’s movement in any direction.

As proposed in Section 4.4, we suggest that the obstacle’s danger level and how the agent measures the risk could moderately impact how the agent navigates. For example, if the human pedestrian encounters a less dangerous obstacle such as another regular pedestrian, he may alter his navigation just a bit to avoid a collision. On the other hand, if the obstacle is a moving construction machine, the pedestrian should try to steer away from the obstacle to avoid a possible

accident.

Another important factor is the size, which is the affected area of the obstacle. For instance, if the obstacle is a group of multiple pedestrians walking together instead of one, the whole group should be treated as a single large-sized obstacle, as suggested by Yamaguchi et al. [24]. In our model, the size of the obstacle is randomized between 0.5 and 2; the danger level is randomized between 0 and 1 at the beginning of each training episode. When the prediction of the obstacle’s movement is used, the risk of the obstacle will be used instead of its danger level. The formulation of risk is presented in Section 6.3.3.

6.3.2 Agent’s observations and actions

In each step, the agent will observe various states of the environment before taking actions. We have considered two possible approaches to the design of the agent’s observations and actions. The first approach is using Euclidean coordinates. This means the agent will observe the relative position of the obstacle and the destination as well as the obstacle’s direction in Euclidean coordinates. For example, assuming the current agent’s position and obstacle’s position are (x_a, y_a) and (x_o, y_o) respectively. The agent needs to observe the related position of the obstacle, in this case, that would be the two values $(x_o - x_a)$ and $(y_o - y_a)$. Since a neural network is used for training, this could lead to a problem of finding a relationship between the coordinates and the rewarding. For instance, when the agent moves, the x (or y) coordinate may increase or decrease. However, the increment or decrement of the value does not have a direct correlation with the increment or decrement of the cumulative reward. Increasing the number of network’s hidden layers could be more effective, but even then it would be more complicated for the neural network to find an optimal policy.

The second approach, using radial coordinate, could resolve this problem. Instead of using the coordinates in x and y values, the agent’s observations and actions would instead use the distance and angle (relative to the local position and heading of the agent). This is helpful for the neural network to specify the relationship between the input and the output. For instance, a low angle and a short distance to the obstacle mean the obstacle is close, therefore going straight (angle close to 0) could lead to a lower reward value.

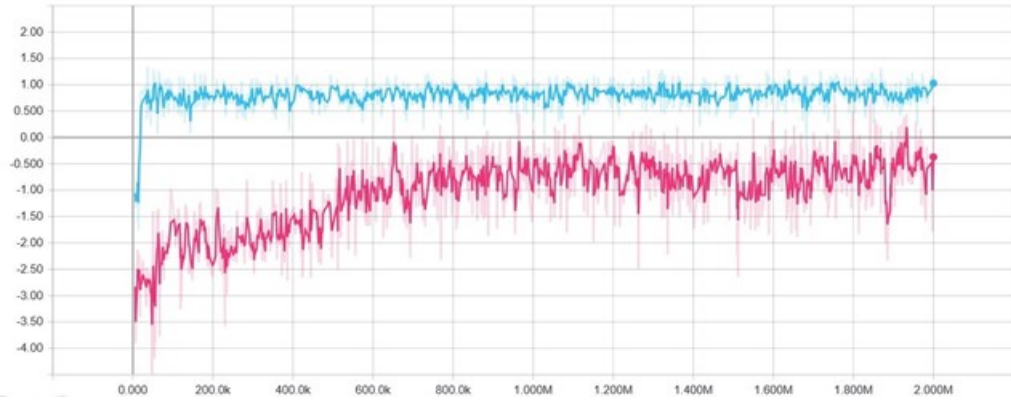


Figure 6.4: Training statistics for radial and Euclidean coordinate methods.

In a comparison between the training of the model using the radial coordinate method and using Euclidean one, we found out that the training using the radial coordinate method is better at both achieved cumulative reward and time to converge. The result is shown in Figure 6.4, in which the blue line represents the reward statistics for the radial coordinate method and the pink line represents the reward statistics for the Euclidean one.

The typical downside of using radial coordinate is angle calculation, e.g. calculation of the change in distance and angle if both the agent and the obstacle are moving. However, in the interacting process, the interval between two consecutive steps is very small, therefore the changes in the distance and angle are minimal. For this reason, we adopt the radial coordinate approach for the observations and actions of the agent.

More specifically, the observations of the environment’s states consist of: (1) the distance to the current destination; (2) the body relative direction to the destination (from agent’s forward direction); (3) The presence of the obstacle. The obstacle is considered present only if it is within the agent’s field of vision. If the obstacle is observable by the agent, the agent will also observe: (4) the distance to the obstacle; (5) the body relative direction to the obstacle; (6) the obstacle’s body relative direction to the agent (from the obstacle’s forward direction); (7) the obstacle’s speed, size, and danger level.

The possible actions which the agent could perform consist of: (1) The desired speed; (2) The angle change in the direction from the current forwarding direction.

Certain constraints are put on the actions of the agent. Firstly, the agent

cannot immediately reach the desired speed, but that speed needs to be gradually increased or decreased. Secondly, the angle change each timeframe is capped at around 10 degrees. The reason for these constraints is the limitation of human locomotion. If the constraints are not set up, it could lead to unnatural walking and turning pedestrian behavior (e.g. the pedestrian turns more than 360 degrees in less than 1 second).

The above step will be repeated until the agent reaches the destination, the agent gets too far from the destination or it takes too long for the agent to reach the destination. After that, the total reward will be calculated to let the agent know how well it has performed the task. The details of the rewarding will be explained in the next section. Finally, the environment will be reinitialized, and the agent will repeat the above steps. The set of agent’s observations and actions, as well as the cumulative reward, is sent to be trained in a neural network aiming at maximizing the cumulative reward value.

6.3.3 Rewarding behavior

Taking the cue from the learning task for the path-planning process, we also designed the rewarding behavior for our model using the idea of *human comfort*, as suggested in Chapter 5. There are numerous factors in the concepts of human comfort. For the training task of this pedestrian interacting process, we adopted the factors listed below, grouped into two categories: *Goal Optimisation* (GO) and *Natural Behavior* (NB).

The category GO consists of the behaviors which encourage the agent to achieve the goal in the most efficient way. The following factors are put under this category:

A. Reaching destination reward

The agent receives a small penalty every step. This is to encourage the agent to achieve the goal as swiftly as possible. The agent also receives a one-time reward when reaching the destination. This also leads to the termination of the current episode and resets the environment. The formula for this reward at time t is calculated by:

$$R_{1,t} = \begin{cases} R_{step}, & \text{if } \delta_{A,D}^{(t)} \geq \delta_{min} , \\ R_{goal}, & \text{if } \delta_{A,D}^{(t)} < \delta_{min} , \end{cases} \quad (6.1)$$

where $\delta_{A,D}^{(t)}$ is the distance between the agent and its destination at the time t ; R_{step} is a small constant penalty value for every step that the agent makes; R_{goal} is the constant reward value for reaching the destination.

B. Matching the intended speed

The agent is rewarded for walking at a desired speed. This value varies between people. For example, a healthy person often walks at a faster speed than the others, while an older person usually moving at a slower speed.

The reward for this is formulated as follow:

$$R_{2,t} = \|v_t - v_{default}\| , \quad (6.2)$$

where v_t is the current speed and $v_{default}$ is the intended speed of the agent. This value may vary depending on different factors like age or gender. For example, a healthy person tends to walk faster, while an old person tends to walk slower.

C. Avoid significant change of direction

Constantly changing direction could be considered unnatural in human navigation. Appropriately, the agent is penalized if the change in direction of the agent is greater than 90° in one second. The reward for this behavior is formulated as follow:

$$R_{3,t} = - R_{angle}, \quad \text{if } \frac{\phi_\Delta}{\Delta t} > 90 , \quad (6.3)$$

where ϕ_Δ is the change in agent's direction, having the same value as action (2) of the agent; Δt is *delta time*, the time duration of each step; and R_{angle} is the constant penalty value for direction changes.

The category NB consists of the behaviors which encourage the agent to behave naturally around humans. As the navigation model in our research is fairly limited, such interactions such as gestures or eye movement cannot be implemented. As a result, the only factor put under this category that is used in our model is:

D. Trying not to get too close to another pedestrian.

The reward for this behavior is formulated as follows:

$$R_{4,t} = \begin{cases} - \text{danger}, & \text{if } size_O \delta_{A,O}^{(t)} < 1 , \\ \left(\frac{\delta_{A,O}^{(t)} - 1}{S - 1} - 1 \right) \text{danger}, & \text{if } 1 \leq size_O \delta_{A,O}^{(t)} < S , \\ 0, & \text{if } size_O \delta_{A,O}^{(t)} \geq S , \end{cases} \quad (6.4)$$

where $\delta_{A,O}^{(t)}$ is the distance between the agent and the obstacle at time t ; *danger* and *size_O* are the danger level and the size of the obstacle respectively; S is the distance to the obstacle which the agent needs to start interacting with. As mentioned in Section 4.1.1, *danger* is the agent's perception of the obstacle's danger. This will be updated with *risk* when a prediction of obstacle' movement is formed, which will be presented in Section 6.3.3.

In normal circumstances, for example, when a pedestrian is walking alone or when he is far away from other people, the pedestrian does not have to worry about how to interact naturally with others. As a result, the behaviors listed in the NB category need less attention than other behaviors listed in the GO category. On the contrary, when the pedestrian is getting close to the other, the GO behaviors should be considered less important. As a result, the cumulative reward for each training episode is formulated as follows:

$$\mathcal{R} = \sum_{t=1}^n h \left(N_{GO}^{(t)}, N_{NB}^{(t)} \right) , \quad (6.5)$$

where h is a heuristic function to combine the rewards for achieving the goal and the rewards for providing the appropriate human behavior; n is the number of steps in that episode; $N_{GO}^{(t)}$ is the sum of the cumulative rewards for all behaviors in GO category at time t and $N_{NB}^{(t)}$ is the sum of the cumulative rewards for all behaviors in NB category at time t .

$$N_{GO}^{(t)} = \kappa_1 R_{1,t} + \kappa_2 R_{2,t} + \kappa_3 R_{3,t} , \quad (6.6)$$

$$N_{NB}^{(t)} = \kappa_4 R_{4,t} , \quad (6.7)$$

where κ_1 is the coefficient for reaching destination rewarding; κ_2 is the coefficient for matching intended speed rewarding; κ_3 is the coefficient for changing direction avoidance rewarding; κ_4 is the coefficient for collision avoidance rewarding.

Different people have different priorities for each previously mentioned behavior. As a result, with different coefficient values of κ_1 , κ_2 , κ_3 and κ_4 , individual pedestrian personality could be formulated.

The heuristic function is implemented in our model as follows:

$$\mathcal{R} = \sum_{t=1}^n \left(\gamma N_{GO}^{(t)} + (1 - \gamma) N_{NB}^{(t)} \right) , \quad (6.8)$$

where γ is a value ranged from 0 to 1, corresponding to how far the agent is from the obstacle and also the size of the obstacle. The reason for including the size of the obstacle in the calculation of γ is that when an obstacle is bigger, it would appear closer to the pedestrian, and the pedestrian would likely stay further away from the obstacle as a result. Therefore, γ is specified in our model as follows:

$$\gamma = \frac{1}{\delta_{A,O} \text{ size}_O + 1}, \quad (6.9)$$

where $\delta_{A,O}$ is the distance between the agent and the obstacle; size_O is the observed size of the obstacle.

6.4 Prediction task

The predictive process happens in almost every part of the brain. This is also the cause of many bias signals sent to the cognitive process, leading to the behavior in which humans act in real life [18]. In the human brain, the prediction is made using information from past temporal points, then it would be forwarded to be compared with actual feedback from sensory systems. The accuracy of the prediction is then used to update the predictive process itself.

The prediction task helps the agent avoid colliding with the obstacle more efficiently. Without using a prediction, the pedestrian might interrupt the navigation of the other pedestrian or even collide with. This behavior is more frequently observable in younger pedestrians, whose prediction capability has not been fully developed.

The prediction task could happen in both the path-planning process and the interacting process. For example, when a person observes another pedestrian walking from afar, he could form a path to avoid the collision. In the implementation result from the path-planning task, as presented in Chapter 5, it is shown that the prediction helps the pedestrian agent to plan a more efficient and realistic navigation path.

The prediction in the interacting process, however, is different from the prediction in the path-planning process. While in the path-planning process, the agent only needs to project an approximate position of the obstacle in order to form a path, in the interacting process the agent will need to carefully observe

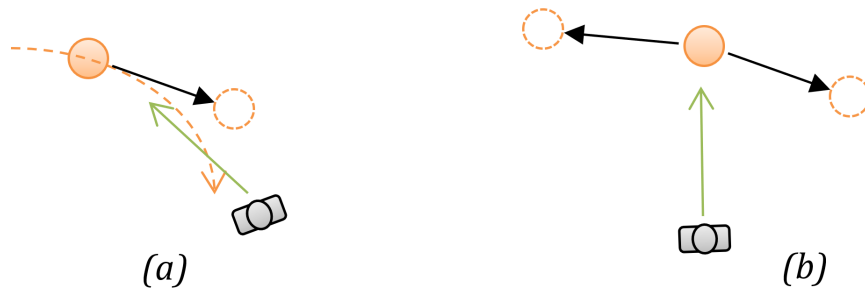


Figure 6.5: The problems with the position forwarding prediction model.

every movement of the obstacle to expect its next actions. This will be carried out continuously when the agent is having the obstacle in sight.

For this reason, a simple position forwarding prediction could not be sufficient. The first problem with this is that when the obstacle is moving with a certain pattern (for example, the obstacle is moving along a curve, as shown in Figure 6.5.a), a position forwarding prediction using only the obstacle's direction is usually incorrect. The second problem is that when the obstacle is uncertain about its orientation and choosing to move in two opposite directions. The agent may see the position in the center is safe to navigate (as shown in Figure 6.5.b), while actually, it is usually the contrary. In order to solve these problems, we had to look into the mechanism of the predictive process. Based on that, we set up three steps for the prediction task, presented as follows:

1. Step 1 – *Estimation*: Based on the previous movement of the obstacle, the pedestrian agent forms a trajectory of its movement. Subsequently, the agent specifies the location in that trajectory that he thinks the obstacle would be at the current moment
2. Step 2 – *Assessment*: The difference between the predicted location and the actual current position of the obstacle is measured. This indicates how correct the prediction was, meaning how predictable the movement of the obstacle was. If the predicted location is close to the actual position, that means the movement of the obstacle is fairly predictable, thus the agent could be more confident in predicting the future position of the obstacle.
3. Step 3 – *Prediction*: The agent forms a trajectory of the obstacle's movement based on the current movement. Combining with the difference calculated in Step 2, the agent predicts the future position of the obstacle on

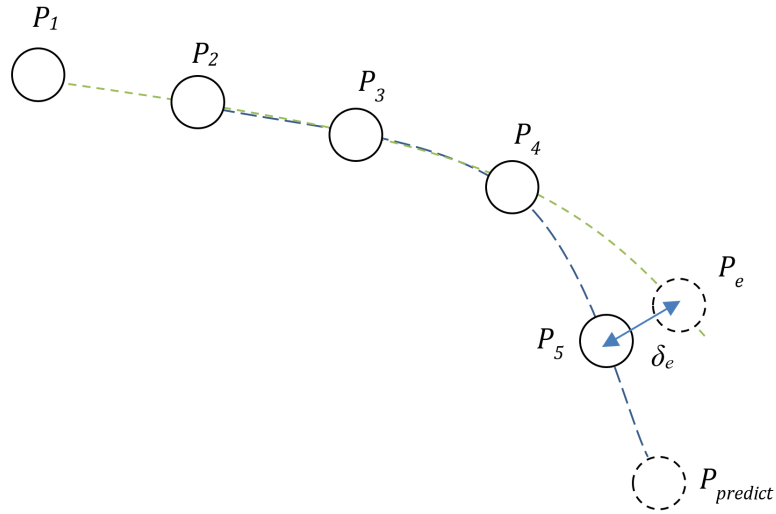


Figure 6.6: Prediction task model.

that trajectory. If the difference is small, meaning that the agent is confident with the prediction, he would predict a position further in the future and vice versa.

Figure 6.6 illustrates the modeling of the prediction task. $P_1, P_2, P_3, P_4,$ and P_5 are the sampled positions of the obstacle's movement, with P_5 is the obstacle's current position. P_e is the projected position of the obstacle from P_1 to P_4 and $P_{predict}$ is predicted position of the obstacle. The flowchart for the prediction process is presented in Figure 6.7.

As an example, if a pedestrian obstacle is going straight in one direction, its movement could be easily figured. Thus, the difference between its predicted location and its actual current position should be primarily small. The agent then will be able to predict the obstacle's position further in the future and will be able to comfortably avoid it. On the other hand, if the pedestrian is moving unpredictably, it will be very difficult for the agent to guess its movement. In this case, the predicted location of the obstacle in the further future would be mostly incorrect. Consequently, avoiding the near future or even the current projection of the obstacle would be a better decision.

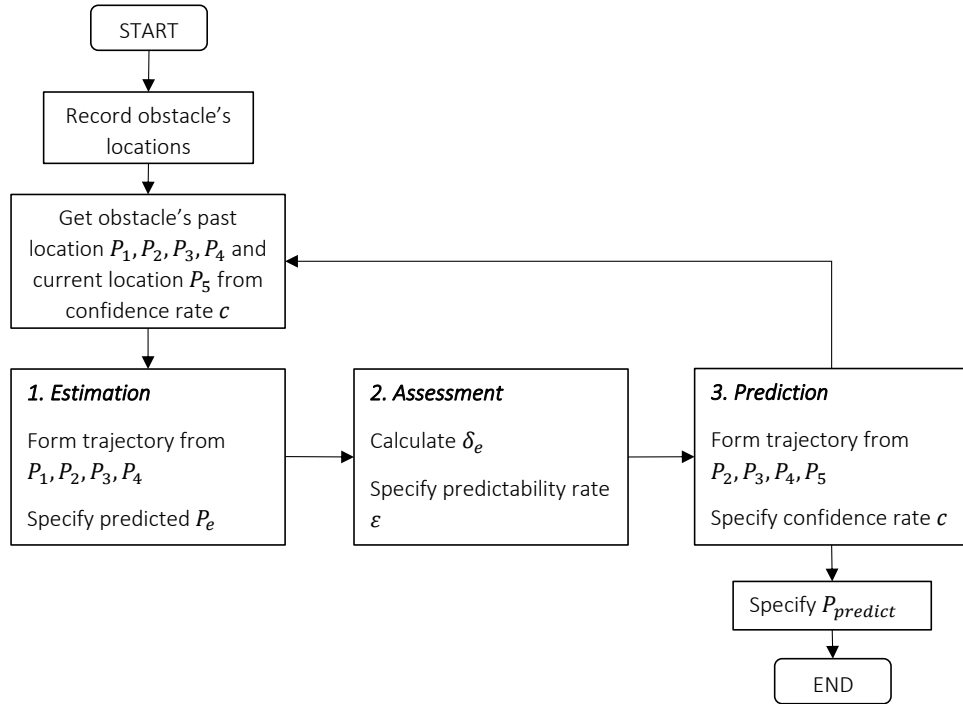


Figure 6.7: Prediction process flowchart.

6.4.1 Estimation

The recent position data of the obstacle is stored together with its respective time information in a data structure by being logged every fixed timeframe. To avoid the incorrect data being logged, the timeframe should be longer than the time duration between two continuous frames.

First of all, the agent needs to form a trajectory of the obstacle's movement from the past positions. To do that, the agent will need to choose some samples from previously recorded location data of the obstacle, then perform interpolation to get a parametric representation of the movement.

To help the agent with choosing the sample and performing interpolation, we propose a concept called *confidence rate*. The confidence rate of the agent, denoted by c , is a value that is dependent on the accuracy of the agent's previous prediction. With a high confidence rate, the agent could be more comfortable interpolating using a wider time span. The confidence rate will be calculated in the Assessment step, presented in Section 6.3.2 below.

For the interpolation process, we used two Lagrange polynomial interpola-

tions. One interpolation is used for the set of (x_i, t_i) and the other is used for the set of (y_i, t_i) . For the interpolating polynomial presented in the form of a cubic function, four sets of samples corresponding to $t_1 \dots t_4$ are required. Given the current time τ , the value t_i is calculated as follows:

$$t_i = \tau - (5 - i)\Delta, \quad (6.10)$$

with

$$\Delta = c \gamma_1, \quad (6.11)$$

where c is the confidence rate ranging from 0 to 1; γ_1 is a time constant discount. For example, if the agent is very confident ($c = 1$) and the samples chosen from the pedestrian obstacle's previous movement of 2 seconds, then γ_1 could be 0.4.

We set a minimum value of 0.3 for Δ as in reality, human perception cannot recognize the object's micro-movement. Therefore, in the case of a low confidence rate (for example, when the previous prediction was greatly incorrect), the pedestrian agent will still use samples from the obstacle's previous 1.5 seconds approximately.

The four sets of the corresponding (x_i, t_i) and (y_i, t_i) are used to specify the $x = x(t)$ and $y = y(t)$ functions using Lagrange interpolation. Specifically, the $x = x(t)$ function are formulated from $(x_1, t_1) \dots (x_4, t_4)$ as follows:

$$x = \frac{(t - t_2)(t - t_3)(t - t_4)}{(t_1 - t_2)(t_1 - t_3)(t_1 - t_4)}x_1 + \frac{(t - t_1)(t - t_3)(t - t_4)}{(t_2 - t_1)(t_2 - t_3)(t_2 - t_4)}x_2 + \frac{(t - t_1)(t - t_2)(t - t_4)}{(t_3 - t_1)(t_3 - t_2)(t_3 - t_4)}x_3 + \frac{(t - t_1)(t - t_2)(t - t_3)}{(t_4 - t_1)(t_4 - t_2)(t_4 - t_3)}x_4. \quad (6.12)$$

The $y = y(t)$ function is similarly specified.

The estimation of the current position of the obstacle $P_e(x_e, y_e)$ at the current time τ is defined by: $(x_e, y_e) = (x(\tau), y(\tau))$.

6.4.2 Assessment

The predictability of the obstacle's movement is calculated using the distance δ_e between the obstacle's current position (x_5, y_5) and the estimated position of the agent (x_e, y_e) as calculated above. If δ_e is small, that means the movement is

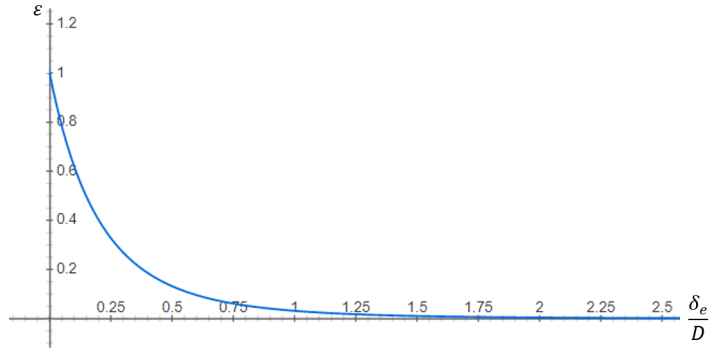


Figure 6.8: Plot of the function $\varepsilon = f\left(\frac{\delta_e}{D}\right)$.

predictable. On the contrary, if δ_e is large, that means the movement is not as the agent expected. An example of this is when a pedestrian encounters an obstacle, which is another pedestrian walking in the opposite direction. When trying to avoid running into the obstacle, the pedestrian observes that the movement of the obstacle was going to his left-hand side. However, the obstacle makes a sudden change and walk to the right instead. This makes the movement of the obstacle seemingly unpredictable, and thus the pedestrian needs to be more careful when planning to interact.

We defined a value predictability rate as ε , determined by:

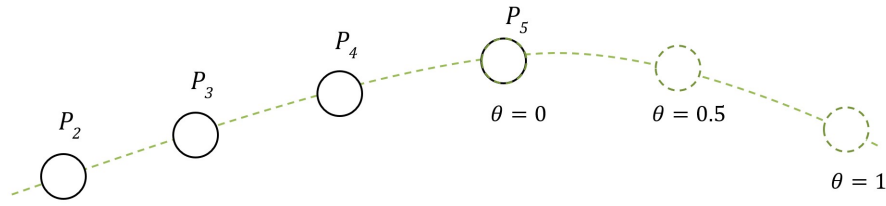
$$\varepsilon = \frac{1}{\left(\frac{\delta_e}{D} + 1\right)^5}, \quad (6.13)$$

where D is the average distance between the first and the last sample points P_1 and P_4 .

The plot of the function $\varepsilon = f\left(\frac{\delta_e}{D}\right)$ is presented in Figure 6.8. As observed from the figure, we could see that when $\frac{\delta_e}{D}$ is close to 0, ε will be approximately 1. The ε value drops steeply when $\frac{\delta_e}{D}$ decreases. For example, when $\frac{\delta_e}{D}$ is around 0.5, the ε value is only 0.13.

The confidence rate c will be then calculated using the predictability rate. The confidence rate gets higher or the agent is more confident when ε is consecutively at a high value and vice versa. The formulation for calculating the confidence rate could be different for each person, as some people could be more confident after several correct predictions than the others.

The formulation for calculating the confidence rate c_t at time t is presented

Figure 6.9: Resulted prediction with different θ .

as follows:

$$c_t = c_{t-1} + \gamma_2 (\varepsilon_t - c_{t-1}) , \quad (6.14)$$

where γ_2 is the discount for the change in confidence rate, with $\gamma_2 = 0$ meaning the confidence rate is not dependable on the prediction rate, and $\gamma_2 = 1$ meaning the confidence rate will always equal the prediction rate. Practically, γ_2 should be from 0.3 to 0.6.

6.4.3 Prediction

Similar to the Estimation step, we also use Lagrange interpolation in the Prediction to form the functions $x = \bar{x}(t)$ and $y = \bar{y}(t)$ for the projection of the movement. In this step, however, the sample positions used are P_2 to P_5 (the current position of the obstacle) respectively. For instance, the function for the four sets of samples $(x_2, t_2) \dots (x_5, t_5)$ is presented as:

$$x = \frac{(t - t_3)(t - t_4)(t - t_5)}{(t_2 - t_3)(t_2 - t_4)(t_2 - t_5)}x_2 + \frac{(t - t_2)(t - t_4)(t - t_5)}{(t_3 - t_2)(t_3 - t_4)(t_3 - t_5)}x_3 + \frac{(t - t_2)(t - t_3)(t - t_5)}{(t_4 - t_2)(t_4 - t_3)(t_4 - t_5)}x_4 + \frac{(t - t_2)(t - t_3)(t - t_4)}{(t_5 - t_2)(t_5 - t_3)(t_5 - t_4)}x_5 . \quad (6.15)$$

The $y = \bar{y}(t)$ function is similarly specified.

The prediction of the obstacle is determined from the functions $x = \bar{x}(t)$ and $y = \bar{y}(t)$ at the time $t_i = \tau + \theta$, where τ is the current time and θ is the forward time duration in the future. Consequently, if the agent wants to predict the location of the obstacle at 1 second in the future, θ would be 1. Figure 6.9 demonstrates how different θ value affects the resulted prediction of the obstacle.

The value θ depends on the confidence rate c . If the agent is confident with the prediction, he will predict an instance of the obstacle at a further point in

the future. On the contrary, if the agent is not confident, for example when the obstacle is moving unpredictably, he would only choose to interact with the current state of the obstacle (θ close to 0). The estimation of θ in our model is formulated as follows:

$$\theta = c \varepsilon \gamma_3 , \quad (6.16)$$

where γ_3 is a time constant discount. For example, when the agent is confident, the current prediction is correct and the forward position of the obstacle could be chosen at 1 second in the future, γ_3 could be set to 1.

To summarize, the function to calculate the predicted position (x_p, y_p) of the obstacle could be formulated as follows:

$$(x_p, y_p) = (\bar{x}(\tau + c \varepsilon \gamma_3), \bar{y}(\tau + c \varepsilon \gamma_3)) . \quad (6.17)$$

Finally, the predicted position of the obstacle will be assigned to the observation of the agent as presented in Section 6.2. More specifically, instead of observing the current position of the obstacle, the agent will use the predicted position (x_p, y_p) of the obstacle.

The risk of the obstacle, as mentioned in Section 4.1.1, will be updated depending on the confidence rate of the agent. One reason for this is when an obstacle is moving unpredictably, it could be hard to expect where it could go next, which leads to a higher risk assessed by the agent. The relation between the obstacle's risk and danger level is defined as follows:

$$r = danger + (1 - danger)(1 - c) , \quad (6.18)$$

where r is the risk, $danger$ is the danger level of the obstacle perceived by the agent and c is the confidence rate of the agent. That means if the agent is confident with the movement of the obstacle, the perceived risk will be close to the danger level observed by the agent. However, if the confidence rate is low, the risk will be increased correspondingly.

6.5 Implementation and discussion

Our proposed model was implemented using Unity 3D. We prepared two separate environments for the implementation. One environment is used for the agent



Figure 6.10: A screenshot from the implementation application.

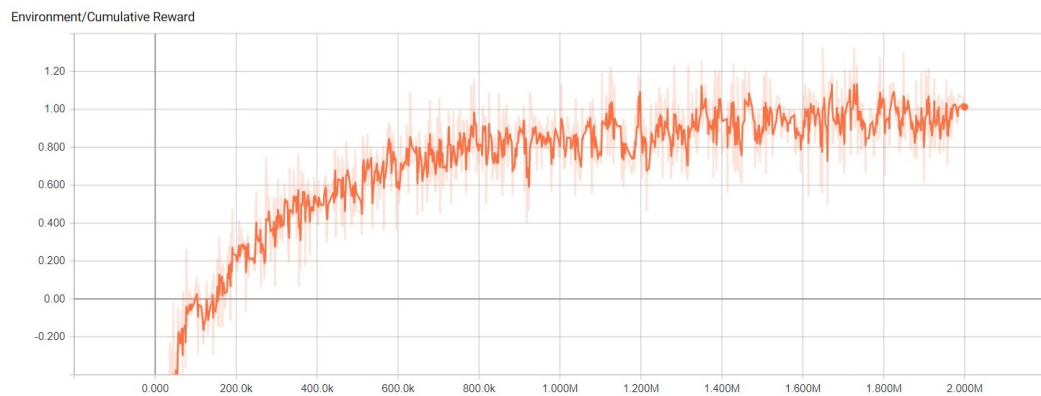


Figure 6.11: Learning task training statistics.

training of the learning task and the other for implementing the prediction task as well as to validate our model. The source code for our implementation could be found at <https://github.com/trinhthanhtrung/unity-pedestrian-rl>. The two environments are placed inside the Scenes folder by the names *InteractTaskTraining* and *InteractTaskValidate*, respectively. Figure 6.10 presents our implementation application running in the Unity environment.

For the training of the learning task, we also used the Unity-ML library. For our designed training environment, the pedestrian agent has the cumulative reward converged after 2 million steps, using a learning rate of 3×10^{-4} . The computer we used for the training is a desktop computer equipped with a Core i7-8700K CPU, 16GB of RAM and NVIDIA GeForce GTX1070 Ti GPU. With this configuration, it took 1 hour 40 minutes to complete the process. The statistics for the training are shown in Figure 6.11.



Figure 6.12: Screenshot from interacting model experimental dataset.

For the predicting task, we created a script called Movement Predictor and assign it to the pedestrian agent. The position records of the obstacle are stored in a ring buffer. The advantage of using a ring buffer is the convenience of accessing its data: with the confidence rate specified, the time complexity to get the data of the obstacle’s past locations is always $O(1)$. The values γ_1 , γ_2 and γ_3 in (6.11), (6.14), (6.16) are set to 1.7, 0.45 and 1.1, respectively.

Similar to the path-planning process, the parameters used in our model are calibrated to resemble the human movement in our conducted experiments on pedestrian interacting behavior. In our experiment, one person acts as the agent to interact with another person acting as an obstacle. In each situation, the obstacle person navigates following a script that was predefined using certain real-life scenarios. The other person could move and interact in the same way as in normal situations. Figure 6.12 shows a screenshot from our pedestrian interacting experimental dataset.

The demonstration of our pedestrian interacting behavior could be observed from <https://github.com/trinhthanhtrung/unity-pedestrian-rl/wiki/Demo>. The user could freely control an obstacle and interact with the agent. In our experiment, we controlled the obstacle to walk and interact with the agent in similar behavior as an actual person using existing pedestrian video datasets. From the demonstration, it could be seen that the movement of the pedestrian agent bears many resemblances with the navigation of actual humans. The pedestrian agent is able to successfully avoid the obstacle most of the time and reach the des-

tionation within a reasonable amount of time. This result suggests that basic navigation behavior could be achieved by the agent by utilizing reinforcement learning, thus confirming this study’s hypothesis as well as the suggestion by other researchers [62]. By incorporating the prediction process, the agent also expressed avoidance behavior by moving around the back of the obstacle instead of passing at the front, similar to how a human pedestrian moves. In case of an obstacle with unpredictable behavior, the agent shows certain hesitation and navigates more carefully. This also coincides with human movement behavior when encountering a similar situation, consequently introducing a more natural feeling when perceiving the navigation, corresponding to our expectations.

On the other hand, several behavioral traits of human navigation were not presented in the navigation of our model’s implementation. An example is that a human pedestrian in real life may stop completely when the collision is about to happen. This is for the pedestrian to carefully observe the situation and also to make it easier for the other person to respond. In our model, the agent only slightly reduces its velocity. Another example is when interacting with a low-risk obstacle, the agent may occasionally collide with the obstacle.

To evaluate our model, we compared our results with a Social Force Model implementation and the built-in NavMesh navigation of Unity. Some examples of the implementation are demonstrated in Figure 6.13. In each situation, our cognitive reinforcement learning model is on the left (blue background), the Social Force Model implementation is in the middle (green background), and the Unity NavMesh implementation is on the right (yellow background). The green circle represents the agent and the red circle represents the obstacle. The green and the red spots are the periodically recorded positions of the agent and the obstacle, respectively.

Upon observation of each model’s behavior, the difference in the characteristics of its movement could be noticed. As the SFM model is realized using a force-based method, the movement of the pedestrian agent in SFM is very similar to a magnetic object. The appearance of an obstacle could push away the agent when it is being close. The agent in the Unity NavMesh implementation often takes the shortest path approach. However, as the agent only considers the current state of the environment, it may occasionally take a longer path when the obstacle moves. On the other hand, the behavior of the agent in our model is more unpredictable, although certain factors such as taking the shorter path and

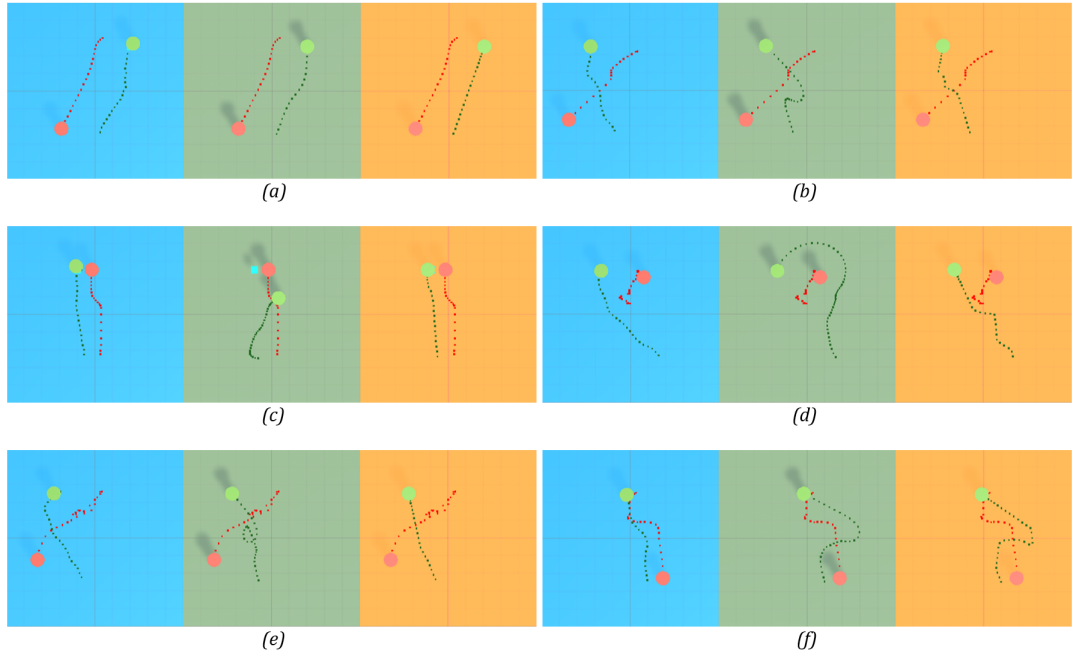


Figure 6.13: Example interacting situations between agent and obstacle in comparison with Social Force Model and Unity NavMesh. The green circle represents the agent and the red circle represents the obstacle.

collision avoidance are still considered. Except for the NavMesh implementation, both implementations of our model and SFM could demonstrate the behavior of changing the agent’s speed. While the agent in SFM often changes the speed to match the obstacle’s velocity, the agent in our model tends to slow down when being close to the obstacle.

In the most basic situations, when there are two pedestrians walking in opposite directions as simulated in (a), all models could demonstrate acceptable navigating behavior. These are also the most common situations observed in real life. However, the difference between the implementations is most evident when in certain scenarios in which the obstacle does not follow the usual flow of the path, such as in other situations presented in Figure X. These are modeled from the real-life pedestrians in the cases when, for instance, a person crossed the path, a person was walking while looking at his phone without paying much attention to the others, or a person suddenly noticed something and changed his path toward that place. While our implementation shows natural navigation in all test scenarios, the SFM and NavMesh implementations show many unnatural behaviors. This could be seen in situation (f) for NavMesh implementation, where the

agent takes a wide detour to get to the destination. For SFM implementation, the agent demonstrates much more inept behavior, notably seen in situations (b), (d), (e) and (f). Another problem of the SFM’s implementation could be seen in (c). In this circumstance, the pedestrian agent is unable to reach its destination, as the force from the obstacle keeps pushing the agent away. On the contrary, the problem with NavMesh’s agent is that the agent continuously collides with the obstacle. This is most evident in the situation (d) and (e), in which the agent got very close to the obstacle, then walked around the obstacle, greatly hindering the obstacle’s movement. Arguably, this behavior could be seen in certain people, however, it is still considered impolite or ill-mannered. The agent in our implementation suffers less unnatural behavior compared to the others. Take the situation (f) for example, while the obstacle was hesitant, the agent could change the direction according to how the obstacle moves.

We also compared our implementation with SFM and NavMesh using the following aspects: the *path length* to reach the destination, the *navigation time* and the *collision time* (i.e. the time duration that the agent is particularly close to the obstacle). These are some common evaluation criteria, which are used in many studies to evaluate the human likeness of the navigation. To evaluate these aspects, we ran a total of 121 episodes of the situations modeled from similar settings from real life. Each episode starts from when the agent starts navigating to when the destination is reached, or when the end time limit of the simulated situation has been reached. The collision time is specified by measuring the time that the distance between the agent and the obstacle is less than the sum of the radius values of the agent and the obstacle. The average results are shown in Table 6.1. Compared to our model, the Social Force Model agent took a considerably longer path as the agent always wanted to keep a long distance from the obstacle. Consequently, the average time to complete the episode of the Social Force Model agent is much higher than ours. Understandably, the collision time of the Social Force Model is the lowest, as avoiding the obstacle is its top priority. This figure seems to be too ideal in practical situations, particularly when the obstacle is moving unpredictably. The agent in the Unity NavMesh implementation has the shortest path length and fastest time to reach the destination on average, as the agent only avoids the obstacle when the distance is really close. However, this also leads to a slightly higher collision time with the obstacle than in our model.

	InteractingRL	SFM	NavMesh
Average path length (meter)	5.134	5.608	4.987
Average navigation time (second)	4.142	4.965	3.881
Average collision time (second)	1.182	0.291	1.267

Table 6.1: Comparisons with Social Force Model and Unity NavMesh in average length, time and collisions.

This finding shows that while certain measurements by SFM and NavMesh are more positive, this result is not reflected in the implementation results, as could be seen in the actual results. This is consistent with our initial suspect, the optimization of such factors as shortest path or least collision may not provide the most human-like behavior in pedestrian navigation. This result consequently validates the questions raised from the experiments of pedestrian behavior in other studies [34]. However, to specify the factors that determine human likeness in pedestrian navigation is a difficult problem. This will be addressed in our future research.

There are still many issues and improvements we need to address in future research. One problem is that our pedestrian agent still ignores many social rules in the case of being close to the other. Partially, the problem is caused by the lack of any gesture implementations in our research, such as eye gestures (e.g. glance, gaze or focusing on something) or body language (e.g. nod, bow). Supplementing different rewarding behaviors could help, such as adding rewarding behavior for passing the right-hand side (left-hand side for countries using left-hand traffic) or when the pedestrian is in a hurry or not, as suggested by Daamen et al. [27]. Another problem is the interaction process in our research is limited to between the agent and an obstacle only. The interactions of the agent could be particularly different with the addition of other pedestrians, expanded to various behaviors like grouping or speed matching [24]. On the other hand, our study might still be applicable to multiple pedestrians by forming two pedestrian groups, as human pedestrians often navigate in groups and following the leaders, as suggested by Pelechano et al. [28].

To accurately evaluate the model is a challenging task as there is not an ideal solution for any specific scenarios. While the pedestrian behavior data can be extracted from a data source such as a video recording, the interactions in this

data are not the only applicable approach. As a result, it is necessary to have a separate extensive study to comprehensively propose the evaluation method for such models.

6.6 Summary

In this chapter, we presented a novel approach to a model of simulating the human-like pedestrian interacting behavior. The model consists of the learning task and the prediction task. In the learning task, we employed deep reinforcement learning to train the agent to learn the interacting behavior with another obstacle. This is done by providing the agent with appropriate rewarding behaviors subjected to several *human comfort* factors. We also proposed the concept of *risk*, which has been demonstrated to moderately affect how the agent navigates to the destination. In the predicting task, we explored the mechanism of the predictive system in human neuroscience and proposed a predicting model to incorporate with the learning task. This model consists of three steps. Firstly, in the estimation step, the position of the obstacle at that moment is projected from the past movements of the obstacle. This is followed by the assessment step, which determines the predictability of the obstacle's movement by comparing the projection with the obstacle's actual position. Finally, in the prediction step, the agent predicts the position of the obstacle at a specific time in the future, depending on the agent's confidence.

The empirical result of the model has presented a striking resemblance to the interacting behavior of human pedestrians. Although the model still lacks certain aspects in social rule conformity, many pedestrian navigation behaviors are present. In the future, we will need to address the problems related to standard social behaviors as well as the inclusion of multiple obstacles.

This model demonstrates the effectiveness of reinforcement learning in general, especially in pedestrian simulation. In particular, when the practices in human cognition are considered, the agent could show more realistic performance. The studies in other application domains could as well benefit from this with appropriate adaptation.

Chapter 7

Discussion

In this chapter, the findings of our results are discussed. Specifically, we discuss the contribution of the concepts in human factors and human cognition. Risk is also an important factor that contributes to our behavioral navigation model. We also express our justification in different approaches, using reinforcement learning or using a traditional method. Finally, we discuss the principle of natural or realistic behavior, particularly navigation behavior.

From the results of our studies, many improvements in the navigation behavior of the pedestrian agent could be observed. This could be achieved thanks to the contribution of several realizations of the ideas in *human factors* and *human cognition*. More specifically, we have proposed a path-planning model and also incorporated the concept of cognitive prediction in our model. These aspects in cognitive science are important for designing a realistic pedestrian model. As previously discussed, many human factors could also greatly affect navigation behavior, such as age or gender [77]. While these factors are not explicitly presented in our model, they could be regulated from other parameters such as the coefficient parameters in the learning model or the confidence rate in the interacting task. These parameters help our model to present a variety of characteristics in human pedestrians, which is certainly necessary for a realistic pedestrian simulation model.

In our model, the obstacle's danger and its risk assessment are also our focus. This aspect is often overlooked in other studies. In many application domains, especially safety-related, these are important issues that need to be addressed. The

real-life observation indicates that pedestrians would navigate differently when assessing different risks from surrounding obstructions. Our implementation has successfully demonstrated this behavior to a certain extent. The navigation by the pedestrian could respond to the risk of the obstacle, caused by several factors like its harm and speed, similar to how humans observe the danger. Consequently, the navigation results are considerably similar to the movement by humans. The implementation results are tolerable, which possibly benefits other studies in the safety application domain.

In both models, we compared ours with other related models, mainly Social Force Model. As is observable from each corresponding evaluation, pedestrian navigation is significantly more realistic than in the SFM model. The pedestrian agent demonstrates many social conforming behaviors, even though the rewarding mechanism given to the agent is still fairly limited. While SFM has shown to be better than our model in certain aspects (e.g. collision avoidance), this result is also better than actual pedestrian humans. Therefore, when being evaluated by actual humans, our model is mostly seen as a higher accurate simulation than the SFM model. This finding shows that a human-like navigation behavior may not be achieved by just optimizing certain factors in the pedestrian model.

Reinforcement learning method has proven to be a viable solution to replicate a natural pedestrian behavior. By providing a reasonable rewarding formulation, the agent could learn to act appropriately. Designing the rewarding formulation could be difficult, similar to how the teacher needs to provide effective teaching methods to students. In our model, we have been able to instruct the agent to navigate in a human-like manner by providing the rewarding formulation similar to how humans feel when observing the navigation. Using a reinforcement learning method also gives the agent a sense of unpredictability when taking actions. This may provide the same unpredictability in human actions; however, this could also lead to unknown actions in unforeseeable situations.

Arguably, the adjustment in giving the rewards is somewhat similar to constructing a rule set in a rule-based model. However, it is significant to note that the pedestrian behavior's agent is the result provided by the output of a neural network. This is different from a rule-based model, in which the rules are constructed manually, the rules or mechanisms in our model are fully transparent. Because of this reason, the result of a rule-based model could be restrained by the designed rules. For instance, a pedestrian could observe state s_1 and s_2 of the

environment and make the action a1. When designing the ruleset, a statement like “*IF state s1 is X THEN do Y*” could ignore the value of s2 entirely. In real life situation, the pedestrian could do differently if the state of s1 is X but the state of s2 is a certain value. This situation could be an oversight, but when it happens, it could lead to unnatural behavior, or at worst, it may lead to serious consequences. A model utilizing neural networks is not prone to this problem, as the model has already been trained through a vast number of states of the environment.

While other models could implement the aspects in human factors and human cognition into their models similar to ours, the results will be strictly restricted to the rules which are manually applied. Specifying an exact quantitative value for any behavior could be challenging. Accordingly, a slight deviation of the value could cause the behavior to be unnatural or falls into the category of “uncanny valley” (i.e. when certain aspects are very close to that of humans but still slightly different, the actions would be observed as highly unnatural even compared to when the aspects are less identical).

To fully evaluate the model is considerably difficult. Even when comparing with real-life situations, if the navigation of the pedestrian differs from the real-world data, that still does not mean the accuracy of the model is low. There are many different criteria to assess a pedestrian simulation model, depending on the aspects of the navigation. We concentrated on creating realistic and natural behavior for the pedestrian agents, however, correctly defining “realistic” and “natural” is hard to be accomplished. In real life, even if actual humans see a pedestrian with unrealistic behavior, they might not be able to indicate which behaviors are unnatural.

Within the scope of our study and considering the human factors, we indicate several aspects that could contribute to a human-like navigation behavior. These include: *following the flow of the path*, *maintaining a normal speed*, and *avoiding getting too close* to other pedestrians and objects. The common idea behind these aspects is *avoiding the risk of collisions*. In other words, those behaviors are the realization of the requirements of risk avoidance. Consequently, we believe the idea of risk avoidance is essential for the construction of natural pedestrian behavior. This idea could also be extended in other human behavior-related studies, such as automated automobiles or robotics.

Chapter 8

Conclusion and Future Work

In this chapter, we conclude this dissertation by summarizing the overview of the model together with its components. Subsequently, the primary finding of the study is presented, followed by the suggestions of our study's future work.

8.1 Summary of the model

The behavioral pedestrian model employed several ideas in cognitive science to create believable human behavior for the pedestrian agent. The reinforcement learning technique, as the name suggests, is also a great instrument to realize the ideas thanks to the similarities with many different areas in cognitive science.

Obstacle's danger and its risk assessment are also primary focuses in our study, as they could considerably affect how the agent navigates. These aspects were considered in many parts of the model, for instance, designing the rewarding behavior and specifying the prediction rate for the pedestrian's interacting prediction.

Based on that conception, we designed our behavioral pedestrian model consists of three component tasks:

1. *Pedestrian path-planning task*: This task imitates the initial planning task in the pedestrian's mind. This process happens when the pedestrian aims to reach a certain destination. The path-planning model uses reinforcement learning for the primary plan of the path, considering various common navigation rules including walking along the path, follow the traffic laws, and

avoid getting too close to the boundaries. With the presence of an obstacle, the pedestrian also needs to avoid colliding with. These rules are realized by giving the appropriate rewarding behavior in the reinforcement learning system. To further improve the model, we also incorporate a prediction mechanism in case of a moving obstacle. Two prediction methods are introduced: the single diagonal method for the obstacle moving in a straight direction, and the pedestrian prediction for the pedestrian obstacle. From that, a point-of-conflict is determined, which subsequently substitutes the original obstacle in the learning process.

2. *Pedestrian interacting task*: This task simulates the interaction of the pedestrian agent with the other obstruction. This obstruction is usually a moving obstacle with unpredictable movement, like another pedestrian for instance. Similar to the planning task, this process also utilizes reinforcement learning for fundamental navigation and collision avoidance. The prediction of the obstacle's movement, however, is much different from the prediction in the path-planning task as in this process, the agent needs to carefully act correspondingly to the movement of the obstacle. To do this, we proposed an interpolation method to determine the trajectory of the obstacle's movement as well as the confidence of the agent in that prediction.
3. *Pedestrian decision planner*: This task helps the agent decide when to use the path-planning task and when to use the interacting task. We designed the decision planner as a sufficient rule-based model from the agent's observation of the environment.

The behavioral pedestrian simulation model benefits from adopting different theories in cognitive science, for example, the strategic thinking process and the predictive system of the human brain. Accordingly, the model has proven to be capable of constructing much more realistic human behavior in pedestrian navigation.

8.2 Conclusion

We proposed a novel behavioral pedestrian simulation model that can replicate a remarkable realistic human navigation behavior. This could be achieved thanks

to the utilization of reinforcement learning, considering several ideas in cognitive science. This results in more realistic pedestrian behavior compared to other force-based or agent-based pedestrian simulation models, such as Social Force Model. The implemented application has demonstrated a favorable result, with the pedestrian agent's capabilities of many human demeanor like following social rules or avoiding possible danger.

8.3 Future work

Although our model could perform well in many settings similar to that in real life, several problems need to be addressed to further improve our model. This section discussed these problems and deliberate their possible approaches.

8.3.1 Considering the development of humans

Human beings cannot do everything since birth. Instead, they need to learn gradually through their lives. This is particularly true for navigation behaviors. Babies need to learn to walk and find the way to their goal before they can avoid different obstacles. The process of learning continues until they are fully grown. Before people could navigate naturally, they need to learn many different skills, such as recognizing different obstacle types, learning to predict people's behavior or adapt to different cultures.

To realize natural navigation behavior, it is essential to study the development of humans. This could also be beneficial when it is necessary to realize the navigation behavior of young children, for example. In this circumstance, using a method like *curriculum learning* could be appropriate. Curriculum learning is a reinforcement learning technique by providing training environments with different difficulties. By aligning the difficulty to the development of humans, we could have more understanding of the aspects that contribute to natural human behavior.

8.3.2 Approaching reinforcement learning using concepts in neuroscience

Besides cognitive and behavioral science, many other scientific fields in neuroscience are also the inspiration for the research in the human behavioral model, pedestrian navigation model included. Researchers have been adopting the insights of computational neuroscience into the realization of navigation models and have achieved promising results [78]. In the human brain, the cognitive system, which is handled by the hippocampus, is separated from the reinforcement learning process, which is primarily managed by the basal ganglia and some related brain regions. In this study, several ideas of the human hippocampus were considered, however, the reinforcement learning concepts of the basal ganglia have been mostly ignored. In the future, it would be better to design a model structure that separates the cognitive and reinforcement learning process by adopting the computational models of the hippocampus and the basal ganglia. This is particularly beneficial for the research in navigation models as well as other related studies.

8.3.3 Designing a cognitive decision planner

In our model, the cognitive decision planner is implemented as a simple rule-based model. As discussed in Section 4.3, this is much simpler than the actual human thinking process. The reasons we have mentioned are the interlinks between different parts of the brain, the interpretation of the inputs as fuzzy data, and the compliance with the predefined instinct of humans.

Reinforcement learning is also a reliable option for designing the cognitive decision planner model. However, to efficiently realize the model, its component tasks (i.e. the path-planning task and the interacting task in our study) must be highly accurate in simulating navigation behavior. Otherwise, the result could be easily *overfitting*, meaning it could produce incorrect behavior when the component tasks are improved.

The decision planner task could also employ fuzzy logic for its observation and output data. Specifically, concepts such as near, far or minor, major change of the environment could be modeled as fuzzy input. This could create a sense of uncertainty in the way the pedestrian agent behaves.

8.3.4 Increasing the number of obstacles

Our current model has a limitation of having only 1 obstacle at maximum. In many situations in real life, the pedestrian needs to interact with more than 1 obstacle, especially when the pedestrian is walking in a crowded environment. While in most basic cases, this problem could be settled by having the pedestrian observe only the nearest obstacle, the simulation could be unnatural in other complex situations.

To solve this problem, a method that could be implemented is to increase the number of obstacles in the pedestrian path-planning task. However, this also means an increase in the number of inputs of the neural network. The environment also needs to be redesigned in a way that the difficulty of the training would be gradually increased. Otherwise, it would be hard for the agent to learn how to act in a noisy environment.

The model also needs to specify which group the obstacles are in, as in real life, many people often walk in groups. In the human brain, when looking at several people walking in groups, the navigation would be much different from when looking at multiple separated people.

Another issue that needs to be focused on is the “leader following” effect when a pedestrian is walking in an environment with multiple obstacles. Aside from collision avoidance, the pedestrian agent also needs to demonstrate the capability to follow other pedestrians’ behavior, such as speed controlling and distancing. Recently, studies in imitation learning have made immense progress, which could be a great candidate for realizing the leader following mechanism.

Bibliography

- [1] Schadschneider, A., Klingsch, W., Klüpfel, H., Kretz, T., Rogsch, C., & Seyfried, A. (2008). Evacuation dynamics: Empirical results, modeling and applications. arXiv preprint arXiv:0802.1620.
- [2] Foltête, J. C., & Piombini, A. (2007). Urban layout, landscape features and pedestrian usage. *Landscape and urban planning*, 81(3), 225-234.
- [3] Zacharias, J. (2001). Pedestrian behavior pedestrian behavior and perception in urban walking environments. *Journal of planning literature*, 16(1), 3-18.
- [4] Rasouli, A., & Tsotsos, J. K. (2019). Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE transactions on intelligent transportation systems*, 21(3), 900-918.
- [5] Sewalkar, P., & Seitz, J. (2019). Vehicle-to-pedestrian communication for vulnerable road users: Survey, design considerations, and challenges. *Sensors*, 19(2), 358.
- [6] Crociani, L., Vizzari, G., Yanagisawa, D., Nishinari, K., & Bandini, S. (2016). Route choice in pedestrian simulation: Design and evaluation of a model based on empirical observations. *Intelligenza Artificiale*, 10(2), 163-182.
- [7] Helbing, D., & Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, 51(5), 4282.
- [8] Henderson, L. F. (1974). On the fluid mechanics of human crowd motion. *Transportation research*, 8(6), 509-515.

- [9] Lv, W., Wei, X., & Song, W. (2015). Experimental study on the interaction mechanism of cross-walking Pedestrians. In *Traffic and Granular Flow'13* (pp. 219-226). Springer, Cham.
- [10] Farina, F., Fontanelli, D., Garulli, A., Giannitrapani, A., & Prattichizzo, D. (2017). Walking ahead: The headed social force model. *PloS one*, 12(1), e0169734.
- [11] Seyfried, A., Steffen, B., & Lippert, T. (2006). Basics of modelling the pedestrian flow. *Physica A: Statistical Mechanics and its Applications*, 368(1), 232-238.
- [12] Bonneaud, S., & Warren, W. H. (2012). A behavioral dynamics approach to modeling realistic pedestrian behavior. In *6th International Conference on Pedestrian and Evacuation Dynamics* (pp. 1-14).
- [13] Teknomo, K., & Millonig, A. (2007, June). A navigation algorithm for pedestrian simulation in dynamic environments. In *Proceedings 11th World Conference on Transport Research*. Berkeley, California.
- [14] Prescott, T. J., & Mayhew, J. E. (1992). Obstacle avoidance through reinforcement learning. In *Advances in neural information processing systems* (pp. 523-530).
- [15] Everett, M., Chen, Y. F., & How, J. P. (2021). Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9, 10357-10377.
- [16] Chen, Y. F., Everett, M., Liu, M., & How, J. P. (2017, September). Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1343-1350). IEEE.
- [17] Trinh, T. T., Vu, D. M., & Kimura, M. (2020, March). A pedestrian path-planning model in accordance with obstacle's danger with reinforcement learning. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System* (pp. 115-120).
- [18] Bubic, A., Von Cramon, D. Y., & Schubotz, R. I. (2010). Prediction, cognition and the brain. *Frontiers in human neuroscience*, 4, 25.

- [19] Ikeda, T., Chigodo, Y., Rea, D., Zanlungo, F., Shiomi, M., & Kanda, T. (2013). Modeling and prediction of pedestrian behavior based on the sub-goal concept. *Robotics*, 10, 137-144.
- [20] Trinh, T. T., Vu, D. M., & Kimura, M. (2020, June). Point-of-Conflict Prediction for Pedestrian Path-Planning. In *Proceedings of the 12th International Conference on Computer Modeling and Simulation* (pp. 88-92).
- [21] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- [22] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- [23] Hoogendoorn, S. P., & Bovy, P. H. (2004). Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological*, 38(2), 169-190.
- [24] Yamaguchi, K., Berg, A. C., Ortiz, L. E., & Berg, T. L. (2011, June). Who are you with and where are you going?. In *CVPR 2011* (pp. 1345-1352). IEEE.
- [25] Kruse, T., Pandey, A. K., Alami, R., & Kirsch, A. (2013). Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12), 1726-1743.
- [26] Juliani, A., Berges, V. P., Teng, E., Cohen, A., Harper, J., Elion, C.,... & Lange, D. (2018). Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*.
- [27] Daamen, W., Hoogendoorn, S., Campanella, M., & Versluis, D. (2014). Interaction behavior between individual pedestrians. In *Pedestrian and Evacuation Dynamics 2012* (pp. 1305-1313). Springer, Cham.
- [28] Pelechano, N., & Badler, N. I. (2006). Modeling crowd and trained leader behavior during building evacuation. *IEEE computer graphics and applications*, 26(6), 80-86.
- [29] Piaggio, M. (1999). An efficient cognitive architecture for service robots. *Journal of Intelligent Systems*, 9(3-4), 177-202.

- [30] Rehder, E., Wirth, F., Lauer, M., & Stiller, C. (2018, May). Pedestrian prediction by planning using deep neural networks. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5903-5908). IEEE.
- [31] Trinh, T. T., & Kimura, M. (2021). The Impact of Obstacle's Risk in Pedestrian Agent's Local Path-Planning. *Applied Sciences*, 11(12), 5442.
- [32] Ijaz, K., Sohail, S., & Hashish, S. (2015, March). A survey of latest approaches for crowd simulation and modeling using hybrid techniques. In 17th UKSIMAMSS international conference on modelling and simulation (pp. 111-116).
- [33] Teknomo, K., Takeyama, Y., & Inamura, H. (2000). Review on microscopic pedestrian simulation model. In Proceedings Japan Society of Civil Engineering Conference.
- [34] Golledge, R. G. (1995, September). Path selection and route preference in human navigation: A progress report. In International conference on spatial information theory (pp. 207-222). Springer, Berlin, Heidelberg.
- [35] Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481), 933-942.
- [36] Martinez-Gil, F., Lozano, M., & Fernández, F. (2011, May). Multi-agent reinforcement learning for simulating pedestrian navigation. In International Workshop on Adaptive and Learning Agents (pp. 54-69). Springer, Berlin, Heidelberg.
- [37] Kretschmar, H., Spies, M., Sprunk, C., & Burgard, W. (2016). Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research*, 35(11), 1289-1307.
- [38] Chung, W., Kim, S., Choi, M., Choi, J., Kim, H., Moon, C. B., & Song, J. B. (2009). Safe navigation of a mobile robot considering visibility of environment. *IEEE Transactions on Industrial Electronics*, 56(10), 3941-3950.

- [39] Melchior, P., Orsoni, B., Laviolle, O., Poty, A., & Oustaloup, A. (2003). Consideration of obstacle danger level in path planning using A* and fast-marching optimisation: comparative study. *Signal processing*, 83(11), 2387-2396.
- [40] Trung, T. T., & Kimura, M. (2019). Reinforcement learning for pedestrian agent route planning and collision avoidance (安全性). 電子情報通信学会技術研究報告= IEICE technical report: 信学技報, 119(210), 17-22.
- [41] Keen, H. A., Nelson, O. L., Robbins, C. T., Evans, M., Shepherdson, D. J., & Newberry, R. C. (2014). Validation of a novel cognitive bias task based on difference in quantity of reinforcement for assessing environmental enrichment. *Animal cognition*, 17(3), 529-541.
- [42] Guangzuo, C., Xuefeng, W., & Boling, L. W. (2011, July). A cognitive model of human thinking. In 2011 Seventh International Conference on Natural Computation (Vol. 2, pp. 992-996). IEEE.
- [43] Naveed Uddin, M. (2019). Cognitive science and artificial intelligence: simulating the human mind and its complexity. *Cognitive Computation and Systems*, 1(4), 113-116.
- [44] Werner, S., Krieg-Brückner, B., Mallot, H. A., Schweizer, K., & Freksa, C. (1997). Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation. In *Informatik'97 Informatik als Innovationsmotor* (pp. 41-50). Springer, Berlin, Heidelberg.
- [45] Tripp, S. (2001). Cognitive navigation: Toward a biological basis for instructional design. *Journal of Educational Technology & Society*, 4(1), 41-49.
- [46] Stites, M. C., Matzen, L. E., & Gastelum, Z. N. (2020). Where are we going and where have we been? Examining the effects of maps on spatial learning in an indoor guided navigation task. *Cognitive research: principles and implications*, 5(1), 1-26.
- [47] Giovannangeli, C., & Gaussier, P. (2008, September). Autonomous vision-based navigation: Goal-oriented action planning by transient states prediction, cognitive map building, and sensory-motor learning. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 676-683). IEEE.

- [48] Levita, L., & Muzzio, I. A. (2010). Role of the hippocampus in goal-oriented tasks requiring retrieval of spatial versus non-spatial information. *Neurobiology of Learning and Memory*, 93(4), 581-588.
- [49] Epstein, R. A., Patai, E. Z., Julian, J. B., & Spiers, H. J. (2017). The cognitive map in humans: spatial navigation and beyond. *Nature neuroscience*, 20(11), 1504-1513.
- [50] Behrens, T. E., Muller, T. H., Whittington, J. C., Mark, S., Baram, A. B., Stachenfeld, K. L., & Kurth-Nelson, Z. (2018). What is a cognitive map? Organizing knowledge for flexible behavior. *Neuron*, 100(2), 490-509.
- [51] Carton, D., Nitsch, V., Meinzer, D., & Wollherr, D. (2016). Towards assessing the human trajectory planning horizon. *Plos one*, 11(12), e0167021.
- [52] Andreev, S., Dibbelt, J., Nöllenburg, M., Pajor, T., & Wagner, D. (2015). Towards realistic pedestrian route planning. In 15th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2015). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [53] Zhang, L., Liu, M., Wu, X., & AbouRizk, S. M. (2016). Simulation-based route planning for pedestrian evacuation in metro stations: A case study. *Automation in Construction*, 71, 430-442.
- [54] Reitter, D., & Lebiere, C. (2010). A cognitive model of spatial path-planning. *Computational and Mathematical Organization Theory*, 16(3), 220-245.
- [55] Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological review*, 55(4), 189.
- [56] Ludvig, E. A., Bellemare, M. G., & Pearson, K. G. (2011). A primer on reinforcement learning in the brain: Psychological, computational, and neural perspectives. *Computational neuroscience for advancing artificial intelligence: Models, methods and applications*, 111-144.
- [57] Ampofo-Boateng, K., & Thomson, J. A. (1991). Children's perception of safety and danger on the road. *British Journal of Psychology*, 82(4), 487-505.

- [58] Stoker, P., Garfinkel-Castro, A., Khayesi, M., Odero, W., Mwangi, M. N., Peden, M., & Ewing, R. (2015). Pedestrian safety and the built environment: a review of the risk factors. *Journal of Planning Literature*, 30(4), 377-392.
- [59] Robin, T., Antonini, G., Bierlaire, M., & Cruz, J. (2009). Specification, estimation and validation of a pedestrian walking behavior model. *Transportation Research Part B: Methodological*, 43(1), 36-56.
- [60] Elliott, J. R., Simms, C. K., & Wood, D. P. (2012). Pedestrian head translation, rotation and impact velocity: The influence of vehicle speed, pedestrian speed and pedestrian gait. *Accident Analysis & Prevention*, 45, 342-353.
- [61] Goh, P. K., & Lam, W. H. (2004). Pedestrian flows and walking speed: a problem at signalized crosswalks. *Institute of Transportation Engineers. ITE Journal*, 74(1), 28.
- [62] Botvinick, M.; Weinstein, A. Model-based hierarchical reinforcement learning and human action control. *Philos. Trans. R. Soc. B Biol. Sci.* 2014, 369, 20130480
- [63] Karasev, V., Ayvaci, A., Heisele, B., & Soatto, S. (2016, May). Intent-aware long-term prediction of pedestrian motion. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2543-2549). IEEE.
- [64] Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A.,... & Srinivasa, S. (2009, October). Planning-based prediction for pedestrians. In 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3931-3936). IEEE.
- [65] Møgelmoose, A., Trivedi, M. M., & Moeslund, T. B. (2015, June). Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations. In 2015 IEEE Intelligent Vehicles Symposium (IV) (pp. 330-335). IEEE.
- [66] Goto, K., Kidono, K., Kimura, Y., & Naito, T. (2011, June). Pedestrian detection and direction estimation by cascade detector with multi-classifiers utilizing feature interaction descriptor. In 2011 IEEE Intelligent Vehicles Symposium (IV) (pp. 224-229). IEEE.

- [67] Dominguez-Sanchez, A., Cazorla, M., & Orts-Escolano, S. (2017). Pedestrian movement direction recognition using convolutional neural networks. *IEEE transactions on intelligent transportation systems*, 18(12), 3540-3548.
- [68] Yi, S., Li, H., & Wang, X. (2016, October). Pedestrian behavior understanding and prediction with deep neural networks. In *European Conference on Computer Vision* (pp. 263-279). Springer, Cham.
- [69] Quintero, R., Almeida, J., Llorca, D. F., & Sotelo, M. A. (2014, June). Pedestrian path prediction using body language traits. In *2014 IEEE Intelligent Vehicles Symposium Proceedings* (pp. 317-323). IEEE.
- [70] Schneider, N., & Gavrila, D. M. (2013, September). Pedestrian path prediction with recursive bayesian filters: A comparative study. In *German Conference on Pattern Recognition* (pp. 174-183). Springer, Berlin, Heidelberg.
- [71] Asahara, A., Maruyama, K., Sato, A., & Seto, K. (2011, November). 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).
- [72] Kahneman, D., & Tversky, A. (1996). On the reality of cognitive illusions.
- [73] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T.,... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928-1937). PMLR.
- [74] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.
- [75] Koh, P. P., & Wong, Y. D. (2013). Comparing pedestrians' needs and behaviours in different land use environments. *Journal of Transport Geography*, 26, 43-50.
- [76] Jaros, M., Di Angelo, M., & Ferschin, P. (2016, July). Modeling and simulation of pedestrian behaviour: As planning support for building design. In *2016 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH)* (pp. 1-8). IEEE.

- [77] Barton, B. K., & Schwebel, D. C. (2007). The roles of age, gender, inhibitory control, and parental supervision in children's pedestrian safety. *Journal of pediatric psychology*, 32(5), 517-526.
- [78] Sukumar, D., Rengaswamy, M., & Chakravarthy, V. S. (2012). Modeling the contributions of Basal ganglia and Hippocampus to spatial navigation using reinforcement learning. *PLoS One*, 7(10), e47467.
- [79] ISO/IEC Guide 51: 2014. (2014). Safety Aspects—Guidelines for their Inclusion in Standards.