A Generic Approach to Modelling Individual Behaviours in Crowd Simulation

Quanbin Sun

PhD thesis

University of Salford

School of Built and Environment

2013

TABEL OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF FORMULAS

DECLARATION

I hereby declare that the research study presented in this thesis was solely carried out by me (i.e. Quanbin Sun). It has not been previously submitted to this or any other institute for the award of a PhD degree or any other qualification.

Printed Name: QUANBIN SUN

Signature: ______________

Date: August 2013

ABBREVIATIONS

1	AI	Artificial Intelligence
2	CA	Cellular Automata
3	CPG	Cell and Portal Graph
4	FPS	Frames Per Second
5	IDE	Integrated Development Environment
6	NPC	Non-player Character
7	SDK	Software Development Kit
8	UDK	Unreal Developer Kit
9	UE	Unreal Engine
10	UI	User Interface

ABSTRACT

Crowd simulation has been widely used to simulate crowd dynamics and their behaviours. However, majority of existing studies can only simulate a specific scenario or behaviour. Although recent developments have attempted to integrate different individual behaviours in order to achieve a more realistic simulation result, it is still very complex and those crowd models often require significant modifications.

This study is therefore aimed to develop a generic crowd model, which provides the flexibility to configure and represent different scenarios, as well as the ability to demonstrate individual differences on crowd behaviours. The theoretical principle of the proposed crowd model is based on the combination of force-based modelling and agent-based modelling. A unified core mathematical formula, which contains seven key parameters, is developed to represent the generic behaviour effects. In addition, a Behaviour Library is developed to present a set of basic behaviours by using the unified formula and subsequently, more complex behaviours could be formed by combining the basic behaviours. The proposed crowd model is implemented in a simulation environment by using Microsoft XNA framework. A number of well-known crowd behaviours are tested with the crowd model for validation. The proposed crowd model is further validated by simulating real life experiments and comparing its results.

This research study presents a novel approach to simulate crowd behaviour at individual level by introducing a generic crowd model that can be configured into specific scenarios. It introduces a theoretical concept, through which different behaviour effects could be quantified by a unified mathematical formula. As a result, crowd modelling and simulation of different scenarios can be significantly simplified. For future work, the proposed crowd model can be tested under complex environment in order to fine-tune its theoretical model and to expand the Behaviour Library.

Chapter 1 INTRODUCTION

1. 1 Research Context

1. 1. 1 Introduction to Crowd Modelling and Simulation

Many studies (M. Liu & S. M. Lo 2011; Kobes, Helsloot, de Vries, et al. 2010; Drury et al. 2009; Kobes, Helsloot, Vries, et al. 2010) on emergency events suggest that crowded environments (e.g. shopping malls, football stadiums) can cause crowd panic which can result in fatalities. However, studies using real-life experiments are usually expensive, both in time and resources. As an alternative approach, with less requirements for time and resources, crowd simulation is introduced to observe and analyse the movements and the behaviours (relating to movement) of a large number of people through the aid of computer programmes which can be represented in the 2D or 3D environment. The simulation usually consists of a crowd model and its implementation.

A crowd model can be categorised as a macro scope model or a micro scope model based on the level at which it describes the crowd. As macro scope models do not provide details of individuals, majority of the existing studies fall into category of the micro scope model. In the past 20 years many crowd models (microscopic) and simulations (Santos & Aguirre 2004; E. D. Kuligowski & R. D. Peacock 2005; X. Zheng et al. 2009; Chu 2009; Ng et al. 2010) are developed to assist designers and the emergency services to have a better understanding of crowd behaviour in emergency events. Several typical crowd phenomena (e.g. clogging, pushing, and "faster-is-slower" effect) are demonstrated by various models (X. Zheng et al. 2009; Cheng et al. 2008; S.R. Musse & D Thalmann 1997; Ebihara et al. 1992). Generally, the modelling approaches of these crowd models can be mainly divided into three categories: force-based models, Cellular Automata (CA) models, and agent-based models.

The force-based models consider that individuals in a crowd are affected by some forms of forces and their motions are determined by the total effects of those forces which are calculated through mathematical methods. This concept was first introduced in the 'Boids' programme (Reynolds 1987) in 1986 which simulates the motion of a flock of birds. In the flock, each bird updates its position by applying a steering force. In 1995, the Social Force model (Dirk Helbing & Peter Molnar 1995) was proposed to describe the movements of pedestrians that are determined by the forces which are generated from nearby crowd and physical objects. This model was further developed (D. Helbing et al. 2000) to simulate panic situations by interpreting social psychology issues, and was then tested by Parisi and Dorso (2007) in a room exit scenario. Heigeas et al. (2003) also introduced a physically-based particle system to model emergent crowd behaviours such as jamming. The force-based models can provide precise position and orientation information on individuals as they have continuous time and spatial representations of a crowd. However, individual behaviours (e.g. following, communications, or interactions) are often ignored in the force-based models as the processes of thinking and decision-making are difficult to be interpreted by mathematical equations.

The Cellular Automata (CA) model was originally invented by Von Neumann (1966) in order to create self-replicator machines in 1966. It was subsequently introduced into crowd modelling by Wolfram (Wolfram 1983; Wolfram 1986; Wolfram 2002). In the CA model, the fields (e.g. buildings, streets, etc.) are represented by a collection of equal size cells. Each cell can only be occupied by one individual at one time and a cell updates its state depending upon the states of adjacent cells. The CA modelling approach are widely used in the simulations of evacuation processes (Kirchner & Schadschneider 2002; Perez et al. 2002; D. Zhao et al. 2006) and in the studies of crowd movement in a bi-directional counter flow (YF Yu & WG Song 2007; Z. Wang et al. 2012; Yue et al. 2010; Jian et al. 2005). Although the CA model has the strength of simplicity in its representation of field and crowd movement, because of its fixed-size cells, it has some limitations. For example, the maximum crowd density is limited by the total number of cells; flow rates through doors could be inaccurate because the cells may not totally align with the environment geometrically (Nuria

Pelechano & a Malkawi 2008); and an individual's physical size has to be the same size as the cell thus his/her movement is not continuous in terms of time and space.

Agent-based modelling is introduced to integrate the human decision making process in crowd simulation (J. Dijkstra et al. 2000; Macal & North 2007; Stefania Bandini et al. 2007; Luo et al. 2008; Bonabeau 2002). It is considered as an appropriate approach because the agents are designed to be autonomous, independent, interactive and intelligent. Agent-based models are usually combined with CA modelling to represent the movements of agents (Hamagami & Hirata 2003; Stefania Bandini et al. 2007). They can also be combined with force-based modelling in order to take into account individual behaviours. For example, intelligent autonomous agents can be implemented on top of steering behaviours (Reynolds 1999). Or agents can be used to simulate group behaviour alongside with the Social Force model (Braun et al. 2003). It has been suggested (Nuria Pelechano & Norman I Badler 2006) that an agent-based model can be created at the top level for communication and navigation, while the Social Force model can be applied at the bottom level to represent the crowd local motions.

1. 1. 2 The Need for New Research

In most of the existing studies, crowds are usually treated as homogeneous but some research studies (Nuria Pelechano & Norman I Badler 2006; Braun et al. 2003; Shendarkar et al. 2008) show that individual behaviours can affect crowd behaviours, i.e. a heterogeneous crowd exhibit a different performance. Several recommendations (Nuria Pelechano & a Malkawi 2008; X. Zheng et al. 2009) are suggested to improve crowd modelling. For example, it is crucial to include physical interactions between individuals in order to interpret crowd behaviours; further research should consider combining different modelling approaches; and models should increase crowd heterogeneity and demonstrate it through. Although these requirements have been achieved to some extent in some studies (Dirk Helbing & Peter Molnar 1995; D. Helbing et al. 2000; Nuria Pelechano & Norman I Badler 2006; Stefania Bandini et al. 2007), there is still a lack of crowd models which are able to describe the relationships between behaviours and movement systematically and enable crowd heterogeneity.

Furthermore, existing crowd models are usually designed for specific scenarios or for certain crowd behaviours. These models use different methods or mechanisms to represent individual and crowd behaviour. It is difficult for them to represent new behaviours because to simulate new crowd behaviours may necessitate the introduction of additional methods or rules which require further modifications to existing models. There have been some partial attempts to address this issue. For example, Pelechano et al. (2008) proposed a framework (HiDAC + MACES + CAROSA) to offer a configurable crowd simulation environment but it focused mainly on behaviour animations and on graphic representation. Moussaid et al. (2011) introduced a solution that combined cognitive heuristic rules and contact forces to simulate crowd dynamics but this solution did not consider individual differences. There is still a challenge in building a model which can integrate different crowd behaviours and interpret how such behaviours affect individuals' movement under a unified mechanism that has the flexibility to represent different scenarios.

To summarise, two research needs have been identified in order to simulate crowds in a more realistic way:

- A need to increase the heterogeneity in a crowd simulation and to model individual behaviours;
- A need to develop a generic behavioural model in order to represent complex individual behaviours in different scenarios.

1. 1. 3 The Research Scope in this Study

This PhD study mainly focuses on the crowd behavioural modelling aspects of crowd simulation. More specifically, the following topics are discussed in this thesis:

- How to interpret and calculate the effects of different behaviours on individuals' movements.
- How to represent heterogeneity in a crowd and their influences on crowd behaviours.

- How to design a crowd model that provides the flexibility in future expansion, e.g. integrating high level artificial intelligence, improving graphical representation, etc.

1. 2 Aim and Objectives

The aim of this research is to develop and implement a crowd model which provides the flexibility to configure individual behaviours (i.e. increasing heterogeneity) and the ability to represent the interactions between individuals in order to simulate and analyse crowd movement. In order to achieve this aim, the following objectives need to be accomplished:

Objective 1. To identify the key elements and research needs in crowd modelling and simulation.

Objective 2. To review crowd modelling approaches, crowd models, simulation applications, crowd behaviours, model design technologies, and simulation software in the context of crowd simulation.

Objective 3. To define a unified method of representing individual behaviours by taking into account crowd heterogeneity.

Objective 4. To design a crowd model that can represent human behaviours and the complex effects of these behaviours on movements.

Objective 5. To implement a prototype simulation system for the proposed crowd model.

Objective 6. To evaluate and validate the crowd model with a series of simulations.

1. 3 Research Methodologies

Research methodology is an attempt to validate the rationale behind the selected research design and provides a justification of why it is appropriate in solving the selected research problem (Bell 2010). It is agreed that the effective use of suitable research strategies in the right way at the right time is always essential for good research (Robson 2002). Given that the purpose and nature of this PhD study is to identify new research needs and to design an appropriate crowd model, the research methods employed in this research study consist of a literature review, software prototyping, and case studies.

1. 3. 1 Literature Review

A literature review provides a solid background and comprehensive knowledge on the subject of crowd modelling and simulation which are essential for this research study. The main purposes of conducting a literature review for this study are:

- To review the studies on the subject of crowd simulation and to identify the needs for developing new research.
- To identify the existing approaches to, and the methods and technologies of, crowd modelling and crowd simulation.
- To provide simulation studies and experimental data of crowd behaviours for crowd model validation.

1. 3. 2 Software Prototyping

In this research study, the design of the crowd model cannot be fully specified at the beginning stage, as the specifications need to be developed during the testing of the proposed model to meet the aim of the PhD study. However, the test and the evaluation of a crowd model require the implementation of that model. Due to the similar nature of crowd model design and simulation programme implementation, the software prototyping research method is selected in this research study.

In order to demonstrate and validate the crowd model, examples of selected scenario simulations are presented by using the final prototype.

1. 4 Organisation of the Thesis

This thesis consists of eight chapters, each of which is broken down into a number of sections and subsections that present the research in detail. The content of each chapter is summarised as follows:

Chapter 1 introduces the background and the aim and objectives of this PhD study. It also provides a brief description of the research methodology that is adopted in this research study. At the end of this chapter, it outlines how the thesis is structured and organised.

Chapter 2 describes the research methodology of the study. It introduces the selection of the appropriate research methods and explains the rationale behind the judgements as to which research methods are used. It also presents the research procedure of the study and describes in detail the outcomes of each element in the methodology.

Chapter 3 includes a comprehensive review of crowd modelling and simulation within the scope of this research study. In the first section of this chapter, it introduces different aspects of crowd simulation and its relationship to crowd modelling, followed by detailed reviews on crowd modelling approaches, and typical crowd models and their represented crowd behaviours. In the second section, it mainly reviews the technologies that have been, or can be employed, in crowd simulation.

Chapter 4 presents the detailed design of the crowd model. The contents of this chapter include a theoretical view of the crowd model, the model structure, the representation of the behaviours and the calculation of their effects, the design of agents, the concept of a Behaviour Library, and the procedure for the simulation of the crowd model.

Chapter 5 describes the implementation of the proposed model. It first introduces the software environment and the technologies employed to implement the simulation system. Then it presents the representation and implementation of each element in the crowd model.

Chapter 6 demonstrates the applications of the proposed crowd model. Three scenarios (i.e. walking through a corridor, exiting from a building, and evacuation from a shopping mall) are selected to show the capabilities of the proposed crowd model in representing crowd heterogeneity, interactions of individuals' behaviours, and environmental influences and constraints on crowd movement.

Chapter 7 provides the validation of the crowd model in this study. Three series of simulations are conducted. The first simulation is to reproduce the well-known "lane formation in bi-directional crowd flow” phenomena. The latter two analyse the studies of “consensus decision making” in small and large human groups in real-life, and present these crowd behaviours in a simulation environment. The results of these three simulations indicate that the proposed model can present consistent and reliable crowd behaviours which are found in both simulation and real-life studies. Furthermore, the latter two simulations also provide additional findings to the original real-life experiments, which further demonstrate the application of the proposed model.

Chapter 8 summarises the whole PhD research work. It firstly provides a summary of the proposed crowd model and confirms the accomplishments of the objectives. Then, it presents the contributions that this research study has made to the knowledge of this subject area. Last but not least, it discusses possible future work.

Chapter 2 LITERATURE REVIEW: CROWD MODELLING & SIMULATION

This literature review on crowd modelling and simulation briefly introduces different aspects from within the relevant studies. It mainly focuses on the modelling approaches when designing the crowd model and on the applications of current crowd simulation studies. Discussions and analysis have been imbedded in the review process. New research trends and requirements are presented at the end as a summary.

2. 1 Overview

This section provides an overview of studies in crowd modelling and simulation. It firstly briefly describes what crowd models and crowd simulations are. Then, it introduces the purposes and applications of such studies, followed by the benefits and the limitations of crowd modelling and simulations. Finally, the relevant terminologies used in this thesis are listed.

2. 1. 1 What is a Crowd Model?

Although no formal definition can be found in the literature, it is possible to summarise that a crowd model represents a system that describes crowds' behaviours and their movements via some predefined mechanisms (e.g. a set of formulas, a collection of rules, etc.).

2. 1. 2 What is Crowd Simulation?

Crowd simulation usually refers to the representation of crowds' movements and their behaviours via 2D or 3D computer graphics. A crowd simulation system usually includes a crowd model that determines the behaviours and movements of a crowd, a graphic engine that is used to represent the crowd, and a virtual environmental 2D/3D model.

Although most studies have focused on the designs (i.e. physically interactions, behaviour rules, artificial intelligence, etc.) of crowd models, there are also many studies which have studied graphic representation and simulation system hierarchy (e.g. how to increase computer programme efficiency). In this thesis, the purpose of crowd simulation is to represent the movement and behaviour of a crowd in a virtual environment and the crowd model is considered as a more important part. In other words, crowd simulation system serves as a tool or a method in order to demonstrate the crowd model in the context of this PhD study.

2. 1. 3 Purposes and Usages of Crowd Modelling and Simulations

2. 1. 3. 1 An Alternative Way in Crowd Behaviour Studies

The main purpose of using simulations (instead of real-life observations and experiments) to study crowd behaviours is to save time and resources. Compared to the traditional research approach (i.e. studying real people), computer simulation offers an alternative way of carrying out studies with less requirements for time and resources.

2. 1. 3. 2 Usages

Crowd modelling and simulations have been used in many fields but are used mainly in the following three areas:

Simulations of Emergency Events

It has been reported that emergency events can cause crowd panic and result in fatalities (M. Liu & S. M. Lo 2011; Kobes, Helsloot, de Vries, et al. 2010; Drury et al. 2009; Kobes, Helsloot, Vries, et al. 2010). However, due to the unexpected nature of these events, data collection and post-analysis are usually difficult and limited. The fire drills and other emergency grills can provide valuable in studying such events but they require large resources and are unlikely to have large numbers of experiments. Aiming to provide a more economic and efficient approach, crowd models and simulations (Santos & Aguirre 2004; E. D. Kuligowski & R. D. Peacock 2005; X. Zheng et al. 2009) were developed to assist designers and the emergency services to have a better understanding of crowd behaviour at such events.

Studies of Collective Behaviours

In crowd modelling and simulations, a collective behaviour refers to the crowd acting in a way that has not been explicitly defined in the crowd model. For example, there are many studies on the bi-directional counter-flow of walking pedestrians (Z. Wang et al. 2012; W. Fang et al. 2003; Yue et al. 2010; Jian et al. 2005; Lam et al. 2003). There are also studies on leadership or grouping behaviour (M. Zheng et al. 2002; Braun et al. 2003; Nuria Pelechano & Norman I Badler 2006; X Pan et al. 2006). Additionally, there are many studies on the pedestrian behaviour in the streets (Moussaïd et al. 2009; S. Bandini et al. 2002; Stern & Richardson 2005; D'Ambrogio et al. 2009; R. Lee & R. Hughes 2007). These studies modelled the behaviours or crowd phenomena studied by social psychologists and tried to explain why and how those behaviours happened.

Building Layout or User Behaviour Evaluations

Crowd simulations can be used to evaluate the effect of the layout design of buildings on crowd behaviours (e.g. finding bottlenecks, testing exits' usages or whether the corridor/stair widths are sufficient) (Tang & X. Zhang 2008; S Lo et al. 2008; X. Zheng et al. 2009; J. Yuan et al. 2009). Crowd simulations can also be used to study crowd behaviours and their effects in given buildings (e.g.) (M. Zheng et al. 2002; J. Dijkstra 2008; Hoes et al. 2009; W. Shen et al. 2012).

2. 1. 4 Benefits and Limitations of Crowd Simulations

2. 1. 4. 1 Benefits

Crowd simulations represent crowd behaviour via computer programmes, which have many advantages over traditional real-life studies:

Requires Less Resources

Computer simulations do not require experimental venues or participants. Everything happens in the virtual world and can be observed on the computer screen. The costs of computer hardware are significantly less than the cost of conducting traditional real-life studies (In fact, a large number of crowd simulations can be run on a modern computer that used for daily working). In addition, alongside the rapid developments in computer technology, dedicate software programmers become dispensable in the studies of crowd simulation as many crowd modelling and simulation tools require only basic programing knowledge or even can be used by

Consumes Less Time

Apart from the expensive resources that are required in traditional studies, such studies also require time to complete the experiments that are required. For example, a fire drill may take ten minutes until all the people have evacuated from the building. A computer simulation can dramatically increase the speed of this type of experiment (e.g. the computer can calculate the period of one minute in the real world within one second). Additionally, real-life studies usually do not have large number of repeated experiments (e.g. Dyer et al.'s findings on consensus decision making (J. R. G. Dyer et al. 2009) were based on less than twenty). Computer simulations can repeat a lot more times than most real-life studies. For example, in this thesis, one evaluation simulation simulated Dyer et al.'s experiments (2009) one thousand and six hundred times and was able to reveal new findings.

Easy to Collect Data

Because the simulations are computer programmes, all the data can be easily captured for post-analysis. Furthermore, the simulations can be paused at any stage of the simulation which increases the experience of during-simulation observations.

Flexibility on Configuration

The individuals' personalities and abilities are totally controlled by the computer programme which means that simulations can configure the compositions of the crowd with various combinations while real-life studies may be restricted to the available participants.

2. 1. 4. 2 Limitation

Crowd simulation has one primary limitation: **it is a virtual simulation of the real world based on a theoretical crowd model**. This means there is no way to guarantee that the findings from the crowd simulation can be found in real life.

Although many crowd models have been carefully calibrated with real life data and can provide accurate results in specific scenarios, it still remains doubtful when applying them in other scenarios as the environment, the crowd, and the situations have been changed.

2. 1. 5 The Terminology in this Study

In the literature, many terms have been used to describe crowd models and crowd behaviours. However, some terms may have specific meanings in crowd modelling studies and, alternatively, various terms in different crowd models could refer to the same thing. It is necessary to define the terminology used in this research study to provide clarification. Additionally, this terminology also includes some terms that are rarely seen in any other subject other than crowd modelling.

2. 1. 5. 1 Crowd

Definition

The definition of the word "crowd" is "*a large number of people gathered together in a disorganized or unruly way*" (Oxford Dictionaries 2013b), and this indicates the three features of a crowd in crowd modelling: *large number, together, and disorganised*. Although there is no strict criterion as to how many is a large number of people in the existing studies, most of the test simulations of crowd models have fifty to one hundred and fifty people. "*together"* indicates that crowd modelling should consider the interactions between crowd members and their influences on each other. "*disorganized*" means that crowd movements and behaviours are modelled under certain physical laws and social rules rather than by some pre-defined computer animations.

Synonyms

“pedestrians”, “particles”, “agents”, and ”group”.

2. 1. 5. 2 Individual

Definition

In crowd modelling, the term “individual” refers to a single entity within the whole crowd which is the same as the definition of the word “individual”: “a *single human being as distinct from a group*” (Oxford Dictionaries 2013e).

Synonyms

“pedestrian”, “particle”, “person”, and “agent”.

2. 1. 5. 3 Behaviour

Definition

The term “behaviour” has two meanings in the studies of crowd models. The first one refers to the behaviour at group level which is about the performance of the whole crowd (e.g. evacuation) or some emergent phenomenon in the crowd (e.g. queuing at exits, automated lane formation in a bi-directional walking flow). The second one refers to the behaviour at an individual level which is covered by the definition of “*the way in which one acts or conducts oneself, especially towards others*”(Oxford Dictionaries 2013a).

Synonyms

“crowd behaviour”, “collective behaviour”, “crowd phenomenon”, “crowd dynamics”, and “individual behaviour”.

2. 1. 5. 4 Field

Definition

In this study, the term “field” is used to describe the places where the crowd is moving on.

Synonyms

“cells”, “venues”, “street”, “floor”, “corridors”, “buildings”, “spatial structure”, and “environment”.

2. 1. 5. 5 Movement

Definition

In this study, the term “movement” refers to the position change of an individual during the process of the simulation.

Synonyms

“motion”, “displacement”.

2. 1. 5. 6 Update

Definition

In this study, the term “update” refers to the repeating process of the individuals/crowd in deciding their behaviour and performing relevant actions.

2. 1. 5. 7 Heterogeneous

Definition

The definition of “heterogeneous” is *“diverse in character or content”* (Oxford Dictionaries 2013c). In crowd modelling, the term “heterogeneous” indicates that a crowd is composed of different types of people. In other words, individuals in a crowd can have different characteristics and may have different behaviours under the same circumstances.

Derivative

The noun of heterogeneous is “heterogeneity”.

2. 1. 5. 8 Homogeneous

Definition

The definition of "homogeneous" is "*of the same kind*" (Oxford Dictionaries 2013d). In crowd modelling, the term "homogeneous" indicates a crowd that consists of the same type of people. In other words, every individual in the crowd is exactly the same. They have the same parameters and will execute identical behaviour under the same situation.

Derivative

The noun of homogeneous is "homogeneity".

2. 2 Categorisation of Crowd Models

2. 2. 1 Categorisation Criteria

Existing crowd models can be divided into many categories, by applying different criteria. For example, crowd models can be categorised based on the following criteria:

- **Based on field representation**: course network crowd model, fine crowd network model.
- **Based on the movement representation of individuals**: continuous model and discrete model.
- **Based on crowd composition**: homogeneous model and heterogeneous model.
- **Based on the number of people in the crowd**: small sized crowd model, medium sized crowd model and huge sized crowd model.

From the literature, the most popular categorising method for crowd models is to divide them based on how they model individuals in the crowd - the crowd modelling approaches. However, different studies appear to have different definitions on modelling approaches for existing crowd models. The following section briefly introduces these approaches and presents the categorising method for the crowd model in this research study.

2. 2. 2 Crowd Modelling Approaches

2. 2. 2. 1 In Existing Studies

Despite the relatively short history of crowd modelling, many crowd models (e.g. (D. Helbing et al. 2000; Reynolds 1999; Santos & Aguirre 2004; X. Zheng et al. 2009; Nuria Pelechano et al. 2008)) have been developed for crowd simulation. These crowd models have been divided into many categories in existing reviews or studies. For example:

Zheng et al. (X. Zheng et al. 2009) divided crowd evacuation models into seven approaches:

- Cellular automata models
- Lattice gas models
- Social Force models
- Fluid-dynamic models
- Agent-based models
- Game theoretic models
- Approaches based on experiments with animals

In Kuligowski and Peacock's study (E. D. Kuligowski & R. D. Peacock 2005) of building evacuation models, the modelling methods were categorised as:

- Behavioural models
- Movement models
- Partial behaviour models

Pelechano and Malkawi (Nuria Pelechano & a Malkawi 2008) considered that building evacuation models could be classified as:

- Macroscopic models
- Fluid-dynamic models
- Flow-based models

- Regression models
- Route choice models
- Queuing models
- Gas-kinetics models
- Microscopic models
- Social Forces (particle systems) models
- Rule based models
- Cellular automata models

In Santos and Aguirre's review of emergency evacuation simulation models (Santos & Aguirre 2004), these simulations models were divided into:

- Flow based models
- Cellular automata models
- Agent-based models
- Models that incorporated social dimensions

2. 2. 2. 2 In this Thesis

Based on the literature, it can be seen that there are no universal rules in the classification of crowd models. In this thesis, the crowd models are divided into two categories based on the scope of how they model individuals in a crowd: macro scope crowd models and micro scope models. The macro scope crowd models consider the whole crowd as one entity while the micro scope crowd models consider each individual as a single entity in the crowd. Macroscopic models treat the crowd as a whole and use flows to characterise crowd behaviour and movement while Microscopic models consider movement and behaviour individually and the crowd is formed through the interactions between individuals.

The modelling approaches in this research study are listed and reviewed as follows:

➢ **Macroscopic models**

- Fluid-dynamic models
- Regression model
- Route choice model

- Queuing model

➢ **Microscopic models**

- Force-based models
- Cellular Automata (CA) models
- Agent-based models
- Hybrid crowd models

2. 3 Macro Scope Crowd Models

The macro scope crowd models describe crowd movement at the macro level. They consider the crowd as a whole entity (usually known as a flow) that moves through the environment by following some global rules.

2. 3. 1 Typical Macro Scope Crowd Models

2. 3. 1. 1 Fluid-dynamic Models

The gas-kinetics model (Henderson 1971) can be viewed as the first fluid-dynamic model. It used an analogy with fluid to describe how crowd density and velocity change over time using partial differential equations. In 2000, Hughes (2000) proposed modelling crowd motion based on the hypotheses of ‘‘thinking fluids’’. Later, Helbing et al. (2002) stated the motion of pedestrians as medium and high densities was analogous to the motion of fluids. Based on previous studies, Colombo and Rosini (2005) presented a crowd model describing some typical features of pedestrians such as possible over compressions in a crowd and the fall due to panic in the outflow of people through a door. Because the crowds were treated as fluid flowing in fields (e.g. in buildings), such models were known as the fluid-dynamic models (or the flow-based models)

2. 3. 1. 2 Regression Model (Milazzo et al. 1998)

The regression model predicted the motions of pedestrian flow under certain circumstances by using the statistical relations between flow variables. The characteristics of the flow were decided by the environment (e.g. stairs, corridors, etc.)

2. 3. 1. 3 Route Choice Model (Hoogendoorn 2003)

In the route choice model, the wayfinding of pedestrians was based on the concept of utility. Utility refers to the feeling of comfort, travel time, etc. Pedestrians chose destinations in order to maximize the utility of their trip.

2. 3. 1. 4 Queuing Model (LOVAS 1994)

The queuing model represented the environment (e.g. doors, rooms) by nodes and employed the Markov Chain model to describe how pedestrians move from one node of a network to another (based on transition probabilities and rules).

2. 3. 2 Discussion

Macro scope crowd models present the overall behaviour and movement of a crowd in a given environment. They cannot present the interactions or the individual behaviours within a crowd. Because the present trend of studies of crowd modelling is to focus on the details of individuals, there are currently not many studies at the macro scope level unless there is some special requirement (e.g. the study is not concerned with individuals) or there is a limitation to computing resources. Zhou et al. (2010) pointed out that a macro scope crowd model should be considered if the number in a crowd is huge (e.g. thousands): *"A model for the huge-sized crowd may have to opt for this type of approach due to the computational cost involved. Executing thousands of virtual individuals demands tremendous computing resources, especially when each virtual individual is an intelligent and autonomous entity rather than a simple object".*

2. 4 Micro Scope Crowd Models

Micro scope crowd models consider a crowd at the micro level which means each individual in the crowd has his/her presence. These models are concerned with the behaviour and decisions of individual pedestrians and their interactions with other pedestrians in the crowd.

In this thesis, these models are categorised based on how they explain the relationships between the individuals. They are reviewed in the following categories:

- Force-based models
- Cellular Automata (CA) based models
- Agent-based models
- Hybrid models

2. 4. 1 Force-based Models

Force-based models are those models which interpret the motions of individuals in the crowd as being determined by forces or force-format effects. This type of model has a common feature: **one or more mathematic formulas are used to calculate such forces or force-format effects.**

As there are no standards or guidelines on how to design the formulas to describe individuals' motions in the crowd, the physical laws or motions' rules differ within each study. This section presents some typical models using the force-based modelling approach to demonstrate how these forces can be calculated in different ways. The following models have been reviewed in this section:

- The "Boids" model (Reynolds 1987)
- The generalised Social Force model (Dirk Helbing & Peter Molnar 1995)
- The Social Force model for panic situations (D. Helbing et al. 2000)
- The modified Social Force model (integrated personalities) (M. Zheng et al. 2002)
- The physically-based particle model (Heïgeas et al. 2003)
- The modified particle swarm optimization-based model (Cheng et al. 2008)

2. 4. 1. 1 The "Boids" Model

Overview

In 1987, Reynolds published a computer programme (Reynolds 1987) to simulate the motion of bird flocks which was known as "Boids". This "boids" model introduced three simple steering behaviours to describe how an individual bird manoeuvres based on the positions and velocities of its nearby flock mates. Although strictly speaking, this was not a model to describe crowd motions, it was the earliest computational model that described the motions of an animal group and was based on the concept of a force format effect (which was named the steering behaviour in this model).

Three types of steering behaviours were defined:

- Separation: steering to avoid crowding nearby birds.
- Alignment: steering towards the average heading of nearby birds.
- Cohesion: steering to move toward the average position of nearby birds.

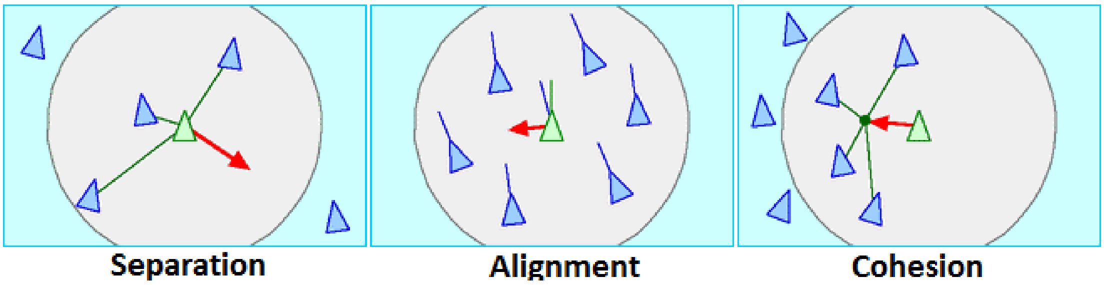


Figure 1 Steering behaviours in the "boids" model (Reynolds 1987)

When looking at one bird in the simulation, it will choose one of the behaviours from the above to match the velocity of the nearby birds.

Details

In the original paper (Reynolds 1987), the algorithms of the steering behaviours were not provided. They were given by simple descriptive rules whichwere listed in the overview section. In other words, this "boids" model introduced a guideline to simulate artificial flocking behaviours. The various implementations of algorithms in "boids" model could result in the different motions of the simulated flocks. For example, a "boids" model with a larger cohesion effect implemented should have a more compact flock than one considered with a smaller cohesion effect.

Discussions

The aim of this model was to introduce a new approach to simulate the flocking behaviour of birds instead of using traditional animation with scripted paths. More importantly, it presented the idea of simulating flock motions by following certain force format-based rules which are comparable to the later force-based models of crowds.

2. 4. 1. 2 Social Force Model (the Generalized Version) (Dirk Helbing & Peter Molnar 1995)

Overview

The origin Social Force model was first proposed in 1995. In the Social Force model, the behaviours of pedestrians are considered to be represented by an equation of motion ("it is possible to put the rules of pedestrian behaviour into an equation of motion.") because the pedestrian's reactions are "rather automatic and determined by his/her experience" (Dirk Helbing & Peter Molnar 1995). Therefore, the term "Social Force" was used to describe the effects that caused the velocity change of a pedestrian. Such effects contained the following aspects:

- The desire to reach a destination comfortably;
- The repulsive effects from other pedestrians and environmental objects (e.g. walls, obstacles, etc.);
- The possible attractive effects from some pedestrians (e.g. friends) and environmental objects (e.g. window displays);

Plus:

- There should be some fluctuations in the behaviours.

In summary, the Social Force model was defined by Formula 1 (which will be introduced in more detail in the following section):

$$\begin{aligned}\frac{d\vec{w}_\alpha}{dt} &:= \vec{F}_\alpha^0(\vec{v}_\alpha, v_\alpha^0\vec{e}_\alpha) + \sum_\beta \vec{F}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta) \\ &\quad + \sum_B \vec{F}_{\alpha B}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_B^\alpha) + \sum_i \vec{F}_{\alpha i}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_i, t) \\ &\quad + \text{fluctuations}.\end{aligned}$$

Formula 1 Social Force model (Dirk Helbing & Peter Molnar 1995)

Details

This section explains the effects that compose the “Social Force” in Formula 1 in more detail:

➢ The desire to reach a destination comfortably

This effect describes a pedestrian α who wants to reach a certain destination in the shortest possible route. In a simple environment (e.g. an open field) without any obstacles, such a route would be a direct line to the destination. In a complex environment (e.g. a building), such a route would be a polygonal line which consists of several sub-destinations because of the environmental constraints (e.g. walls). If $\vec{e}_\alpha^k$ was the next destination or sub-destination, the pedestrian’s desired direction $\vec{e}_\alpha(t)$ his/her next motion could be calculated by:

$$\vec{e}_\alpha(t) := \frac{\vec{r}_\alpha^k - \vec{r}_\alpha(t)}{\|\vec{r}_\alpha^k - \vec{r}_\alpha(t)\|},$$

Formula 2 Desired direction of an pedestrian when he/she moves towards a destination in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In this formula, $\vec{r}_\alpha^k$ denotes the position the destination k. $\vec{r}_\alpha(t)$ denotes the actual position of the pedestrian at time t. Considering that the pedestrian would take a relaxation time τ (it was set as 0.5 s in the model) to change the direction of his/her actual velocity, τ_α denotes the relaxation time of pedestrian α, $\vec{v}_\alpha(t)$ to the desired direction $\vec{e}_\alpha(t)$, v_α^0 represents the desired speed, such an acceleration effect could be described as:

$$\vec{F}_\alpha^0(\vec{v}_\alpha, v_\alpha^0\vec{e}_\alpha) := \frac{1}{\tau_\alpha}(v_\alpha^0\vec{e}_\alpha - \vec{v}_\alpha).$$

Formula 3 effect of reaching a destination in the Social Force model (Dirk Helbing & Peter Molnar 1995)

➢ The repulsive effects from other pedestrians and environmental objects

This type of effect describes a pedestrian α who wants to keep a certain distance from other pedestrians and environmental objects (e.g. walls, obstacles).

The strength of the repulsive effect from other pedestrians depends on the nearby crowd density and on pedestrian α*’s* desired speed v_α^0. It is considered that pedestrian α will feel increasingly uncomfortable while he/she is getting closer to another pedestrian β. Such a feeling can be measured by the repulsive effect via the following formula:

$$\vec{f}_{\alpha\beta}(\vec{r}_{\alpha\beta}) := -\nabla_{\vec{r}_{\alpha\beta}} V_{\alpha\beta}[b(\vec{r}_{\alpha\beta})]$$

Formula 4 Repulsive effect from pedestrian β to pedestrian α in the Social Force model (Dirk Helbing & Peter Molnar 1995)

The formula $V_{\alpha\beta}(b)$ represents repulsive potential and is assumed as "*a monotonic decreasing function of b with equipotential lines having the form of an ellipse that is directed into the direction of motion*" (Dirk Helbing & Peter Molnar 1995). In the simulations of this Social Force model, $V_{\alpha\beta}(b)$ was defined as:

$$V_{\alpha\beta}(b) = V^0_{\alpha\beta} \mathrm{e}^{-b/\sigma}$$

Formula 5 Calculation of $V_{\alpha\beta}(b)$ in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In this formula, $V^0_{\alpha\beta}$ was given the value of 2.1 m^2s^{-2} and σ was given the value of 0.3 m. e was a mathematical constant which approximately equals to 2.71828. b denoted the semi-minor axis of the ellipse and could be calculated through the equation below:

$$2b := \sqrt{(\|\vec{r}_{\alpha\beta}\| + \|\vec{r}_{\alpha\beta} - v_\beta \,\Delta t\, \vec{e}_\beta\|)^2 - (v_\beta \,\Delta t)^2}$$

Formula 6 Equation to calculate b for repulsive potential function in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In this formula, $\vec{r}_{\alpha\beta}$ was defined as $\vec{r}_\alpha - \vec{r}_\beta$. $\vec{r}_\alpha$ denotes the location of the pedestrian α and $\vec{r}_\beta$ denotes the location of pedestrian β. Δt was defined as 2 seconds.

Similar to the repulsive effects from other pedestrians, a pedestrian also feels repulsive effects from environmental objects (e.g. walls, obstacles). The repulsive effect from such an object can be measured by:

$$\vec{F}_{\alpha B}(\vec{r}_{\alpha B}) := -\nabla_{\vec{r}_{\alpha B}} U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$$

Formula 7 Repulsive effect from an environmental object to pedestrian α in the Social Force model (Dirk Helbing & Peter Molnar 1995)

$U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$ was a function which denoted the monotonic decreasing potential between $\vec{r}_\alpha$ and $\vec{r}^{\alpha}_{B}$. In the simulations of this Social Force model, $U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$ was defined as:

$$U_{\alpha B}(\|\vec{r}_{\alpha B}\|) = U^0_{\alpha B} e^{-\|\vec{r}_{\alpha B}\|/R}$$

Formula 8 Calculation of $U_{\alpha B}(\|\vec{r}_{\alpha B}\|)$ in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In this formula, $U^0_{\alpha B}$ was given the value of 10 m^2/s^2 and R was given the value of 0.2 m. e was a mathematical constant which approximately equals to 2.71828. $\vec{r}_{\alpha B}$ was defined as $\vec{r}_\alpha - \vec{r}^\alpha_B$. $\vec{r}_\alpha$ denotes the location of the pedestrian α and $\vec{r}^\alpha_B$ denotes the location of the border B of that environmental object which was closest to the pedestrian α.

- **The possible attractive effect from some pedestrians and environmental objects**

This type of effect can occur when a pedestrian is attracted by other pedestrians (e.g. friends) or environmental objects (e.g. window displays). Such an attractive effect is also dependent on the distance factor, plus it decreases with time t because of the decline of the interest. This attractive effect was modelled by:

$$\vec{f}_{\alpha i}(\|\vec{r}_{\alpha i}\|, t) := -\nabla_{\vec{r}_{\alpha i}} W_{\alpha i}(\|\vec{r}_{\alpha i}\|, t)$$

Formula 9 Attractive effect from another pedestrian or environmental object in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In the formula, $W_{\alpha i}(\|\vec{r}_{\alpha i}\|, t)$ was a function that represented the monotonic increasing potentials. $\vec{r}_{\alpha i}$ was defined as $\vec{r}_\alpha - \vec{r}_i$. $\vec{r}_\alpha$ denotes the location of the pedestrian α and $\vec{r}_i$ denotes the position of the person or object that causes the attractive effect.

(Note: this actual effect was mentioned but not considered in the simulation with Helbing and Molnar's Social Force model (1995)*. Therefore, the suggested equation of $W_{\alpha i}(\|\vec{r}_{\alpha i}\|, t)$ was not present in that research study.)*

- **Repulsive and attractive effects weakened due to perception**

The above repulsive and attractive effects calculations assume that the pedestrian has full awareness of other pedestrians and environmental objects. In reality, a person or an object located behind the pedestrian should have a weaker influence because the pedestrian's perception is limited by his/her sight and sense angle. To take this effect into account, a direction dependent weight was introduced to adjust the repulsive and attractive effects:

$$w(\vec{e},\vec{f}) := \begin{cases} 1 & \text{if } \vec{e}\cdot\vec{f} \geq \|\vec{f}\|\cos\varphi \\ c & \text{otherwise.} \end{cases}$$

Formula 10 Direction dependent weight to adjust the repulsive or attractive effects in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In this formula, $\vec{e}$ denotes the pedestrian's desired direction of motion. 2φ denotes the angle of sight and the angle φ was proposed as $100°$. c was a coefficient to adjust the weakening influence, whose value should be a number between zero and one (0.5 was used in this model).

By taking the direction dependent weights into account, the repulsive and attractive effects on a pedestrian's behaviour becomes:

$$\vec{F}_{\alpha\beta}(\vec{e}_\alpha,\vec{r}_\alpha-\vec{r}_\beta) := w(\vec{e}_\alpha,-\vec{f}_{\alpha\beta})\vec{f}_{\alpha\beta}(\vec{r}_\alpha-\vec{r}_\beta)\,,$$
$$\vec{F}_{\alpha i}(\vec{e}_\alpha,\vec{r}_\alpha-\vec{r}_i,t) := w(\vec{e}_\alpha,\vec{f}_{\alpha i})\vec{f}_{\alpha i}(\vec{r}_\alpha-\vec{r}_i,t)\,.$$

Formula 11 Calculations of repulsive and attractive effects that influence a pedestrian's perception in the Social Force model (Dirk Helbing & Peter Molnar 1995)

➢ Fluctuations in pedestrian's behaviours

The purpose of adding fluctuations is to reflect the random variations in pedestrian's behaviours. On the one hand, such fluctuations can alter the pedestrian's behaviours in the case of a behaviour has equivalent forms (e.g. a pedestrian can walk via the right-side or the left-side in order to avoid the collision with an obstacle in front of him/her). On the other hand, such fluctuations can provide deviations from the given formulas of motion calculations (either deliberate or accidental). (*Note: Helbing and Molnar* (1995) *only raised this concept in their study but did not implement this effect in the simulation for simplicity.)*

➢ Velocity limit on a pedestrian's motion

In addition to the above rules, a relationship between a pedestrian's actual velocity $\vec{v}_\alpha(t)$ and preferred velocity $\vec{w}_\alpha(t)$ has also been introduced. Therefore, the actual speed of a pedestrian cannot exceed his/her maximal speed v_α^{max} (which was limited to $1.3v_\alpha^0$) and the actual velocity $\vec{v}_\alpha(t)$ is defined as:

$$\frac{d\vec{r}_\alpha}{dt} = \vec{v}_\alpha(t) := \vec{w}_\alpha(t)\, g\left(\frac{v_\alpha^{\max}}{\|\vec{w}_\alpha\|}\right)$$

Formula 12 A pedestrian's actual velocity calculation in the Social Force model (Dirk Helbing & Peter Molnar 1995)

In the formula, function *g(x)* is defined as:

$$g\left(\frac{v_\alpha^{\max}}{\|\vec{w}_\alpha\|}\right) := \begin{cases} 1 & \text{if } \|\vec{w}_\alpha\| \le v_\alpha^{\max} \\ v_\alpha^{\max}/\|\vec{w}_\alpha\| & \text{otherwise.} \end{cases}$$

Formula 13 Function g(x) to constrain the pedestrian's actual velocity in the Social Force model (Dirk Helbing & Peter Molnar 1995)

Discussion

This generalized Social Force model can be treated as the first crowd model that stated the velocity change of a pedestrian can be measured by the form of "force" which is generated from a pedestrian's behaviours. This approach presented a way to connect the human behaviours looked at in social psychology studies with the motion change of pedestrians looked at in physics' studies.

This model proposed that the velocity of a pedestrian is determined by the "Social Force" which consists of three behavioural effects: the pedestrian's own desire, the repulsive effects from the surroundings, and the attractive effects from some special targets. The model presented a guideline as to how to interpret the effects of behaviour on pedestrians' motions and it became the foundation of many later studies. Although the functions and some parameters were designed arbitrarily, the results of the test simulations did show that this model can simulate similar crowd behaviours and movements as those presented by social studies.

2. 4. 1. 3 The Social Force Model (Panic Crowd Version) (D. Helbing et al. 2000)

Overview

This model describes a crowd in a panic situation. It mixes the influence of nine socio-psychological features and physical forces with behaviour (Social Forces) by using mathematical formulas. It can be treated as a further development of the original Social Force model (Dirk Helbing & Peter Molnar 1995). In this model, three Social Force effects were considered which were:

- A pedestrian tends to move with a desired velocity which is different from his/her actual velocity.
- A pedestrian wants to keep a velocity-dependent distance from other pedestrians.
- A pedestrian tries to keep a velocity-dependent distance from walls.

Therefore, the formula of the change of a pedestrian's velocity was given by an acceleration equation:

$$m_i \frac{d\mathbf{v}_i}{dt} = m_i \frac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i} + \sum_{j(\neq i)} \mathbf{f}_{ij} + \sum_W \mathbf{f}_{iW}$$ **, while**

$$\mathbf{f}_{ij} = \{A_i \exp[(r_{ij} - d_{ij})/B_i] + kg(r_{ij} - d_{ij})\}\mathbf{n}_{ij} + \kappa g(r_{ij} - d_{ij})\Delta v_{ji}^t \mathbf{t}_{ij}$$ **, and**

$$\mathbf{f}_{iW} = \{A_i \exp[(r_i - d_{iW})/B_i] + kg(r_i - d_{iW})\}\mathbf{n}_{iW} - \kappa g(r_i - d_{iW})(\mathbf{v}_i \cdot \mathbf{t}_{iW})\mathbf{t}_{iW}$$

Formula 14 A pedestrian's velocity change in the Social Force model (D. Helbing et al. 2000)

In the right hand side of the equation, the first part represents the velocity change of a pedestrian's own desire. The latter two parts f_{ij} and f_{iW} are the "interaction forces" from the other pedestrians and the walls respectively (the Σ symbol donates the summation operator).

Details

In particular, the following nine socio-psychological features of a panic crowd were considered in this Social Force model:

- *"People move or try to move considerably faster than normal.*
- *Individuals start pushing and inter-actions among people become physical in nature.*
- *Moving and, in particular, the passing of a bottleneck becomes uncoordinated.*
- *At exits, arching and clogging are observed.*
- *Jams build up.*
- *The physical interactions in the jammed crowd add up and cause dangerous pressures of up to 4,450Nm^{-1} which can bend steel barriers or push down brick walls.*

- *Escape is further slowed by fallen or injured people acting as `obstacles'.*
- *People show a tendency towards mass behaviour, that is, to do what other people do.*
- *Alternative exits are often overlooked or not efficiently used in escape situations."*

----(D. Helbing et al. 2000)

The following section explains the three aspects of the Social Force effects in Formula 14 A pedestrian's velocity change in the Social Force model (D. Helbing et al. 2000) in detail:

➢ Velocity change caused by a pedestrian's own desire

This first aspect of the Social Force effects that influence a pedestrian's behaviour was the desire to change his/her actual velocity to the desired velocity during a characteristic time. The calculation of this effect is given by:

$$m_i \frac{v_i^0(t)\mathbf{e}_i^0(t) - \mathbf{v}_i(t)}{\tau_i}$$

Formula 15 Social Force effect of a pedestrian's own desire to change velocity in the Social Force model (D. Helbing et al. 2000)

In the above formula, *i* refers to pedestrian i. m_i denotes the mass of the pedestrian (and was given the value of 80 *kg* in this model). v_i^0 denotes the desired speed of the pedestrian. e_i^0 denotes the direction of the desired speed. v_i denotes the pedestrian's actual velocity. $\tau_i$is the characteristic time (which was estimated as 0.5 *s* in this model) during which the pedestrian could change his/her actual velocity to the desired velocity.

➢ Repulsive interaction forces between pedestrians

The second aspect of the social effects was the summation of the repulsive interaction forces from other pedestrians which were distance dependent (between two pedestrians). For pedestrian *i* and pedestrian *j*, such a repulsive force normally described the tendency of them staying away from each other. In the case of the two pedestrians touching each other, a "body force" and a "sliding friction" force were introduced to describe the counteraction of body compression and the tangential motion respectively. Whether the two pedestrians are touching each other can be judged by whether the distance between their centres of mass is shorter than the sum of their radii (Note: the shapes of pedestrians are represented by circles in this Social Force model. The pedestrians' diameters were assumed to be uniformly distributed in the range of [0.5, 0.7] metres). As a result, the formula for the repulsive interaction force that pedestrian *i* feels from a pedestrian *j* is:

$$\mathbf{f}_{ij} = \{A_i \exp[(r_{ij} - d_{ij})/B_i] + kg(r_{ij} - d_{ij})\}\mathbf{n}_{ij} + \kappa g(r_{ij} - d_{ij})\Delta v^t_{ji}\mathbf{t}_{ij}$$

Formula 16 Repulsive interaction force from a pedestrian in the Social Force model (D. Helbing et al. 2000)

The explanation for this formula can be broken down into three parts:

- The desire for staying away from pedestrian *j* was given by $A_i exp[(r_{ij} - d_{ij})/B_i]n_{ij}$, where *A* and *B* are constant. (In this model, A was assigned to 2000 *N* and B was assigned to 0.08 *m*.) r_{ij} is the sum of the radii of the two pedestrians. d_{ij} is the distance between the pedestrians' centres of mass. n_{ij} denotes the normalized vector pointing from pedestrian *j* to *i*.
- If two pedestrians were in touch with each other, the "body force" was modelled as $kg(r_{ij} - d_{ij})n_{ij}$, where *k* is a constant (and was given the value of 12000 *kg/s²* in this model). r_{ij} and d_{ij} are the same as they are denoted in the first part of the equation above. *g(x)* is a function and it equals the argument *x* only if the pedestrians have touched each other. Otherwise, it equals zero, i.e. no "body force" exists. n_{ij} denotes the normalized vector pointing from pedestrian *j* to *i*.

  If two pedestrians were in touch with each other, the "sliding friction" was modelled as $\kappa g(r_{ij} - d_{ij})\Delta v^t_{ij}t_{ij}$, where *κ* is a constant (and was given the value of 24000 *kg/m·s* in this model). r_{ij}, d_{ij} and *g(x)* have the same meanings as defined above. Δv^t_{ij} represents the tangential velocity difference and can be calculated via $\Delta v^t_{ij} = (v_j - v_i)t_{ij}$. v_i and v_j denotes the velocity of pedestrian *i* and *j* respectively. t_{ij} is the tangential direction. (If the normalized vector n_{ij} pointing from pedestrian j to i was represented as (n_{ij}^1, n_{ij}^2), the tangential direction is defined as $t_{ij} = (-n_{ij}^2, n_{ij}^1)$).

- **Repulsive interaction forces from the walls**

The repulsive interaction forces from the walls were modelled similarly to the repulsive interaction forces from other pedestrians, which is also the summation of the repulsive forces from the applicable walls and contains three aspects of effects. The formula was given as:

$$\mathbf{f}_{iW} = \{A_i \exp[(r_i - d_{iW})/B_i] + kg(r_i - d_{iW})\}\mathbf{n}_{iW} - \kappa g(r_i - d_{iW})(\mathbf{v}_i \cdot \mathbf{t}_{iW})\mathbf{t}_{iW}$$

Formula 17 Repulsive interaction force from a wall in the Social Force model (D. Helbing et al. 2000)

The explanation of this formula can be also broken down into three parts:

- The effect that described a pedestrian receiving a perpendicular repulsive force from a wall was given by $A_i exp[(r_i - d_{iW})/B_i] n_{iW}$, where *A* and *B* are constant. (In this model, A was assigned to 2000 *N* and B was assigned to 0.08 *m.*) r_i is the radius of the pedestrian. d_{iW} is the pedestrian's distance to wall *W*. n_{iW} denotes the perpendicular direction that points to the pedestrian from the wall.
- If two pedestrians were in touch with each other, the counteracting compression effect was modelled as $kg(r_i - d_{iW}) n_{iW}$, where *k* is a constant (and was given the value of 12000 *kg/s^2* in this model). r_i and d_{iW} are the same as they are denoted in the first part of the equation above. *g(x)* is a function which equals its argument *x* only if the pedestrians have touched each other. Otherwise, it equals zero, i.e. no such counteracting effect exists. n_{iW} denotes the perpendicular direction that points to the pedestrian from the wall.

  If two pedestrians were in touch with each other, the tangential friction effect was modelled as $-\kappa g(r_i - d_{iW})(v_i \cdot t_{iW}) t_{iW}$, where κ is a constant (and was given the value of 24000 *kg/m·s* in this model). r_i, d_{iW} and *g(x)* have the same meanings as defined above. v_i denotes a pedestrian's velocity. t_{iW} is the tangential direction to the direction of n_{iW}.

Simulation Results

The simulations by this model present three phenomena of escape panic: (1) Transition to incoordination due to clogging. (2) The "faster is slower effect" due to impatience. (3) Mass behaviour. For all simulations, the parameters were identical for all pedestrians for "*calibration and robustness, and to exclude irregular outflows of parameter variations*" (D. Helbing et al. 2000). Details of these simulations will be introduced in the "model applications" section.

Discussion

➢ Interacted social psychological issues

This model has configured the original generalised Social Force model (Dirk Helbing & Peter Molnar 1995) in a panic situation. It has specialised the formulas that calculate the effects from other pedestrians and walls to reflect the social psychological issues. The effects of these issues are reflected through the relevant functions and constants in the formula. The model demonstrated that it is possible to represent the effects of social psychological issues in a physical laws' based system. However, although this model has shown that the proposed formulas can represent crowd movements in panic, it did not explain how the social psychological issues were translated into the functions and parameters in the formulas. In other words, this model did not include a mechanism or guideline as to how to represent other social psychological issues if the simulation context was changed.

➢ Data calibration

The Social Force model has suffered from scarcity of data; this fact was pointed out by Helbing et al. (2000) in their study. The parameters were set by empirical observations or kept being modified until the simulation was close to real life, which was mainly dependant on the authors' knowledge and judgements. Johansson et al. (2007) introduced an efficient way to calibrate the Social Force model. They proposed to track the trajectory data from the pedestrians via video and apply it to the Social Force model. In this approach, the pedestrian trajectory data were gathered through tracking the head movement of each pedestrian. The tracking algorithm was stated to be suitable for handling more than 1000 pedestrians simultaneously.

- **Limitations**

This Social Force model has simulated “pushing” behaviour and variable flow. However, it has been pointed out that, in high-density crowds, the pedestrians appear to “shake” or “vibrate” because they are affected by numerous Social Forces which result in unrealistic crowd behaviours and movements (N Pelechano et al. 2007; Nuria Pelechano et al. 2008).

Additionally, the traditional Social Force model treats the individual homogenously, for the reason of ease in the simulation. In the real world, people have different characters and their action abilities can vary. Adding individual characters and roles into the Social Force model can make it more realistic. Some research studies have already taken place in this area but they have required support from other modelling approaches. For example, Braun et al. (2003) added two parameters (Dependency Level and Altruism level) to model individual differences and grouping behaviour *(This model has actually combined the two modelling approaches. This modelling approach will be discussed in section 2. 4. 4 Hybrid Models)*.

2. 4. 1. 4 A Modified Social Force Model (M. Zheng et al. 2002)

Model Overview

In the generalized Social Force model (Dirk Helbing & Peter Molnar 1995), the desired velocity of a pedestrian is determined by the positions of the destination and such desired velocity possesses a global value for all pedestrians. Zheng et al. (2002) considered the model should reflect the differences within individual personalities to obtain a more realistic desired velocity and presented a model to consider the influences of pedestrians’ personalities on desired velocities.

Two kinds of personalities (patient and impatient) were taken into account when looking at individual behaviours. The patience of a pedestrian determines his/her behaviour when he/she is behind another pedestrian whose velocity is less than his/her desired velocity. A pedestrian with an impatient personality is likely to overtake the person in front while a patient pedestrian will probably slow down and follow (*Note: there are no quantitative definitions as to how much less velocity would trigger these behaviours in Zheng et al.'s presented research study*). The behaviours of these pedestrians are illustrated below:

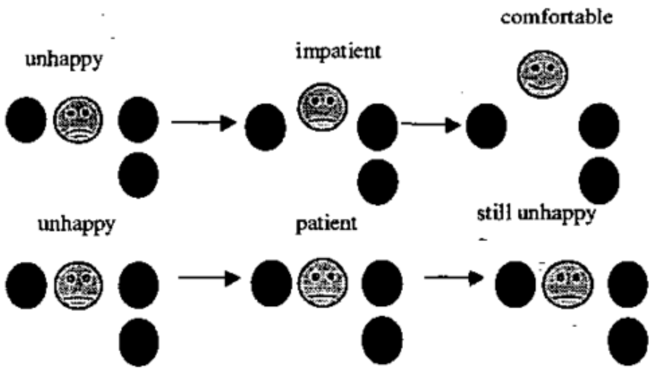


Figure 2 Behaviours of pedestrians (impatient and patient) (M. Zheng et al. 2002)

Additionally, a concept of pedestrians' amenity was introduced and defined as:

$$amt = \frac{1}{N}\sum_i\left[\frac{1}{T_i}\sum_t\left(1 - \frac{|v_i^0\vec{e}_i(t) - \vec{v}_i(t)|}{v_i^0}\right)\right],$$

Formula 18 amenity of a pedestrian group (M. Zheng et al. 2002)

In this formula, *N* is the number of the pedestrians and T_i donates the time for passing from the entrance to the exit. v_i^0 dentoes the desired speed of the pedestrian. $\vec{e}_i$ denotes the direction of the desired velocity. $\vec{v}_i$ denotes the actual velocity of the pedestrian.

This model was tested in a scenario of pedestrians walking bi-directionally on a straight road, shown in Figure 3. Empty circles represent the pedestrians moving towards the right while the solid ones represent those moving towards the left.

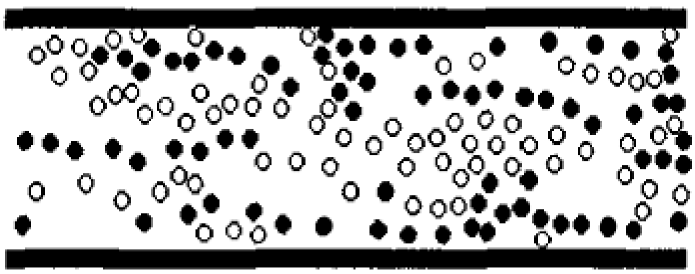


Figure 3 A snapshot of the simulation of Zheng et al.'s (2002) modified Social Force model

The simulation results of this model found out that, in high-density cases (the amount of pedestrians > 90) when the P:I (Patient pedestrians: Impatient pedestrians) ratio was 1:1, the group amenity had the highest values (the simulations were tested at three P:I ratios: 4:1, 1:1, and 1:4) whereas there were no differences in low-density cases (the amount of pedestrians <= 90). The results also showed that the average passage time become the shortest in high-density cases when the P:I ratio was 1:1 while it was independent of the P:I ratio in low-density cases.

Discussions

This model demonstrated that individual personalities (patient and impatient) could be interpreted within the generalized Social Force model (Dirk Helbing & Peter Molnar 1995) to achieve more realistic desired velocities of pedestrians in different circumstances. However, the conclusions drawn on the differences between pedestrians' P:I ratios and crowd densities seem unreliable and insufficient. They can be challenged on two points:

- Firstly, the total number of conducted simulations was not stated. One cannot tell whether these statistics were the results of only one simulation or were the average of a reasonable amount of simulations.
- Secondly, to compare the differences between groups, statistical test methods should be used. For example, a t-test (comparing a group of two) or an ANAOVA test (comparing a group of three or more).

This model was reviewed in order to show the attempt to interpret different individual behaviours by adapting the Social Force model.

2. 4. 1. 5 A Physically-Based Particle Model of Emergent Crowd Behaviours (Heïgeas et al. 2003)

Overview

This model simulated crowd collective behaviour by using particles to represent individuals. The interaction force between two particles was defined by a piecewise linear function based on the distances between the two particles:

$$\text{If } \mathbf{D} < \mathbf{D_1}, \qquad \text{then } \mathbf{F} = (\boldsymbol{K_1 D} + \boldsymbol{Z_1 V})\vec{\mathbf{u}}$$

$$\textbf{If } D_1 < D < D_2, \quad \textbf{then } F = (K_2 D + Z_2 V)\vec{u}$$
$$\textbf{If } D_2 < D < D_3, \quad \textbf{then } F = (K_3 D + Z_3 V)\vec{u}$$
$$\textbf{If } D_3 < D, \quad \textbf{then } F = 0$$

Formula 19 A piecewise linear function for the interaction force between two particles (Heïgeas et al. 2003)

In the formula, *D* is the distance between the two particles and *V* denotes the norm of their relative velocity. $\vec{u}$ denotes the unit vector between the two. D_1, D_2, D_3 are the thresholds for the piecewise function. K_1, K_2, K_3 denote the stiffnesses and Z_1, Z_2, Z_3 are the viscosities.

Details

Based on the piecewise function, the distance between the two particles has been divided into four zones. They are (as shown in Figure 4):

- Zone A ($D < D_1$): the anticipation zone with a low stiffness.
- Zone B ($D_1 < D < D_2$): the avoidance zone with a medium stiffness.
- Zone C ($D_2 < D < D_3$): the impenetrable zone with a high stiffness

The zone where $D_3 < D$ has not been defined explicitly. The interaction force within this zone was defined as zero.

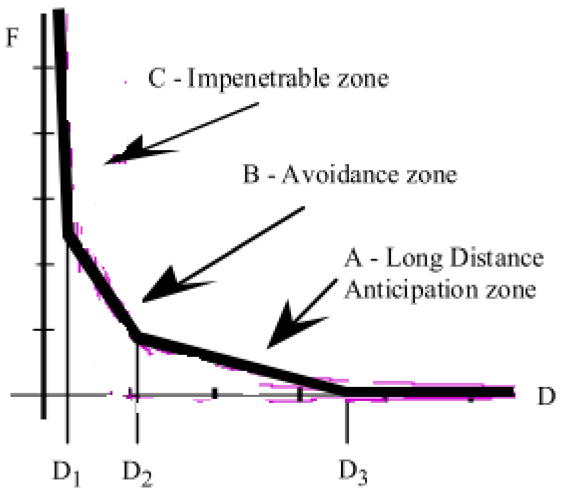


Figure 4 The piecewise linear function approximation (Heïgeas et al. 2003)

From the model description, it can be seen that the interaction force decreases alongside the increase of distance. In this research study, the suggested Zone thresholds were given as: $D_1 = 1$ metre, $D_2 = 3$ metres, and $D_3 = 5$ metres. However, no specific values were given to stiffnesses and viscosities (The study only mentioned that these values should be chosen from a range of pre-defined values).

In addition, this model proposed using sets of fixed particles to represent buildings (e.g. walls, obstacles) instead of having geometrical representations of buildings.

Discussions

This model presented a physically-based particle system (particles can either represent individuals or buildings) for crowd simulation. A piecewise linear function was proposed to measure the interacting forces between two particles. This function is a decreasing function which is comparable to the repulsive effect in Social Force model (Dirk Helbing & Peter Molnar 1995).

However, this model only presented the interacting forces between individuals (or individuals and buildings). There is a lack of description as to how such interacting forces could blend into individuals' movements. In a real-life scenario, the model should consider more forces rather than only the interacting force; for example, the individuals' own desires or possible attraction effects from others. This model cannot represent those complex behaviours because it was only designed to simulate one type of crowd behaviours.

Additionally, the lack of definitions of, and explanations on, the constants (stiffnesses and viscosities) in the piecewise function leaves a lot of uncertainty in the representations of crowd behaviour in this model.

2. 4. 1. 6 Modified Particle Swarm Optimization-based Model (Cheng et al. 2008)

Foundation

A particle system is a collection of a large number of individual particles, each having its own behaviours. A particle swarm optimization model (Kennedy & Eberhart 1995) was proposed to simulate the actions of flocks of birds and schools of fish. The model was based on the algorithm that particles swarming in a search space will move to the best positions according to their knowledge. This system only involves primitive mathematical operations and it is computationally inexpensive in terms of both memory requirements and speed.

Overview

As the algorithm of seeking a best position by birds is considered similar to the behaviour of finding the exit in an emergency evacuation of a crowd, the above particle swarm model (Kennedy & Eberhart 1995) was adapted to simulate crowd emergency evacuation and became the Modified Particle Swarm Optimization-based Model (MPSO model). The velocity of an individual (defined as a particle) was defined as:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1(t)(p_i(t) - x_i(t)) + c_2 r_2(t)(p_g(t) - x_i(t))$$

Formula 20 A particle's velocity calculation in MPSO model (Cheng et al. 2008)

The next position of the particle could be calculated via:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Formula 21 A particle's updated position in the MPSO model (Cheng et al. 2008)

Details

In Formula 20, ω is the inertia weight which can be adjusted in the direction of linear decrease. v_i is the velocity of particle *i*. p_i is the best position for particle *i*. x_i is the position of particle *i*. p_g denotes the *global best* position. c_1, c_2 are constants and r_1, r_2 are stochastic factors (the value of *c* multiple *r* was defined to have a mean of 1).

More specifically, ω helps the particle to maintain its inertia: "*With large ω, the algorithm provides preferable global convergence, while with small ω, the algorithm provides preferable local convergence*" (Cheng et al. 2008). ω was introduced to solve the pre-mature convergence phenomenon when the original particle swarm optimization model (Kennedy & Eberhart 1995) was used to simulate a crowd. In this MPSO model, ω was defined as (where MaxNumber is the maximal iterative time):

$$\omega(t) = 0.9 - \frac{t}{MaxNumber} \times 0.5$$

Formula 22 Implementation of ω in the Modified Particle Swarm Optimization Model (Cheng et al. 2008)

Simulation results

In this MPSO model, the cell (that can be occupied by one individual) size was defined as 0.4*m* × 0.4*m*. A scenario of 200 occupants evacuating from a single exit room (dimension: 18*m* × 22*m*) was simulated. The results have been compared to the Social Force model and a Cellular Automata model (data were from Weiguo et al.'s (Weiguo Song et al. 2006) work) and were presented as below:

Table 1 Simulation results of the MPSO model and comparisons with the Social Force model (Cheng et al. 2008)

The width of exit	Evacuation Time (s)		
	Social Force Model	Cellular Automata Model	MPSO Model
0.8	241.0	131.8	200.2
1.2	85.7	74.2	81.9
1.6	45.3	49.4	47.5
2.0	31.2	39.2	35.7
2.4	24.2	30.0	28.1
2.8	19.4	29.7	25.5
3.2	17.1	27.5	23.1
3.6	15.0	27.0	22.0
4.0	14.6	26.4	21.6

The MPSO model has also been implemented into a simulation system for the emergency evacuation of a two-floor building. Congestions in the corridor and stairs were demonstrated (see Figure 5).

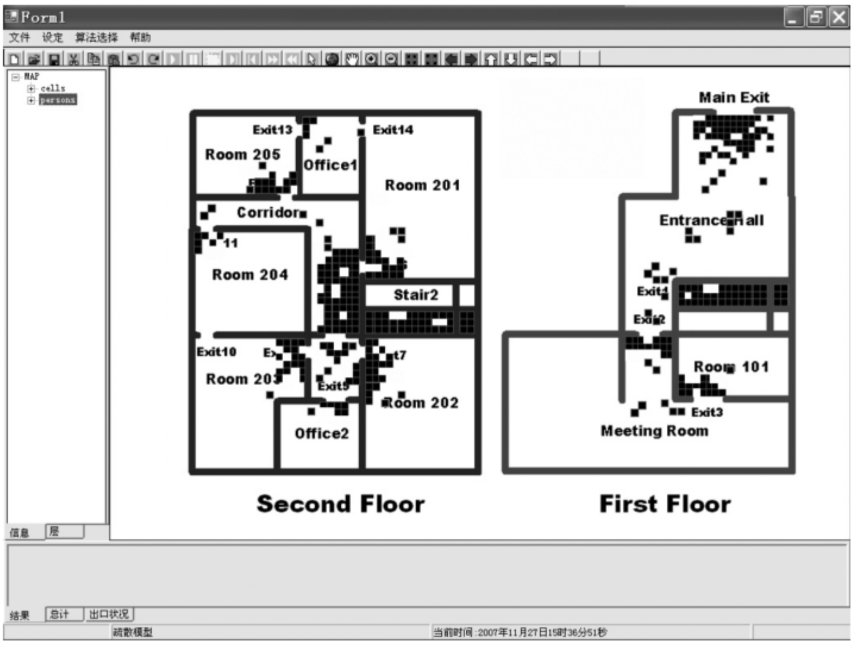


Figure 5 Congestions during an evacuation in the MPSO model (Cheng et al. 2008)

Discussions

This MPSO model presented a solution to simulate crowd emergency evacuation by using a modified formula from the particle swarm optimization model (Kennedy & Eberhart 1995). The simulations showed that it was an effective way to simulate crowd evacuations and the presented results were comparable to those of the existing study (Weiguo Song et al. 2006).

However, the original particle swarm optimization model was categorised as a force-based model, whereas the movements of the particles were decided by their own status and by the neighbour particles. The positions of the particles were defined in continuous representations. In the study of the MPSO model, the implementation of the model chose to use fixed size cells to represent the positions of particles, which was also the approach of CA modelling. This implementation introduced a fundamental inconsistency that was not clearly explained in the study, which was that CA modelling is not compatible with continuous position representation. I.e. when using Formula 21 to calculate the next position of a particle, it cannot guarantee that the next position is exactly aligned with the cells which were designed to be occupied by individuals. For example, assuming an environment which has only two cells, if the particle's initial position was in Figure 6 (a), its next position should either remain as Figure 6 (a) or become Figure 6 (b). It cannot become Figure 6 (c), but the result of $v_i(t+1)$ may only change the particle's position by half a cell.

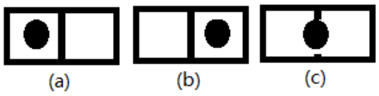


Figure 6 Possible and impossible an individual's positions in a two cells' CA model

Thus it is required to have some rules to decide the particle's next position in the above situation. Such rules were not presented in the MPSO model. When the relevant rules have been provided, it will make this MPSO model become a CA model and it should no longer be called a particle swarm optimization model.

2. 4. 1. 7 Features of the Force-based Modelling Approach

In the above sections, the discussions were focused on each model individually. This section will discuss the overall benefits and limitations of using the force-based modelling approach to design a crowd model.

The Features of Force-based Models

- **Continuous position representation of individuals**

Because the motions of individuals are determined by the results of the calculations from the mathematical formulas defined in the force-based models, the positions of the individuals are usually represented by the Cartesian coordinate system. Such position representation provides continuous and precise information on the positions of the individual. In addition, the precise position information can help to improve the accuracies of the interactions between the individuals in the crowd.

- **Force-based models can take advantage of physical laws**

The motions of individuals in the force-based models follow the mechanisms of kinematics which provide these models with a reliable physical foundation. Furthermore, the Social Force model and its derivatives have made use of real force (i.e. the force in physics) and the motions of individuals are based on Newton's laws. In reality, the crowd are moves in a world that obeys Classical Mechanics and physical laws. In crowd modelling, force-based models represent individuals' motions by adapting physical laws, which provides a similar approach to simulate the crowd movement.

Limitations of Force-based Models

- **Homogeneous behaviour of the crowd**

The formulas defined in the force-based models are universally applied to every individual to ensure consistent behavioural results from all individuals. This approach is a simplification of the complex crowd behaviour in real-life. However, recent studies (Nuria Pelechano & Norman I Badler 2006; Braun et al. 2003; Shendarkar et al. 2008) have shown that individual differences can change a crowd's overall behaviour and studies of crowd modelling has come to focus on creating an intelligent crowd that can reflect the decision making process of human beings. As a matter of fact, the force-base modelling approach by itself cannot achieve such a requirement *(Many studies have combined force-based modelling with other approaches which will be reviewed in section 2. 4. 4 Hybrid Models)*.

- **Force based models can be computationally expensive**

The formulas in force-based models usually consider all the interactions between the individuals and such an approach requires a time complexity of $O(n^2)$ to run the simulation. This has put great pressure on a computer's computational power in the real time simulations undertaken in the early (1990s) studies of crowd modelling. During the rapid development of computer technology over the past two decades, such a requirement has become a minor issue in most studies (except the models which involve thousands of people). However, theoretically, the force-based modelling approach has a larger time complexity than other crowd modelling approaches (e.g. the CA models introduced in the next section typically require a time complexity of $O(n)$).

2. 4. 2 Cellular Automata (CA) based Models

Cellular automata (CA) based models refer to those models that utilise cellular automata as the foundation to describe the movements of a crowd. A Cellular Automata model is a mathematical model that contains a set of cells which change their states based on a set of pre-defined rules. It was adopted to represent individual movements in a crowd and such models are known as Cellular Automata (CA) based models.

(Note: the Cellular Automata based crowd models are usually called CA models in the study of crowd modelling and simulation although they refer to the models that adopt the mathematical CA modelling method. In this thesis, the term "CA models" refers to Cellular Automata based crowd models unless the mathematical CA model is indicated explicitly.)

2. 4. 2. 1 History of Cellular Automata

Von Neumann's Cellular Automata (Von Neumann & Burks 1966)

The term "Cellular Automata" (CA) was first introduced in Von Neumann's Universal Constructor in 1940s (Wolfram 2002). The Universal Constructor was a self-replicating machine which transited the states of cells within it synchronously. The cells were orthogonally located in a two-dimensional Cartesian system. Each cell was a finite state automation that contained 29 states. The cell state would be updated depending upon the states of the four adjacent (up, down, left, and right) cells. All the cells were identical with the same state-update rules applied. The purpose of this model was to design a finite state machine which could build copies of itself, i.e. the same pattern of cells' states could be reproduced when the cells keep updating their states by following the predefined rules.

Game of Life by Conway (Gardner 1970)

The Game of Life was a cellular automaton devised by a British mathematician John Horton Conway in 1970. It was also a model with square cells located in a two-dimensional Cartesian system but it had much more simplified cell states and update algorithm than Von Neumann's Cellular Automata. In the Game of Life, the state of a cell could only be either dead or alive. The cell state update algorithm contained four rules (the adjacent cells counted the eight surrounding cells, thus horizontally, vertically, and diagonally):

Any live cell with fewer than two live adjacent cells becomes dead.

Any live cell with two or three live adjacent cells remains alive.

Any live cell with more than three live adjacent cells becomes dead.

Only the dead cells with exactly three live adjacent cells become alive.

Within a certain initial pattern, the cells can replicate their states. Many types of replicator patterns have been found in the Game of Life, for example:

- Still lifes (Figure 7): a pattern that does not change during the update.
- Oscillators (Figure 8): a pattern that repeats the states of cell with a certain period.
- Spaceships: a pattern like Oscillators but keep moving one direction during the update.

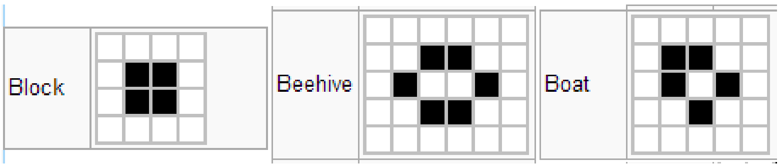


Figure 7 Examples of "still lifes" patterns in the Game of Life. Images source: (Wikipedia 2013).

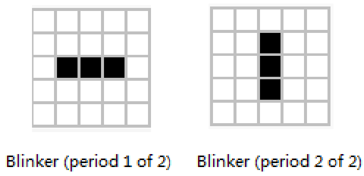


Figure 8 An example of the "oscillators" pattern in the Game of Life. Images source: (Wikipedia 2013).

A more detailed introduction to game of life patterns can be found Hogg's Illustrated Guide (2009) which contains 306 patterns in 73 types.

Cellular Automata in Detail

Comprehensive reviews and discussions on the Cellular Automata model as a mathematical model can be found in Wolfram's studies (Wolfram 1983; Wolfram 1986; Wolfram 2002).

2. 4. 2. 2 Cellular Automata in Crowd Modelling

Introduction

A crowd can be treated as a dynamic system in which individuals change their positions by following certain movement/behaviour rules. Based on this premise, the Cellular Automata (CA) modelling approach can be a very suitable method to model individuals' movements in the crowd because individuals can be modelled to move in cells based on a set of rules.

More specifically, in a Cellular Automata based crowd model, the cells present the positions that an individual occupies. The cell states can either be empty or occupied. When an individual leaves a cell its state become empty and when an individual enters a cell its state becomes occupied. A cell can only be occupied by one individual at one time and an individual can only move into another adjacent cell while it is empty. In this case, the rules of how the individuals move are translated into rules that decide how the cells update their states. In the mathematical CA model, the boundary of the cell system is defined as infinite (i.e. has infinite cells). The amount of cells in a Cellular Automata based crowd model is constrained by the environment (e.g. for a given cell size, the size of the room determines how many cells it has). This means that in a Cellular Automata based crowd model not all cells have eight adjacent cells.

Typical Studies of CA models

Unlike the force-base models which may have different mechanisms to interpret the forces or their effects, CA models, in contrast, share the same mathematical foundation - the mathematical CA model. The only difference between CA models is how they define the rules for updating the states of the cells, i.e. the rules that determine the individuals' movements. There are many studies of crowd modelling based on Cellular Automata (Blue & Adler 2001; Burstedde et al. 2001; Perez et al. 2002; Z. Lin et al. 2005; D. Zhao et al. 2006; Georgoudas et al. 2006; Varas et al. 2007; YF Yu & WG Song 2007; Schultz et al. 2007; W. Yuan & Tan 2007; W. Fang et al. 2003). In the reviews of CA models in this section, the studies of CA models in two typical scenarios have been presented: emergency evacuation (Kirchner & Schadschneider 2002) and bi-directional walking flow (Yue et al. 2010).

➢ A bionic-inspired CA model for pedestrian dynamics (Kirchner & Schadschneider 2002)

✧ ***Overview***

This model described the interaction among pedestrians and simulated an evacuation from a large room with one or two doors. In this CA model, a pedestrian is considered to move to one of its unoccupied neighbour cells (the four adjacent cells) with certain transition probabilities (see Figure 9). The position update of all the pedestrians happened at the same time, i.e. this model uses parallel update.

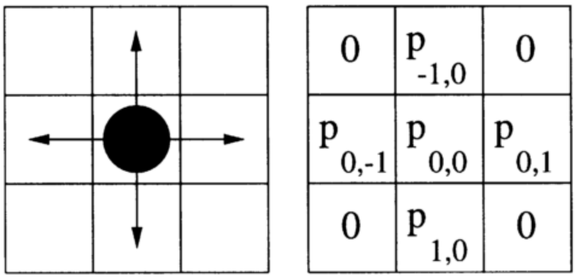


Figure 9 a pedestrian's possible movements and their probabilities in the bionic-inspired CA model (Kirchner & Schadschneider 2002).

Each cell contains two types of information. The first one is named *the static floor field (S)*. It reflects the property of the cell as a position which does not change over time or by the presence of the pedestrians. It is used to specify which cell is more attractive in terms of an evacuation process. The second one is the *dynamic floor fields (D)*, which represents the virtual trace left by the pedestrians. Such a trace is time dependent and has its own diffusion and decay.

The rules to update the cell state (pedestrian movement) are listed below:

- The *dynamic floor field (D)* can decay with a probability of $\alpha \in [0,1]$ and can diffuse the pedestrian's trace to its neighbour cells with a probability of $\delta \in [0,1]$.
- The probability p_{ij} of a pedestrian moving to an unoccupied neighbour cell *(i, j)* is calculated by the following formula (K_D and K_S are the weightings of the two fields):

$$p_{ij} = N \exp(k_D D_{ij}) \exp(k_S S_{ij})(1 - n_{ij})\xi_{ij} ,$$

with

$$\text{occupation number}: \; n_{ij} = 0, 1 ,$$

$$\text{obstacle number}: \; \xi_{ij} = \begin{cases} 0 & \text{for forbidden cells, e.g. walls ,} \\ 1 & \text{else ,} \end{cases}$$

$$\text{normalisation}: \; N = \left[\sum_{(i,j)} \exp(k_D D_{ij}) \exp(k_S S_{ij})(1 - n_{ij})\xi_{ij} \right]^{-1}$$

Formula 23 Calculation of the probability of the pedestrian moving to an unoccupied cell in the bionic-inspired CA model (Kirchner & Schadschneider 2002)

- Each pedestrian chooses its target cell based on the p_{ij} in the previous update step.
- Only one pedestrian is allowed to move into one cell and any conflicts are resolved by a probabilistic method.
- The value of *D* is increased by all moving pedestrians.

✧ ***Discussion***

This is a typical demonstration of a CA model, with some modifications. It simplifies the effects of neighbour cells as it only considers four adjacent cells instead of eight. The concept of taking into account the trace of a pedestrian does introduce a certain complexity into the update rules.

➢ **A CA model for bi-direction pedestrian flow (Yue et al. 2010)**

✧ ***Overview***

This CA model was designed to simulate the bi-directional walking flow of pedestrians. In this model, a pedestrian decides his/her movement in a 3×3 matrix according to the corresponding *transition payoff* P_{ij} (see Figure 10).

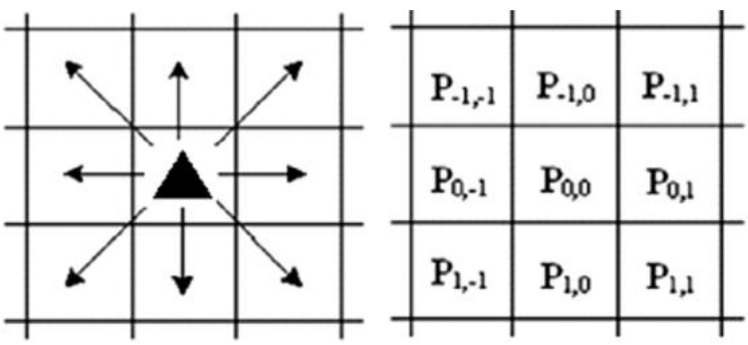


Figure 10 A pedestrian's movements and the associated transition payoffs matrix (Yue et al. 2010)

The *transition payoff* P_{ij} was determined by the four parameters (Direction-parameter D_{ij}, Empty-parameter E_{ij}, Forward-parameter F_{ij} and Category-parameter C_{ij}) of each cell. The formula was given by the following (for the detailed calculation of each parameter, please refer to Yue et al.'s paper (2010)):

$$P_{ij} = D_{ij} + E_{ij} + F_{ij} + C_{ij}$$

Formula 24 Formula to calculate the transition payoff in Yue et al.'s CA model (2010)

In addition, by taking into account the habit of walking on the right-hand side of the road, a Right-hand parameter R_{ij} was introduced and Formula 25 is introduced to calculate *transition payoff* P_{ij}. For the cells on the right-hand side, R_{ij} provides a positive value. For the cells on the left-hand side, R_{ij} provides a negative value. For the cells in the middle, R_{ij} returns zero.

$$P_{ij} = D_{ij} + E_{ij} + F_{ij} + C_{ij} + R_{ij}$$

Formula 25 Formula to calculate the transition payoff with a right-hand walking preference in Yue et al.'s CA model (2010)

In this model, the pedestrian can only move one cell in one update step. The update rules of the pedestrians' movements were given as follows:

- The pedestrian moves to the cell which has the highest *transition payoff* P_{ij} (chosen from the nine values in the matrix). In the case of multiple highest P_{ij} values, the target position is chosen randomly with equal probability.
- In the case of conflict where more than one pedestrian attempts to move into the same cell, one pedestrian will be randomly chosen with equal probability. All the un-chosen pedestrians stay at the original position.
- If two pedestrians choose each other's cell as the target position, they switch their positions.

✧ ***Discussion***

The model described pedestrians' movements in a bi-directional walking scenario with several simple mathematical formulas by using the CA model. The comparisons with empirical data and experimental observations showed that this model had similar results in simulations at various crowd densities. It is possible to fine-tune the values of the parameters in the formula to achieve more accurate simulation results.

However, it is very difficult to link these parameters directly to other human behaviours (other than walking in a bi-directional scenario). In other words, the design of this model reflects its original purpose but lacks the possibility of further expansion.

- **Lattice gas models**

As reviewed in Zheng et al.'s (2009) studies, lattice gas models were considered as "*a special case of cellular automata, and were popularized in the 1980s by Fredkin and Toffoli and by Wolfram*". The lattice gas model has been applied in modelling crowds by many studies (Y Tajima & T Nagatani 2002; Nagai et al. 2005; D Helbing et al. 2003).

In this PhD thesis, all the crowd models utilising the concept of Cellular Automata (or similar, e.g. the lattice gas model) are categorised as "Cellular Automata models".

2. 4. 2. 3 Features of the Cellular Automata Modelling Method

CA is an artificial intelligence approach to simulation modelling defined as mathematical idealizations of physical systems in which space and time are discrete and physical quantities take up a finite set of discrete values.

CA Model Features and Benefits

- **Simplified crowd movements and field representation**

The main feature of a CA model is that it divides the fields (e.g. rooms, corridors, streets, etc.) into equal size cells and represents the movement of a crowd upon those cells. Figure 11 demonstrates a typical crowd simulation by a CA model. The room is represented by cells (with the grey ones indicating the walls). A circle in a cell indicates that the cell is occupied by a pedestrian. Because the pedestrians' positions are designed to be within the cells, their movement rules only need to consider how they travel from one cell to an adjacent cell.

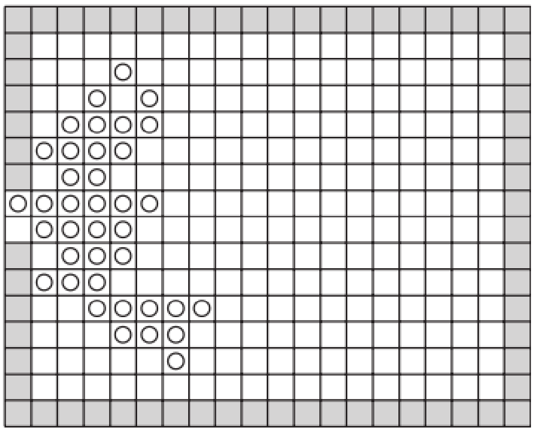


Figure 11 A snapshot of the simulation of a CA model (Figure from the study of Varas et al. (2007))

➢ **Mapping between crowd behaviour rules and cell state update rules**

In CA models, the movement of a crowd is described through how the cells update their state. The behaviour rules of how an individual chooses his/her route under certain conditions are transferred to the update rules of how a cell changes its state based on the states of its neighbouring cells. Compared to the force-base models (which usually involve physical concepts and equations and aim to reflect behaviours through formulas), CA modelling provides a direct mapping between behaviour rules and cell state update rules.

CA Model Limitations

➢ **Discrete in time and space**

Because the movements of individuals are within cells in the CA models, the period to update the cell state is decided by the size of the cell and the average speed of the crowd. For example, if the cell size of a CA model is $0.4\mathrm{m} \times 0.4\mathrm{m}$ (in order to reflect the space of one person occupying it) and the average speed of the crowd is 1 $\mathrm{m/s}$, that CA model needs to be updated every 0.4 second to ensure that the individuals can move exactly into another cell during one update. This approach introduces some limitations:

- **Loss of some detail**: It cannot provide details within that 0.4 second because the CA model can only simulate crowd behaviour within a fixed update period. If there is a requirement to simulate this scenario with a 0.2 second update period, CA modelling is incapable of achieving this.

- **Homogeneous speed of the crowd**: The individuals in the crowd cannot have different speeds, because in a CA model the individual can only move one cell per update period.

➢ **Fixed cell size**

In CA models, every cell has a fixed size and can only be occupied by one individual at one time. The size of cell is determined by the space a person would occupy. This approach also has some limitations:

- **Fixed maximal crowd density (when every cell is occupied)**: For a given CA model (e.g. the cell size is $0.5\mathrm{m} \times 0.5\mathrm{m}$) the crowd density cannot exceed 4 $\mathrm{person/m^2}$ which means a CA model is not applicable to represent situations where the space that one person can occupy is changing. For example, a report (Vassalos 2004) pointed out that a crowd can still move when the density increases to 7.4 $\mathrm{person/m^2}$ (meaning that the cell size becomes 0.135 $\mathrm{m^2}$) in some situations.
- **The cells may not totally align geometrically with the fields:** Because the cell size is determined by the space that one person would occupy, it cannot be guaranteed that the field can be exactly covered by the cells. This issue becomes crucial at doors or exits. For example, if the cell size is $0.4\mathrm{m} \times 0.4\mathrm{m}$ and the door width of a room is 1 m (which requires 2.5 cells), the CA model needs to choose two cells or three cells to represent the door. However, no matter what the choice is, it will not provide a precise representation of the door and the simulation results (e.g. the flow rate through that door) could be doubtful.
- **Homogeneous individual body size:** The CA model assumes that everybody occupies the same space and, thus, it is incapable of representing some typical cases. For example, a fire-fighter who is carrying equipment takes up more space which has an impact on the crowd movement. A study (Averill et al. 2005) showed that, in the rescues undertaken after the World Trade Centre Attack (known as “911”), the fire-fighters carrying large equipment dramatically slowed down the flow of people moving downstairs (the actual speed was half that normally undertaken in fire drills).

➢ **Lack of individuals' characters**

In the CA models, the existence of individuals is indicated through the cell states and the individual movement is determined by the states of the neighbouring cells. The CA model only represents the positions of individuals and ignores their characters. In the real world, one individual may act differently from another under the same situation and studies (M. Zheng et al. 2002; Braun et al. 2003; Nuria Pelechano & Norman I Badler 2006) have suggested taking this into account. Unfortunately, this is a requirement that cannot be achieved by CA modelling as the update rules in a CA model are bound to static cells rather than moving individuals (actually no individual has been defined in the CA model).

2. 4. 3 Agent-based Models (ABM)

There is no formal definition of an agent-based model in crowd modelling. Usually, an agent-based crowd model refers to a model which utilises agents (the modelling method of agents) to represent individuals in a crowd.

(Note: Agent-based models can exist in subjects other than crowd modelling. In this thesis, agent-based models refer to studies on crowd modelling and simulation.)

2. 4. 3. 1 What is an Agent?

Although many studies on crowd modelling (Stefania Bandini et al. 2009; Heliövaara et al. 2012; Stefania Bandini et al. 2007; Bonabeau 2002; H. Zhang & Huang 2006; Bai et al. 2008; Luo et al. 2008; Macal & North 2007; N Pelechano et al. 2007; Kruszewski 2005; S.R. Musse & D Thalmann 1997; Rossmann et al. 2009) have stated that they have used agent-based models, there is no universal definition of "agent" in these studies. However, from these studies of crowd modelling, it is possible to summarise the features that an agent may have:

- Autonomous: This indicates an agent behaves on its own behalf and this is the most important feature of an agent. An agent is an independent entity and can make its own decisions. An agent is self-contained and is able to function on its own.

- Interactive: An agent has the ability to perceive its surroundings and communicate with other agents. It can interchange information with others or influence others' behaviours.
- Intelligent: An agent is able to make decisions in different situations. Its behaviour would take the surroundings, other agents, and its own status into account.
- Individualised: As an agent represents an independent individual, it can contain a set of attributes to distinguish it from others. Such diversities in characters and abilities will result in differences in decision making and behaviours.
- An agent may have the ability to learn and adapt its behaviours based on its experiences. Individual learning and adaptation requires an agent to have memory, usually in the form of a dynamic agent attribute.

2. 4. 3. 2 Multi-Agent System (MAS)

A multi-agent system refers to a crowd simulation system that employs an agent-based model to represent individuals. Furthermore, studies of the multi-agent system (J. Dijkstra et al. 2000; Zoumpoulaki et al. 2010; Davidsson 2001; Xiaoshan Pan et al. 2007; Sud et al. 2008; Tang & X. Zhang 2008; HELBING et al. 2005; X Pan et al. 2006) usually include an implementation of the simulation environment. However, models of the agents were still treated as the core part in these studies but were wrapped in a simulation environment for simulation purposes. In fact, most of the studies with the keyword "agent-based model" have included the implementation of a simulation environment as well. In other words, the terms "agent-based model" and "multi-agent system" are usually interchangeable in the study of crowd modelling. If a study states itself to be using an "agent-based model", it could emphasis the method of agent modelling which has been adopted in the study. If a study claims to be using a "multi-agent system" it may want to indicate that the crowd in the study are modelled by many independent agents. Both terms sound having slightly different focuses but refer to the same thing - agent.

In this thesis, the term "agent-based model" will be used to refer to the models or the simulation systems that adopt the agent modelling method.

2. 4. 3. 3 The Design of the Agents

In agent-based models, the agent model usually consists of two parts: the part which reflects how the agent interacts with the virtual world and makes decisions based on its perceptions; and the part that describes the agent's own character and abilities. Although the methods that are used to create the agent and the rules that the agents use to make their decisions usually differ from study to study, there is one modelling method that has been adopted in many studies which is known as the Belief-Desire-Intention (BDI) agent.

BDI (Belief-Desire-Intention) Agent

➢ **Introduction**

The Belief-Desire-Intention (BDI) agent was a software modelling method that divided a system into many independent modules (also known as agents) that can function on their own and can interact with other modules to achieve tasks at the system level. The idea come from Bratman's theory of human practical reasoning (Bratman 1987). A well known general architecture of the BDI model (although more often known as Procedural Reasoning Systems) was presented by Georgeff and Ingrand (1989). Basically, the architecture of an BDI agent describes the process of how the agent makes decisions; this has been discussed in many studies (Singh 1998; Z. Lin et al. 2005; T. I. Zhang et al. 2003). Such architecture consists of three sections:

- **Belief**: Belief is the information which is possessed by the agent. It includes the agent's own states and the perceptions from the system (i.e. the virtual world in a crowd simulation).
- **Desire**: Desire presents the agent's goal or motivation. It is the target that the agent wants to achieve.
- **Intention**: Intention indicates what the agent will do next. It is the result of the rational thinking by the agent by analysing the information from the Belief and Desire section.

The BDI agent has been used in many studies (Cho et al. 2008; Shendarkar et al. 2008; Stefania Bandini et al. 2009; Moradi et al. 2008; Zoumpoulaki et al. 2010; Luo et al. 2008; Qiu & Hu 2010; McKenzie et al. 2006). For a more detailed demonstration, the following section presents a typical study (Zoumpoulaki et al. 2010) on the simulation of an emergency evacuation by using the BDI agent.

- **An example of a BDI agent model (Zoumpoulaki et al. 2010)**

This study proposed a BDI agent model to take into account individual personalities and emotions during an emergency evacuation. The architecture of the BDI agent is showed in Figure 12.

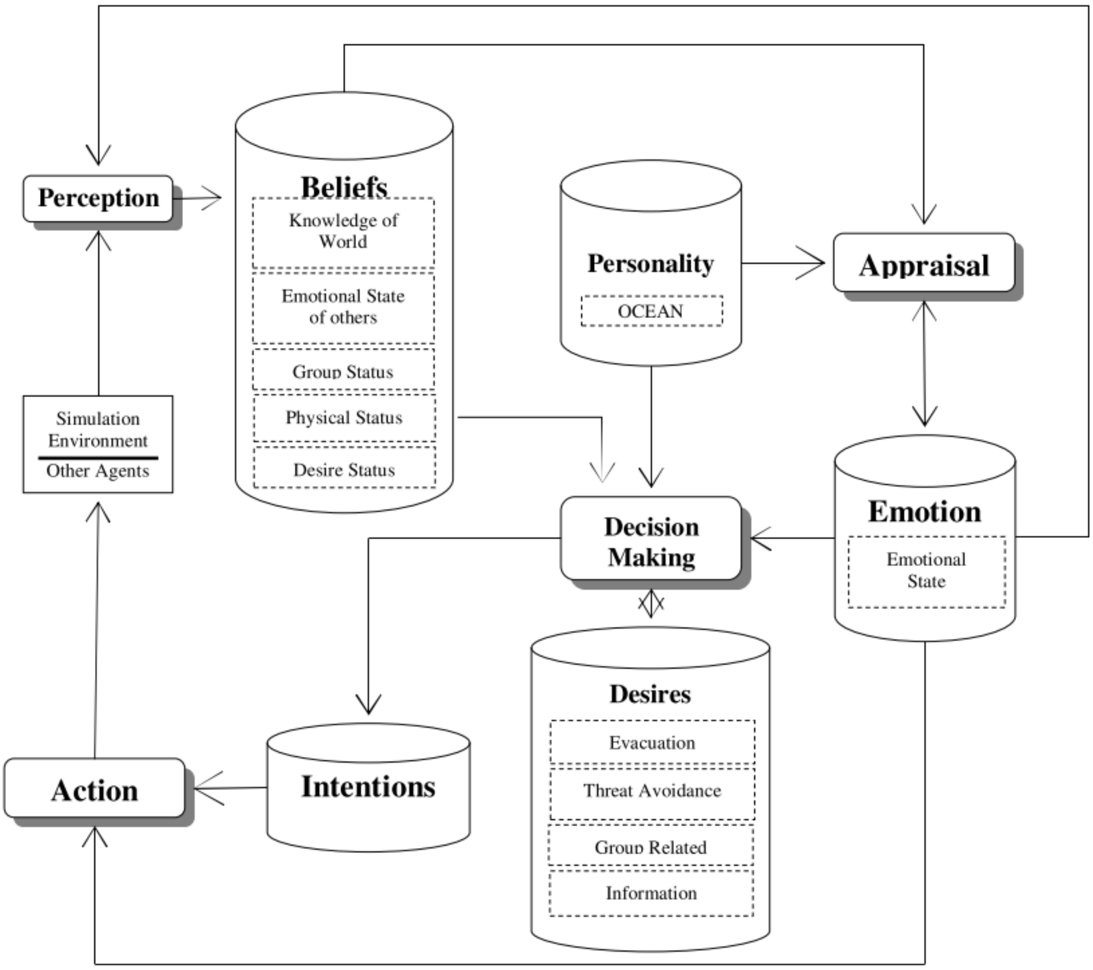


Figure 12 The BDI agent architecture in Zoumpoulaki et al.'s study (2010)

In the simulation, the agent behaves by following the process defined in the above BDI agent model. This operation circle begins with the *Perception* phase where the agent perceives the information from the virtual world. The agent's emotional state at that time would influence the result of the perception. The perceived information consists of the agent's *Belief*. This *Belief* is firstly used in the appraisal process to update the agent's emotional state. During this process, the agent's personality could affect the changes of emotions. Then the agent's *Desire* will be generated during the decision making process, whereby the beliefs, the personality, and the emotions are all taken into account. As a result, the agent's *Intention* is decided. Then this intention is converted into the agent's actions in order to update the status of the simulation environment.

- **Discussions**

The BDI agent architecture was considered as a paradigm to reflect folk psychology in the simulation of humans making decisions and which maps the plain language that describes people's reasoning and actions to how agents work (H. Zhang & Huang 2006; Shendarkar et al. 2008). Additionally, the BDI agent model is easy to implement into a programme as it was designed as a software model.

2. 4. 3. 4 Features of the Agent-based Modelling Approach

Compared to the force-based models and the CA models (which focus on crowd movement), the agent-based models observe the individuals in a crowd from a different point of view. That is, the agent-based models describe how individuals make decisions based on their knowledge and their movements are thus behaviour results.

Advantages in Using an Agent-based Model

The concept of using agents to represent individuals has introduced many benefits:

- **Natural Mapping between Agents and Individuals**

An agent behaving on its own is comparable to a real person in reality. When taking into account the social psychological issues in crowd modelling, agents can easily be treated as individuals as they are designed to be intelligent and autonomous.

- **Individual Heterogeneity**

Because an agent is a self-contained entity, it can have different attributes to distinguish it from other agents. This approach enables individuals' characteristics and personalities to be reflected in the model thus creating a heterogeneous crowd to achieve more precise results when compared with reality.

- **Information Interchange**

As an agent possesses certain information of its own and has the ability to interact with other agents, it is possible to pass information around a crowd and the individuals in the crowd may possess different degrees of knowledge of the surrounding environment. Agent-based models can create a simulation environment that is closer to reality than the force-based models and the CA models where global settings are applied.

Limitations of Agent-based Models

However, agent-based models also have some limitations:

- **Require Large Computer Resources**

Because agent-based models are usually more complex than force-based models and CA models, they require and consume larger computer resources (in terms of computer hardware) when running the simulations. This was quite a big issue in the early days of using this type of modelling (i.e. 1990s or early 2000s). But this factor has become less and less important as computer hardware has developed very quickly in the past two decades.

- **Arbitrary in Agent Design**

As introduced above, agent-based modelling only describes the concept of the agent rather than defining the rules as to how agents work. The ways to design how the agents behave differ from study to study. For example, agent behaviours can be determined by a finite state machine or agents can act with behaviours under certain probabilities. Such varieties in agent design make comparisons and evaluations of agent-based models very difficult.

- **Lack of a Movement Representing Foundation**

The agent-based models usually focus on the decision making process of the agents and often simplify the process of transferring the decided behaviours into movement.

An agents-based model requires a movement representing system. As a matter of fact, rather than design new movement representing systems, many studies on agent-based models have chosen to use the movement representing systems based on the force-based models or CA models. These types of modelling approaches will be discussed in more detail in the “Hybrid Models” section.

2. 4. 4 Hybrid Models

As different modelling approaches have strengths in different areas, it has been suggested to combine different approaches (X. Zheng et al. 2009) to model a crowd. Actually, many research studies (Braun et al. 2003; Nuria Pelechano & Norman I Badler 2006; N Pelechano et al. 2007; Stefania Bandini et al. 2007; J. Dijkstra et al. 2000) have already combined modelling approaches to design a crowd model. This type of crowd model is categorised as “Hybrid Models” in this thesis and they are sub-categorised as follows:

- Agent-based CA Models
- Force-based Agent Models
- Force-based CA Models

2. 4. 4. 1 Agent-based CA Models

Introduction

Agent-based CA models combine the agent-based modelling approach and the CA modelling approach to simulate individuals’ decision-making and their movements. Unlike the update of the cell state in the CA models, the agent-based CA models consider that the agents are moving between the cells. More specifically, in the agent-based CA models, the fields and the individuals are represented respectively by the cells and the agents. The agents’ movements are decided by behaviour rules based on the states of the neighbouring cells and those agents in the neighbouring cells. The cells purely serve as the field representation to provide position information on the agents.

Typical Agent-based CA models

Many studies (Heliövaara et al. 2012; Hamagami & Hirata 2003; J. Dijkstra et al. 2000; Stefania Bandini et al. 2007; S. Bandini et al. 2002; Giuseppe Vizzari et al. 2008) have built crowd models by using this agent-based CA modelling approach. The following sections present two studies of agent-based CA models to demonstrate how these two modelling approaches are combined in a crowd model. The first study (J. Dijkstra et al. 2000) focused on improving the CA approach and the second study (Stefania Bandini et al. 2007) emphasised the design of the agent.

➢ **Dijkstra et al.'s multi-agent cellular automata system** *(J. Dijkstra et al. 2000)*

✧ ***Model overview***

In this crowd simulation system, the environment is represented by cellular automata. The cells are defined with width *W* and length *L*. The pedestrians move between the cells during each update step (it is possible for a pedestrian to move more than one cell in one update step). The cells have been divided into three types which affect the decision making:

- Empty: means that this cell belongs to the walkway.
- Decision: indicates that this cell represents a decision-point area (e.g. a T-junction).
- Wall: means that this cell is part of a wall.

The pedestrians are modelled as agents who have respective roles. They make their decisions and conduct their movement based on a set of predefined rules and the states (include occupation and type) of their neighbouring cells. The neighbouring cells are defined as the cells within the radius of *r* and are demonstrated in Figure 13. Taking the cell circled in the centre for example, when r= 1, its neighbouring cells are consisted of the cells numbered 1. When r = 2, the neighbouring cells consist of the cells numbered both 1 and 2.

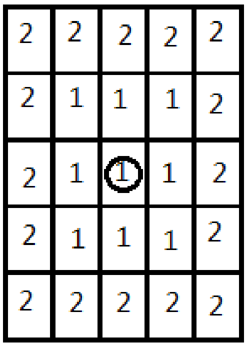


Figure 13 Illustration of neighbouring cells in Dijkstra et al.'s CA model (2000)

The rules that determine the agents' behaviours are defined as:

- **Rule 1**: **If** passed the decision point, then go to Rule 3. **Else** go to Rule 2.
- **Rule 2**: **If** the cell type is "decision", then determine a preferred direction and skip rule 3 and 4. **Else** go to Rule 3.
- **Rule 3**: **If** the cell is not occupied and the cell type is not a "wall", walk through. **Else** go to Rule 4.
- **Rule 4**: **If** the left/right adjacent cell is not occupied, move to the left/right cell. **Else** wait.

✧ ***Model discussion***

This model raised two new issues when using a CA model as the environment representation. One was the shapes of the cells could be triangles instead of the traditional squares. The other was that the neighbouring cells could include a cell whose distances were larger than one cell. However, the paper did not state how these two issues were handled and there was a lack of introduction to the decision making process of the agents. The proposed crowd model had demonstrated some novel concepts in order to improve the traditional CA modelling but a lack of detail and test simulations makes it difficult to evaluate the accuracy and reliability of this model.

➢ **Bandini et al.'s situated cellular agent approach** *(Stefania Bandini et al. 2007)*

✧ ***Model overview***

In this model, the agent (named Situated Cellular Agent) is defined by a 3-tuple $< s, p, \tau >$. τ represents the agent type which determines the agent state, perceptive capabilities and behaviour. s denotes the agent state which is one of the values specified from its type. p defines the field/space where the agent is situated. The agent behaviours are defined by four primitives: emit (s, f, p), react$(s, a_{p1}, a_{p2}, \cdots, a_{pn}, s')$, transport$(p, q)$, and trigger$(s, s')$. (As the agents and their behaviour representation are designed in complex mathematical methods and languages, for detailed descriptions of those primitives and their relationships to the agent types, please refer to Bandini et al.'s original paper (Stefania Bandini et al. 2007)). However, the field representation is quite simple and standard Cellular Automata has been adopted (the cell size is 40 cm × 40 cm).

✧ ***Model discussion***

This model presented a comprehensive agent design and behaviour representation method which used extensive mathematical languages and symbols. It demonstrated that complex agent models could be built upon a simple CA model.

Discussions on the Agent-based CA Modelling Method

Through combining agent-based modelling and CA modelling, the agent-based CA models possess the advantages of having intelligent agents and the simplified movement representation from both approaches. However, this combined modelling method does not cope with the limitation of imprecise position representation that is introduced by the CA models. In other words, the agent-based CA models do present a smarter crowd but the precision of the presented behaviour and crowd movement have not been improved.

2. 4. 4. 2 Force-based Agent Models

Introduction

Force-based agent models combine the agent-based modelling approach and the force-based modelling approach to simulate individuals' decision-making and how behaviours can affect movements. In force-based agent models, the agents and their decision-making process are designed the same as in the ordinary agent-based models. The effects of decided behaviours on the agents' movements will be calculated by the formulas defined via use of the force-based modelling approach.

There are two clear layers in a force-based agent model: the agent model sits on the top to decide the agents' behaviours while the force-based model lies on the bottom to calculate the corresponding actions of those behaviours.

As a summary, in force-based agent models, the movements of agents are still modelled through the representation of formulas such as with traditional force-based models. However, it is considered that such calculations should reflect crowd heterogeneity thus individuals' differences are taken into account via the agent.

Typical Force-based Agent Models

There are several research studies (Braun et al. 2003; Nuria Pelechano & Norman I Badler 2006; Qiu & Hu 2010; Heliövaara et al. 2012) on force-based agent models. The following sections present some typical force-based agent models to demonstrate how these two modelling approaches are combined and how the agents' attributes and state affect the calculation of the agents' movements.

➢ Steering behaviours for autonomous characters (Reynolds 1999)

✧ ***Model overview***

This model was proposed by Reynolds (1999) as a solution in order to create autonomous characters in animations and games. This model defined eighteen behaviours that could be performed by an agent and presented detailed calculation methods as to how those behaviours affect agents' movements. Additionally, Reynolds presented the concept of combining existing behaviours to create complex patterns of behaviours.

✧ ***Model details***

✦ ***Agent design***

The model described the agent itself which is called the "Simple Vehicle Model" in Reynolds' study. It represents the steering behaviours which are defined to describe agent action and movement. This Simple Vehicle Model contained the following parameters:

Table 2 The parameters of Reynolds' Simple Vehicle Model (1999)

Parameter	Type
mass	scalar
position	vector
velocity	vector
max_force	scalar
max_speed	scalar
orientation	N basis vectors

The agents defined by this Simple Vehicle Model update their position by applying the steering forces at each simulation step (the physics is based on forward Euler integration). The formulas are given as follows:

$$\boldsymbol{steering\_steering\_force} = \boldsymbol{truncate}\,(\boldsymbol{steering\_direction}, \boldsymbol{max\_force}),$$
$$\boldsymbol{acceleration} = \boldsymbol{steering\_force} \,/\, \boldsymbol{mass},$$
$$\boldsymbol{velocity} = \boldsymbol{truncate}\,(\boldsymbol{velocity} + \boldsymbol{acceleration}, \boldsymbol{max\_speed}),$$
$$\boldsymbol{position} = \boldsymbol{position} + \boldsymbol{velocity}$$

Formula 26 Formulas of the updating agent's position by applying the steering force (Reynolds 1999)

✦ ***Steering behaviours***

The steering behaviours are the agent's movement actions. The results of the steering behaviours on the agents' movements are represented through the geometric calculation of the desired steering force. The following steering behaviours have been defined: seek, flee, pursuit, evasion, offset pursuit, arrival, obstacle avoidance, wander, path following, wall following, containment, flow field following, unaligned collision avoidance, separation, cohesion, alignment, and leader following.

Seek (or pursuit of a static target) is the behaviour that steers the agent towards a specified position in a global space. This behaviour produces a steering that aligns the agent's velocity with the direction of the target (see Figure 14).

Flee is a behaviour that inverses the behaviour "seek". It steers the agent to move away from the target (see Figure 14).

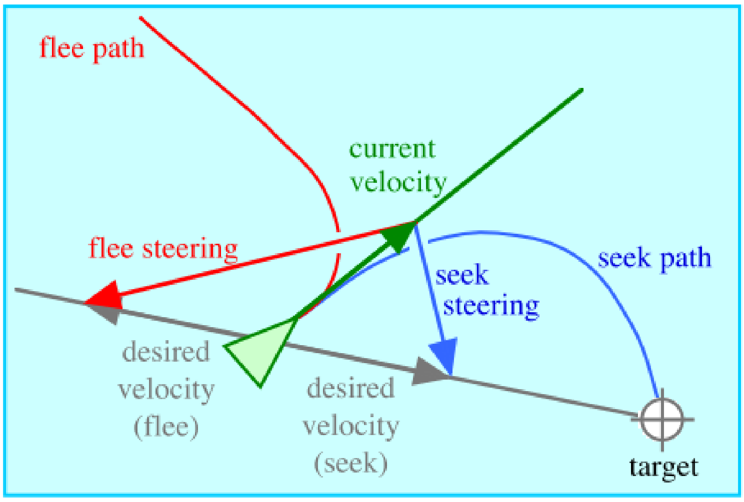


Figure 14 Demonstrations of the "seek" and "flee" steering behaviours (Reynolds 1999)

Pursuit is the behaviour that steers the agent towards a moving target. The steering on agent's velocity will be based on the prediction of the future position of the target (see Figure 15).

Evasion represents the behaviour that steers the agent to the opposite direction of a moving target. The steering on agent's velocity will be based on the prediction of the future position of the target (see Figure 15).

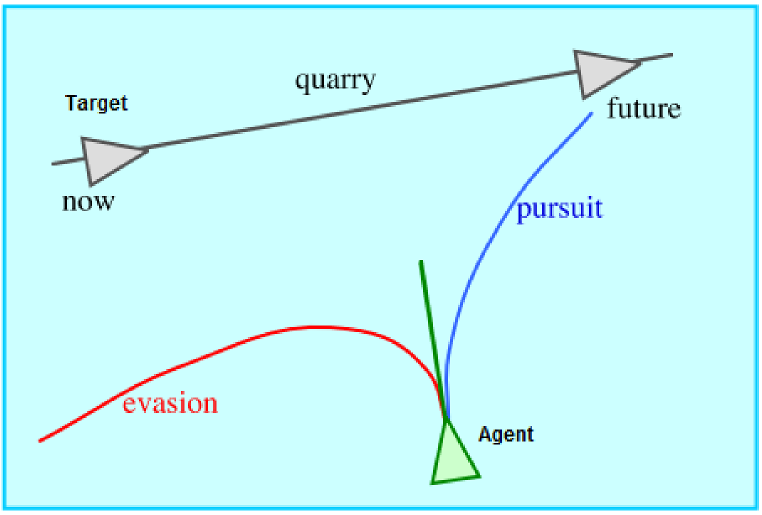


Figure 15 Demonstrations of the "pursuit" and "evasion" steering behaviours (Reynolds 1999)

Offset pursuit refers to the behaviour that steers the agent to a path near (by a given radius) a moving target (see Figure 16).

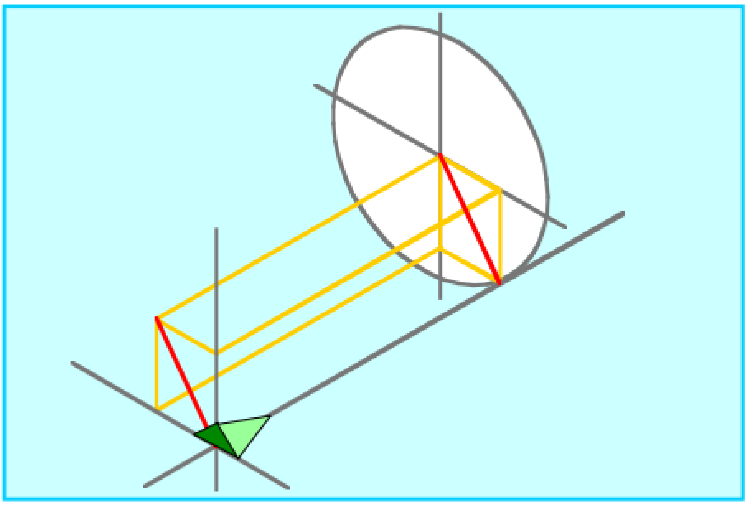


Figure 16 Demonstration of the "offset pursuit" steering behaviour (Reynolds 1999)

Arrival represents a behaviour that diverts from the "seek" behaviour. This behaviour causes the agent to slow down when it is approaching the target and to eventually stops at the target position.

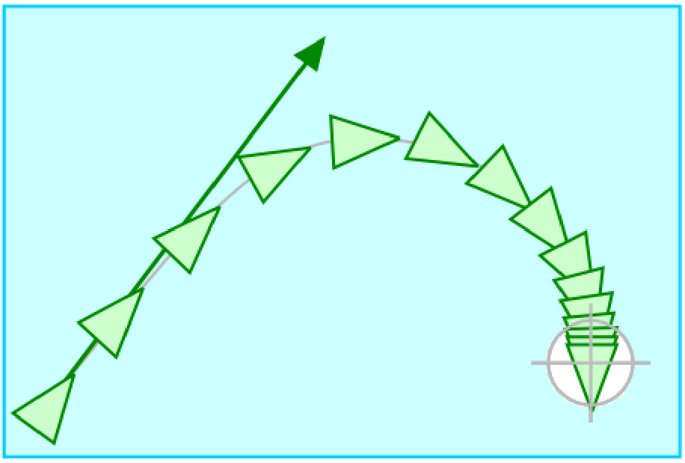


Figure 17 Demonstration of the "arrival" steering behaviour (Reynolds 1999)

Obstacle avoidance is the behaviour when an agent manoeuvres in a cluttered environment by dodging around obstacles. The obstacle avoiding strategies in this model are based on the assumption that both the agents and obstacle can be reasonably approximated as spheres (see Figure 18).

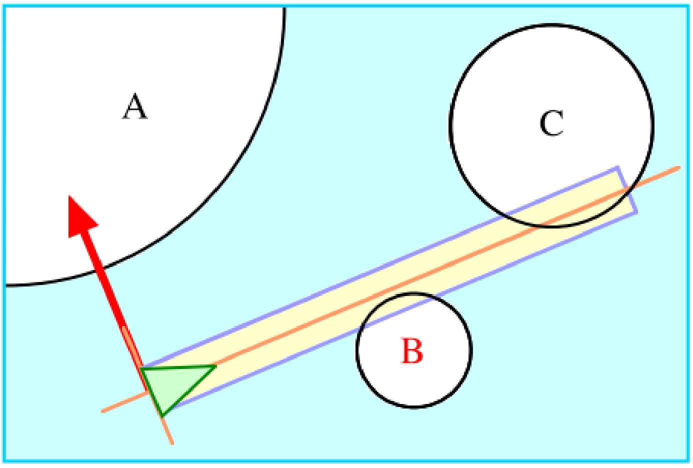


Figure 18 Demonstration of the "obstacle avoidance" steering behaviour (Reynolds 1999)

Wander is a behaviour which the agent steers with random directions. This model proposed defining this behaviour in order to retain the steering of the agent's direction state and to make small random displacements to it in each frame (see Figure 19). The agent is likely to turn in the same direction consecutively.

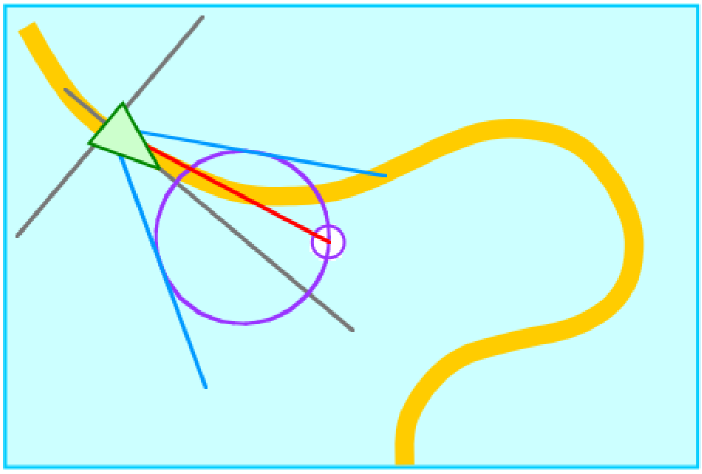


Figure 19 Demonstration of the "wander" steering behaviour (Reynolds 1999)

Path following refers to the behaviour whereby an agent follows a predetermined path, such as a roadway, corridor or tunnel (see Figure 20). Variations on this behaviour include "**wall following**" and "**containment**" (see Figure 21).

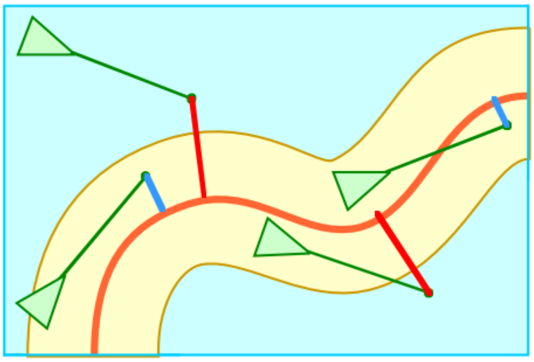


Figure 20 Demonstrations of the “path following” steering behaviour **(Reynolds 1999)**

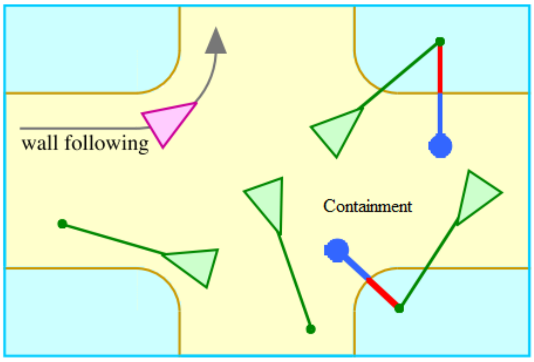


Figure 21 Demonstrations of the “wall following” and “containment” steering behaviours (Reynolds 1999)

Flow following describes the behaviour whereby the agent’s motion is affected by its position within an environment. Every position in the environment contains direction information known as the flow field (imagine the floor has arrows on it to direct the agents) (see Figure 22).

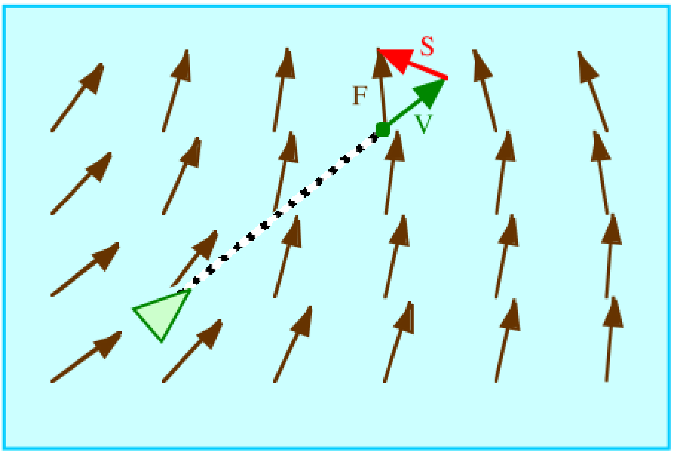


Figure 22 Demonstration of the “flow following” steering behaviour (Reynolds 1999)

Unaligned collision avoidance refers to the behaviour whereby the agent tries to avoid a possible collision with another agent by predicting their future positions. As demonstrated in Figure 23, the agent coming from the right will slow down and turn to its left, while the agent approaching from the left will speed up and turn to its left as both of them have sensed a potential collision at a future position.

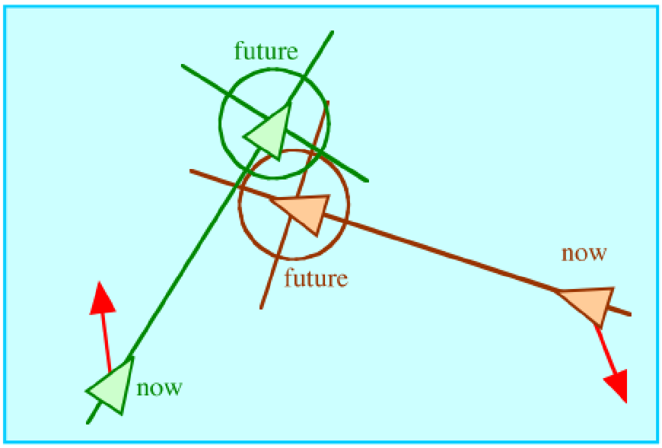


Figure 23 Demonstration of the "unaligned collision avoidance" steering behaviour (Reynolds 1999)

Leader following is a behaviour that describe the situation when one or more agents follow another moving agent defined as the leader. In this behaviour, the followers want to stay near the leader's back without getting to close as well as staying out of the leader's way (in case they happen to find themselves in front of the leader) (see Figure 24).

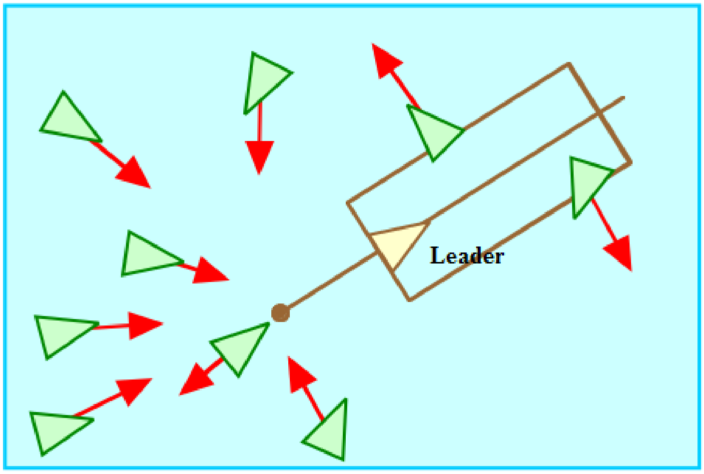


Figure 24 Demonstration of the "leader following" steering behaviour (Reynolds 1999)

This model has defined three group related steering behaviours: **separation**, **cohesion** and **alignment**. Because the definitions and descriptions of these behaviours are same as the behaviours proposed in Reynolds' "Boid" model (Reynolds 1987) which has already been reviewed in this PhD thesis, "*section 2. 4. 1 - Force-based Models*" can be referred to for details.

✦ ***Combining behaviours***

As it stated by Reynolds, "*Unless an autonomous character exists in a very simple world, it would seldom make sense for the character to continually execute a single steering behaviour*", thus an agent should be able to switch between behaviours sequentially as well as having the ability to perform multiple behaviours in parallel. For example, "**flocking**" behaviour can be achieved by combining the "separation", "cohesion" and "alignment" behaviours.

This model proposed to combine the steering forces of multiple behaviours by computing each of the component steering behaviours and adding the steering forces together, possibly with a weighting factor for each of them. This linear combining approach is simple but works well. However, it may introduce two shortcomings: it is not the most computationally efficient approach, and the component behaviours may cancel out each other's steering forces in the end. To cope with these shortcomings, Reynolds considered that the computation load could be decreased by observing that a character's momentum serves to apply a low-pass filter to the changes in the steering force and the problem of components cancelling each other out can be addressed by assigning a priority to components.

✧ ***Model discussion***

The strengths of Reynolds' model are the detailed descriptions and implementations of the agents' behaviours and their effects on the agents' movements. It has defined eighteen steering behaviours to describe the movements of the agents and presents the movement calculation formulas by using the parameters that are defined in the simple agent model. In addition, this model raised a new concept in combining the basic behaviours to create complex patterns of behaviours. The author considers this is a good approach to building new behaviours for an existing crowd model as no crowd model can define every behaviour within the crowd. If a mechanism that combines new behaviours can be established, the possible behaviours from such a crowd model can be unlimited and unpredictable as the new combined behaviours can be combined again to create other new behaviours.

However, because Reynolds' steering behaviours' model aimed to define autonomous characters for games and animations, the agent model in his study has been designed very simply and only with physical parameters that related directly to movement. The decision making process has also been skipped.

➢ Braun et al.'s Social Force based agent model (Braun et al. 2003)

✧ *Model introduction*

Braun et al. considered, when using Helbing et al.'s Social Force model (D. Helbing et al. 2000) to simulate emergency evacuations, that it was unreal for a crowd to react in the same way. They proposed that individual characteristics and group structure (i.e. a number of people moving together as a group) in the crowd should be taken into account.

The approach of agent-based modelling was introduced to create heterogeneous individuals. The agents were designed to have the following parameters:

- **Id** - An identifier of the agent.
- **IdFamily** - An identifier of the family. A family consists of a number of agents who know each other and tend to stay as a group while moving.
- **DE** - Dependence level of the agent, which indicates the degree of need for help.
- **AL** - Altruism level of the agent, which indicates the agent's tendency to help others. It is designed as an in-family parameter.

The agents' movements were based on the Social Force model plus two additional rules. (This section only presents the formulas of these rules. For a detailed parameters' description, please refer to the original paper (Braun et al. 2003).) The first rule is that the desired speed of the agent is determined by its maximum velocity and the DE value, given by the function:

$$\boldsymbol{v_0 = (1 - DE)v_{max}}$$

Formula 27 The formula of the agent's desired velocity (Braun et al. 2003)

The second rule is that, in addition to the forces defined in the Social Force model, the agents in the same family are affected by the altruism forces $\overrightarrow{Fa_i}$ from their family members (see Figure 25). The altruism force is determined by the positions of the family members and their AL and DE values. The calculation of $\overrightarrow{Fa_i}$ is given by:

$$\overrightarrow{Fa_i} = K \sum_j AL_j DE_j \left| \overrightarrow{d_{ij}} - \overrightarrow{d_{ip}} \right| \overrightarrow{e_{ij}}$$

Formula 28 The formula to calculate the altruism force (Braun et al. 2003)

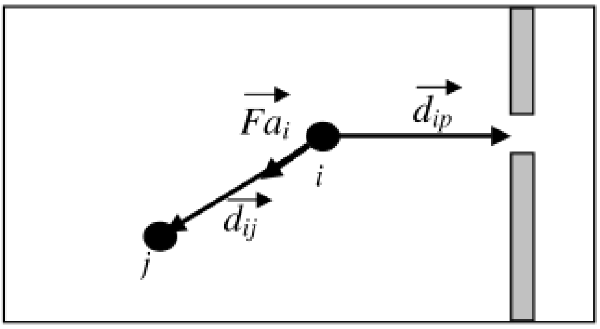


Figure 25 Representation of the altruism force on agent i (Braun et al. 2003)

✧ ***Model discussion***

The contributions of this model centre on two aspects. From the model design point of view, this model demonstrated how to integrate the agents' characteristics and their behaviour preferences into the calculations of the force-based models. For example, the DE parameter can affect the agent's desired velocity.

From the crowd simulation point of view, the simulation results from this study did show that a heterogeneous crowd could have different behaviours. For example, Figure 26 shows the transition of individuals' positions during the simulation of a room evacuation. It can be seen that the agents in the same family (represented in the same colour) have formed groups, leaving their initial separate positions, during the evacuation. It also demonstrates that the flow rates of people exiting from the door are dependent on the DE and AL values.

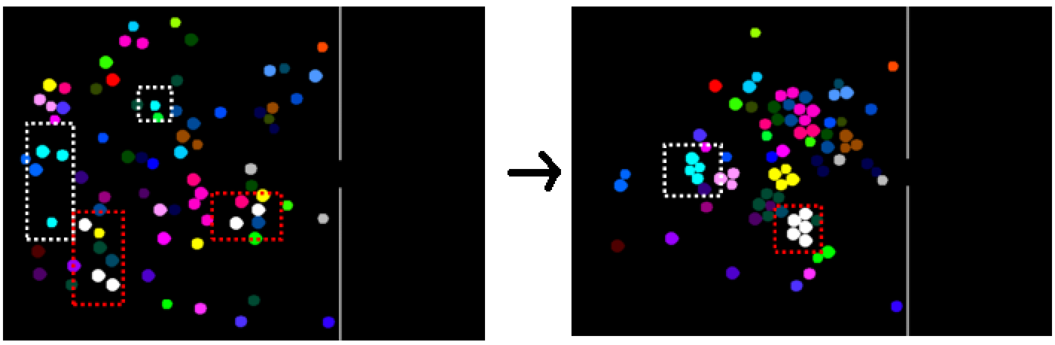


Figure 26 The grouping behaviours of families during evacuation (Braun et al. 2003)

Discussions on the Force-based Agent Models

Through combining the force-based modelling approach and the agent-based modelling approach, the strengths of both approaches have been taken advantage of and the weaknesses have been compensated for. More specifically, as reviewed before, the force-based models have precise representations of movements but have difficulty in representing the decision making process. In contrast, the agent-based models focus on the intelligent behaviours of individuals but lack the theoretical foundations of movement representation. The force-based agent modelling approach has offered a solution by absorbing the benefits of these two approaches.

However, it has to be pointed out how the parameters of the agents affecting the calculations of the formulas vary in the different force-based agent models. Because there is no standard mechanism to measure the effects of the agents' parameters in the forces' calculation formulas, it is very difficult to combine the different behaviours from separate studies to create a more comprehensive crowd model.

2. 4. 4. 3 Force-based CA models

Introduction

Not many studies have considered combining the force-based modelling approach and the CA modelling approach together to design crowd models as both approaches focus on the movement representation of individuals. Song et al. (2006) considered that by combining these two approaches, such a crowd model can have the computational efficiency of the CA model but can also retain the ability to represent complex interaction behaviours in the crowd as the presented in force-based models. In their study, they presented a CA model with force essentials to simulate crowd behaviours at an exit.

A CA Model with Force Essentials (Weiguo Song et al. 2006)

This model was designed to simulate crowd behaviour during emergency evacuations. The movement representation is based on the traditional CA modelling method. In the model, each cell is defined as a square of $0.4\text{m} \times 0.4\text{m}$ in size. The cell updates its state by considering three types of interactions: attraction, repulsion and friction.

"Attraction" represents the behaviour whereby people always move toward the exit during an evacuation. During each update, a pedestrian will move to an adjacent cell that is closer to the exit until it reaches the exit. The target cell is defined as "first choice".

The forces' essentials are reflected through "repulsion" and "friction". A pedestrian will modify his/her "first choice" by taking into account "repulsion" and "friction". Repulsion represents the effects from nearby pedestrians or walls (see Figure 27, the arrow indicates the pedestrian's "first choice"). Friction represents the slowing down effects caused by the two touching pedestrians (or wall) (see Figure 28, the arrow indicates the pedestrian's "first choice"). As a result, the effects of repulsion and friction are transferred into probabilities that can affect the pedestrian's "first choice" (*For the detailed formulas and calculations, please refer to Song et al.'s original paper (2006)*).

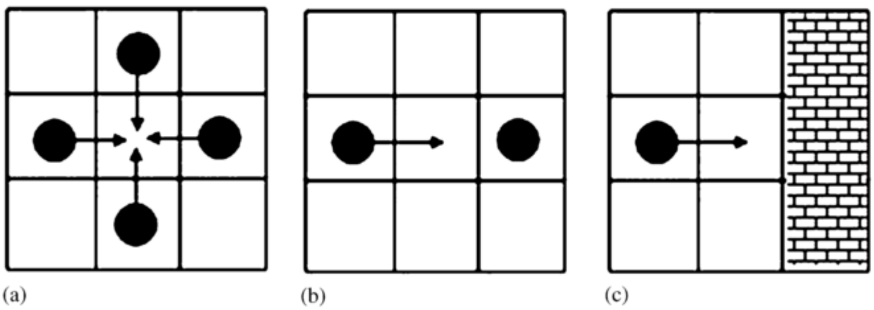


Figure 27 Occurrence of repulsion (Weiguo Song et al. 2006)

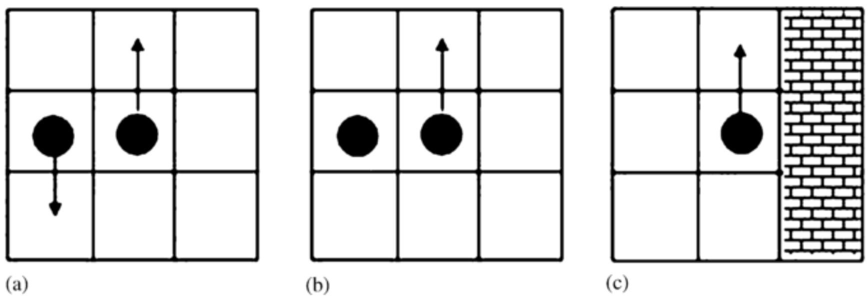


Figure 28 Occurrence of friction (Weiguo Song et al. 2006)

Discussion

The simulation results shown in this CA model with force essentials can produce similar crowd behaviours to the Social Force model (D. Helbing et al. 2000). Because this model was based on CA modelling, it clearly has computational advantages.

Although Zheng et al. (2009) considered this model as being "*based on the lattice gas model and the Social Force model*", the effects of "attraction", "repulsion", and "friction" were calculated in a different way. This model used the stochastic method to take into account the force essentials in designing the rules that update the cell states. This approach was based on Kirchner and Schadschneider's studies (2002) which used purely a CA model. As a conclusion, this model had integrated the force essentials in modelling crowd movement, but the method was not adopted from the force-based modelling approach.

Generally speaking, the force-based modelling approach and the CA modelling approach are mutually exclusive because they have different representations of individuals' movements. However, the concept of such a modelling approach can be used to inspire this other modelling approaches. Song et al.'s model (2006) is one of the examples of this.

2. 4. 5 Other Modelling Approaches

2. 4. 5. 1 The Integrated Network Approach (J. Yuan et al. 2009)

In the simulations of very large buildings with a huge number of people, the micro scope models can become unsuitable due the limitations of computational power while macro scope models cannot provide adequate information on crowd behaviour in some important positions (e.g. on main corridors and on stairs). Yuan et al. therefore proposed (2009) to build a mixed scopes model for such occasions. The presented crowd model uses an integrated network approach to describing crowd movement and is tested in a complex environment (the evacuation of ten thousand people in a large shopping mall of four floors). The integrated network consisted of coarse grids and fine grids. Grids represent the zones (e.g. rooms, corridors, stairs, etc.) in the building. The crowd in the coarse grids are modelled by a fluid-dynamic model and the crowd in the fine grid are modelled by a CA model.

This modelling approach presents a solution to modelling extremely large numbers of people in a complex environment. The trade-off between simulation efficiency and detailed representation is considered well worth in the case of the computational power is limited.

2. 4. 5. 2 A Cognitive Approach based on Heuristics (Moussaïd et al. 2011)

Moussaid et al. (2011) argued that force-based models could lead to sophisticated mathematical expressions when representing complex behaviours and that the integration of behavioural forces could raise many theoretical issues (e.g. adjusting the weightings of these forces) as well. They proposed a crowd model based on two heuristics to describe a pedestrian's motion. The first heuristic determined how a pedestrian adjusted his/her walking direction which was given as "*A pedestrian chooses the direction* α_{des} *that allows the most direct path to destination point* O_i *, taking into account the presence of obstacles*". The second heuristic determined a pedestrian's desired walking speed which was given as "*A pedestrian maintains a distance from the first obstacle in the chosen walking direction that ensures a time to collision of at least* τ". In addition, this model adopted the concept of the interaction forces from the Social Force model (D. Helbing et al. 2000) but only in the case of pedestrians touching each other (the same rule was applied to the walls).

In this model, the pedestrians determined their motions actively through their perceptions and were only repelled by the passive forces from other pedestrians or walls in the overcrowded environment (i.e. touching each other). This was considered as a more reliable description of real-life situations (Moussaïd et al. 2011). To model crowd behaviour in two circumstances (normal and crowded) was an improvement on the traditional force-based models. However, the two heuristics to determine pedestrians' motions can only prove an alternative representation of pedestrians' movement. Because they were applied globally to all pedestrians they have the same limitations as the formulas in the force-based model. Additionally, this model also presented a solution to integrate the effects of the forces from multiple targets into the pedestrian's vision rather than combining those forces as binary interactions (an approach that is used in the force-based models). However, such a solution could only be treated as an alternative, not as a better approach, as it was theoretically based on the heuristics.

To conclude, this heuristics based cognitive crowd model has some advantages over the traditional force-based models. For example, it presents simpler rules to determine pedestrians' motions. It describes crowd behaviours in two circumstances. However, like the force-based models, the heuristics were globally applied which means that heterogeneity and individual intelligence were not presented.

2. 5 Summary of Crowd Models

This section summaries the reviewed crowd models in previous sections which aims to provide an overview of comparisons between different modelling methods.

Table 3 Summary of crowd models

Modelling Methods		Advantage	Disadvantage
Macro Scope Crowd Models		• Require less computing resources when comparing to Micro Scope Models	• No details of individuals
Micro Scope Crowd Models		• Detailed information of individuals	• Require more computer resources when comparing to Macro Scope Models
Micro Scope Crowd Models	Force-based Models	• Provide Continuous position representation of individuals • Built foundation on physical laws	• Crowd is homogeneous • Is computationally expensive
	Cellular Automata (CA)	• Simplified crowd	• Discrete in time and

	Models		• movements and field representation • Direct mapping between individual behaviours and cell state update rules	• space • Fixed cell size can cause unrealistic behaviours • Lack of individuals' characters
	Agent-based Models (ABM)		• Natural mapping between agents and individuals • Individual heterogeneity • Information interchange between agent	• Require more computer resources than most other models • Arbitrary in agent design • Lack of a movement representing foundation
	Hybrid Models	Agent-based CA Models	• Intelligent agents • simplified movement representation	• Still have the disadvantages of CA models
		Force-based Agent Models	• The strengths of both approaches have been taken advantage of and the weaknesses have been compensated for	• Lack a unified mechanism to link agents' parameters and movement calculations
		Force-based CA Models	• Have computational advantages by using CA model	• Two approaches are mutually exclusive because of different representations on individuals' movements

	Other	The Integrated Network approach	• Can simulate huge number of crowd	• The trade-off between simulation efficiency and detailed representation is only well worth in the case of the computational power is limited.
		Cognitive Approach based on Heuristics	• Provides simpler rules over traditional force-based models	• heterogeneity and individual intelligence were not presented

2. 6 Crowd Model Applications and Simulations

Crowd models are created to represent crowd behaviour in many areas and situations. They can look at crowd behaviour in buildings or on streets. It can be under emergency circumstances or in normal conditions. The following sections represent the most popular applications of crowd simulations.

2. 6. 1 Emergency Evacuations

The most often applied area of crowd modelling is for emergency evacuations. A lot of studies (Santos & Aguirre 2004; Simpson 2004; Cheng et al. 2008; Georgoudas et al. 2006; Kobes, Helsloot, Vries, et al. 2010; Núria Pelechano & A. Malkawi 2007; Tang & X. Zhang 2008; D Helbing et al. 2002; Parisi & Dorso 2007; Nuria Pelechano & Norman I Badler 2006; Varas et al. 2007; Kirchner & Schadschneider 2002; Weiguo Song et al. 2006; Zoumpoulaki et al. 2010; Nuria Pelechano & a Malkawi 2008; J. Yuan et al. 2009; Aguiar 2010; D. Zhao et al. 2006; W. Yuan & Tan 2007; D Helbing et al. 2003) have been carried out to simulate crowd behaviours during emergency evacuations. These studies have presented many empirical crowd behaviours in emergency evacuations and have tried to interpret the causes of such behaviours through modelling them.

2. 6. 1. 1 Congestion

Congestion has often been observed in emergency evacuations . It happens when a large number of individuals try to pass through one exit and they are impatient while waiting. This behaviour is often modelled as one individual considering moving to an exit and ignoring others' existences (D Helbing et al. 2002; Cheng et al. 2008).

Figure 29 shows the individuals are trying to push each other in order to pass through the door and the congestion is found near the door which is circled in red.

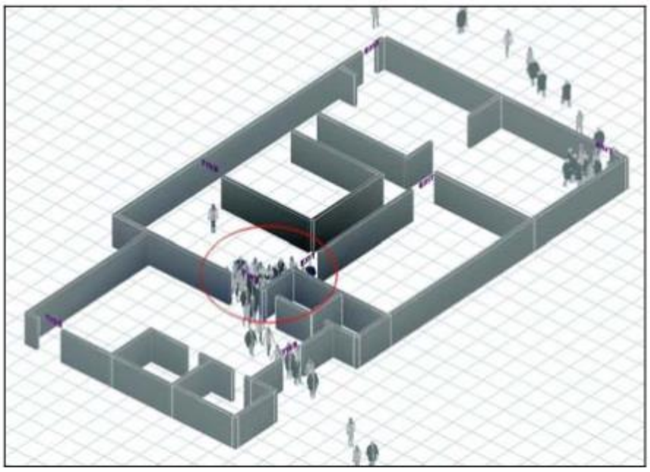


Figure 29 Competitive behaviour (Xiaoshan Pan et al. 2007)

Figure 30 demonstrated that potential congestion areas in buildings can be identified through crowd simulations.

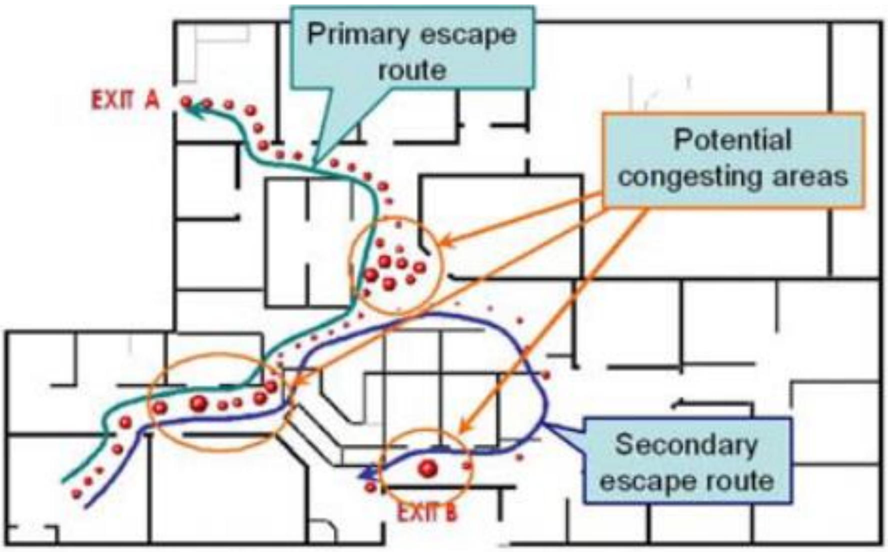


Figure 30 Potential congestion areas have been highlighted through simulation (Xiaoshan Pan et al. 2007)

In addition, crowd formation can transit into an arch-like shape when the congestions happen at a small exit. The crowd spreads into such a formation because all the individuals move to positions that are close to the exit. This phenomenon is illustrated in Figure 31 and Figure 32.

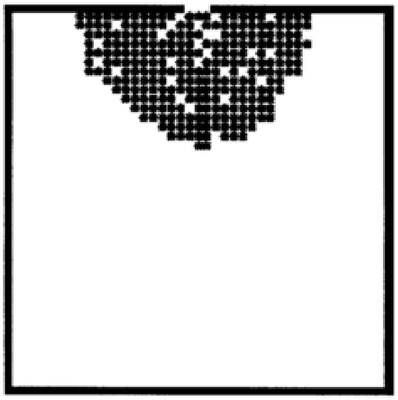


Figure 31 Congestion is observed when a large number of people escape via one exit (Kirchner & Schadschneider 2002).

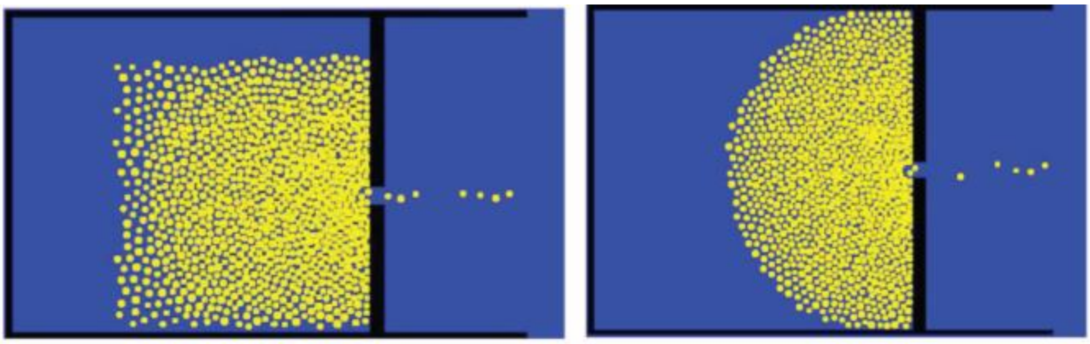


Figure 32 Crowd transits into an arch-like formation when congesting at an exit

2. 6. 1. 2 Queuing Behaviour

Queuing behaviour is when a crowd take turns to pass through an exit and is usually opposite to the congesting behaviour. It is considered as a more effective evacuation behaviour than everybody rushing for the exit (Xiaoshan Pan et al. 2007). Queuing behaviour happens when a crowd are patient and imperturbable (D Helbing et al. 2002; Cheng et al. 2008). Some crowd models (Xiaoshan Pan et al. 2007; Kruszewski 2005; Cheng et al. 2008) have included rules to represent queuing behaviour (see Figure 33 and Figure 34).

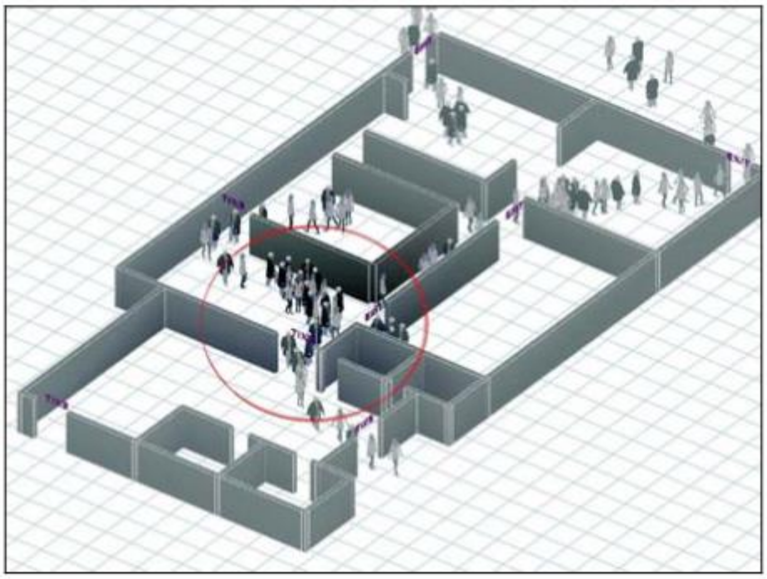


Figure 33 The crowd are imperturbable enough to queue in the corridor (Xiaoshan Pan et al. 2007)

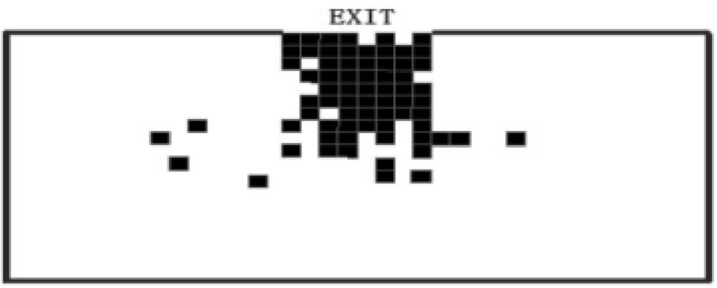


Figure 34 Queuing behaviour can happen when the individuals are not pushing each other (Cheng et al. 2008)

2. 6. 1. 3 Herding Behaviour

Herding behaviour describes a phenomenon whereby in a multi-exit environment (e.g. in a room with more than one door) one exit is clogged while other exits may not be fully utilized. There are several interpretations as to what results in this behaviour. Some studies (Cheng et al. 2008; Kobes, Helsloot, Vries, et al. 2010) considered this behaviour was caused by a crowd tending to make use of their familiar exit during an evacuation process. Some studies (D Helbing et al. 2002) modelled this behaviour as individuals tending to follow the nearby crowd. Thus the clogged crowd at one exit would attract more people while the other under-used exits could be fully utilised but are not.

The following snapshots (Figure 35, Figure 36, and Figure 37) show the herding behaviour in simulations.

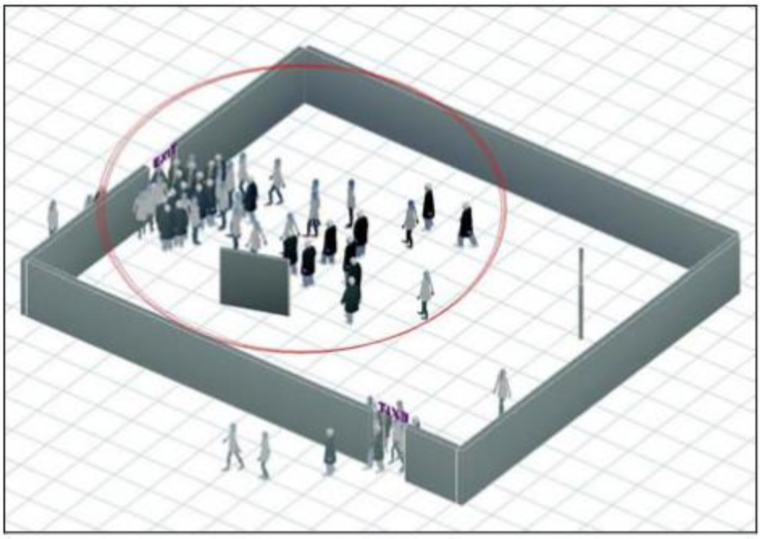


Figure 35 Herding - one exit more used than the other (Xiaoshan Pan et al. 2007)

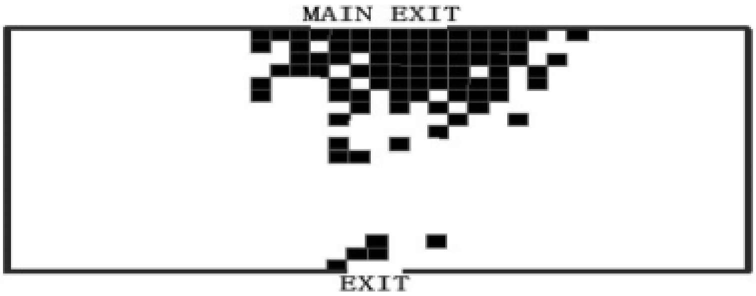


Figure 36 Herding - one exit is more used than the other (Cheng et al. 2008)

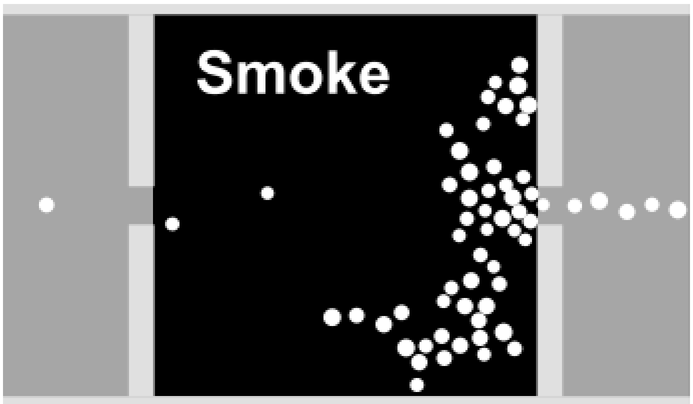


Figure 37 Herding - crowds tend to move to the exit with more people (D Helbing et al. 2002)

2. 6. 1. 4 Grouping Behaviour

Grouping behaviour represents the phenomenon whereby some individuals prefer to stay together and move in a group (Kobes, Helsloot, De Vries, et al. 2010). Pelechano and Badler (2006) modelled this behaviour through leadership. People with strong leadership attributes can attract others (but this factor can only affect the people within a certain range) to follow them thus the groups are then formed

Figure 38 (the individuals are represented by the small colour dots. The big red bunches represent fire in the room) shows leadership can result in different sizes of groups during a maze-like building evacuation. Figure 38(a) presents the simulation with low leaderships' crowd and the groups with small numbers of individuals are observed. Figure 38(b) presents simulation with high leaderships' crowds and the groups are consisted of larger number of individuals compared to the previous case.

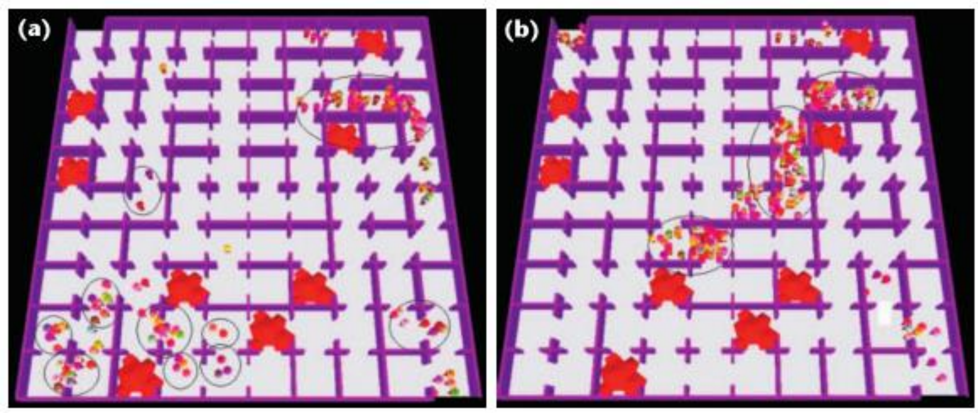


Figure 38 Leadership can cause individuals to stay in groups (Nuria Pelechano & Norman I Badler 2006)

2. 6. 1. 5 The Effects of Better Route Choices

In emergency evacuations, people cannot usually choose the shortest route to escape. They generally like to evacuate by using their familiar route and tend to ignore the short exit route offered by the exit signs (Kobes, Helsloot, Vries, et al. 2010). Or they prefer the clear and long route over a zigzag shortcut (Simpson 2004). There some studies on how to increase evacuation efficiency during such emergencies. For example, Pelechano and Badler (2006) demonstrated that the number of the evacuees could increase if there was communications within the crowd (i.e. individuals could exchange route information on finding the best route) (see Figure 39).

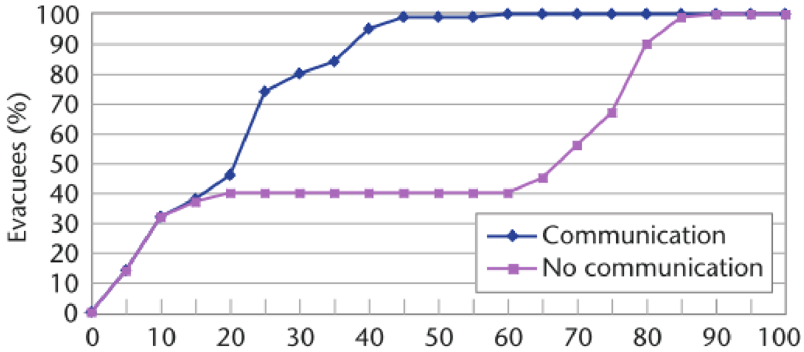


Figure 39 Communication in the crowd can increase evacuation efficiency (Nuria Pelechano & Norman I Badler 2006)

Shendarkar et al. (2008) showed that where policemen could guide a crowd in a fire situation a quicker evacuation was achieved but they also showed that such an effect had a cap when the policemen reached an excessive number.

2. 6. 1. 6 "Faster-is-slower" Effect

The "faster-is-slower" effect describes the situation where during an emergency evacuation. The faster individuals want to move, the more time would be taken for the whole crowd to escape. This phenomenon was first found in the Social Force model (D. Helbing et al. 2000). It was considered that an increase in the desired velocity could result in large frictions which could slow down the crowd movement. Figure 40 shows the desired velocities and the correspondent evacuation times in Song et al.'s (2006) study.

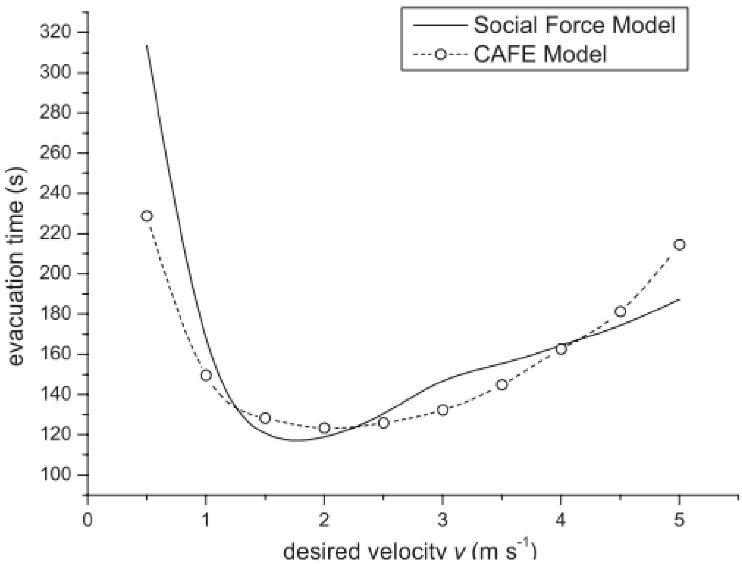


Figure 40 Demonstration of the "Faster-is-slower" effect (Weiguo Song et al. 2006)

2. 6. 1. 7 Other Findings or Crowd Behaviours

The following sections list more findings from the crowd simulations:

Limited Effects of Wide Doors

Song et al. (2006) showed that increasing the door width only had a limited effect on evacuation time. They found that in the evacuation of two hundred people from a room ($15\text{m} \times 15\text{m}$), the evacuation times were almost no different when the door width was above 2.4 meters.

Irregular Crowd Flow through the Exit

This is a phenomenon represented by the Social Force model (D. Helbing et al. 2000). It has been reported that in the situation of a large number of people exiting through a small exit, the flow rate through the exit became irregular when the crowd's desired velocities were high (above 1.5 m/s). In other words, the crowd sometimes got stuck at the exit and no-one could exit. This was caused by the frictions between individuals becoming high even to the point of stopping their movement.

One Side Usage of a Narrow Door

This phenomenon was simulated by the Social Force model (Dirk Helbing & Peter Molnar 1995). Once a pedestrian has passed through a door, the others on the same side are more likely to follow while the pedestrians on the other side have to wait (see Figure 41, the black circles represent the pedestrians going to the right-hand side and the white circles represent the pedestrians going to the left-hand side).

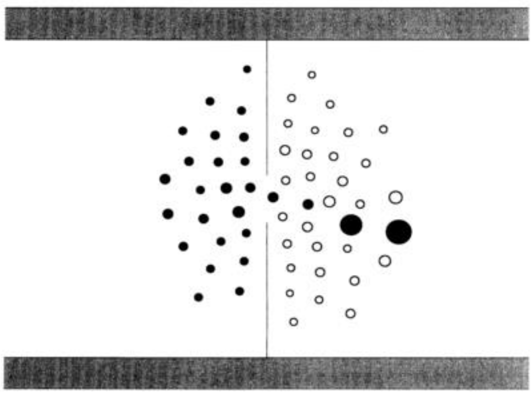


Figure 41 A narrow door usually has one-direction traffic (Dirk Helbing & Peter Molnar 1995)

2. 6. 2 The lane formation of a bi-directional crowd flow

It has been found that, in the situation where a crowd moves in bi-directions in a contained environment (e.g. the crowd coming from both sides of a corridor or a street), although there are no explicit rules or signs to guide the crowd, the movements of the crowd will form into lanes spontaneously and eventually reach a stable state (see Figure 42 and Figure 43 for a demonstration). This phenomenon is usually known as “lane formation in a bi-directional crowd flow”. There are many studies that have specifically designed crowd models to simulate (Blue & Adler 2001; W. Fang et al. 2003; Jian et al. 2005) or were capable of simulating (Dirk Helbing & Peter Molnar 1995; N Pelechano et al. 2007; X Pan et al. 2006; Zoumpoulaki et al. 2010) this phenomenon.

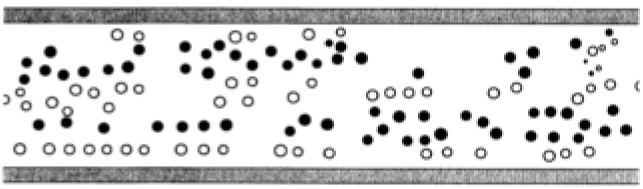


Figure 42 Lane formation will be formed spontaneously in a bi-directional crowd flow (Dirk Helbing & Peter Molnar 1995)

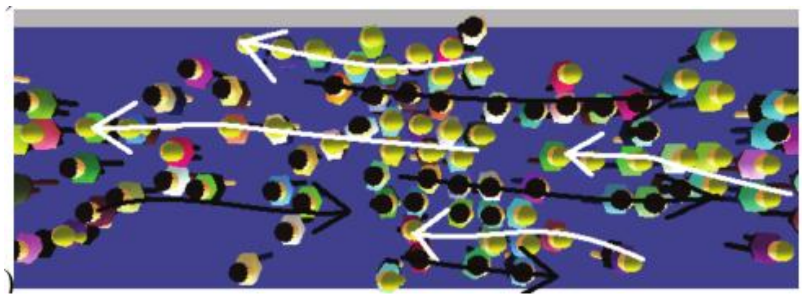


Figure 43 The bi-directional flow in the HiDAC model (N Pelechano et al. 2007)

There are many studies that have further investigated what could affect lane formations. For example, Tajima et al. (2002) demonstrated that different manoeuvre strategies could affect lane formations in a bi-directional walking flow. Lane formation is more prominent in the case of pedestrians trying to coordinate their movements with those walking in the same direction (see Figure 44(a)) than in the case of pedestrians aiming to avoid others walking in the opposite direction (see Figure 44(b)).

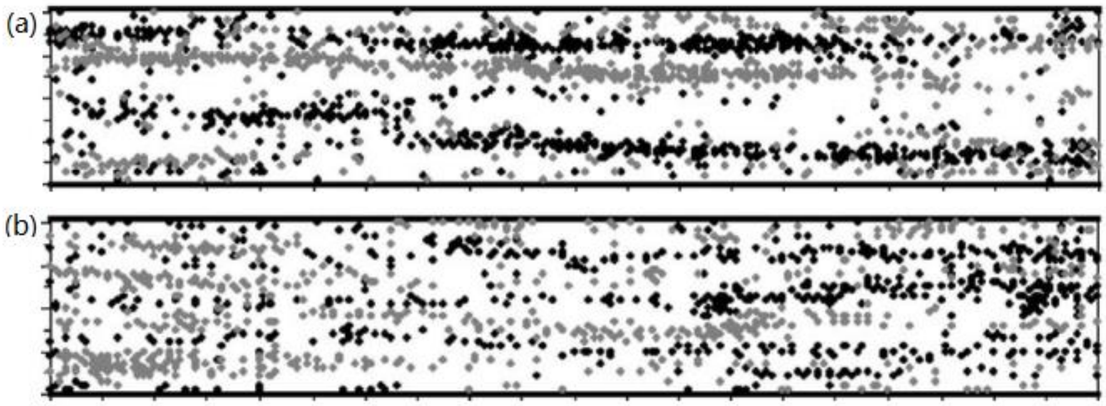


Figure 44 Lane formation is dependent on pedestrians' moving preferences (Yusuke Tajima et al. 2002)

Yue et al. (2010) considered adding the moving custom of pedestrians (preferring walking on the right-hand side) in the simulation of a bi-directional flow and reported that pedestrians with a higher composition of the same moving custom would experience a better performance in terms of the crowd velocity–density and flow–density. Figure 45(a) shows the simulation with no walking preference and Figure 45(b) with the preference to walk on the right-hand side in order to avoid collisions. The pedestrians are moving in the up-down (white triangles) and down-up (black triangles) directions.

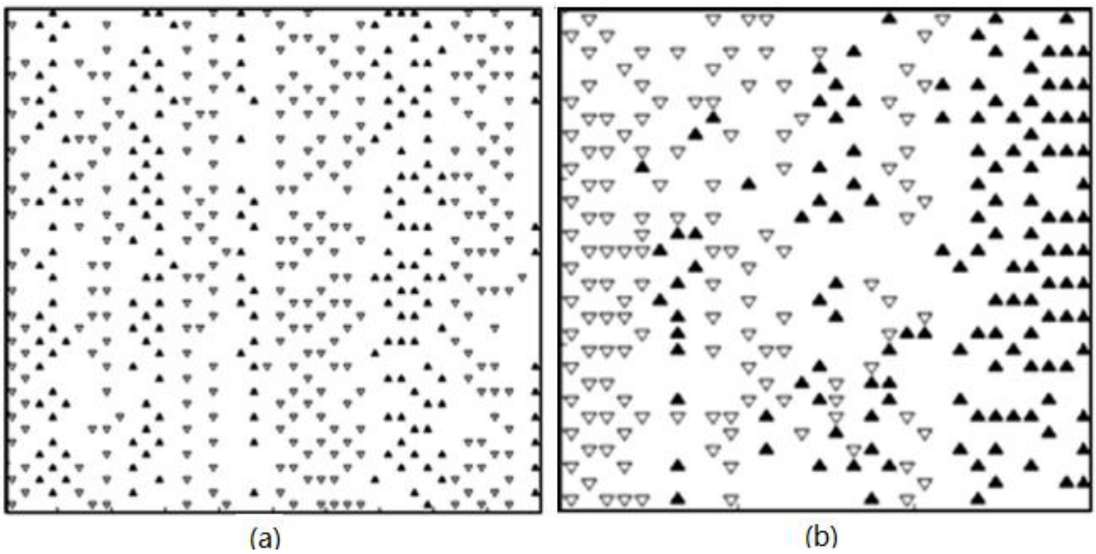


Figure 45 Simulations of bi-directional crowd flow (Yue et al. 2010)

Wang et al. (2012) simulated team-moving behaviour in a bi-directional pedestrian flow as they considered that pedestrians are usually "*in a team-moving state*" in reality. The study was based on the CA model and the pedestrians in the same group would try to maintain the group's formation during the simulation. The simulation demonstrated that groups of teams would create blocks in the crowd (see Figure 46, the pedestrians are moving in the up-down (green triangles) and down-up (red triangles) directions). It has also been pointed out that the jamming caused by team-moving behaviour is related to the teaming manner. Traverse teaming would result in more blocks than the lengthways or the diagonal teaming.

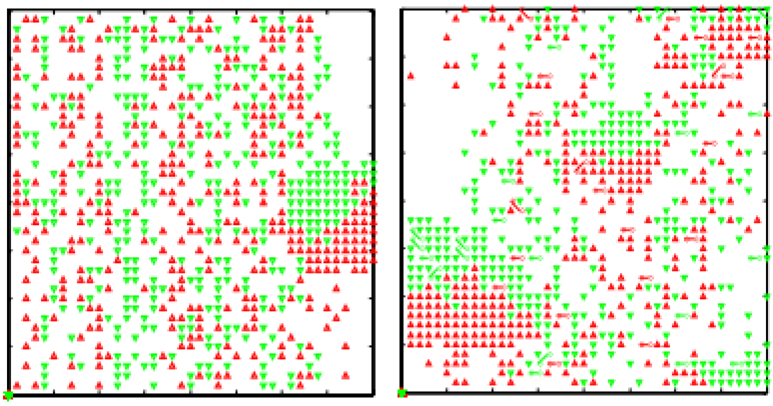


Figure 46 Blockages caused by the team-moving behaviour (Z. Wang et al. 2012)

2. 7 Research Trends and New Requirements Identified from the Literatures

The most popular research field for crowd simulation is for emergency evacuations, for example, evacuation in a building during a fire. The aim of such research (E. D. Kuligowski & R. D. Peacock 2005; Santos & Aguirre 2004; Leggett 2004; E. Kuligowski 2005) is to provide an estimated evacuation time and predictions of crowd movements. However, crowd simulation can also be used to gain a more detailed look at different aspects of a crowd. For example, it can focus on describing the movement of a crowd (Dirk Helbing & Peter Molnar 1995; D Helbing et al. 1997; D. Helbing et al. 2000; Kennedy & Eberhart 1995; Heïgeas et al. 2003; Cheng et al. 2008); it can focus on behaviour modelling (Stern & Richardson 2005; S.R. Musse 2001; Torrens 2007; S.R. Musse & D Thalmann 1997); it can emphasize the effects of different individuals (Nuria Pelechano & Norman I Badler 2006; Braun et al. 2003); it can stress the importance of the intelligence of individuals (Davidsson 2001; Macal & North 2007; Stefania Bandini et al. 2007; Seidel et al. 2008; Stefania Bandini et al. 2009; Bonabeau 2002; Shendarkar et al. 2008; Sung et al. 2004), and it can explore solutions for large-scale crowd simulation (Q. Zhang et al. 2009; J. Yuan et al. 2009).

Several key requirements have been identified in order to improve crowd models: "it is important to consider the physical interactions between individuals and the resulting impact of these interactions in the behaviour of the virtual humans" (Nuria Pelechano & a Malkawi 2008). Further research should consider combining different modelling approaches and increasing the heterogeneity in crowd simulation (X. Zheng et al. 2009). These requirements have been achieved to some extent by previous studies (Dirk Helbing & Peter Molnar 1995; D. Helbing et al. 2000; Nuria Pelechano & Norman I Badler 2006; Stefania Bandini et al. 2007). There is still a lack of crowd models to describe the relationships between behaviours and movement systematically and to integrate crowd heterogeneity into these relationships.

More specifically, three needs have been identified through reviewing the literature:

2. 7. 1 Heterogeneous Crowd

"A crowd is not simply a collection of individuals. The behaviour of an individual may be affected by others in the crowd, which may depend on various physiological, psychological, and social factors. That is, an individual may be forced to behave in a manner that is deemed proper by the crowd in a given situation. Therefore, a crowd may exhibit highly complex dynamics."

- (Zhou et al. 2010)

For most existing crowd models, it is a common approach to treat the crowd as homogeneous. In other words, all the individuals in the crowd obey the same rules or their movements are determined by some global formulas. They have the identical movements and behaviours in the same situations. To design a homogeneous crowd model there are usually three considerations:

- One consideration is the complexity of the modelling and simulation. As studying from the simple to the complex is a common research strategy, it is rational to design a homogeneous crowd model to reduce complexity at the beginning (given the short history of crowd modelling and simulation). The homogenous crowd can be considered as an approximation to a crowd in the real world.

- The second consideration is that this approach can model the resultant group behaviours of a crowd based on social psychology findings. It has been found that individuals could lose their individualities and adapt their behaviours to those of the whole crowd (Soraia Raupp Musse et al. 2005; Heïgeas et al. 2003; Stoot & Stephen Reicher 1998; Villamil et al. n.d.). The simulation of a homogeneous crowd can successfully produce similar crowd behaviours to those social psychology findings.
- The last consideration is the limitation of computer processing power. The simulation of a heterogeneous crowd was constrained in 1990s, but it becomes less and less prominent now as computer technology has developed. Nowadays (since late 2000s), the requirement of powerful computers only needs to be considered in the simulations of extremely large crowds, e.g. of many thousands.

However, as research in crowd simulation has developed, many studies have modelled a crowd from a heterogeneous perspective and have demonstrated that crowds with different compositions have different performances. For example, Pelechano and Badler (2006) introduced the leader role into a crowd and showed that the crowd could have different group patterns. Shendarkar et al.'s study (2008) showed how policemen could affect the choices of escape routes by individuals during fire excavations.

Furthermore, although many crowd theories in social psychological studies have considered that a crowd tends to have homogenous behaviours; individual presences are never ignored by those studies. For example, the classic (contagion) crowd theory (Le Bon 1895) indicated that people in a crowd would tend to think and act in the same way. That means that personalities are decreased within the group but they do not vanish. The convergence theory (Wright 1987) considered that people who wish to act in a certain way come together to form crowds which suggests that a crowd consists of individuals (although in this case the individuals were similar). The emergent-norm theory (Turner & Killian 1957) stated that crowds were composed of people with mixed interests and motives yet behaved as a homogenous crowd as an overall result.

To summarise, crowds are naturally composed of heterogeneous individuals and appear to have homogenous behaviours as a result. Although homogeneous crowd models can simulate the crowd behaviours that are observed in social psychological studies, they cannot reflect the true nature of a crowd's composition. Compared with the homogenous crowd models, the heterogeneous crowd models can represent a crowd more precisely and thus can provide more realistic simulations. Many recent studies on crowd simulations (Schultz et al. 2007; Nuria Pelechano & Norman I Badler 2006; Shendarkar et al. 2008) have demonstrated the different performances of the heterogeneous crowd.

To conclude, in further studies, crowd models should increase crowd heterogeneity in order to close the gap between simulation and reality. This need has also been suggested in a recent survey (X. Zheng et al. 2009) of crowd models.

2. 7. 2 Individual Behaviours

Santos et al (2004) recommended that emergency evacuation simulation models should take more social science into account as most of these models focus on the rules that describe the overall crowd movements. Until now, not many studies have integrated specific individual behaviours into crowd modelling.

Integrating individual behaviours into crowd models can produce complex crowd behaviour thus achieving more accurate simulation results. Some studies have demonstrated the differences shown in crowd behaviours when integrating individual behaviours. For example, Braun et al. (2003) showed how grouping behaviour (a number of individuals tending to move together as a sub-group) affected a crowd's overall movement speed. The findings (Figure 47) revealed that the average speed of the crowd was not affected by the willingness to maintain sub-groups if the requirements were fixed while the average speed would decrease if more individuals were required to move as sub-groups with a fixed willingness. Pelechano and Badler (2006) modelled the communication behaviour in the evacuation from a complex structured building. The results (Figure 48) suggested that communications (the exchange of information on the building) between individuals can help them choose a better escape route and thus increase evacuation efficiency. The AL values indicate the willingness (in probability) of individuals to walk in a sub-group. The DE values indicate the requirement (in probability) of individuals to walk in a sub-group. For the graph at the Left-hand side of Figure 48, DE is fixed at 0.5. For the graph at the right-hand side of Figure 48, AL is fixed at 0.5.

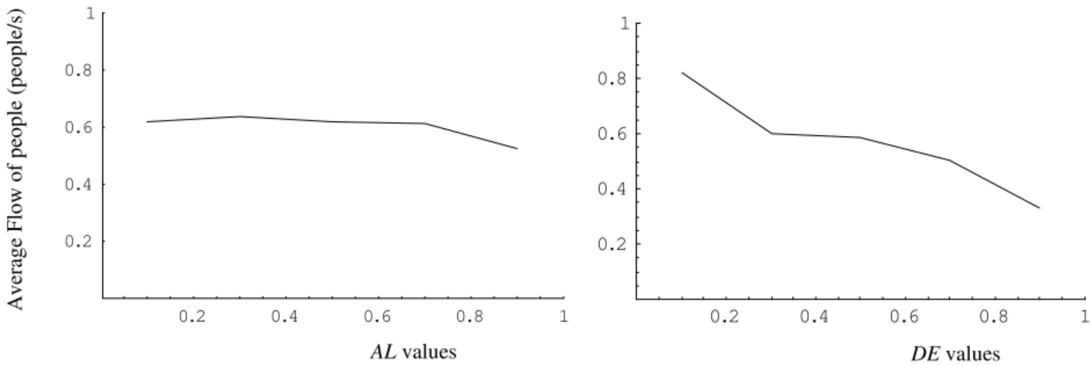


Figure 47 The influence of grouping behaviour on crowd average speed (Braun et al. 2003)

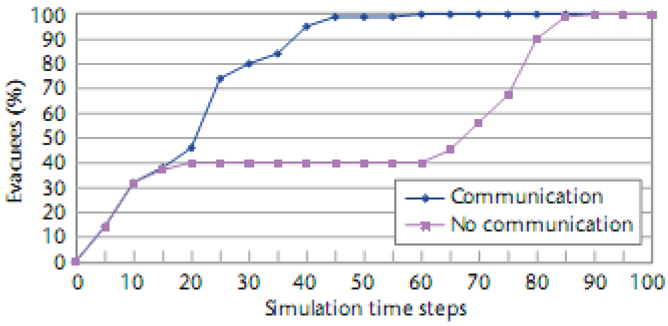


Figure 48 Evacuation with/without communication (Nuria Pelechano & Norman I Badler 2006)

To model more individual behaviours in a crowd can also increase crowd heterogeneity within the model as individuals have more behaviours available to choose from or may have multiple behaviours at the same time.

2. 7. 3 Generic Crowd Modelling

According to the literature, the studies of crowd modelling usually focus on some specific scenarios. The majority of the crowd simulations are related to panic or chaos situations (such as fire/emergency evacuation). There are also a large number of studies on the walking behaviour of pedestrians in a counter-flow scenario. These crowd models have been specially designed and fine-tuned to represent relevant crowd behaviours in targeted scenarios/situations. This approach to designing a crowd model has its own advantages as the social psychological issues can be preliminarily integrated into the crowd model and then carefully calibrated to produce the optimal simulation results.

However, if it were considered to extend such model applications into a border situation or into other scenarios, these scenarios specialised crowd models might suffer from several issues:

- Because social psychological issues have been integrated into the crowd movement mechanism (e.g. formulas, rules), it could become very difficult to use one calibrated crowd model configuration to represent the different scenarios.
- As the crowd is likely to have different behaviours in different scenarios, to represent new crowd behaviours or movement may require additional formulas or rules. As these extensions may not fit the original design, the modifications and supplements will increase the complexity of the crowd model.
- If new crowd behaviour and movement mechanisms need to be introduced, the further development is equal to the study of creating a new model. Such an inconsistency mechanism can also increase the work required for crowd model validation.

To cope with the above issues in the further development of an existing crowd model, one possible solution is to design a generic crowd model and then configure the crowd heterogeneity and the influences of the social psychological issues. If a unified crowd behaviour and movement mechanism has been established in a generic crowd model, future modifications and supplements can follow the same mechanism. In addition, the validation work of the extended crowd model is also easier because the foundation of the crowd model has not been changed.

To summarise, although the requirement to create a generic crowd model has not been explicitly proposed by the existing studies yet, a generic crowd model offers a more comprehensive and flexible approach when considering extending the application of a crowd model in advance.

2. 7. 4 Summary of the Research Needs

This section has summarised the requirements for the further development of crowd modelling through the existing studies. In addition, a potential and more flexible crowd model designing approach has been proposed. To conclude, three research needs have been identified:

- Further studies on crowd modelling should reflect heterogeneity in the crowd. Individuals can have different attributes, behaviour preferences, etc.
- More individual behaviours should be modelled in crowd simulations. The individuals could have personalised behaviours during movement in order to create variations in the crowd collective behaviours.
- For the future extensions on the crowd model and for consistency on crowd behaviour representation, the design of a generic crowd model should be considered.

2. 8 Summary of the Chapter

This chapter reviews the studies of crowd modelling and crowd simulation. It firstly provides descriptions of, and terminologies for, crowd modelling and simulation. Then it introduces the categorisation of crowd models and discusses the modelling approaches used in current crowd modelling studies. It critically reviews relevant crowd models using these modelling approaches. This research study particularly focuses on micro scope crowd models. Furthermore, this chapter also introduces the applications of these crowd models and correspondent crowd simulations in the studies of crowd behaviours and building layout evaluations.

Chapter 3 LITERATURE REVIEW: TECHNOLOGIES FOR CROWD SIMULATION

Carrying out a crowd simulation requires the implementation of a theoretical crowd model. This chapter provides a review on the existed and existing technologies that have been used or can be used to implement the crowd model via the computer simulation approach.

3. 1 Overview

Briefly speaking, the implementation of a crowd model can be achieved in several ways:

- The most common practice is to develop the simulation software application (simulation system / simulation environment) from scratch. This approach provides the most flexibility in implementation but it requires very good knowledge and skills in programming. It has been used in a large number of crowd modelling studies (Dirk Helbing & Peter Molnar 1995; Heïgeas et al. 2003; Kirchner & Schadschneider 2002; Qiu & Hu 2010).
- Another approach is to use existing simulation packages (These packages (e.g. UDK, Quest 3D) normally include a graphic engine (usually specially designed for crowd simulation). Some packages (e.g. A.I.implant) even have artificial intelligence support). This approach can save work in implementing a basic simulation environment but the features of the crowd model may not be fully represented through the existing packages or simulation environments.
- The most convenient way to undertake a crowd simulation is to use crowd simulation software. Such software has integrated a crowd model into the simulation system. Although most parts of crowd models are fix-designed in such software, some software provide the features to modify the basic crowd parameters or the environmental structures.

A comparison of these three approaches is presented in the table below:

Table 4 Summary of crowd simulation approaches

	Approach 1	Approach 2	Approach 3
Descriptions	Create the simulation system from scratch to implement a crowd model through programming	Implement a crowd model via existing simulation packages.	Use crowd simulation software.
Programming skills' requirements	Requires primary programming skills: the knowledge of system design, and the knowledge to create graphic engines.	Requires primary programming skills: the knowledge of system design, and the knowledge to use the existing graphic engines.	Does not require programming skills. Only needs the relevant knowledge to use the simulation software.
Implementation Workload	Heavy. Need to implement the crowd model and create a simulation environment to suit the model.	Medium. Need to implement the crowd model and adjust the existing simulation environment to suit the model.	None, because the simulation software is the result of the implementation.
Representation of the crowd model	The simulation system can work exactly as the crowd mode being designed.	The representation depends on the features that the simulation environments provided.	The simulation software is implemented to represent the pre-designed crowd model.
Flexibility & Extendibility	Everything can be changed and updated in further research.	The further development of the crowd model may be limited by the simulation environment.	Updating the model and further development cannot be undertaken by the users of the software.

As the implementations of the crowd simulation system via approach 1 is totally dependent on individual studies, this section only presented the reviews of existing simulation software and some available simulation packages.

3. 2 Crowd Simulation Software

3. 2. 1 Surveys on existing Crowd Simulation Software

Many crowd models have been implemented into software to simulate emergency evacuations. Such software can be used to observe crowd movements and behaviours in different environments with pre-defined individuals. Several reviews (Santos & Aguirre 2004; E. Kuligowski 2005; E. D. Kuligowski & R. D. Peacock 2005) have been conducted to evaluate and demonstrate the purposes, modelling approaches and applicable fields of existing or previously existing crowd simulation software. For example, Santos and Augirre (2004) presented a survey on emergency evacuation simulation models (e.g ECACNET4, EESCAPE, EgressPro, the Magnetic model, EGRESS, SIMULEX, EXIT89, GirdFLOW, ALLSAFE, EXODUS, BFRIES, FIRESCAP, etc) by briefly analysing the strengths and limitations of those models. Kuligowski and Peacok (2005) published a more comprehensive review which included thirty crowd evacuation models.

(Note: In this section, the review of existing crowd models are primarily based on the study of Kuligowski and Peacok (2005) because the most of these crowd models are not available to the author of this PhD thesis.)

Kuligowski and Peacok (2005) summarized the features of existing crowd evacuation simulation software. This study reviewed this software through eleven categories. It can be used as a comprehensive guidance on how to select suitable simulation software to meet one's requirements. The reviewed software is represented as follows (detailed explanations are presented in the next section):

Table 5 Features of crowd simulation software (E. D. Kuligowski & R. D. Peacock 2005)

Model	Available to public	Modeling Method	Purpose	Grid/ Structure	Perspective of M/O	Behavior	Movement	Fire data	CAD	Visual	Valid
FPETool	Y	M	1	N/A	G	N	UC	N	N	N	N
EVACNET4	Y	M-O	1	C	G	N	UC	N	N	N	FD
TIMTEX	Y	M	4	C	G/I	N	D	N	N	N	PE
WAYOUT	Y	M	5	C	G	N	D	N	N	2-D	FD
STEPS	Y	M/PB	1	F	I	N/I	P, E	N	Y	3-D	C
PedGo	Y	M/PB	1	F	I	I	P,E (CA)	N	Y	2-D	FD
PED/PAX	Y/N3	PB	3	C	G	I	D	N	Y	2,3-D	N
Simulex	Y	PB	1	Co.	I	I	ID	N	Y	2-D	FD,PE
GridFlow	Y	PB	1	Co.	I	I	D	N	Y	2,3-D	FD, PE
ASERI	Y	B-RA	1	Co.	I	R/C, P	ID	Y1,2	N, F	2,3-D	FD
BldEXO	Y	B	1	F	I	R/C, P	P, E	Y1,2	Y	2,3-D	FD
EXITT	Y	B	2	C	I	R/C	C	Y1,2	N	2-D	N
Legion	Y	B	1	Co.	I	AI	D,C	Y2	Y	2,3-D	FD,OM
PathFinder	N1	M	1	F	I/G	N	D	N	Y	2-D	N
EESCAPE	N1	M	5	C	G	N	D	N	N	N	FD
Myriad	N1	M	1	N/A	I	N	D	N	Y	2-D	3P
ALLSAFE	N1	PB	5	C	G	I	Un F	Y1,2	N	2-D	OM
CRISP	N1	B-RA	1	F	I	R/C, P	E,D	Y3	Y	2,3-D	FD
EGRESS 2002	N1	B	1	F	I	R/C, P	P,D (CA)	Y2	N	2-D	FD
SGEM	N2	M/PB	1	F	I	N/I	E,D (CA)	N	Y	2-D	FD, OM
Egress Complexity	N2	M/PB	5	C	G/I	N	Ac K, FA	N	N	N	OM
EXIT89	N2	PB	1	C	I	I/C(smk)	D	Y1	N	N	FD
BGRAF	N2	B	1	F	I	R/C, P	UC?	Y1,2	N, F	2-D?	FD
EvacSim	N2	B	1	F	I	R/C, P	D	Y2	N	N	N
Takahashi's Fluid	N3	M-O	1	C	G	N	FA-D	N	N	2-D	FD
EgressPro	N3	M	5	C	G	N	D	Y2	N	N	N
BFIRES- 2	N3/U	B-RA	4	F	I	R/C, P	UC	Y2	N	N	N
VEgAS	N3/U	B	1	F	I	AI	ID	Y1?	Y	3-D	N
Magnetic Model	U	M	1	F	I	I	FA	N	N	2-D	N
E-SCAPE	U	B	1	C	I	R/C, P	OML	Y2	N	2-D	N

3. 2. 1. 1 Explanations of the labels in the table

The crowd evacuation software was reviewed in eleven categories:

Availability to the Public

This category describes the status of a specific simulation software, i.e. how it can be accessed.

- **Y**: This model is available to public, either free or at a charge.
- **N1**: This model has been used by a company on a consultancy basis.
- **N2**: This model has not been released to the public yet.
- **N3**: This model is no longer in use.
- **U**: The status of the model is unknown.

Modelling Methods

This category indicates how the crowd model calculates the evacuation times (i.e. what issues have been considered).

- **B**: "behaviour models". This type of model has incorporated individuals' behaviours onto their movements and is able to present the decision-making process.

- **B-RA**: Indicates that the model also has risk assessment capabilities.
- **M**: "movement models". This type of model describes the crowd purely based on their movements.
- **M-O**: Indicates that the model is designed to optimise the evacuation times.
- **PB**: "partial behaviour models". This type of model focuses on the movement of the occupants but also considers the effects of behaviours.

Purpose

This category explains the target simulation scenario of the software.

- **1**: A model for all types of buildings.
- **2**: A model that specialises in residences.
- **3**: A model that specialises in public transport stations.
- **4**: A model designed for low-rise buildings (under 22.9 metres) only.
- **5**: A model that is only capable of simulating 1 route / 1 exit buildings.

Grid/Structure

This category is about the representation of the field in the software.

- **F**: The fine network divides the floors into a number of small cells which can be occupied by individuals *(Note: The same presentation as is used in the CA models)*.
- **C**: The coarse network divides the floor into rooms, corridors, etc. and the individuals move from one place to another *(Note: The same approach is used in the fluid dynamic models).*
- **Co**: The continuous network utilises a 2D Cartesian coordinate system to represent the crowd position.

Perspective of the Model/Occupant

This category explains how the simulation software monitors the individuals and how the individuals view the simulation environment.

- **G**: No individual's details are presented during the simulation and the crowd possess all the information on the environment.

- **I**: Individuals are represented separately during the simulation and an individual's behaviours and movement are based on its own knowledge of the environment.
- **I/G**: Individuals are represented separately during the simulation and they possess all the information on the environment.
- **G/I**: No individual's details are presented during the simulation but the crowd movement is dependent on the crowd member locations.

Behaviour

This category explains how behaviours are modelled in the software.

- **N**: No behaviour. This model only represents crowd movements.
- **I**: Implicit behaviour. This model assigns behaviours implicitly in the crowd to affect the movement.
- **C**: Conditional behaviour (rule). This model designs behaviours which are affected by structural or environmental conditions.
- **AI**: Artificial Intelligence. This model aims to simulate human intelligence during an evacuation.
- **P**: Probabilistic. This model represents behaviours through a stochastic approach.

Movement

This category describes how the crowd model decides the movements of the occupants during an evacuation.

- **D**: Density correlation. In the model, the speed of the individuals in a place is decided by the crowd density in that place.
- **UC**: User's choice. The users of the software decide the speed and density values in the buildings before the simulation.
- **ID**: Inter-person distance. Each individual has a minimum distance from others, obstacles, etc.

- **P**: Potential. If the field was represented by cells, each cell is given a value (also known as potential) to guide the movement of the occupants. i.e.. the occupants will always move to the cells with lower values.
- **E**: Emptiness of the next cell. In this model, the occupant moves to an empty cell.
- **C**: Conditional. This type of model moves the crowd depending on the conditions of the environment, the structure, etc.
- **FA**: Functional analogy. The occupants' movements are calculated through equations in this type of model.
- **OML**: Other model link. The movement of the occupant is calculated through another model. This simulation software only represents the animation of the movement.
- **Ac K**: Acquiring knowledge. This type of model considers the movements of a crowd are dependent on their knowledge of the environment instead of being determined by movement algorithms.
- **Un F**: Unimpeded flow. This type of model calculates the evacuation time by combining the evacuation time taken in unimpeded conditions and the delay or improvements' time
- **CA**: Cellular automata. This is a cellular automata model.

Fire Data

This category indicates whether the model can simulate the effect of fire during the simulation.

- **Y1**: The fire data needs to be imported from another model.
- **Y2**: Users can input specific fire data at certain times during the evacuation simulation.
- **Y3**: The model has built-in fire data.
- **N**: The simulation of fire is not supported in this model.

CAD Support

This category indicates whether the model supports CAD models as the simulation environment (i.e. represents the CAD layout structure).

- **Y**: CAD models are supported.
- **N**: CAD models are not supported.
- **F**: CAD model imported features are under development.

Visual

This category indicates the type of visualisation in the simulation software.

- **2-D**: The simulation is represented in a 2-D environment.
- **3-D**: The simulation is represented in a 3-D environment.
- **N:** This software does not have any visualisation.

Validation

This category explains how the crowd model is validated.

- **C**: The simulation results are validated through code requirements.
- **FD**: The simulation results are validated against fire drills or experiments.
- **PE**: The simulation results are validated through the literature.
- **OM**: The simulation results are validated against other models.
- **3P**: The simulation results are validated by a third party.
- **N**: No validations have been provided.

3. 2. 2 Discussions on Crowd Simulation Software

Using existing crowd simulation software usually does not require any special knowledge of software design or development. Thus they are very convenient in the evaluation of the layout of buildings.

However, in a crowd model that has already built-in simulation software, crowd behaviours and the movement algorithms cannot be changed. Although some software (e.g. EXODUS) provides the flexibility to adjust some parameters (e.g. speed) of the crowd and the environment (e.g. smoke), the theory fundamentals are not modifiable, which prevent adjusting the crowd model to suit specific situations or presenting different crowd compositions (e.g. including the roles of fire fighters, disabled people). Furthermore, much evacuation software do not usually expose the model's details to the public, which means that these models cannot be improved or modified except by the companies or research groups who release them.

Although majority of crowd evacuation simulation software have been validated and can provide reliable results for a designed simulation environment, crowd behaviours and performance in new environments many need fine-tuning in order to represent environmental effects as much software has the feature of importing structural layouts from CAD models. However, it is not possible to adjust the crowd/individual behaviours in these built-in crowd models so the simulation results may not be accurate if the new environments incorporate already known influences that will affect the behaviour of the crowd. d

As a conclusion, crowd simulation software can be very useful for social psychologists or emergency services to study well-modelled crowd behaviour in specific environments. However, they are not considered suitable for the studies of crowd modelling, as these studies usually require adjustment of the crowd model itself.

3. 3 Crowd Simulation Packages

3. 3. 1 Reviews of Crowd Simulation Packages

In this study, a crowd simulation package is defined as a software development kit or preliminary simulation software (a graphical engine or a simulation environment) for the implementation of a crowd model. The following simulation packages have been evaluated: AI Implant, Quest 3D, UDK (Unreal Development Kit) and Microsoft XNA framework

3. 3. 1. 1 AI.implant (Version 5.4)

Introduction

AI.implant is a commercial software solution (Presagis USA Inc. 2009) for real-time simulation which features artificial intelligence support. It models intelligent movement and behavioural manifestations of humans for a simulation and supports dynamic navigation mesh (automatic way-finding). AI.implant provides a built-in editor to configure the virtual environment and the intelligent agents for the simulation. It also support to input building plans from Auto CAD and 3ds Max.

Evaluation

- **Testing Simulations**

The author has created some testing scenarios to evaluate this software. The version was 5.4 when such evaluation happened in 2009.

The first scenario was to test the automatic way-finding function in AI.implant. As it demonstrated in Figure 49, an agent has been located in the right-hand side of the building and its destination is set at the left-hand side room (the arrow pointed position). The line indicates the automatic generated route. As a main feature of AI.implant, it can analyse the layout of a building and generate a navigation mesh for it. This simulation shows such function works very well in all the tests of different starting positions and destinations.

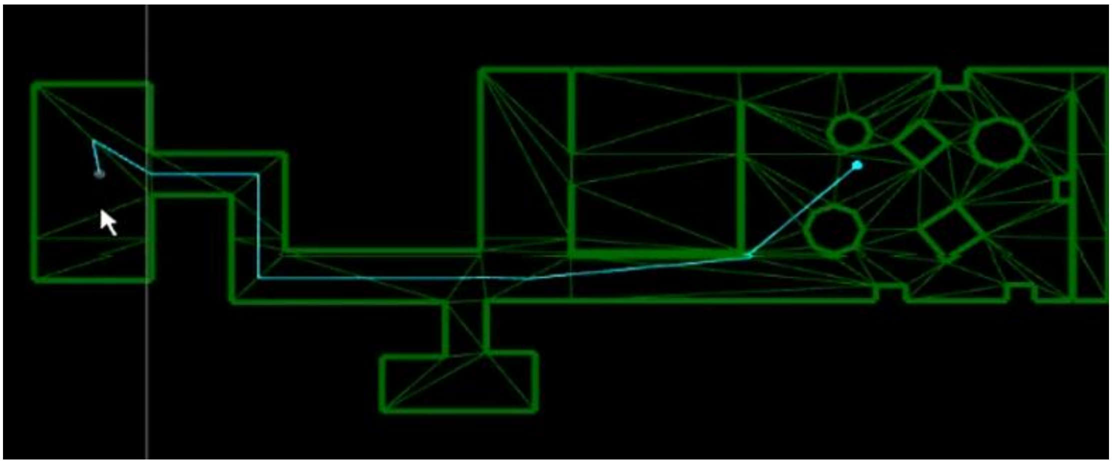


Figure 49 Automatic way-finding in AI.implant 5.4

AI.implant also supports user-defined navigation meshes. Figure 50 shows another scenario in a big room. The meshes formed by the blue lines are the user-defined meshes and the white circles (look like ⌖) represent the way-points. The agents will use these routes (have to enter and exit the routes via the way-points) whenever possible. The white dot in the left-hand side represents an agent and its destination is located at the right-hand side. The red line indicates the automatically calculated route by AI.implant. In the tests of different starting positions and destinations, the routes can be very cleverly generated by AI.implant.

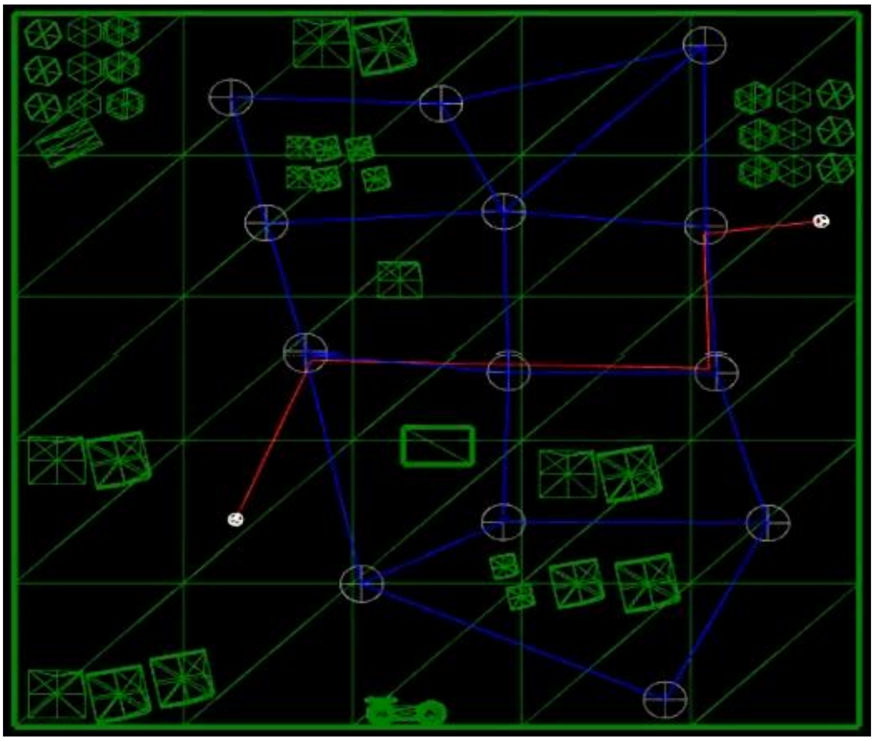


Figure 50 User-defined navigation meshes in AI.implant

➢ Discussion

In the tests with different building models, AI.implant has demonstrated its strength to recognise the layout of a building and can automatically provide navigation to the agents. In some other tests of collision avoidance, the built-in AI performs very well. The authors tried to import several building plans into the AI.implant's build-in editor, all of them were recognised accurately.

However, AI.implant does not provide many options in configuring the agents' behaviours in its built-in editor. It requires one to use its SDK to implement such functions. At the moment of testing, the author was unable to do so due to lack of special knowledge of AI.implant's SDK.

(Note. At the last time of 08/13 when the author visited Presagis USA Inc.'s official website, AI.implant was removed from its products.)

3. 3. 1. 2 Quest 3D (version 4.3)

Introduction

Quest3D is a tool for producing real-time 3D multimedia productions. (e.g. 3D Virtual Reality, simulators, etc.) (Quest 3D 2009). Quest3D itself does not create any materials such pictures, 3D meshes and sounds for a simulation. These materials need to be created in external programmes and imported into Quest 3D. Then Quest 3D can assembles these materials into interactive experiences.

The Quest3D editor is provided to complete the assembling tasks and is divided in three sections: Channels, Object and Animation.

- The channels section draws the logic and dependencies of the virtual world.
- The object section defines the looks of your 3d objects.
- The animation section defines animating motions and values. Additionally, it also preview of the end result.

Evaluation

➢ **Creating 3D testing scenarios**

By using the Quest3D editor, the author has created three different virtual 3D environments:

- 3D crowd rendering (see Figure 51): many agents are display with different textures.
- A maze-like environment (see Figure 52): an agent walks in a maze.
- A street scenario (see Figure 53): an agent walks on the street.

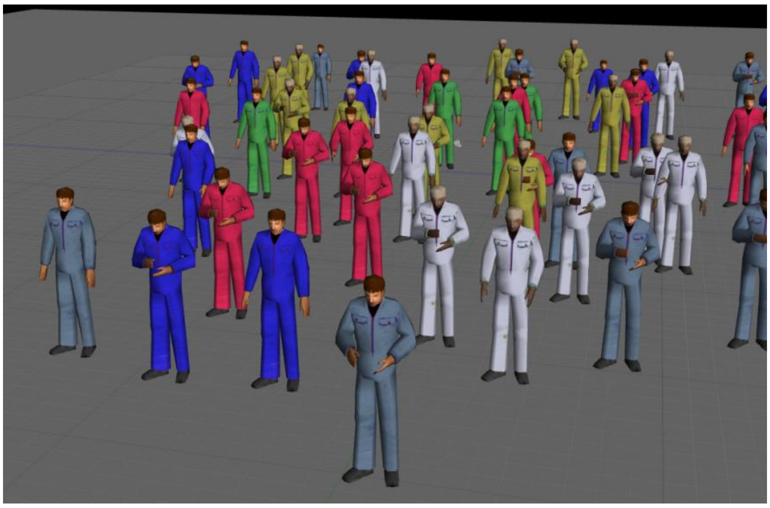


Figure 51 Crowd rendering in Quest 3D (v4.3)

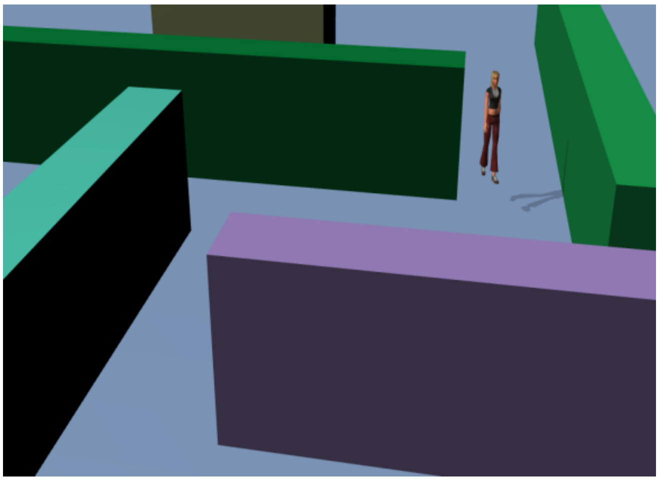


Figure 52 Simulation in a maze-like environment

Figure 53 Simulation in a street environment

In Quest 3D, the navigation need to explicitly defined by the users. The movements of agents are controlled via graphical programming (see Figure 54) within the Quest3D Editor.

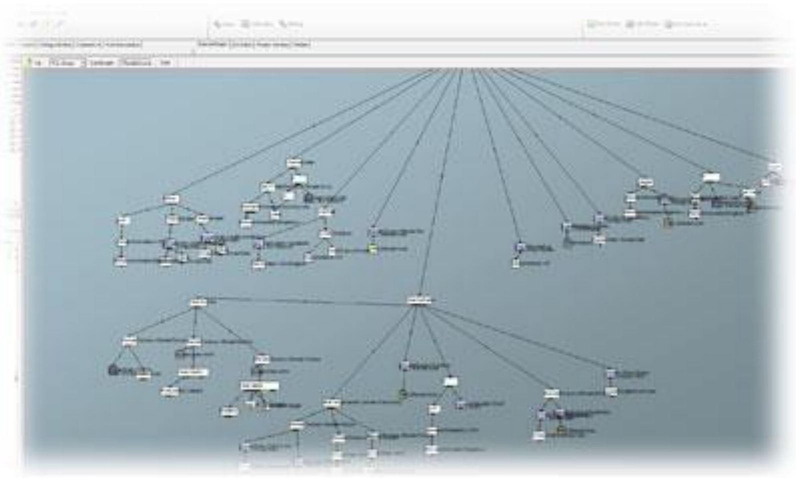


Figure 54 Graphical programming in Quest 3D (v4.3)

- **Discussion**

Through the evaluation of Quest 3D (v4.3), the author has found that it is a very convenient tool in creating 3D animations or 3D demonstrations. The Quest3D editor provides efficient management of the imported materials. The feature of graphical programming enables non-programmers to define behaviours of the 3D models (e.g. can be the movement of a person). However, as a programmer, the author considers this graphical programming is not a good choice for developing complex behaviours of intelligent agents. Mainly, the reason is the structure of the programming tree becomes very complex when adding new behaviours. In Figure 54, only some simple movement paths are defined and the structure already looks bulky. If an advanced agent model was defined and complex behaviours were defined, such structures will have very poor readability which will increase the difficulty in further development as well.

3. 3. 1. 3 UnReal Engine 3 (UDK version)

Introduction

Unreal Engine (UE) 3 is a comprehensive and leading development framework in the game industry for creating stunning and complex 3D games, which has been developed and licensed by Epic Games (2009). Some well-known games utilised UE 3 are Gear of Wars, Mass Effects, and Unreal Tournament. In November 2009, Epic Games released a UE 3 version named Unreal Developer Kit (UDK) which is totally free for non-commercial user. The author considers crowd simulation, in a sense, has the same nature like a self-running computer game as it consists of NPC in a virtual environment. Therefore, an evaluation on employing UDK to implement the crowd model in this study has been carried out.

- **Virtual Environment Build Support**

UDK provides a built-in editor (UnrealEd) to create the virtual environment for simulation. The UnrealEd provides the following tools or functions: Terrain Editor, Material Editor, Mesh Editor, Animation Editor, Foliage Editor, Unreal PhAT, Unreal Cascade, Unreal Matinee, Unreal Kismet, UI Editor, Sound Cue Editor, Post-process editor, Unreal Content Browser, Scene Manager, Reference Graph viewer.

- **AI and Navigation Support**

The AI system in UE3 provides two ways of navigating the AI characters. One is that the UnrealEd can automatically generate navigation mesh from a given virtual world. Another is to assign a route network which is node-based to the virtual world manually. Then the AI characters will calculate the optimal route and take action. Additionally, the UDK also support a large number of crowd animations through its flocking technology.

Evaluation

- **Building a simulation prototype**

The author successfully created a simulation prototype showing an empty environment where a crowd move by following a pre-defined navigation mesh. This simple crowd representation is showed in Figure 55 (because this simulation was created on top of the default First-person shooter game mod in UDK, a player holding a gun is displayed).

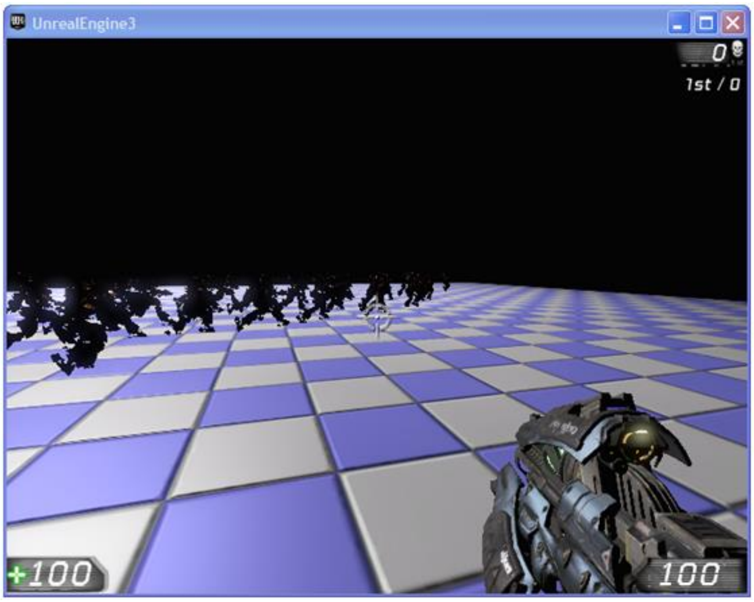


Figure 55 Simulate crowd movement in UDK

Figure 56 illustrates the virtual environment setup for the simulation. The icons with a person in a green background are the points that the agents enter/exit the simulation environment. The mesh (consists of different colours of lines) is the navigation map for the agents. Several walls are placed in the environment as well. One wall on the left-hand side has been placed on the routes (red lines) of the agents. During the simulation, it can be observed that the built-in AI can detect the collision with the wall and calculate a walk around route.

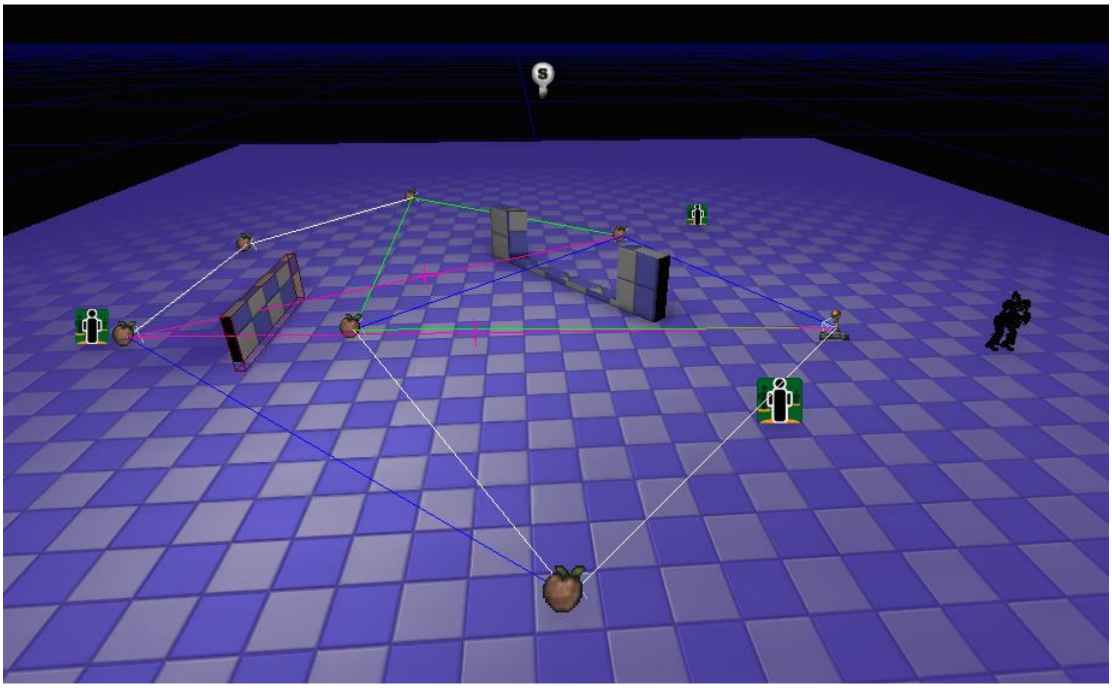


Figure 56 Environment setup for a simulation with UDK

The workflow of the simulation and the control of the crowd agents are configured through scripts in Kismet, which is demonstrated in Figure 57.

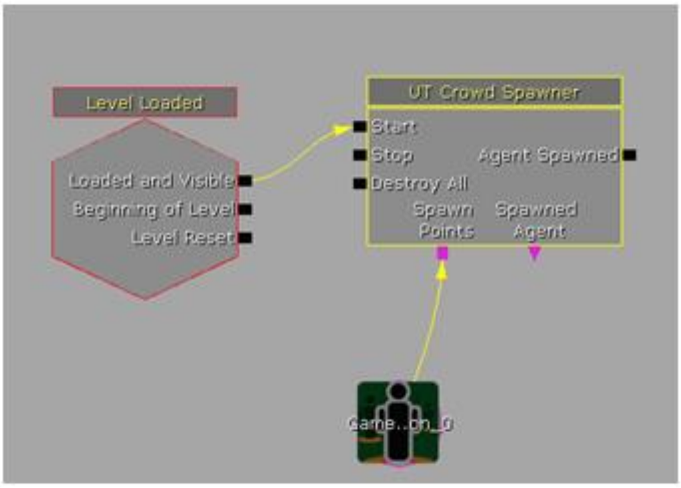


Figure 57 Script to generate crowd via Kismet in UDK

- **Discussion**

To create an initial crowd simulation environment with UDK was quite simple and the built-in AI also had clever performance. However, the author was unable extend the existing agents model in UDK because the extremely lack of documentation on behaviour scripting or programming in UDK. At the time (11/2009 - 12/2009) when the author evaluated the UDK, there were only some basic introductions and the further documents required a $2500 standard license fee (there was no budget for the author to purchase a license for the evaluation purpose).

To sum up, UDK has been found to be a complete and mature framework to develop a game. However, it requires very specific knowledge of Unreal Engine.

3. 3. 1. 4 Microsoft XNA Framework 4.0

(Note. When the author evaluated the Microsoft XNA Framework at early 2010, it was at version 3.0 and the 4.0 version was released later. Because the XNA framework has been selected for the crowd model implementation in this study and the latest 4.0 version was used later on, this evaluation will be based on version 4.0.)

(Further Note. In early 2013, Microsoft has decided to bring an end to XNA development (*Microsoft-News.com* 2013; *The Escapist* 2013)*. Alternatively, Microsoft divided the relative game development into Windows Phone Apps and Xbox Live Indie*

Game. There is no Microsoft official XNA framework website anymore, although relevant downloads and documents still exist in Microsoft MSDN.)

Introduction

Microsoft XNA (XNA is not an acronym) is a managed runtime environment for video game development. The first version of the XNA toolset was announced on 24 March 2004 and version 4.0 was released on 16 September 2010. The XNA framework is implemented based on Microsoft .Net Framework (v2.0 and later v3.5) and it comes with an integrated development environment (IDE) - XNA Game Studio.

XNA framework provides a built-in game engine and automatically handles the game lifecycle. Generally speaking, by using XNA framework, a developer only needs to handle two aspects during the game development, i.e. game logic and graphical representation.

A game developed via XNA framework can be ran on Windows OS platform (XP, Vista, and 7), Windows Phone platform, and XBOX 360 platform. In this study, only the Windows OS platform will be evaluated.

Evaluation

➢ Create an simple XNA game

By using XNA Game Studio (which is integrated into Microsoft Visual Studio), one can create an empty game (see Figure 58) without any effort. This empty game does nothing but refreshes the graphical representation at a frame rate of 60 FPS and updates the game logic. Because no game logic and display has been defined yet, this running game only shows a blank window.

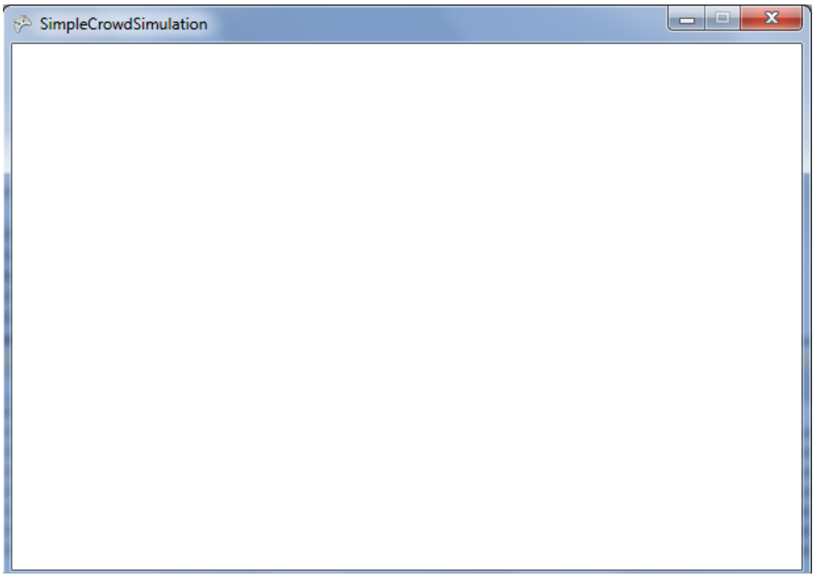


Figure 58 An empty XNA game

The next step is to add some agents into the game, define their movement logic, and display them in the simulation window. According the XNA game lifecycle, the agents are created at the initialisation stage of the XNA engine. Their movement logic is handle in the Update() method and their graphical representation is defined in the Draw() method.

For the evaluation purpose, a scenario that shows agents moving randomly has been implemented. The snapshot of the simulation is showed in Figure 59. The agents are presented by circles with small dots indicating their orientation. Additionally, some text is displayed on the top of the screen to show relevant information at real-time. Furthered, the author has found the data (e.g. simulation information, agents' position, etc.) generated during the simulation can be easily export to a Text file or an Excel file for further analysis.

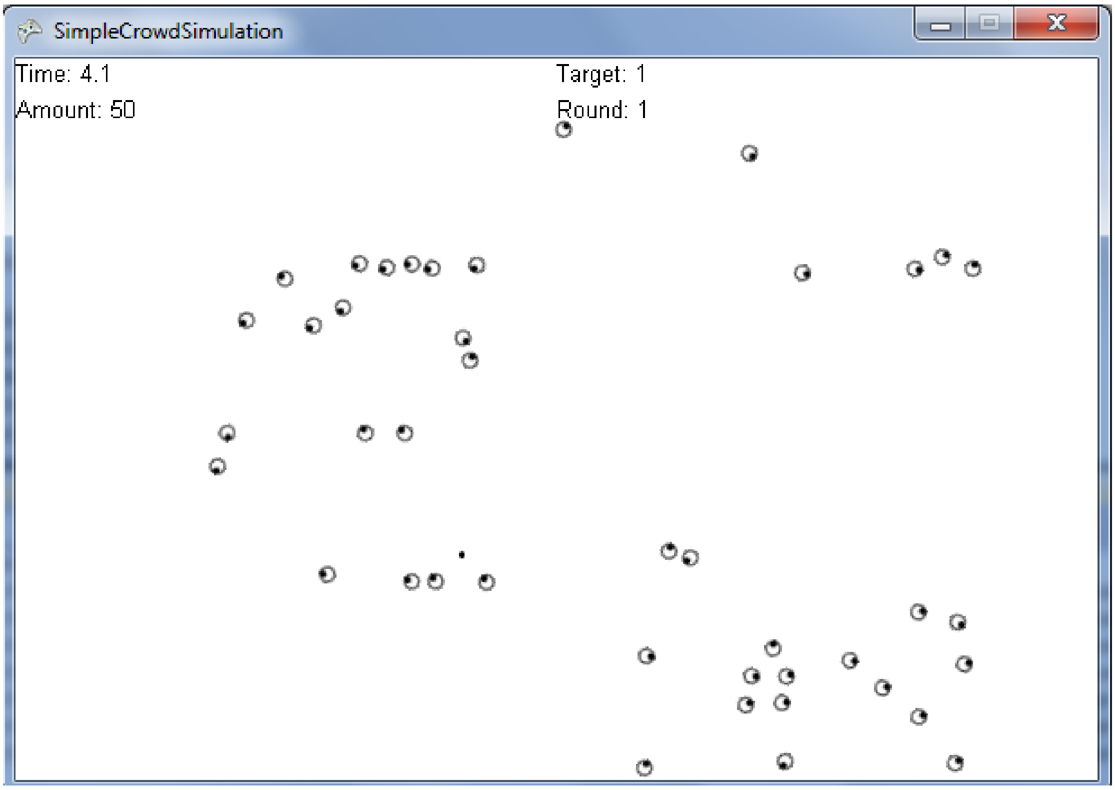


Figure 59 A simple XNA game showing agents move randomly

➢ **Discussion**

Through the development of a simple crowd simulation prototype, XNA framework has demonstrated its simplicity in implementing a crowd simulation system. Because it provides a programming environment for developments, the logic of agents' behaviours and their movement can be easily handled. The XNA game lifecycle also supports the integration of high level agent models. However, the graphical representation in an XNA game needs to be handled entirely by developers, with which the author could only manage to demonstrate the simulation in very basic graphics.

3. 3. 2 Discussions on Simulation Packages

3. 3. 2. 1 Comparisons

Compared to simulation software, the crowd simulation packages can offer more flexibility for configuring an individual. Considering the aim of this study is to develop and implement a new crowd model, the author believes employing a simulation package is a more suitable choice for this PhD study.

The reviewed simulation packages in this study are compared below:

Table 6 Comparisons of crowd simulation packages

Name	Developing Environment	Graphics	AI and agent modelling support
Quest 3D	built-in editor	built-in 3D representation, supports advanced textures	no built-in AI, behaviours need to be defined via graphical programming, separate models are not supported.
AI.implant	built-in editor	built-in 2D simulation	built-in AI and agent, no additional agent models
UDK	built-in editor, scripting	built-in 3D simulation, supports advanced textures	has built-in AI and agent, additional agent model may achieve via scripting
XNA	Programming	provides graphical representation support but users need to manually draw the graphics	no AI, additional agent models can be added into the game lifecycle

In this study, the crowd model to be designed will focus on present individuals'' movement and how their behaviours affect their movements. It requires a simulation package which can control the individuals'' movements at a low level and a package which supports integration of additional agent models or other high level AI. Graphics of the crowd simulation is considered as a mean to present the crowd's and the individuals' movement and behaviours, which does not requires 3D display (however, a better visualisation in crowd simulation is always preferred if available).

3. 3. 2. 2 This study's choice

As a result, Microsoft XNA framework was considered the most appropriate package to develop and implement the crowd model in this study. Such a decision does not imply that Microsoft XNA framework is the best solution for crowd simulation but it presents a choice to make the best use of the author's expertise in programming in order to accomplish this study's aim and objectives.

3. 4 Navigations in Crowd Simulation

3. 4. 1 Introduction

Navigation, also called way finding, is the implementation of how individuals find their way to their decided destinations in the simulation environment. It is a necessary and important part of crowd simulation. Depending on the scenario of the simulation, the navigation can be either simple or complicated. For example, in the scenario of leaving from a one-exit room, the navigation is simple and it only needs to set the desired walking direction of all the individuals to the direction of the exit. In contrast, navigation could become more complicated in a more complex environment. For example, in the scenario of an evacuation from a shopping mall, the navigation may consist of a mathematical representation of the environmental geometry structure (known as the navigation map) and the method to calculate the possible routes to the destination.

In the following sections, it firstly presented different points to view for viewing navigation. Then it introduces some popular navigation methods in crowd simulations.

3. 4. 2 Navigation from Different Points of View

3. 4. 2. 1 The Scope of Navigation: Global or Local

Navigation is required at both the "**global level**" and the "**local level**" during crowd simulation. Navigation at the global level is required in a complex environment (e.g. a shopping mall). Navigation usually consists of a navigation map and an algorithm of how to select a route. For example, the passages and shops in a shopping mall can be represented by a note-based network and the Dijkstra's shortest path algorithm (E. W. Dijkstra 1959) can be applied to calculate the path. In this case, the route to the destination is usually constrained by the environmental geometry structure.

Navigation at local level deals with the movement of the crowd/individual in a relatively small area (e.g. a room or a corridor). It usually contains the information on a direct route to the desired exit in the area. Furthermore, this level of navigation is usually integrated with crowd models (e.g. the Social Force model (D. Helbing et al. 2000) assigned all the individuals with a pre-defined velocity which caused the individuals to move to the exit).

3. 4. 2. 2 The State of Navigation: Static or Dynamic

Depending on whether the simulation environment can be changed (e.g. a blockage of corridors due to fire) during the simulation, navigation can be either "**static**" or "**dynamic**". The "static" approach is effective and precise when the research is located in some specific scenario. For example, to find out the maximum capacity of a building, evacuating time can be obtained by simulations with different numbers of occupants. In the reviewed literature, most studies used a static layout to represent the simulation environment. In these studies, the navigation map was generated before the simulation started and remained unchanged during the simulation.

In contrast, in dynamic navigation, the navigation map will be constantly updated to reflect any changes in the simulation environment, e.g. in a fire accident, the fire may spread and block some possible paths. Dynamic navigation can be used in real-time visual simulation to support a more real scenario.

Compared to static navigation, dynamic navigation consumes more computational resources. However, such a difference can usually be ignored in modern studies because of the rapid development in computer technology.

3. 4. 2. 3 Knowledge of the Environment: Shared or Individually-based

Usually, navigation uses the same navigation map to calculate the routes for all the individuals which means that the crowd has "**shared**" knowledge during the simulation. This type of navigation provides the same route choice to all the individuals. This approach is easy to implement but ignores the differences in individual knowledge of the environment.

Another approach is to use the "**individually-based**" navigation map in navigation. Each individual can improve his/her navigation map when his/her knowledge of the environment has increased (e.g. through exploring or communication). This type of navigation enables knowledge-based route choices for individuals.

3. 4. 3 Navigation Methods

3. 4. 3. 1 Cell and Portal Graph (CPG)

Cell and portal graph (CPG) is an abstract presentation of the environmental geometry structure and is usually used for global level navigation. CPG was firstly introduced by Teller in 1992 (Nuria Pelechano et al. 2008) and since then has been used in many studies (Lerner et al. 2006; Pettre et al. 2005; Nuria Pelechano & Norman I Badler 2006). In CPG (see Figure 60 for example) the places and areas (e.g. rooms, corridors, passages, etc) are represented by cells which represent real spaces. The connections or links between those places and areas (e.g. doors) are represented as portals, which do not occupy any space

Through transferring the environmental geometry structure (i.e. the floor plan) into CPG, navigation becomes the problem of travelling from one node to another in the graph (*Note. The algorithms of visiting the notes in a graph are studied in graph traversal which is beyond the scope of current PhD study. Two widely known algorithms in searching a path could be Dijkstra's algorithm for the shortest path and the A* search algorithm.*).

.

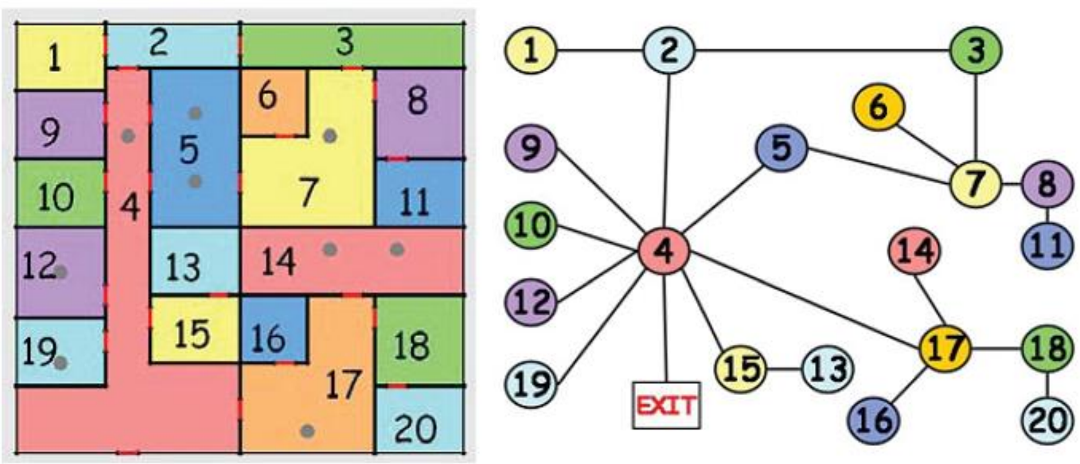


Figure 60 A floor plan of a building and its representation in CPG (Nuria Pelechano et al. 2008)

3. 4. 3. 2 Potential Field

The potential field method divides the simulation environment into regular size grids and assigns each grid with a numerical potential. In this navigation method, the individual at a higher potential grid moves to the adjacent lower potential grid. The potential field navigation method naturally fits the CA modelling approach as both make use of grids (cells). Figure 61 demonstrates a potential map of a room in a CA model. The grey cells represent the walls and have a potential of 500 which is far larger than the normal cells (ranging from 1 to 22).

500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500
500	7.5	8	8.5	9	9.5	10	11	12	13	14	15	16	17	18	19	20	21	22	500
500	6.5	7	7.5	8	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	500
500	5.5	6	6.5	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	500
500	4.5	5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	500
500	3.5	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	500
500	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	500
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	500
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	500
500	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	500
500	3.5	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	500
500	4.5	5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	500
500	5.5	6	6.5	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	500
500	6.5	7	7.5	8	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	500
500	7.5	8	8.5	9	9.5	10	11	12	13	14	15	16	17	18	19	20	21	22	500
500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500	500

Figure 61 The potential fields of a room in a cellular automata model (Varas et al. 2007)

3. 4. 3. 3 Directional Vectors

This type of navigation uses invisible vectors on the field to guide the moving directions of individuals. As the vectors are usually equally located on the field of the simulation environment, this type of navigation is also known as flow tiles (Chenney 2004). Figure 62 demonstrates a typical navigation map showing the directional vectors.

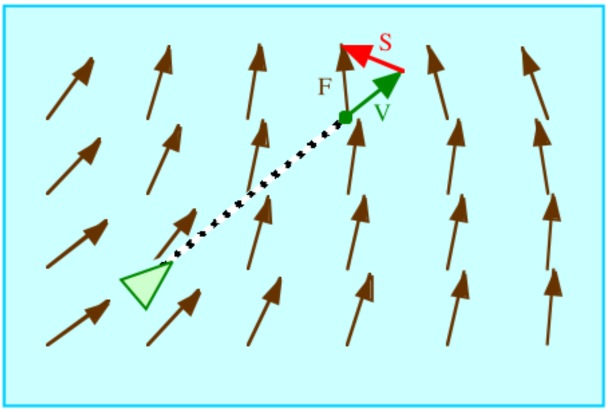


Figure 62 Navigation using directional vectors (Reynolds 1999)

3. 4. 3. 4 Discussions

It should be noted that the navigation methods introduced above are purely mathematically based and the resulted route is optimum. In crowd simulation, human intelligence, mental issues and action ability may be taken into account. For example, in the situation of an emergency evacuation, it is reasonable to consider that most of the evacuees may not be able to make a decision concerning an escape route because they are panicking or because of their insufficient knowledge of the environment (additionally, psychology research points out that people tend to use familiar routes to escape a building, such as the route by which they enter the building, and ignore the signs to safety exits).

3. 5 Summary of the Chapter

This chapter reviews the technologies that could be used in the implementations of crowd simulation. It firstly introduces the technologies that can be used in the implementation of crowd simulation. Then, it reviews popular crowd simulation software and the available crowd simulation packages. Finally, the concept of navigation and relevant technologies in crowd simulation is presented.

Chapter 4 RESEARCH METHODOLOGIES

4. 1 Overview

A research methodology is the attempt to validate the rationale behind the selected research design and provide justification of why it is appropriate in solving the selected research problem (Bell 2010). It is agreed that the effective use of suitable research strategies in the right way at the right time is always essential for good research (Robson 2002). To sum up, the research methodologies are guidelines of the research, which provide the rational process to achieve the research aim and objectives.

4. 1. 1 Research Paradigm

The research paradigm of this study is positivism which incorporates the realism ontology and the epistemology of empiricism. On the one hand, generally speaking, realism, as an ontological position, is "*we perceive objects whose existence and nature are independent of our perceptions*"(Oxford Companion to Philosophy, 1995). It confirms the existence of reality which can be observed or experienced (Blaikie 2007). On the other hand empiricism is based on the idea that knowledge comes from "observing" the world around us and then is produced by the use of the human senses (Blaikie 2007). As realist ontology claims that reality exists independently from actors, empiricism bases on this idea and suggests that the way of knowing reality is via human's sensory perceptions.

Positivism claims that only phenomena experienced by the senses can be regarded as real knowledge (Bryman, 2008). Research methods associated with positivism include quantitative or experiment-based research (Denzin & Lincoln 2000). It is considered that positivism can provide precise measurements and objective interpretation of the results (Kolakowski 1972; Comte 1988). Moreover, positivism tends to provide a "pattern model" of explanations, typically statistical associations.

4. 1. 2 Research Framework and Key Methods

The key methods employed in this research study are: literature review and software prototyping. **Error! Reference source not found.** shows the research methodology model of this research:

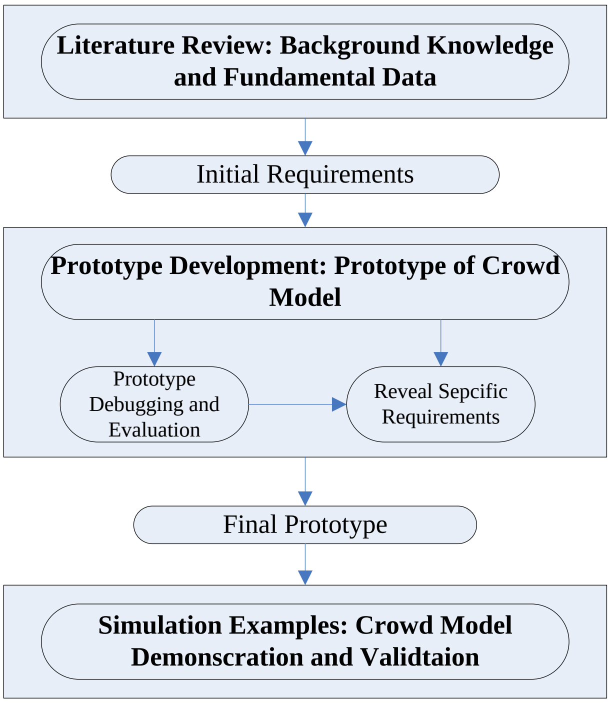


Figure 63 Model of Methodologies

4. 2 Literature Review

4. 2. 1 Introduction

Literature review is a widely used research method to review current knowledge on a particular subject. Through information seeking and critical analysing, literature review gives researchers background and knowledge of the research area and identifies possible questions which can lead to further research (Taylor, 2009). The purposes of literature review are summarised as (Bourner et al. 1996):

- To enrich personal knowledge in the research fields.
- To avoid doing a research that has already been done.
- To identify the area where to carry on further research.
- To find out other researchers who are working in the same area.
- To seek out available resources that can help the research.

4. 2. 2 Outcomes of Literature Review

4. 2. 2. 1 Overview

In this research, the literature reviews were conducted in the following area:

- Crowd simulation in general
- Crowd modelling
- Implementation of crowd models
- Evaluations on crowd models

The knowledge gained from literature was used in the following fours aspects of this PhD research:

- Helped to identify the need of further research
- Provided methods, technologies, and relevant knowledge to design a crowd model that meet the needs that were identified in this study
- Guided the implementation of the proposed crowd model
- Provided a scientific approach to evaluate the proposed crowd model

4. 2. 2. 2 Detailed Outcomes

Detailed outcomes of the literature review are listed below:

✧ **General**

- Identify the purposes that drive researches in crowd simulation. Identify the goals of crowd simulation and the outcomes that have been achieved in the existing studies.
- Review the scenarios that have been used to carry out crowd simulation. Find out crucial cases that have been considered in crowd simulations.
- Survey crowd behaviours that have been present in crowd simulation and how they are represented in the simulation.
- Clarify outcomes and requirements of current crowd simulation researches.
- Identify research aspects of crowd simulation and current focuses in terms of academic research.
- Find out research groups or network of crowd simulation researchers.

✧ **Crowd Modelling**

- Review the crowd models that represent crowd at different scopes and identify the usages of both macroscopic models and microscopic models.
- Identify key components of crowd models and find out the aspects that are considered in designing crowd models.
- Find out existing crowd modelling approaches and related crowd models in theory.
- Review the principles for force-based models, CA models and other models that describe movement of crowd.
- Review the approach to integrate artificial intelligence in crowd modelling and the technique of applying agent-based models.
- Identify the individual behaviours that have been considered in crowd behavioural models.
- Review the methods that can be used in representing human behaviours.
- Identify key parameters that can be used to represent generic individual behaviours.
- Identify the key issues of the generic crowd model.
- Clarify the relationship between a crowd behavioural model and a generic crowd model.
- Identify human behavioural theories which have already been used or can be used in crowd behavioural models.
- Identify the process of human making decisions and the issues affect such process and how.
- Identify mathematic models and theories in coordinating multiple parameters.

✧ **Model Implementation**

- Identify the process of implementing a theoretic crowd model.
- Survey existing crowd simulation frameworks.
- Survey existing software of crowd simulations.

- Survey graphic engines and software packages that can be used to implement a crowd model.
- Identify technologies of autonomous agent in software engineering.
- Identify technologies about implementing artificial intelligence.
- Identify software development methodologies and the process that can be applied to implement a crowd model.
- Review algorithms which are used in crowd navigation.

✧ **Evaluation/Validation**

- Survey applicable methods to validate a crowd model in both mathematical and physical scopes.
- Review the methodologies that can be applied to evaluate a crowd model.
- Identify the gaps between simulation and the real world.
- Identify the scenarios that can be used in evaluation/validation.

4. 3 Software Prototyping

4. 3. 1 Introduction

Software prototyping is a term of software engineering. The prototyping method is to develop a prototype which implements initial requirements and through user evaluation to gain further specifications (Courage & Baxter 2005). The process of prototyping development method has four steps (Naumann & Jenkins 1982):

- Collect initial requirements from the user.
- Develop a working prototype which implements the already-known features.
- User evaluates the prototype and provides feedback.
- Improve the prototype to suit the newly identified specifications.

This methodology is particularly useful when specifications cannot be fully identified at the beginning. Usually, the end-user cannot provide accurate requirements at the early stage because of the complexity of the system (Defence Science Board, 1987). In terms of this research which aims to design a generic crowd model for crowd simulation, the key requirements are still not very clear at the early stage as well as which technologies are better and easier to use to implement the crowd model. Prototyping can reveal the detailed requirements by testing prototypes and provide feedbacks for further development. This iteration also suits the natural process of this research. To be more specific, the software prototyping method provides the following benefits to this research study:

- The crowd model cannot be fixed or fully decided until the very late stage of research. Specifications are not clearly identified at the early stage. Feedbacks collected from prototype evaluation can help reveal the requirements.
- To test the validity of the designed crowd model as well as the accuracy of implementation need a lot of validating work. Using prototype development methodology will cost less time on correcting the model and its implementation.
- The technologies that best suit implementing the crowd model are unknown at the beginning. Through developing some simple prototype can help find out the pros and cons of each technology.
- The process of crowd model design is to build a simple model into a multiple individual parameters supported model. The models in each step can be implemented by developing a prototype for validation and evaluation.

4. 3. 2 Discussions on Similar Research Methods

Other software development models have been reviewed but considered not suitable. One of them is the traditional waterfall model (Royce 1970). The waterfall model divides the development process into seven sequential phrases and starts with identify the requirements specifications. It is quite impossible to follow such a process in this research as the full requirements are need to be clarified through evaluation. Another one is the spiral model (Boehm 1986) which is an iteration progress model. It contains steps of collect requirements, preliminary design, first prototype and then second prototype and keep iterating until reach the user's satisfaction. The spiral model somehow combines the prototyping model and the waterfall model and it aims to large and complex projects so it's not necessary to use this model in this research.

4. 3. 3 Process of Prototyping

In this study, prototyping are used in two aspects:

- To design the crowd model
- To select of a suitable implementation tool

4. 3. 3. 1 Prototyping in Crowd Model Design

The design of the crowd model employs evolutionary prototyping and prototyping process includes three development cycles. Evolutionary prototyping refers to build a robust prototype and constantly refine it over the development cycles. The three development cycles can be summarized below:

- **First development cycle**: design a crowd model which can represent various individual behaviours into measurable effects on individuals' movements.
- **Second development cycle**: integrate an agent model into the existing model to create crowd heterogeneity and provide a decision making process for high level artificial intelligence representation.
- **Third development cycle**: through analysing the behaviour effects calculation methods and their employments of the agent information in agent model, identify a generic representation of different individual behaviours and explain how the agents' parameters can be integrated into the behaviour effect calculations.

4. 3. 3. 2 Prototyping in Tool Selection

As there are many existing tools (a tool can be a crowd simulation software, a crowd modelling package, a framework for further development, etc.) for the study in crowd modelling and simulation, the author needs to select an appropriate tool for the implementation of the crowd model in this study. Through developing some simple prototypes can not only test the capability of one tool but also demonstrate to what extent the author can make use of it.

In this study, the following tools have been tested during the selection process:

- A.I. implant (version 5.4)
- Quest 3D (version 3.2)
- UnReal Engine (UDK version)
- Microsoft XNA Framework 4.0

4. 3. 4 Using Examples to Demonstration and Validation

4. 3. 5 Introduction

In this research study, examples of some typical scenarios will be presented through the simulations of the final prototype of the crowd simulation system to test and evaluate the proposed crowd model. Observations and data collected from the simulation will be analysed. In order to accomplish the requirements in the objectives of this study, these examples should be able to demonstrate several different types of human behaviours and show their effects on crowd. These behaviours will be interpreted and represented through the proposed crowd model. With appropriate configurations of agents, the crowd behaviours and movement will be observed from the simulation. The result of the simulation need to be compared with the experimental data in real-life in order to validate and evaluate the prototype thus the crowd model. Modifications could be made based on the evaluation result in order to improve the simulation prototype.

4. 3. 5. 1 Discussions on other Methods

In terms of demonstrating the prototype and evaluating the results of simulation, the workshop method is also considered as an alternative way. As it may be difficult to obtain sufficient experimental data from a real case, the results of simulation can be validated and evaluated through a group of experts. For example, in the case of a fire emergency evacuation, it would be nearly impossible for the authors to observe such evacuation from own experience. And it would be costly to organise an event such a fire drill as an experiment. In this study, the author considers the examples of three scenarios for model demonstration and another three simulation scenarios to reproduce the existing real-life experiments are sufficient for such purposed.

4. 4 Summary of the Chapter

This chapter outlines the research methodologies of this PhD study. It briefly introduces the nature and process of this research study and presents the appropriate research methods that have been chosen: literature review and software prototyping. Then it introduces each of the selected research method in detail, explains the rationale behind the selection, and discusses how to apply these research methods and the outcomes of each stage of the research process.

Chapter 5 CROWD MODEL DESIGN

5. 1 Overview of the Crowd Model

This overview section aims to provide an overview of this study's crowd model from a design perspective. It briefly describes the proposed crowd model in three aspects:

- How the crowd and individuals are viewed and represented in this crowd model;
- Followed what approaches, this crowd model is designed;
- The compositions and structure of this crowd model.

5. 1. 1 Research Scopes of this Crowd Model

The research scope defines how the crowd is modelled, how the individuals are modelled, and how they are represented in this study.

5. 1. 1. 1 Representing Crowd at a Microscopic Level

The proposed crowd model is categorised as a microscopic crowd model. The crowd is modelled as a collection of individuals. Each individual in the crowd is independent and can make his/her own decision as well as can react to the others who surround him/her. An individual conducts his/her behaviours based on his/her perceptions, abilities, and preferences. The crowd behaviour (including collective behaviour) is presented as a result of reactions and interactions between individuals.

5. 1. 1. 2 Heterogeneous Individuals

The individuals in this crowd model are designed to be heterogeneous and have unique sets of their own abilities, knowledge, characteristics and behaviour preferences.

The heterogeneity in this crowd model is demonstrated in two aspects:

- Firstly, it enables individuals to make independent decisions in the same situation, which means individuals will act accordingly to their own interests and abilities.

- Secondly, it refers to the variances in individuals conducting a same behaviour due to the differences between them, i.e. a behaviour may be performed slight differently by individuals.

5. 1. 1. 3 A Crowd Model with 2-Dimensional Representation

In this crowd model, individuals are located in a plain (in terms of terrain) simulation world which is represented as a 2D virtual environment. The Cartesian 2D coordination system has been adopted to represent the positions of the individuals. In this study, all the calculations and discussions relating to individuals' positions and their behaviours are based on this premise.

5. 1. 2 Modelling Approaches

The design of this crowd model combines the force-based modelling and the agent-based modelling approach. In this model, the movement of each individual is determined by behaviour effects (i.e. the forces generated from its behaviours). The agent is used to represent individual with independent physical and psychological attributes who can make independent decisions, which enables the crowd heterogeneity.

(The term 'agent' will be used to refer to an individual in the crowd from now on)

These two approaches represent individual/crowd behaviours at two different levels. At the lower level, the force-based modelling method interprets how the behaviours affect the movements of agents. Such behaviour effects are calculated through a set of pre-defined behaviour rules (via derivations of a unified formula) and the continuous positions of the agents are represented in the Cartesian coordinate system. At the higher level, the agent-based modelling approach is adopted to model the intelligent individuals (known as agents) and their decision-making process. It determines the selection of the agent's behaviour configuration. The effects of those behaviours are then calculated at the lower level by the corresponding formulas.

5. 1. 2. 1 Bottom level: Force-based Modelling

This model adopts the concept of force-based modelling that considers that the effects between entities (including individuals and other physical objects) can be represented in the form of forces. Furthermore, it proposes that each behaviour can generate an effect that determines the movement of individuals by taking into account the agents' heterogeneities.

The force-based modelling level provides guidelines on two aspects:

- Firstly, it contains a unified formula to calculate the effect of any behaviour that is related to the agent movement.
- Secondly, it explains a mechanism of how to combine the effects generated from different behaviours.

5. 1. 2. 2 Top level: Agent-based Modelling

The agent-based modelling approach aims to create intelligent individuals in a crowd simulation who can make their own decisions based on their status and the surrounding environment. In this study, the agents are designed by taking into account the functions of the bottom force-based modelling level.

The agent-based modelling level provides guidelines on three aspects:

- Firstly, it defines the attributes of an agent (such as physical abilities, personalities and behaviour preference) which enables the crowd to have heterogeneity.
- Secondly, it defines how an agent should act during the simulation and how this agent perceives and interacts with the surroundings.
- Thirdly, it provides a human-like decision making process to create an intelligent, automatous and independent agent.

5. 1. 3 Crowd Model Structure

In order to emphasise the different aspects of crowd modelling, to clarify the functions of each part in a crowd model, and to ease the complexity of implementation, this study's crowd model is designed as a loosely coupled system which contains of four modules (i.e. Agent Information, Agent Action Engine, Behaviour Library, and Simulation World). Each module is self-contained and operates independently. As a system, each module serves certain functions and interacts with other modules via standard protocols.

5. 1. 3. 1 Compositions of the Crowd Model

The functions of these four modules are briefly explained below:

- **Agent Information**: The Agent Information module provides all the information that relates to the agent for decision-making and simulation representation. It contains not only the attributes that are defined in the agent model but also the agent's status and perceptions during the simulation.
- **Agent Action Engine**: The Agent Action Engine acts as the brain of an agent. It follows a predefined process to control the agent's actions. Firstly, it decides the preferred behaviours based on the information from Agent Information module. Secondly, it calculates the behaviour effects by applying corresponding behaviour rules (via the formulas that presented in the Behaviour Library). Lastly, it updates the agent's information and the possible interactions with the simulation world.
- **Behaviour Library**: The Behaviour Library is a collection of behaviours. It explains how to calculate the behaviour effects of different behaviours by utilising a unified formula and relevant information (personal attributes, status, and perceptions) of the agent. It can be treated as a reference or a resource pool for the Agent Action Engine.
- **Simulation World**: The Simulation World can be viewed as a container of all the objects in the crowd model. Those objects consist of two categories: one is the agents which form the crowd in the simulation; the other one is the environmental objects which include rooms, gates, obstacles, etc.

5. 1. 3. 2 Model Structure Outline

Figure 64 outlines the overall structure of the crowd model and the communications between them.

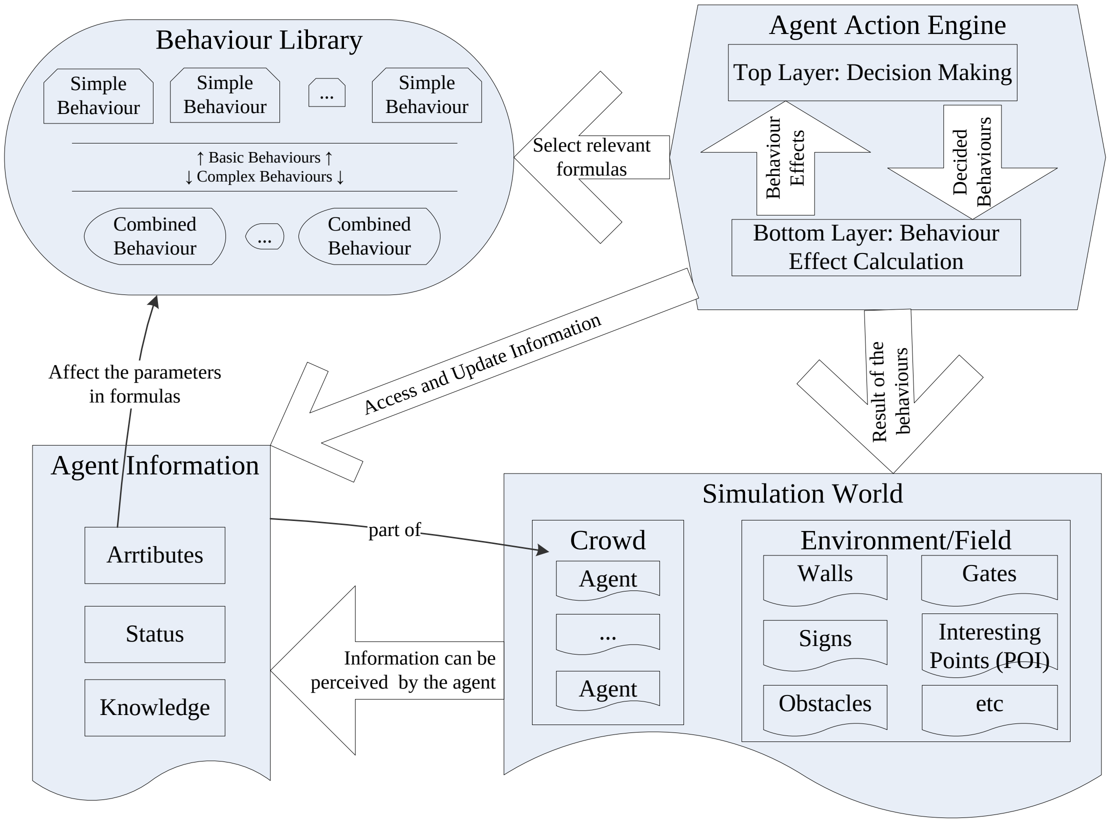


Figure 64 Structure of the crowd model in this PhD study

5. 1. 3. 3 Benefit of such a Design

By designing the crowd model in a loosely coupled manner, it reduces the dependencies between modules thus the crowd model can be described in a clearer structure. Such an approach not only simplifies the design and implementation of this crowd model as each module is self-contained, but also makes the expansion of the crowd model can focus on one module without changing the working mechanisms of other modules.

5. 2 Details of the Crowd Model

This section provides detailed descriptions of the proposed crowd model in this research study. The contents of this section are listed as follows:

- The types of behaviours that are discussed in this crowd model.
- How to calculation the effect of a single behaviour on an agent's movement. This includes what parameters are used and how to calculate them.
- How to combining single behaviours.
- The design of the agent model which includes: the attributes that are used to describe an agent, the knowledge an agent can possess, and the agent decision making and action process.
- A Behaviour Library which explains: how to interpret the behaviour with the unified formula, how to map an agent's attribute into the formula that is used to calculate behaviour effect, how to represent complex or advance behaviour via combining single behaviours.
- How the environment is represented and perceived by the agents. In the crowd model, the environmental information is considered part of an agent's knowledge.

5. 2. 1 Types of Behaviour in this Crowd Model

This research study discusses crowd behavioural modelling and the representations of different types of behaviour. Based on the scope where these behaviours happen, they can be placed in three levels:

- **Behaviours at the modelling level**: These types of behaviours refer to a single and specific action that an agent performs. These behaviours are directly modelled and represented in this crowd model. For example, "walk randomly" is a behaviour at modelling level.
- **Behaviours at the individual level**: These types of behaviours refer to the decision or the overall action that an agent will take under a given situation. The representations of these behaviours usually consist of several behaviours at modelling level. For example, an agent will "evacuate from a building" during a fire emergency. This behaviour could contain a collection of behaviours: "leave the room", "seek the exit", "avoid collision", and "follow other people".

- **Behaviours at crowd level**: These types of behaviours refer to the phenomenon or the behavioural preference that is emergent from a whole crowd or a group of people. These behaviours are not directly modelled in this crowd model but will emerge during the simulation when certain behaviours have been applied to the agents. For example, congestions can be observed at bottlenecks of buildings during an emergency evacuation.

The detailed descriptions of the behaviours in these three levels are presented below:

5. 2. 1. 1 Behaviours at Modelling Level

Behaviour at the modelling level can be seen as a binary action between the agent and its behaviour target. This type of behaviour defines a specific action that the agent will take and the effect of such behaviour on an agent's movement can be calculated through the unified formula (introduced in chapter 5. 2. 2. 6) in this crowd model. For the whole list of behaviours at the modelling level refer to the behaviour rules in the Behaviour Library (see chapter 5. 2. 4 for more details).

Some of the behaviours at the modelling level can find direct corresponding behaviours that are known in common sense. For example, an agent's behaviour can be described as "exit the room" via common sense. Such behaviour can be interpreted by the behaviour at the modelling level as "Seek" the exit of that room. However, not all behaviours at the modelling level have their projected behaviours in common sense. For example, the "repulsive effect" is a behaviour that describes an agent who feels a repulsive effect that pushes it from a nearby entity (either another agent or an object). This will usually be treated as a passive reaction and will not be mentioned explicitly in the description of the behaviour in the real world.

5. 2. 1. 2 Behaviours at Individual Level

In this model, the behaviours at an individual level refer to the decisions and the overall actions of an agent. They are those behaviours that are performed by the agent and are used as the descriptions of what is the agent doing. They are the behaviours that usually are used in the real world.

5. 2. 1. 3 Behaviours at Crowd Level

When the word "behaviour" is used at crowd level, it usually indicates a phenomenon or behavioural preference that can be observed in a crowd (demonstrated by the crowd movement as a result). Such behaviour will only happen when the crowd is under certain circumstances or with specific compositions. For example, when a large group exits at a small gate it can be observed that the crowd form an arch shape formation around the exit. This phenomenon is called "clogging behaviour". Providing another example, studies of consensus decisions (J. R. G. Dyer et al. 2009; Faria et al. 2010; J. R. G. Dyer et al. 2008) in human groups reveal that a group's movement is determined by a small number (about 10%) of individuals in that group. Such effective leadership is known as the "consensus decision making" behaviour of group.

In this crowd model, these types of behaviours are not directly modelled as they are considered as emergent behaviours. The word "emergent" indicates such a behaviour that usually cannot be seen from the descriptions of individual behaviours. For example, when "clogging behaviour" is observed, no individuals were told to queue in the arch formation at the exit. Such a phenomenon emerged because every individual was trying to move closer to the exit and the arch shape formation was the result so that most individuals could achieve their closer distance to the exit. In most cases, a crowd level behaviour can be achieved through the combination of the different individual level behaviours of the agents which means such behaviour represents the results from the interactions between individuals in the crowd.

5. 2. 2 Representation of Behaviour Effect

5. 2. 2. 1 Definition of Behaviour and Behaviour Effect

In this section, the term "behaviour" refers to the behaviours at modelling level. The "behaviour effect" refers to the result of an action performed by an agent which can be measured as a position change of that agent. It is a binary action which only relates to the agent and its behaviour target. For types of behaviours, a behaviour target (either real or virtual) must be presented which causes the behaviour to happen or acts as the source to result in the behaviour effect.

5. 2. 2. 2 Behaviour Effect and the Agent's Movement

Representing behaviour through its effects on agents' movement is not a new concept as similar approaches have been seen in existing studies (Reynolds 1987; Dirk Helbing & Peter Molnar 1995; D. Helbing et al. 2000; Reynolds 1999) but this model takes a novel approach which proposes that all the behaviour effects could be represented and calculated by applying a set of generic parameters and using a unified formula.

In this crowd model, an agent's movement is updated by applying the behaviour effects on its position. This approach considers that a behaviour produces an effect which changes the position of the agent. (It can also be expressed as the agent generating an effect to change its position in order to conduct a behaviour.) As a result, the agent's movement calculation is shown as follows:

$$next\ position\ =\ current\ position\ +\ behaviour\ effect$$

Formula 29 Applying behaviour effect on an agent's position

In order to represent the effect of a behaviour, the form of vector is used and only two elements need to be clarified:

- The strength (magnitude) of the behaviour effect which represents the distance that the agent will move.
- The direction of the behaviour effect which indicates the direction in which the agent will move.

In this research study, the positions of the agents are represented in the 2D coordinate system. The behaviour effect on agent's position is demonstrated in Figure 65:

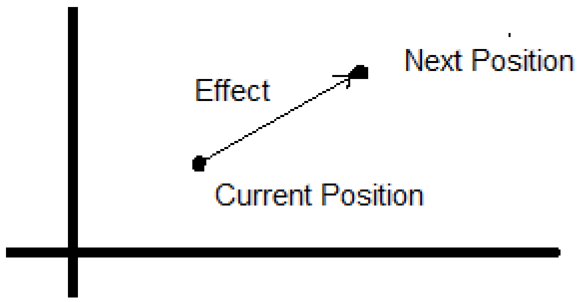


Figure 65 An effect changes agent's position from the Current Position to the Next Position

5. 2. 2. 3 Relationship between Behaviour Effect and the Agent's Speed

In kinematics, an object changes its position because it has velocity. In this crowd model, an agent changes its position because behaviours have effects on it. It is considered that the behaviour effect has the same form of velocity and can be treated as an equivalent of the velocity to some extent in the movement calculation.

More specifically, when an agent is influenced by a behaviour (or the agent performs the behaviour) that can change the agent's position, it is described as moving at speed S m/s in the direction of W because of that behaviour. During a time of Δt, the result of the speed is to move the agent in the direction W for a distance of Δd, where $\Delta d = S \times \Delta t$. In this crowd model, such a result is interpreted as the effect of the behaviour.

The effect that changes an agent's position strictly follows the principles of kinematics which means that the effect of the behaviour and the agent's average speed during time Δt is convertible. When Δt is small enough ($\Delta t = 1/60$ second in this crowd model), the agent's speed during Δt can be considered constant. So the effect of a behaviour can be interpreted as the agent deciding to move at a certain speed (which is equal to the average speed in Δt).

In kinematics, the change of position can be stated:

$$next\ position = current\ position + displacement$$

Formula 30 Position change in kinematics

And

$$displacement = average\ speed\ \times \Delta t$$

Formula 31 Displacement calculation in kinematics

By considering Formula 29, Formula 30 and Formula 31 together, it can be discovered that, in a time Δt, the effect generated from the object to the agent can be calculated through:

$$behaviour\ effect\ = average\ speed \times \Delta t$$

Formula 32 Conversion between speed and the base effect of behaviour

Because Δt is a constant in the crowd model (e.g., Δt =1/60 second in this model), the effect of behaviour is proportional to the average speed. So the key point in this crowd model is how to determine the average speed for an agent during time Δt.

5. 2. 2. 4 Relationship to Classical Mechanics for the Movement Calculation

In Reynolds' models (Reynolds 1987; Reynolds 1999), the approach of using the effect of behaviour to determine the movement of artificial creatures was demonstrated. However, there was a lack of demonstration on the relationship between those models and the traditional force-based models which employed Classical Mechanics.

In force-based models, the agent movement is explained as a force that is generated from the target and is applied on the agent. This force results in a velocity change to agent and thus affects the movement. The whole calculation involves the classical dynamical mechanics whereby the movement of an object is described by Newton's law of motion. It includes complex calculations and requires more parameters.

this section will demonstrate how to simplify this calculation through mathematical conversion and physical laws. Finally, it will prove that Formula 32 is a simplified and equal expression of the force-based models.

For an object with initial velocity $\vec{v}_0$and mass m, if a force of $\vec{f}$ was applied, the displacement in time Δt can be calculated through:

$$\vec{D} = \vec{v}_0 \Delta t\ + \frac{1}{2} \times \frac{\vec{f}}{m} \Delta t^2$$

Formula 33 Displacement calculation in Classical Mechanics

Because acceleration a = $\frac{\vec{f}}{m}$, the velocity of the object $\vec{v}_t$ after Δt is:

$$\vec{v}_t = \vec{v}_0 + a\Delta t = \vec{v}_0 + \frac{\vec{f}}{m}\Delta t$$

Formula 34 the calculation of velocity (after time Δt)

Thus

$$\vec{v}_t - \vec{v}_0 = \frac{\vec{f}}{m}\Delta t$$

Formula 35 Relationship between $\vec{\mathbf{v}}_t, \overline{\mathbf{v}}_0, \overline{\boldsymbol{f}}, \boldsymbol{m}, \boldsymbol{and}\ \Delta \boldsymbol{t}$

By combining Formula 33 and Formula 35, one has:

$$\vec{D} = \vec{v}_0\Delta t \ + \frac{1}{2}(\vec{v}_t - \vec{v}_0)\Delta t = \frac{1}{2}(\vec{v}_0 + \vec{v}_t)\Delta t$$

Formula 36 Displacement calculated through velocities

The average velocity $\vec{v}_a$ equals:

$$\vec{v}_a = \frac{1}{2}(\vec{v}_0 + \vec{v}_t)$$

Formula 37 Average velocity during time $\Delta \boldsymbol{t}$

Then the displacement $\vec{D}$ can be calculated via:

$$\vec{D} = \vec{v}_a\Delta t$$

Formula 38 Displacement calculation through average velocity and time

Formula 38 is identical to Formula 31 which means that using average speed and time to calculate the effect of behaviour (Formula 32) is equivalent to the calculation in the models employ Classical Mechanics and Newtonian forces such as the Social Force models.

5. 2. 2. 5 The Factors that Decide the Behaviour Effect

This section discusses what factors should be considered in the calculation of the behaviour effect and how they influence the result of the behaviour effect.

Factors

- **Position**

Because a behaviour is an action happened between an agent and a target (could be a virtual target) and both could affect agent's movement. Therefore, the agent's position and target's position are two must-included parameters for the calculation.

- **Direction**

Because the agent is located in an environment which includes geometrical information (in this study, 2D geometry is currently used. However, the same concept can be applied in 3D geometry as well), there should be direction information in the behaviour effect on the agent. It is important to include the direction of the behaviour as it is crucial to the action result. For example, the behaviour of "walk away" and "walk towards" may have the same strength (indicates the same scalar value in measurement) but they have opposite directions of behaviour effect.

- **Default Walking Speed**

The result of a behaviour effect is to change the agent's position. The agent's default walking speed should be considered as a base value for reference in calculating any behaviour effect. That is to say, neglecting influences from all the other factors, the distance that the behaviour effect makes the agent move is equal to the distance that the agent will move at its default walking speed.

- **Distance**

The distance between the agent and its target may affect the value of the behaviour effect as the distance issue has been widely considered in physical systems. For example, in Newton's law of universal gravitation, the force is inversely proportional to the square of the distance; In the Social Force model, the repulsive interaction force between two pedestrians is determined by their distance (Dirk Helbing & Peter Molnar 1995; D. Helbing et al. 2000).

- **Agent**

This factor represents the agent's personal desire to conduct a behaviour. In the literature, a need of increasing heterogeneity in crowd modelling and simulation has been identified and this factor is designed to reflect individual differences in this model.

The value of this factor is not static but will be dynamically calculated by taking into account the behaviour, the agent, and the surrounding environment. For a behaviour, it has a unique mechanism to determine the value of this factor. For an agent, its own attributes and the surrounding environment will be used under the same mechanism to calculate this factor for each behaviour. In this way, crowd heterogeneity can be achieved through adjusting the personal attributes of the agents.

In the later section of "*Behaviour Library*", it explains how this factor is determined for each behaviour that has been identified in this crowd model. At the current stage of this study, these mechanisms in calculating behaviour effect are kept simple. However, a mechanism of determining the value of this factor could be referred to advanced high level AI and could involve with a complex process of utilising agent's attributes.

- **Target**

Because the behaviour is a binary action, this crowd model also contains a factor to measure the effect of a different target object on behaviour. This idea is exactly the same as the one presented above.

5. 2. 2. 6 The Formula for Calculation

Based on the above discussion, seven factors have been identified as parameters in the behaviour effect calculation. As a result, the following standard unified formula is proposed to calculate the behaviour effect:

$$Behaviour\ Effect\ = Rotation(Normalise(P_t - P_a)\ ,\alpha)\ E_s F_a F_t F_d$$

Formula 39 The unified formula for the behaviour effect calculation

The functions and parameters defined in the formula are explained in the following:

- **Functions:**

  - $\mathrm{Normalise(vector)}$: refers to the normalise operation on a vector which does not change the direction of the vector but sets the norm to 1. For example, assume $\vec{\mathrm{E}}\,(\mathrm{x},\mathrm{y})$ is the vector E; after the normalise operation it has $||\vec{\mathrm{E}}||$ = 1, i.e. $\sqrt{\mathrm{x}^2+\mathrm{y}^2}=1$). $\vec{\mathrm{E}}$ and E both point to the same direction.
  - $\mathrm{Rotation(vector,\alpha)}$: is defined as turning the direction of the vector anti-clockwise with an angle α.

- **Parameters:**

  - $\mathrm{P_t}$ is the position of the behaviour target. It is a vector which is presented as $\mathrm{P_t(x,y)}$ because this model uses a 2-D coordinate system.
  - $\mathrm{P_a}$ is the current position of the agent. It is a vector which is presented as $\mathrm{P_a(x,y)}$ because this model uses a 2-D coordinate system.
  - α is the behaviour angle which indicates the offset of the direction to the original behaviour direction. The original behaviour direction is determined by the positions of the agent and its behaviour target which is calculated by $P_t - P_a$.
  - $\mathrm{E_s}$ stands for the Effect of the base Speed. It is a scalar value which means the distance agent can move in time Δt under the agent's normal conditions. It has a direct link to the speed of the agent.
  - F_a stands for Self Factor. It reflects the agent's own desire on the behaviour. For example, if the agent decides to walk normally, then SF is set to 1. If the agent decides to run, SF could be set to 3.
  - F_t stands for the Target Factor. It reflects the impact of the target on the agent. For example, in the case of walking away from a smelly agent, that smelly agent should have a high TF so that it can generate a large effect.
  - F_d stands for Distance Factor. It indicates that, when calculating the effect between the agent and its target, the distance between the two should be taken into account. For example, the repulsive effect between two agents becomes less when their distance increases.

Formula Explanation

In this formula (Formula 39), the result of the behaviour effect is represented in the form of a vector. The direction of this vector is calculated via the first part of the formula: “$Rotation(Normalise(P_t - P_a), \alpha)$”. The length (magnitude) of the vector is determined by second part of the formula ”$E_s F_a F_t F_d$”.

This formula provides a unified representation of all the behaviour effects. The differences between the behaviours are reflected in the values of the parameters. For a given agent and its behaviour target, the behaviour effects may have different outcomes on different behaviours as the values of parameters can be behaviour dependant. As a result, the parameters in the formula can be divided into two categories by their natures:

- Behaviour independent parameters: P_t, P_a, and E_s are independent of the behaviours which means their values remain the same during the calculation of the behaviour effects on different types of behaviours.
- Behaviour dependent parameters: α, F_a, F_t, and F_d are dependent on the behaviours which indicates that their value may vary during the calculation of the behaviour effect on different types of behaviours.

To be more specific, the agent’s position P_t and the behaviour target position P_a will always remain the same in all behaviour effect calculations and they will produce the original direction of the behaviour effect which is always be calculated by “$P_t - P_a$”. The Effect of base Speed E_s is a scalar value which indicates the displacement of the agent in that period under normal conditions. The value of E_s is dependent on the agent’s base movement speed but is independent of the types of behaviours that the agent conducts.

The differences between the behaviours are demonstrated through the four behaviour dependent parameters: α, F_a, F_t, and F_d. The behaviour angle α decides the final direction of the behaviour effect. The agent’s Self Factor F_a, its Target Factor F_t and the Distance Factor F_d are scalar values which serve as the coefficients to the base value of behaviour effect. Because their values are behaviour dependant, the calculations of these parameters will be discussed in more detail in the "5. 2. 4 Behaviour Library" section.

- **Relationship between E_s and the agent's walking speed**

As stated above, E_s is directly linked to the agent's speed. In addition, the simulation graphic configuration (the scale of the simulation unit to the real world unit and the frame rate of the simulation) needs to be considered as well. As a result, E_s is calculated via Formula 40. s represents the agent's default walking speed in real world, u denotes the unit scale in the simulation environment (1 pixel : 0.05 metre in this study), and r denotes the frame rate of the simulation engine (60 FPS in XNA framework).

$$E_s = s/ur$$

Formula 40 Calculation for effect of base speed

- **An Alternative for Direction Calculation in the Formula**

In the presented formula, the direction of the behaviour effect is calculated by "$Rotation(Normalise(P_t - P_a)\,, \alpha)$". However, in some cases, this calculation may be simplified into turning a certain angle (anti-clockwise) based on the agent's current orientation (which is a unit vector), e.g. turning left:

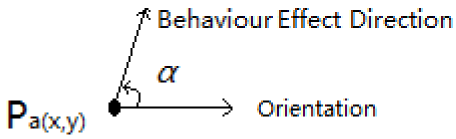


Figure 66 Behaviour Direction based on orientation

$$Behaviour\ Direction\ = Rotation\ (Curerent\ Orientation, \alpha)$$

Formula 41 Behaviour direction alternative calculation

Although this approach appears to introduce a new method to calculate the behaviour direction, it actually can be converted and represented by the standard unified formula (Formula 39).

In order to maintain the unified calculation, a virtual target needs to be created. Its position is in front of the agent and is located in the line of the agent's orientation. Assuming the orientation of the agent is *O(m,n)* and its position is $P_a(x,y)$, the position of the virtual target can be defined as:

$$P_t(x',y') = P_a(x,y) + O(m,n) = P_t(x+m, y+n)$$

Formula 42 Calculation of the virtual target in the alternative behaviour direction calculation

Their positional relationship and the direction of the behaviour effect are demonstrated in Figure 67 below:

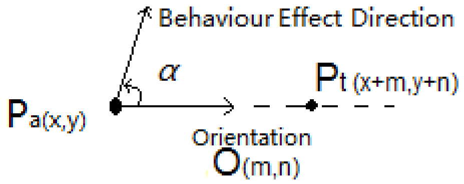


Figure 67 A virtual target in the direction of the agent's orientation

The direction of the behaviour effect is calculated through the original formula:

$$Behaviour\ Direction = Rotation(Normalise(P_t - P_a)\ , \alpha)$$

Formula 43 Behaviour direction default calculation

The above analysis demonstrates that the alternative calculation of behaviour direction can be converted into the original method. This alternative representation can simplify the behaviour effect calculations in those behaviours which involve agent's orientation change but do not have a target of the behaviour. For example, the behaviour of turning left.

5. 2. 2. 7 Combining Behaviour Effects

The method

In this research study, as the effects of behaviours are in the forms of vectors, the author proposes to use the standard vector operation - addition to combine multiple behavioural effects. The vector addition operation is commonly used in physics when calculating net force. As the effects of the behaviours in this model have similar natural forces, it is believed that using such a mechanism to calculate the sum of the effects is a reasonable approach.

The construction of adding together the two effects of behaviours is illustrated as below:

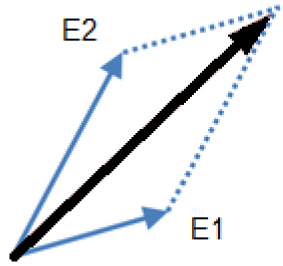


Figure 68 Vector operation: addition of two vectors

In this crowd model, a 2D vector is used to represent the effects of the behaviours (which is decided by the representation of the agent's position). In the above example, E can be calculated by:

$$E = E1 + E2 = E1(x_1, y_1) + E2(x_2, y_2) = E(x_1 + x_2, y_1 + y_2)$$

Formula 44 Addition of two effects

If the combination involves more than two effects, the total effect can be calculated by adding up each effect in sequence (In fact, when calculating the sum of the vectors, the sequence of the addition does not affect the final result; this is known as the commutative law. For example, *E1 + E2 + E3 = (E1 + E2) + E3 = E1 + (E2 + E3)*).

The final step of combining multiple behaviours is to check that the total effect E should not exceed the ability of the agent's movement. In other words, $||E|| \leq$ the effect is equal to the maximum speed of agent.

No Need for Weighting Coefficients

In Reynolds' Steering Behaviour model (Reynolds 1999), a weighting coefficient for each behaviour could be considered during the combination in order to reflect the priority of the behaviours. In this approach, the formula of adding two effects becomes:

$$E = \alpha \times E1(x_1, y_1) + \beta \times E2(x_2, y_2) = E(\alpha \times x_1 + \beta \times x_2, \alpha \times y_1 + \beta \times y_2)$$

Formula 45 Combination of two effects (if behaviour weighting factors apply)

This step is considered inappropriate in this crowd model because such weighting coefficients are implicitly included in formula. In Reynolds' Steering Behaviour model, all the formulas to calculate the steering forces were plain formulas which did not contain any personal preferences. In this model, the parameters F_a, F_t, and F_d can actually reflect the agents' behavioural preferences. If a behaviour is important, it can be reflected through larger values of F_a, F_t, and F_d.

To sum up, the behaviour effects in this crowd model have already been weighted through the parameters in the formula. Therefore, there is no need to introduce another type of weighting coefficient during the combination.

5. 2. 3 Agent Design

In this crowd model individuals are modelled as agents. The agents have their own attributes and status and can make independent decisions. These parameters and status influence how an agent decides and conducts its behaviours. The influences are represented in two aspects:

- The first aspect is during the decision-making process. The agent's perception is subject to its personal attributes and its decision should be made by following its behavioural preferences.
- The second aspect is when conducting the behaviours. The personal attributes will be used in the calculation of the behaviour effects to determine the parameters in the formula.

This section covers all the aspects of the design of the agent in the proposed crowd model in the following sequence:

- The agent's attributes determines the character and personality of an agent.
- The agent's knowledge contains the information/resources that an agent possesses during the simulation. The behaviour library defined in the next section is considered as part of this.
- The agent's status is the information about the agent at the time when it is making a decision. It will affect which behaviour to use from the Behaviour Library as well the values of the parameters in the formula.
- The agent's Action Engine describes the process of making a decision.

Because agent's attributes, agent's knowledge, and agent's status describes the information about an agent in different aspects, they are referred to a general term of "Agent Information".

5. 2. 3. 1 Agent's Attributes

Agent's attributes are used to describe the agent's character, abilities and preferences. They represent the nature of an agent. Their values are pre-defined prior to the start of a simulation.

Agent's attributes can be divided into three categories:

- **Physical attributes**: Physical attributes describe how the agent is presented in the crowd model and its physical abilities. Attributes includes: position, body size, orientation, movement mode, base movement speed, maximum movement speed, and movement speed adjusters.
- **Range attributes**: Range attributes define the ranges within or without which certain behaviours can take effect. Attributes includes: sight range, sense range for group behaviour, desired distance from others, minimum distance from others, desired distance from a wall, minimum distance from a wall, desired distance from obstacles, and minimum distance from obstacles.
- **Personality attributes**: Personality attributes define the character and the behaviour preferences of the agent. Attributes includes: leadership, willingness to follow, willingness to stay in a group, probability of being affected by POIs (point of interests), repulsive feeling towards people, and repulsive feeling towards obstacles.

Physical attributes

(This crowd model represents the agents in a 2-D dimension environment by default. The following attributes are defined on this premise.)

- **Position**

The position of the agent describes where the agent is in the simulation environment. It is a point which is represented as (x, y). Strictly speaking, this attribute is not part of the agent's natural parameters. This attribute only exists when an agent has been deployed in a simulation. It is a reference to a location in the environment. Before the simulation starts, each agent has an initial position based on the configuration.

During the simulation, the position keeps updating to reflect the movement of the agent.

An agent has only one position attribute and this attribute is used to determine the value of the parameter P_a in the unified formula (Formula 39).

- **Shape**

The shape attribute describes the space an agent occupies in the simulation environment. This model uses a circle to represent an agent. The reason for not using a square, an ellipse, or other more accurate polygons to represent the agent is that the circle is much easier to handle when dealing with rotation (turning), collision detection and graphical representation.

This attribute is defined as a static attribute in this model which means all agents are modelled and displayed as a circle and there is no alternative form of representation. This may affect some algorithms that involve shape information. For example, the collision detection algorithm is implemented to detect the collision of two circles.

The centre of the circle is defined as the position of the agent.

- **Size**

The size attribute indicates how large the agent is. It is defined as the diameter of the circle in the shape attribute.

As a default value, the size of the agent is set to 10 pixels.

- **Orientation**

This attribute indicates the facing direction of the agent.

Figure 69 demonstrates a possible implementation of the above geometry attributes. The circle represents the shape of the agent. The centre point of the circle is the agent's position. The size of the agent is the diameter of the circle. The orientation is indicated by a small line segment (showing the agent currently facing right in the picture).

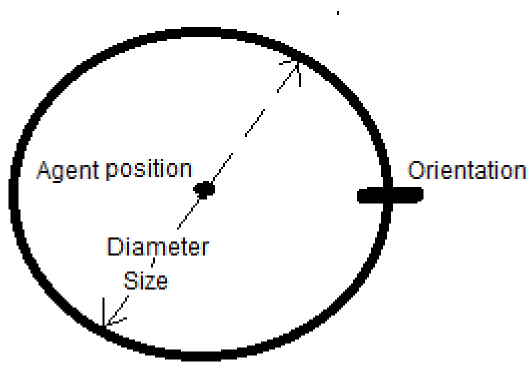


Figure 69 Demonstration of an agent's geometry attributes

- ## Movement Mode

This attribute describes the status of how the agent is moving at that moment. The agent could either be "walking" or "running" in this crowd model.

- ## Default Walking Speed

The default walking speed describes the desirable speed of an agent under normal circumstances.

It is used as a base value in the unified formula to calculate the Effect of base Speed (E_s).

- ## Behaviour Effect Limit (Walking)

Because this crowd model has a combination mechanism for behaviour effects, it is necessary to have an attribute to limit the combined effect. The final combined behaviour effect may exceed the effect that the maximum walk speed can achieve which could produce un-realistic behaviour.

For example, if there is an object which produces a maximum push away effect on the agent, the agent will walk away from this item at its full default walking speed. Assuming that there are three similar objects together, they will produce three times the maximum push away effect on the agent which, in turn, requires a three times increased walking speed. It is obvious that the agent cannot walk three times faster than its maximum speed. The attribute "Behaviour Effect Limitation" is used to limit the combined effect in order to prevent physically unachievable behaviour.

In the case of only one behaviour, the agent may have a desire to move at a higher speed (e.g. want to walk five times faster) which is not achievable based on its physical condition. In this case, the attribute “Behaviour Effect Limitation” provides a physical limitation on the behaviour effect.

This attribute is only used to compare the final combined behaviour effect. If the combined effect is larger than the “behaviour effect limitation”, then the effect should be limited to the value of the behaviour effect limitation.

The value of this attribute is set to 120% of the behaviour effect of the default walking speed.

- **Default Running Speed**

Similar to the “default walking speed”, this attribute defines the speed if the agent is running instead of walking.

- **Behaviour Effect Limit (Running)**

This attribute is used to limit the final combined behaviour effect to ensure that it does not exceed the effect of “default running speed”.

- **Movement Speed Adjuster**

This attribute defines the percentage of the default speed that should be used in the behaviour effect calculation. The default value is 100%. This attribute is used to change the base value of the default speed. The behaviour effect limitation will be changed correspondingly.

Range attributes

The range attributes are designed for those behaviours which only happen within or without certain ranges. In other words, the agent only performs the behaviours to the targets which are within or without the behaviour ranges.

In the unified formula, the range attributes determine the thresholds of the Distance Factor F_d. When the target is out of / within the behaviour range, it means that F_d will be set to zero so that the behaviour takes no effect. When a target is within / out of range, the correspondent value can be set to reflect the distance effect on the behaviour. In the case where F_d is not applicable to the behaviour, F_d should be set to one to indicate no influence on the behaviour effect.

- **Sight range**

An agent is only aware of the objects and agents in a certain range which is defined as the sight range. Sight range is the longest distance that an agent can interact at. All the other range attributes should be equal or smaller than this value.

Map awareness is not affected by the sight range. For example, the agent can plan a route with its knowledge of the map and walk to a position which is out of sight.

- **Sense range for group behaviour**

This attribute describes how far the agent will consider performing group related behaviours, for example, following the majority, following a leader, etc.

- **Desired distance from others**

This attribute defines the ideal distance that an agent would have from the other agents. The agent will try to maintain the desired distance from others if such a distance has not been reached. The effort that the agent will make to maintain the distance is dependent on how close the current distance to the desired distance. The closer to the desired distance, the less effort the agent will make.

- **Minimum distance from others**

This attribute defines the minimum distance that an agent can have from the other agents. The agent will use its full power to alter its position if its distance to others is less than the minimum distance.

- **Desired distance from walls**

This attribute defines the ideal distance that an agent will have from walls. The agent will try to maintain a desired distance from walls if such a distance has not been reached. This attribute has a similar nature of the “desired distance from others”.

- **Minimum distance from walls**

This attribute defines the minimum distance that an agent will have from walls. The agent will use its full power if its distance to walls is less than the minimum distance.

- **Desired distance from obstacles**

The agent will try to walk around an obstacle if it blocks the way. This attributes defines the desired distance from the obstacle while the agent performs collision avoidance behaviour.

- **Minimum distance from obstacles**

This attributes defines the minimum distance from an obstacle while the agent performs collision avoidance behaviour.

Personality attributes

- **Leadership**

This attribute indicates influence on others. The higher the influence the more likely it is that others will follow.

- **Willingness to follow**

The higher the influence of this attribute the more likely it is that the agent will follow somebody.

- **Group behaviour modifier**

The group behaviour modifier is the willingness to perform group related behaviour. Default value is one.

- **Probability of being affected by POIs (points of interest)**

This attribute relates to the probability of being attracted by the points of interest.

- **Repulsion modifier (to self)**

This attribute has a proportional relationship to F_a parameter. The default value is one.

- **Repulsion modifier (to others)**

This modifier has a proportional relationship to F_t parameter. The default value is one.

- **Attraction modifier (to self)**

This modifier has a proportional relationship to F_t parameter. The default value is one.

5. 2. 3. 2 Agent's Knowledge

An agent is an independent entity. Its possession of information on the surrounding environment is modelled in two aspects in this crowd model: map awareness and its perceptions to the environment.

Map Awareness

The map contains information of the simulation environment. It is a collection of the environmental objects.

The map provides target information for an agent. In one simulation, the global map for the environment can be fully or partially possessed by an agent. An agent makes decisions and moves based upon its own map.

- **Path / route**

The path/route provides the waypoint to the agent and enables it to move in the simulation world.

Agent's Perceptions of the Environment

During the process of decision making, the agent's observations of the environment form its perceptions and are used as inputs. This research study does not discuss how an agent acquires its perceptions psychologically. It only defines what objects can be perceived by the agent in this crowd model and how these perceptions are used for decision making and behaviour effect calculations.

- **Obstacles**

An obstacle is an object that an agent needs to walk around. In this model, an obstacle is represented by a circle. It can be treated as a non-moving agent. It has similar attributes to an agent and these attributes will be used in the agent's behaviour effect calculation.

Its attributes includes position, size and a repulsion modifier (to others).

- **Wall**

A wall can be treated as a line of obstacles whose size is 1. A wall is the basic unit to form the geometrical structure of the environment.

Its attributes includes start position, end position and a repulsion modifier (to others).

- **Area with a virtual effect**

An area with a virtual effect is deigned to implement an influence to change relevant parameters during the behaviour effect calculation. For example, an effect to halve the walking speeds of all the agents within the area. This effect is a simplified and abstract representation of the real world as it only describes the end influence on agents' behaviours. For example, the halving walking speed effect can be caused by frog or water on the floor in the real world.

- **Point of Interest (POI)**

A POI refers to an object that can provide information to the agents or can affect the agents' behaviours in the simulation environment. For example, a sign that indicates the emergency exit is a POI.

5. 2. 3. 3 Agent's Status

Crowd simulation is a dynamic process. The agent model should contain both pre-defined static attributes before the simulation and the dynamic status of the agent during the simulation. The agent's status represents the agent's information at a certain time (Δt) in the simulation. The agent's status keeps updating throughout the simulation.

Current position

The current position is the position of the agent at Δt.

Current speed

The current speed is the movement speed at the beginning of Δt. The speed will be updated by the behaviour effect at the end of Δt.

- **Desired speed**

A desired speed equals the agent default speed times the effect factors that come from the agent itself the target, and the distance. It is conveyed by the total behaviour effects calculated through the unified formula.

The agent may want to achieve a very high movement speed mentally (i.e. desired speed) but such a desired speed must be within the ability of its physical strength. Because this crowd model allows the agent to have multiple behaviours at one time, such a combination (integration) may result in a very large total behaviour effect which cannot be performed by the agent. If such an effect is applied on the agent, it will produce un-realistic behaviour. For example, based on agent's own desire and effects of the surroundings, the total behaviour effect may equal a speed of 10 m/s which is a very high speed which cannot easily be achieved by everybody.

The desired speed has no limitation in the crowd model but the agent must have some physical limitation. The maximum speed provides the maximum behaviour effects that an agent can have at one time. If the calculated total behaviour effect exceeds the maximum effect, the maximum value will be used instead of the calculated one.

Current orientation

Current orientation is the moving direction of the agent at the begging of Δt. The orientation will be updated by the behaviour at the end of Δt.

Behaviour mode

The behaviour mode indicates the behaviour status of the agent. The behaviour mode is used for decision making.

➢ **Moving**

Moving indicates the agent moving under normal situations which includes Wander, Follow and Move to a destination.

➢ **Collision avoidance**

The agent tries to avoid collision with other objects (which could be a wall, an obstacle, or another agent). It will return to the moving mode afterwards.

Goal

Some behaviours have a goal. It could be a waypoint, another agent, or any position. It indicates something that the agent wants to achieve. For example, if the goal is to exit, the agent will need to follow several waypoints when it is inside a building. The behaviour of the agent could change during the process of reaching the goal.

5. 2. 3. 4 Agent Action Engine

The action engine can be treated as the brain of an agent. It follows an agent action process (Figure 70) to calculate the steering force that is used to update the agent's position. It interacts with the behaviour library, the agent information module and the simulation world to retrieve relevant information. Information retrieved from the simulation world is called the agent's perception. The agent's status will also be updated during the process. The action engine will notify the outcome of the agent's behaviour to the simulation world. Objects in the simulation world may be affected correspondingly. In each time frame (Δt) the agent will repeat the action process to decide its behaviour and update relevant information.

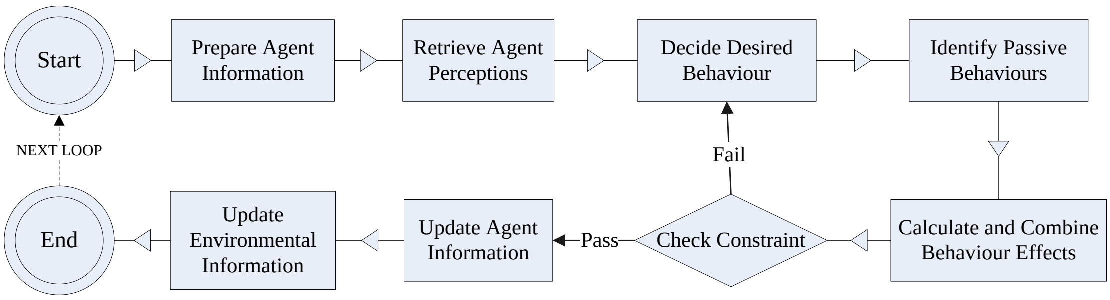


Figure 70 Agent action process

Prepare Agent Information

This step gathers the relevant information of an agent when it starts its action process. Agent information includes its attributes and status at Δt.

Retrieve Agent Perceptions

The agent perceptions are the information on surroundings that can be perceived by the agent. These perceptions are then used in the "decide desired behaviour" step. The maximum range of perceptions is the sight range of the agent. Additionally, the perceptions may also be limited by the agent's range attributes of certain behaviours and the agent's status.

The agent's knowledge of the map is considered as part of the perception in this step. However, because such knowledge is treated as existing knowledge of the agent, it will not be constraint by the range attributes.

In more detail, the perceptions include:

➢ **Information on other Agents**

- Not all other agents have to be perceived during the agent action process.
- When used as perception, only the attributes and status that are marked as public can be accessed.

➢ **Environment Objects**

Environmental objects have currently not been used in the simulation. They should be treated as an agent as well.

➢ **Environment Effect**

When the agent is positioned in a certain area it may receive certain visual area effects.

Decide Desired Behaviours

For a given scenario, an agent will have some purpose(s) and it will make decisions perform relevant behaviours to achieve its goal(s). For example, in the situation of exiting a building, the agent's goal will be to move to the exit of the building. The choice of route is determined by the agent’s position (as well as the agent’s information) at that moment. If the agent is in a room it may need to move to the door of that room first. If the agent is in a corridor, it may directly move to the known exit or it has to walk through a corner first. Or, it can decide to wait until the other agents nearby have escaped. No matter what decision(s) it makes, those behaviours will be interpreted by the Behaviour Library and the effect of behaviour will be calculated through corresponding formula.

At this phase, high level artificial intelligence can be integrated into the decision making process. However, at current stage, this crowd model neither focuses on how an agent makes a decision nor on the factors that could affect the decision, but only represents the end result of the decision making. It emphasises that a decision can be interpreted by the behaviours that are defined in the Behaviour Library (or to configure by the stand formula) and their effect can be calculated and combined via a unified formula mechanism. The expansion on high level AI is considered as an further step of this study.

Identify Passive Behaviours

The previous step describes the passive behaviours that can be performed by the agent in addition to its active behaviour which is decided in the above step. Passive behaviour means the behaviour that could be performed by the agent irrelevant to its goals. It refers to the behaviour that happens spontaneously or subconsciously.

In this crowd model, the behaviours that generate repulsive effects on the user are defined as passive behaviours. The details of when and how to calculate these passive behaviours effects are presented in section 5. 2. 4 Behaviour Library.

Calculate and Combine Behaviour Effects

In this step, the agent will calculate the combined effect of the behaviours that were decided in the previous two steps. This can be done in three steps:

- **Step 1**: Calculating the effect of each individual behaviour that is identified. The calculation methods are defined in the Behaviour Library.

- **Step 2**: Using the combining mechanism that was discussed in 5. 2. 2. 7 and Combining Behaviour Effects to calculate a total effect.
- **Step 3**: Capping the calculated total effect based on the agent's attributes.

Check Constraint

This step tests whether the total behaviour effect calculated by the above step will cause a collision. It simply checks if the new position after the behaviour effect is applied will cause a collision with other objects (e.g. other agents, walls, or obstacles).

If there is a potential collision, then the agent will half its total behaviour effect and the check is redone. If the final effect is less than 1% of the total behaviour effect (equal to when it is halved seven times by seven times) the agent will stop moving at this time frame (which sets the final effect to zero).

If there is no potential collision, then the total behaviour effect will be used at the final behaviour effect.

Update Agent Information

In this step, the agent status will be updated. The final behaviour effect will be used to calculate the new position and orientation of the agent.

If any behaviours could affect the agent's other status, this status will be updated in this step as well.

Update Environmental Information

If the behaviours of the agent interact with the surroundings (other agents and the environment) the relevant status will be updated at this stage; for example, opening a door or removing an obstacle. Whether a behaviour has interaction and how it affects the surroundings are defined in the Behaviour Library.

5. 2. 4 Behaviour Library

In the above sections, a unified formula to calculate the behaviour effects and an agent model to represent the individual have been both introduced. There is a requirement to link these two to reflect different kinds of agent's behaviours. As a result, a Behaviour Library is proposed to contain such information.

The Behaviour Library can be seen as a collection of behaviours and explanations as to how to calculate the behaviour effects of different behaviours by utilising the unified formula and the attributes of the agent. The Behaviour Library is an intermediate layer between the force-based model and the agent-based model. On the one hand, this layer refers to the unified formula that is defined in the force-base model to represent different behaviours. On the other hand, it explains how the agent's attributes and other information in the agent-based model could affect the value of the parameters in the formula.

The values of the parameters in the formula are determined by behaviours and each behaviour can has its own mechanism. The characters, abilities and status of individuals, environmental issues and surrounding situations are also taken into account and demonstrated during the process of calculation.

The Behaviour Library contains following behaviours:

- **Seeking (move to)**: the agent walks towards a target.
- **Walking away from**: the agent walks away from a target.
- **Following**: the agent follows the movement of another agent.
- **Keeping a distance from another agent**: the agent feels a repulsive effect from another agent which causes the agent to keep a certain distance from another agent.
- **Keeping a distance from a wall**: the agent feels a repulsive effect from a wall which causes the agents to keep a certain distance from the wall.
- **Avoiding collision**: the agent avoids the target on purpose by keeping a certain distance when walking around it.
- **Walking towards the group**: the agent walks to a position which has the most agents in density.
- **Aligning direction with the group**: the agent adjusts its heading direction which is determined by the average direction of nearby agents.

- **Handling repulsive effect from nearby crowd:** the agents feels repulsive effects from all nearby crowd.
- **Keeping in a group**: the agent wants to maintain its position within a group.

5. 2. 4. 1 Simple Behaviours

A simple behaviour refers to a straightforward behaviour which usually only include one behaviour target. Currently, the Behaviour Library includes eight simple behaviours.

Seeking (Moving to)

➢ **Behaviour Description**

This behaviour describes the movement whereby an agent finds its way to some destination. It is modelled as the agent moving towards the target directly. Figure 71 demonstrates the walking direction when an agent is seeking a target. In this figure, the circle represents the position of the agent and the dot represents the position of the target. The arrow indicates the walking direction of the "seek" behaviour.

Figure 71 Illustration of "Seeking" behaviour

➢ **Behaviour Effect Calculation**

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is the position of the target which the agent seeks. The target is determined in the "decide desired behaviour" phase during the agent action process.
- α equals $0°$ in this behaviour because the agent is moving directly towards the target.
- F_a indicates the agent's mental state while walking. The default value of F_a is 1 which indicates that the agent is approaching the target under normal

circumstances. The value can be below 1 if the agent wants to slow down or the value can go above 1 if the agent is in a hurry to reach the target.

- F_t reflects the degree of the attraction from the target to the agent. The default value is 1 representing an ordinary target. A value above 1 indicates the target has more weighting to attract the agent. For example, when the agent is following a certain route (consisting of a collection of waypoints), each waypoint is the target of the "seek" behaviour. All the other waypoints will have the F_t value at 1 to create a smooth walking behaviour while the final waypoint can have a high TF value to represent the fact that the goal point has a bigger attraction for the agent.
- F_d equals 1 in the calculation to create an equal behaviour effect at all positions because this behaviour is considered irrelevant to distance.

The formula for "seeking" behaviour is:

$$Seeking\ Effect = Rotation(Normalise(P_t - P_a)\ ,0))E_s F_a F_t$$

Formula 46 Formula for "seeking" behaviour effect

Walking away from

➢ Behaviour Description

This behaviour describes an opposite movement to "seeking" behaviour which introduced above. It is modelled as the agent moving away the target directly. Figure 72 demonstrates the walking direction when an agent is seeking a target. In this figure, the circle represents the position of the agent and the dot represents the position of the target. The arrow indicates the walking direction of the "walking away from" behaviour.

Figure 72 Illustration of "walking away from" behaviour

➢ Behaviour Effect Calculation

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is the position of the target which agent wants to walk away from. The target is determined in the "decide desired behaviour" phase during the agent action process.
- α equals 180° in this behaviour because the agent is moving towards the direction opposite the target.
- F_a indicates the agent's mental state while walking. The default value of F_a is 1 which indicates that the agent is leaving the target under normal circumstances. The value can be below 1 if the agent wants to slow down or the value can go above 1 if the agent is in a hurry to leave the target.
- F_t reflects the degree of the repulsion from the target to the agent. The default value is 1 representing an ordinary target. A value above 1 indicates the target has more weighting to push away the agent. For example, if the target is a fire, the agent may want to leave it faster thus F_t should have a higher value in this case.
- F_d equals 1 in the calculation to create an equal behaviour effect at all positions because this behaviour is considered irrelevant to distance.

The formula for "walking away from" behaviour is:

$$Walking\ away\ from\ Effect = Rotation(Normalise(P_t - P_a)\ \ ,180))E_s F_a F_t$$

Formula 47 Formula for "waking away from" behaviour effect

Wandering

➢ Behaviour Description

Wandering means that the agent moves randomly or moves without a goal. Movement is considered to be a smooth trajectory rather than a totally irregular trajectory. In this model the wandering behaviour is defined as "during each update interval (i.e frame), the agent will turn a random angle between [-ɵ,+ɵ] which happens at a certain probability".

➢ Behaviour Effect Calculation

The behaviour effect of wandering can be calculated using the unified formula with the following settings applying:

- P_t denotes the position of a virtual target which is located in front of the agent. The virtual target motivates the agent to keep moving forward. The distance of this virtual target from the agent does not matter due to the *Normalise* operation.
- α indicates possible walking direction change in the wandering behaviour. Its value is chosen randomly from the interval $[-\theta, +\theta]$ with a certain probability at each update process of the crowd model. According to some studies (Reynolds 1999; Couzin et al. 2005), on modelling this behaviour the random angle should be constrained by a time-dependent function to prevent a twitchy moving trajectory. In the case of the crowd model updating at an interval of 1/60 second, the function to determine α was given by the formula below (*this function and its parameters will be discussed in the following paragraph after the parameters' list*).

$$at\ each\ update,\ \alpha := f(\theta) = \begin{cases} Random\ ([-18\,°,\ +18\,°])\ , & at\ 5\%\ probability, \\ 0 & ,\ at\ 95\%\ probability \end{cases}$$

Formula 48 Default function to determine α at each frame for "wandering" behaviour

- F_a indicates the agent's mental state while wandering. The default value of F_a is 1 which indicates that the agent is moving randomly under normal circumstances. A value larger than 1 means the agent could be anxious while a value less than 1 could indicate that the agent is in a casual status.
- F_t equals 1 because this virtual target does not affect the value of the behaviour effect. The value of the behaviour effect is only affected by the agent's state.
- F_d equals 1 because the behaviour effect is irrelevant to distance. The value of the behaviour effect is only affected by the agent's state.

The formula to calculate the wandering effect is:

$$Wandering\ Effect = Rotation\big(Normalise(P_t - P_a)\ ,f(\theta)\big)E_sF_a$$

Formula 49 Formula for "wandering" behaviour effect

➢ The Function that Decides the Behaviour Angle

The function that decides the behaviour angle can be explained as the agent turning its moving direction at a random angle at a random time while moving. To represent this behaviour through the unified formula it can be treated as the agent selecting a random target and performs the "seek" behaviour while walking. This representation can be simply described as: at each Δt the agent randomly chooses an angle α to turn based on its current facing (walking) direction.

The formula to calculate the wandering effect is:

$$Wandering\ Effect = Rotation(current\ direction, \alpha)E_sF_a$$

Formula 50 An alternative formula for "wandering" behaviour effect

In practice, such a randomly chosen angle α cannot produce a wander behaviour that one would expect. As a result, "it is 'twitchy' and produces no sustained turns" (Reynolds 1999). Reynolds suggested that limiting the angle α to a small value at each time Δt and increasing the probability of turning to the same side could achieve a better and more natural behaviour. A similar implementation (Couzin et al. 2005) has been used in simulating animal movements as well. The author considers this approach to be a good implementation as wandering behaviour should produce a smooth trail rather than a totally random line in statistics.

In this model the wandering behaviour is defined as "at each time Δt (in the simulation, the time Δt is the animation update interval) the agent will turn a random angle between [-ө,+ө] which happens at a certain probability". The walking trajectory is decided by three parameters: time Δt, angle range [-ө,+ө], and the probability of the turning action. Time Δt is determined by the graphic engine. In this study, the simulation has an update rate of 60 frames per second which means Δt is 0.167 seconds. The following configurations have been tested in order to find a combination of angle and probability in order to create a smooth moving trajectory:

Table 7 Wandering settings

Wandering Setting No.	Angle Range	Probability of the turning action
1	$[-180\ °, +180°]$	100%

2	$[-18°, +18°]$	50%
3	$[-18°, +18°]$	5%

By comparing the wandering trajectories in these settings, setting number 3 has been chosen to represent the wandering behaviour in this crowd model. In other words, the angle is set to $[-18°, +18°]$ and in each time frame the probability to change an angle is set to 5% for the wandering behaviour to create a smooth wandering trajectory.

Following

➢ Behaviour Description

Following is the behaviour when an agent tries to keep walking behind somebody. This behaviour can be interpreted as seeking a virtual target position behind the actual target. Figure 73 illustrates the positional relationships of the agent, its target, and the virtual target position in "following" behaviour. The two big circles are the agent and its target (with a dash to indicate its orientation). The small circle is the virtual position that the agent wants to walk toward which is located somewhere behind the target. The distance between virtual position and the target is given by the agent's desired distance to follow the target.

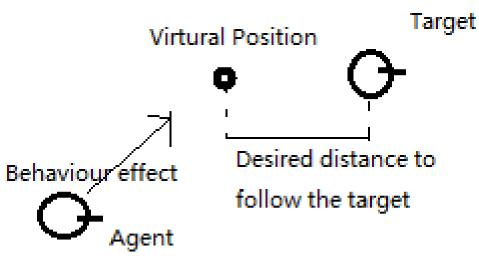


Figure 73 Illustration of "following" behaviour

➢ Behaviour Effect Calculation

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is defined as a virtual position. Its location is given by the following formula (where P_{target} is the position of the agent's following target,

$D_{desire\ follow}$ is the desired following distance, θ is the orientation of the target):

$$P_t = P_{target} - \left(D_{desire\ follow}\, sin\,\theta\, , D_{desire\ follow}\, cos\,\theta\,\right)$$

Formula 51 Calculation for the Virtual position in "following" behaviour

- α equals 0° because the agent is moving directly towards the virtual position.
- F_a has a default value of 1 to reflect normal circumstances. Its value will be larger if the agent is in a hurry and smaller if the agent is not in a hurry.
- F_t has a default value of 1 to represent an ordinary target. A value larger than 1 indicates the target has more weighting to attract the agent and vice versa.
- F_d equals 1 because this behaviour is irrelevant to distance.

The formula is:

$$Following\ Effect = Rotation(Normalise(P_t - P_a)\ \ ,0)E_sF_aF_t$$

Formula 52 Formula for "following" behaviour effect

Keeping a Distance from an Agent (Repulsive Effect)

➢ Behaviour Description

This behaviour describes the agent's willingness to keep a certain distance from another agent. Such behaviour can result in a repulsive effect. This repulsive effect is inversely proportional to the distance between the agent and its target which is the parameter F_d in the formula. The concept of this repulsive effect was inspired by the repulsive forces in the Social Force model (Dirk Helbing & Peter Molnar 1995; D. Helbing et al. 2000).

Differently from the Social Force model, the repulsive effect in this crowd model is considered irrelevant to the agent's mass. In addition, in this study, a boundary is introduced to define a range in which this behaviour can occur because the agent should not receive any repulsive effect from the target once it has reached the comfortable distance from the target. In other words, this behaviour will not be triggered if the target is outside the range of the desired distance because in that case the agent has already kept an adequate (or more than adequate) distance from its target. This boundary is reflected as the desired distance from others in the agent's attributes.

Furthermore, when the distance is closer than a certain value, the repulsive effect will reach its maximum. This represents the situation where the agent feels very uncomfortable and wants to reach a longer distance from the target as fast as it can. This distance is represented by the minimum distance from others attribute of an agent.

Behaviour Effect Calculation

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is the position of "another agent" which is the behaviour target.
- α equals $180°$ because this behaviour represents a repulsive effect and the agent is moving away from the behaviour target.
- F_a has a default value of 1 to reflect normal circumstances. A higher value indicates that the agent is sensitive to the nearby others and wants to pursue the desired distance faster and vice versa.
- F_t has a default value of 1 to indicate an ordinary behaviour target. A higher value indicates that the target has some features that can drive others to move away from it quickly, for example, the target agent could be dirty and smelly so he/she generally produces a larger repulsive effect and vice versa.
- F_d is considered to reflect the agent's following reactions:
    - If the target is too close to the agent, the agent will try its best to move away from the target.
    - If the target is too far from the agent, the agent simply ignores that target and feels no repulsive effect.
    - If the target is within a certain range, the agent will received a repulsive effect from the target. Such an effect is represented by a decreeing function depending on the distance between the two.

  As a result, the parameter F_d is represented by a piecewise function $g(d)$ which has a value between 0 to 1 in which *d* stands for the distance between the agent and the target. The distance that the agent starts to feel too close is defined as the minimum distance from others - $D_{minimum\ agent}$. The distance that the agent starts to ignore the target is defined as the desired distance from others - $D_{desire\ agent}$. It is proposed that the function $g(d)$ is calculated as follows whereby $g(d) = k/d$, where *k* is a constant coefficient (*The appropriate value of k is associated with Δt and the unit of distance in the crowd model. With the default settings in this crowd model, the unit of distance is in pixel (scale: 1 pixel = 0.05 meter) and k is set to 1.).*

$$F_d := g(d) = \begin{cases} 0\,, \left(d \geq D_{desire\ agent}\right) \\ \dfrac{k}{d}, \left(D_{minimum\ agent} < d < D_{desire\ agent}\right) \\ 1\,, \left(d \leq D_{minimum\ agent}\right) \end{cases}$$

Formula 53 Distance function for repulsive effect (agent)

The formula to calculate the behaviour effect becomes:

$$Repulsive\ Effect\ from\ an\ agent = Rotation(Normalise(P_t - P_a)\ ,0)\ E_s F_a F_t g(d)$$

Formula 54 Formula for “Keeping a distance from an agent (repulsive effect) behaviour effect

Keeping a Distance from a Wall (Repulsive Effect)

➢ Behaviour Description

This behaviour describes the behaviour whereby an agent tries to keep a certain distance from a wall. The parameters and calculation in this behaviour is similar to the above behaviour “keeping a certain distance from another agent”.

Because a wall is represented by a line in this model, this means it is a collection of positions. As the formula requires a target position to calculate the behaviour effect, this behaviour requires finding a target position based on the information possessed by the wall. In this research study, the position of the target is proposed as the projection of the agent’s position to the wall. In this way, the behaviour produces an effect which drives the agent away from the wall via the shortest route which is in the perpendicular direction from the wall to the agent. The position of target P_t is demonstrated as follows:

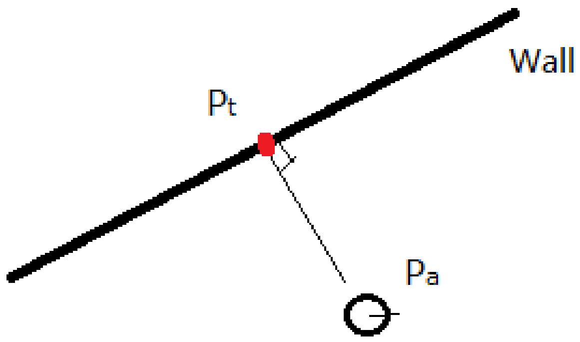


Figure 74 The position of the behaviour target in "keeping a distance from a wall" behaviour

➢ Behaviour Effect Calculation

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is the position of the agent's projection on the wall.
- α equals 180° because this behaviour represents a repulsive effect and the agent is moving away from the wall.
- F_a has a default value of 1 to reflect normal circumstances. A higher value indicates that the agent is sensitive and wants to pursue the desired distance faster and vice versa.
- F_t has a default value of 1 to indicate an ordinary behaviour target. A higher value indicates the wall has some features to drive the agents move away from it. For example, a wall seems about to collapse will make the agent move away more quickly.
- F_d is considered to reflect the agent's following reactions:
  - If the wall is too close to the agent, the agent will try its best to move away from the wall.
  - If the wall is too far from the agent, the agent simply ignores that wall and feels no repulsive effect.
  - If the wall is within a certain range, the agent will received a repulsive effect from the wall. Such an effect is represented by a decreeing function depending on the distance between the two.

As a result, the parameter F_d is represented by a piecewise function $g(d)$ which has a value between 0 to 1 in which d stands for the distance between the agent and the wall. The distance that the agent starts to feel too close is defined as the minimum distance from walls - $D_{minimum\ wall}$. The distance that the agent starts to ignore the wall is defined as the desired distance from walls - $D_{desire\ wall}$. It is proposed that the function $g(d)$ is calculated as follows whereby $g(d) = k/d$, where k is a constant coefficient (*The appropriate value of k is associated with Δt and the unit of distance in the crowd model. With the default settings in this crowd model, the unit of distance is in pixel (scale: 1 pixel = 0.05 meter) and k is set to 1.).*

$$F_d := g(d) = \begin{cases} 0\,, (d \geq D_{desire\ wall}) \\ \dfrac{k}{d}, (D_{minimum\ wall} < d < D_{desire\ wall}) \\ 1\,, (d \leq D_{minimum\ wall}) \end{cases}$$

Formula 55 Distance function for repulsive effect (wall)

The formula to calculate the behaviour effect becomes:

$$Repulsive\ Effect\ from\ a\ wall = Rotation(Normalise(P_t - P_a)\ ,0)\ E_s F_a F_t g(d)$$

Formula 56 Formula for "Keeping a distance from a wall (repulsive effect) behaviour effect

Avoiding Collision

➢ Behaviour Description

This behaviour describes an agent adjusts its moving direction in order to walk around a target by keeping a desired distance. This behaviour only happens if the agent's current behaviour will result in a collision. For example, this behaviour will happen when the agent is moving forward and it is going to collide with an obstacle. Figure 75 illustrates the "avoid collision" behaviour: the agent will adjust its direction with a certain angle α to avoid the collision (The agent can either turn left or turn right, the angle is defined as α or $-\alpha$).

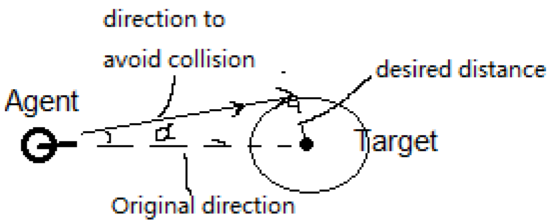


Figure 75 Illustration of 'avoiding collision' behaviour

- **Behaviour Effect Calculation**

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is the position of the object that will collide with the agent.
- α is the angle by which the agent will adjust its moving direction. It is calculated by the following formula:

$$\alpha := h(d) = r() \cdot sin^{-1}\left(\frac{R_a + R_t + D_{avoid\ distance}}{d}\right)$$

Formula 57 Calculation for the behaviour angle in "avoiding collision" behaviour

In this formula, the function $r()$ returns a value of 1 or -1 to indicate whether the agent goes left or right. R_a is the radius of the agent. R_t is the radius of the target. $D_{avoid\ distance}$ represents the desired distance that the agent wants to keep while avoiding collision. d denotes the distance between the agent and the target.

- F_a is set to the same value of the intended behaviour before detecting the collision.
- F_t has a default value of 1 to represent an ordinary obstacle. A value above 1 indicates the obstacle has more weighting to push away the agent.
- F_d equals 1 because the distance factor has been considered in the calculation of behaviour angle α.

The formula is:

$$Avoiding\ Collision\ Effect = Rotation\big(Normalise(P_t - P_a), h(d)\big)E_sF_aF_t$$

Formula 58 Formula for "avoiding collision" behaviour effect

Walking towards a Group

- **Behaviour Description**

This behaviour describes the movement whereby an agent tries to manoeuvre its position to the centre of a group. (A similar behaviour called "cohesion" was presented in Reynolds' study (1987).) The centre of the group is defined as the average position of all the agents in that group rather than the geometric centre. The group contains the people who are within a certain range of the agent. The relationships are illustrated in Figure 76: the small circles with a dot indicating their orientations represent the crowd. The large circle indicates the group boundary and only the agents inside the circle will be considered as a group (on this occasion, the group contains ten agents). The five agents outside the circle are not considered and have no effect on this behaviour. The agent is at the centre of that circle. However, the centre of the group represents the average positions of the group which is highlighted by a solid black dot.

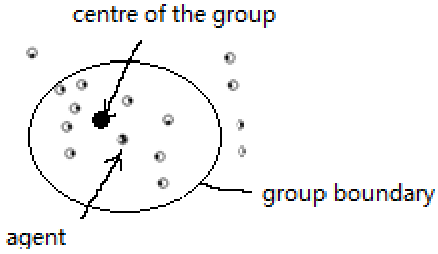


Figure 76 Illustration of the group and its boundary in "walking towards the Group" behaviour

➢ Behaviour Effect Calculation

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is a virtual position that represents the average position of the group. Assuming that the group contains *N* agents and $P(x_i, y_i)$ represents the position of agent *i*, P_t can be calculated by:

$$P_t := P_{average} = \frac{\sum_i^N P(x_i, y_i)}{N}$$

Formula 59 Calculation for the average position of a group

- α equals $0°$ because the agent is moving directly towards the virtual position.
- F_a has a default value of 1 to reflect normal walking circumstances. Its value will be higher if the agent is in a hurry and lower if the agent is not in a hurry.
- F_t equals 1 because this virtual target only has a location effect.
- F_d equals 1 because this behaviour is irrelevant to distance.

The effect of the "walking towards the group" can be calculated by:

$$Walking\ towards\ the\ Group\ Effect = Rotation\big(Normalise\big(P_{average} - P_a\big), 0\big)E_sF_a$$

Formula 60 Formula for "walking towards the group" behaviour effect

Aligning Direction with the Group

➢ Behaviour Description

In this behaviour, the agent changes its walking direction to the average direction of the nearby group. This behaviour describes when an agent aligns its moving direction to the group (A similar behaviour, called "alignment", was presented in Reynolds' study (1987).) The group has the same definition as in "walking towards the group" behaviour. The group direction is defined as the average moving direction of all the other agents in the group. In this model, this behaviour is interpreted as the agent seeking a virtual target that represents the average direction of the group.

➢ Behaviour Effect Calculation

The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is the position of a virtual target that attracts the agent walking in the same direction of the group. Assuming that the group contains *N* agents, where $O(x_i, y_i)$ represents the walking direction of agent *i*, the position of this virtual target must satisfy the following equation:

$$Normalise(P_t - P_a) = Normalise\Big(\sum_{i \neq self}^{N} O(x_i, y_i)\Big)$$

Formula 61 Requirement of P_t in "aligning direction with the group" behaviour

- α equals 0° because the agent is moving directly towards the virtual position.
- F_a has a default value of 1 to reflect normal walking circumstances. Its value will be higher if the agent is in a hurry and lower if the agent is not in a hurry.
- F_t equals 1 because this virtual target only affects the direction of the behaviour.

- F_d equals 1 because this behaviour is irrelevant to distance.

The behaviour effect can be calculated by the formula (where P_t is constrained by Formula 61):

$$Aligning\ Direction\ with\ the\ Group\ Effect\ = Rotation(Normalise(P_t - P_a), 0)E_sF_a$$

Formula 62 Formula for "aligning direction with the group" behaviour effect

➢ Alternative Calculation for the Behaviour Effect Direction

For a group of *N* nearby agents, their average orientation $O_{average}$ is calculated by:

$$O_{average}\ = Normalise(\frac{\sum_{i \neq self}^{N} O(\ x_i\ ,\ y_i)}{N})$$

Formula 63 Average orientation of a group (excluding the agent itself)

The behaviour direction is the average orientation of the group, which implies "$O_{average}$" is an equivalent representation of the behaviour effect direction. As a result, the formula to calculate this behaviour effect can be simplified to:

$$Aligning\ Direction\ with\ the\ Group\ Effect = Normalise(\frac{\sum_{i}^{N} O(\ x_i\ ,\ y_i)}{N})E_sF_a$$

Formula 64 An alternative formula for "aligning direction with the group" behaviour effect

5. 2. 4. 2 Combined Behaviours

A combined behaviour involves more than one of the simple behaviours and usually describes a more meaningful behaviour. There are two combined behaviours in the Behaviour Library at the moment.

Handling Repulsive Effect from nearby Crowd

➢ Behaviour Description

When an agent is in a crowd, behaviour "keeping a distance from an agent" is applicable to all other agents in that crowd. To simplify, these effects are usually treated as an overall repulsive effect from the whole crowd rather than the effects from each agent individually. This behaviour rule describes the agent receiving an overall repulsive effect from the crowd (the combination of the repulsive effects from everybody) which pushes it away from others in the crowd.

➢ Behaviour Effect Calculation

All the parameters used in this behaviour have the same natures as they are in "keeping a distance from an agent" behaviour. Because this behaviour represents a combined effects of a group contains *N* agents, the representations of the parameters may be slightly different:

- P_t is replaced by P_i which denotes the position of the *i*th agent.
- α equals $180°$ because this behaviour represents a repulsive effect and the agent is moving away from the behaviour target.
- F_a still refers to the agent's Self Factor.
- F_t is replaced by F_{ti} which denotes F_t for the *i*th agent.
- F_d is represented by the same function $g(d)$ which is defined in "keeping a distance from an agent" behaviour (refer to Formula 53).

The formula to calculate the behaviour effect becomes:

$$Repulsive\ effect\ from\ nearby\ crowd = \sum_{i \neq self}^{N} Normalise(P_a - P_i) E_s F_a F_{ti} g(d)$$

Formula 65 Calculation for repulsive effect from nearby crowd

Keeping in a Group

➢ Behaviour Description

This behaviour describes the agent trying to position itself in a group. It includes two effects according to literature: (a) a cohesion effect that moves one to the average position of nearby individuals (Reynolds 1987); (b) an alignment effect that adjusts one's walking direction towards the average heading of nearby individuals (Reynolds 1987; Couzin et al. 2005).

As a result, this behaviour can be represented by combing two existing behaviours in the library: "walking towards a group" and "aligning direction with the group". The group in this behaviour has the same definition as in those two behaviours.

Behaviour Effect Calculation

The behaviour effect contains two aspects. However, their calculations will use the same set of parameters. The behaviour effect can be calculated using the unified formula with the following settings applying:

- P_t is a virtual position that represents the average position of the group. The average position $P_{average}$ has the same calculation method as it is in "walking towards a group" behaviour (refer to Formula 59). P_t is only applicable to the first aspect of the behaviour as the alternative calculation formula is used for the second aspect.
- α equals $0°$ because the agent is moving directly towards the virtual position.
- F_a has a default value of 1 to reflect normal walking circumstances. Its value will be higher if the agent is in a hurry and lower if the agent is not in a hurry.
- F_t equals 1 because the virtual target in the first aspect of this behaviour only provides location information and in the second aspect of the behaviour it is not applicable.
- F_d equals 1 because this behaviour is irrelevant to distance.

Additionally, because the second aspect of the behaviour effect is calculated via the alternative formula, the average orientation $O_{average}$ of the group is required (its calculation can be referred to Formula 63).

To sum up, the behaviour effect of "Keeping in a group" can be calculated by the formula:

$$Keeping\ in\ a\ Group\ Effect = \left(Rotation\left(Normalise\left(P_{average} - P_a\right), 0\right) + O_{average}\right)E_sF_a$$

Formula 66 Behaviour effect calculation for keeping in a group

5. 3 How the Crowd Model Works

5. 3. 1 The Workflow of the Crowd Model

In a simulation to represent a crowd phenomenon or behaviour, the agents usually have a different combination of behaviours at different times. An agent will make a decision based on its observation. The decision will be interpreted as a collection of behaviours in the Behaviour Library.

Applying this crowd model in a simulation can be achieved through three stages: pre-simulation, in-simulation and post-simulation.

5. 3. 1. 1 Pre-simulation stage

At this stage, two things need to be identified and interpreted through the model.

Environmental Information

Before the simulation starts, the environmental information must be transferred into a format that can be understood by the agents. It contains two aspects (navigation map and objects' information):

✧ ***Map***

The geometrical information on the environment will be interpreted by a navigation map that can be used by the agents to generate routes and waypoints. This map will become part of the agents' knowledge.

✧ ***Objects***

The objects (excluding agents) in the environment will be modelled and become the agent's perceptions during the simulation. At this point, the attributes of the following objects need to be determined:

- Wall: The simulation environment may contain walls. Their position and length information can be identified. The repulsive modifier will also be determined.
- Obstacle: The position, size and repulsive modifier attributes will be decided at this stage.

- Virtual area effect: A virtual effect is defined by an area and the effect on the agents in this area. The effect can either affect the value of an agent's attributes or can affect the decision making of an agent. In the former case, it will define how the attributes change when the agent is in the area. In the latter case, it works as an additional rule to decide an agent's behaviour.

Agent Information

✧ ***Identifying behaviours***

Prior to the simulation, the simulation scenario should be identified and interpreted by the behaviours that are defined in the model. If a behaviour is not defined in the Behaviour Library, it should firstly be checked if a combination of existing behaviours could achieve that behaviour; if not, it requires using the core behaviour to represent its effect.

However, this crowd model does not provide a comprehensive guide on how to do this as it is beyond the scope of this study. By an example presented later in this chapter and the evaluation and validation simulations in Chapter 7 and Chapter 8, the author will demonstrate the process of how to analyse given scenario.

✧ ***Defining agents' information***

The information on the agents can be identified and determined at this stage. It should include everything defined in the agent design section (5. 2. 3).

5. 3. 1. 2 During-simulation Stage

After the simulation starts, every agent will use its agent action engine to perceive, make decisions and act. Each agent will behave independently and repeat the agent action process at each time frame during the simulation.

During this process, the graphic engine will keep updating the animation of the agents' behaviours (movements) so it can be observed by the real time simulator.

Because everything has been clearly defined in the crowd model and all the information has been determined at the pre-simulation stage, the calculation and information updating are straightforward.

5. 3. 1. 3 Post-simulation Stage

A simulation is the representation of a real scenario. The end of a simulation can be once it reaches a certain situation; for example, when all the people have exited a room the goal has been reached.

The simulation programme can capture the useful information within the simulation for further analysis. All the information that is defined in the agent design and the process of the simulation can be recorded for post analysis.

5. 3. 2 Example: Two Agent Walking Through a Corridor

This example simulates a scenario of two persons walking through a corridor of 1.5 meters' width. The simulation environment is illustrated in Figure 77. The arrow indicates the agents' waking direction. The one in the front is Mr Grey (as he is presented in grey colour). The one in the back is Mr White (as he is presented in white colour). Mr White walks twice as fast as Mr Grey and will be able to overtake Mr Grey before he reaches the end of the corridor. Additionally, Mr White's comfortable position in the corridor is to maintain equal distance from the two side walls while Mr Grey's desired distance to a side wall equals to his initial distance to the top side wall (their initial positions have reached the desired distances).

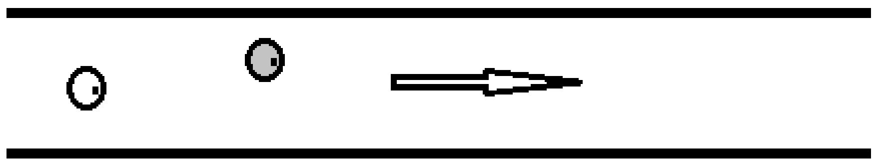


Figure 77 Two agents walking through a corridor

The purpose of this example is to demonstrate how an observed phenomenon is interpreted by the proposed crowd model and the movements of the two agents are represented by the combinations of the behaviours in the Behaviour Library. The positions of the two agents in the corridor are captured to show some key stages during the simulation process. For each stage, their behaviours are analysed and the effects of the behaviours are also demonstrated.

5. 3. 2. 1 Positional Description in this Thesis

Prior to presenting the example, the author considers that it is worthy to explain how the positions and directions are described in this thesis in order to avoid any confusion and misunderstanding.

In the thesis, the terms of “east, south, west, and north” are used to describe the positions and directions in the simulation environment, which are the same as in a standard map. For example, in Figure 78, the agent is facing east. Its back is the position of west. The direction of north is on its left-hand side and south is on it right-hand side. If the agent is described as moving forward with its current orientation, it is moving to the east.

(However, to keep the fluency of natural language, this thesis may still use right-hand side (east), bottom (south), left-hand side (west), and top (north) in describing a figure.)

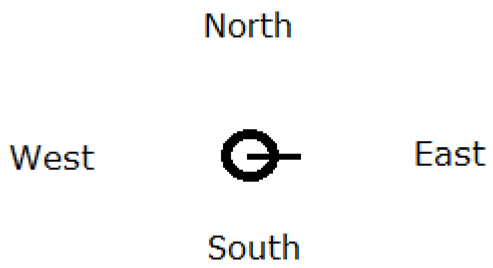


Figure 78 Illustrations on the directions and positions in the thesis

As a result, the two agents in this example can be described as walking through the corridor from west to east.

5. 3. 2. 2 Pre-simulation Stage

Analysing the Simulation Scenario

At the beginning, the two start to move forward (to the east). Because they have reached their comfortable positions in the corridor, they will move straightforward.

After a while, when Mr White gets close to Mr Grey, he cannot walk straightforward to pass Mr Grey because that will cause a collision. In order to prevent the collision, he will walk around and overtake Mr Grey (in this case he will move south to make enough space). In terms of Mr Grey, he soon notices that Mr White is coming from behind with a much higher speed. On the one hand, he wants to step aside (moving north) to give some room to Mr White but, on the other hand, he does not like to get too close to the wall.

After Mr White overtakes Mr Grey, although there is no potential collision, the two will still change their walking direction to adjust themselves in order to reach a comfortable position in the corridor. Mr White wants to stay in the middle between two sidewalls while Mr Grey only needs to keep a shorter distance from the sidewall.

In the end, the two reach their comfortable positions again and proceed to the east end of the corridor.

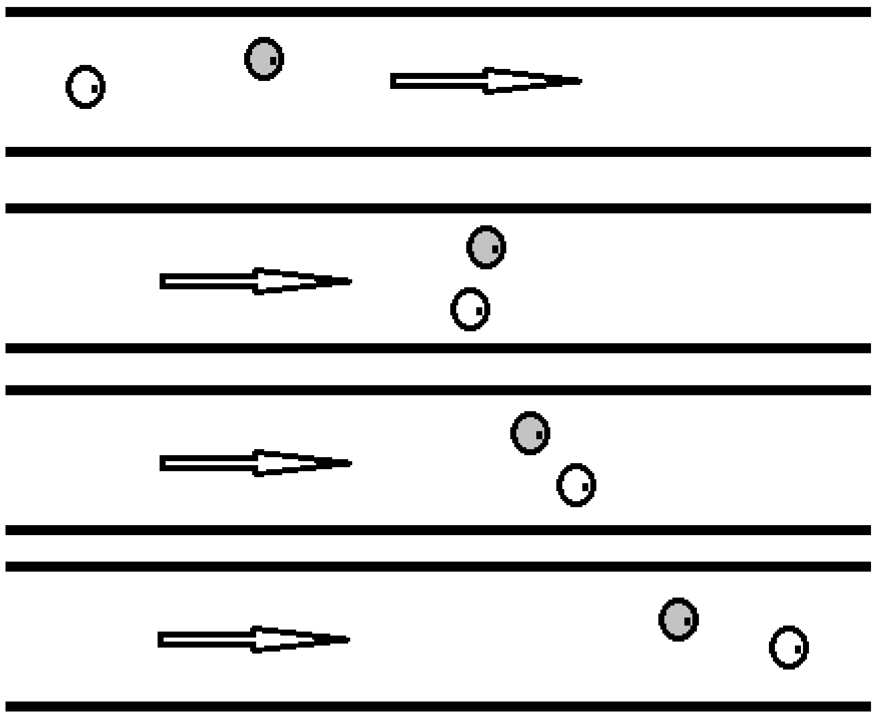


Figure 79 Mr White and Mr Grey walks in a corridor

The trajectories of the movement for both agents are showed in Figure 80.

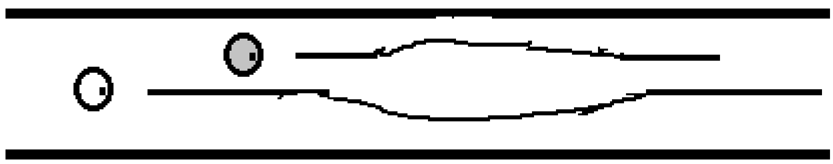


Figure 80 Walking trajectories of the two

To summarise, this corridor walking scenario can be divided into four phases:

- **Phase1**: start walking -> before catching up

- **Phase2**: catching up -> overtaking -> overtaken
- **Phase3**: after overtaken
- **Phase4**: long after overtaken -> reach the destination

Interpreting the Scenario by using the Behaviour Library

To simulate this scenario, the individual behaviours that can be observed form the scenario need to be identified first. These behaviours will be analysed and interpreted with the behaviours in the Behaviour Library. Table 8 below lists both the identified individual behaviours from the scenario and their corresponding behaviours in the Behaviour Library.

Table 8 Individual behaviours and their corresponding behaviours in the Behaviour Library

Behaviour observed in the scenario	Behaviour in the Behaviour Library
Walk to the east end of the corridor	Seeking (Move to)
Keep a desired distance from the sidewall	Keeping a distance from a wall
Avoid collision with others	Keeping a distance from an agent

Deciding Environmental Information

In this example, the simulation environment is very simple. The corridor is formed by two walls and there is no obstacle in the corridor. The width of the corridor is 2 metres.

Determining Agent Information

This step is to assign values to the agents' personal parameters. As different behaviours between the two agents have been observed, their parameters are slightly different. The related parameters are listed below:

Table 9 Settings of personal parameters for Mr White and Mr Grey

Personal parameter	Mr White	Mr Grey
Default speed (m/s)	1.4	0.7

Desired distance from walls (m)	0.5	0.3
Desired distance from others (m)	1	1

5. 3. 2. 3 In-simulation Stage

After configuring the simulation with the identified information, the simulation is ready to begin. This section demonstrates agents' behaviours at each phase of the simulation and how these behaviours affect the movement of the agents. The illustrations on behaviours and the calculations of behaviour effects are mainly **based on Mr White's point of view** as the same principles can be applied to Mr Grey as well.

Phase 1: Start Walking -> Before Catching up

At the beginning, because both agents have reached their comfortable positions in the corridor and are far away from each other, they are only walking under the effect from the behaviour "seeking". As a result, both agents will walk straightforward down the corridor (i.e. to the east end). The behaviour effects on both agents are illustrated in Figure 81 (the arrows indicate the walking direction of the agent).

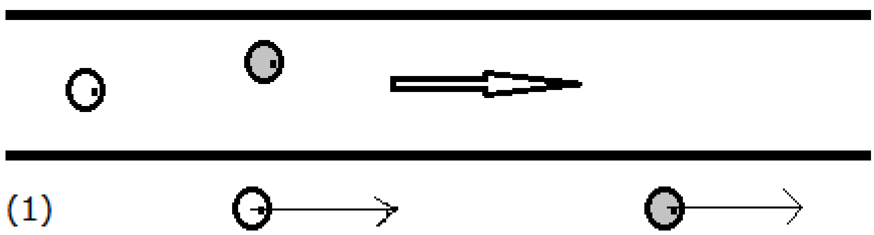


Figure 81 Illustration of the effects that agents received at the beginning

Phase 2: Catching up -> Overtaking -> Overtaken

When Mr White catches up with Mr Grey, they both need to manoeuvre their positions while moving forward (i.e. east). All three behaviour rules have effects on them at this phase. The sum of these three effects will generate a total effect on the agents (the total effect cannot exceed the maximum speed effect). During this phase, according to the relative positions between the two agents and the sidewalls, the possible directions of total effect are illustrated in Figure 82.

- **Just catching up**

This is the point when Mr White just gets close enough to Mr Grey. Because the distance between the two agents becomes shorter than Mr White's desired distance from others, he starts to feel a repulsive effect that pushes him away from Mr Grey. Additionally, Mr White still wants to walk to the east end of the corridor so he is receiving the behaviour effect of "seeking". Furthermore, at this moment, Mr White feels his distance to the sidewall is comfortable thus he feels no repulsive effect from the wall. The total behaviour effects and their overall result are illustrated in Figure 82 situation 1.

- **Overtaking**

As Mr White moves forward, his distance to the sidewall (the wall on his right-hand side, i.e. the bottom wall in Figure 82) decreases and he starts to feel a repulsive effect from the wall. In this case, the three behaviour rules will contribute to the total effect. As a result, Mr White feels the three effects separately and the outcome is the total effect. The first effect is generated by behaviour "seeking" which makes him move directly to the right end of the corridor. The second effect comes from behaviour "keeping a distance from a wall". It makes Mr White move towards north. The third effect is caused by behaviour "keeping a distance from an agent". It makes Mr White move towards southwest (approximately). When adding up these three effects, the total effect can have three possible directions which are shown in situation 2, 3 and 4 in Figure 82.

In situation 2, the distance between the two agents is close and the distance between Mr White and wall is far (but close enough to have a repulsive effect). In this situation, the effect that pushes Mr White to move south is larger than the effect which pushes him towards north. The total effect will cause Mr White to move to southeast (approximately).

In situation 3, the moving up effect and the moving down effect have reached a balanced state. As a result, Mr White will walk straight on (i.e. to the east).

In situation 4, Mr White's position is close to the wall and he feels that the repulsive effect from the wall which makes him move north is larger than the repulsive effect from Mr Grey which makes him move south. The outcome would be an effect that makes Mr White move to northeast (approximately).

Situation 5 represents Mr White feeling a repulsive effect from the wall but no effect from Mr Grey. The total effect will make Mr White move to the northeast (approximately).

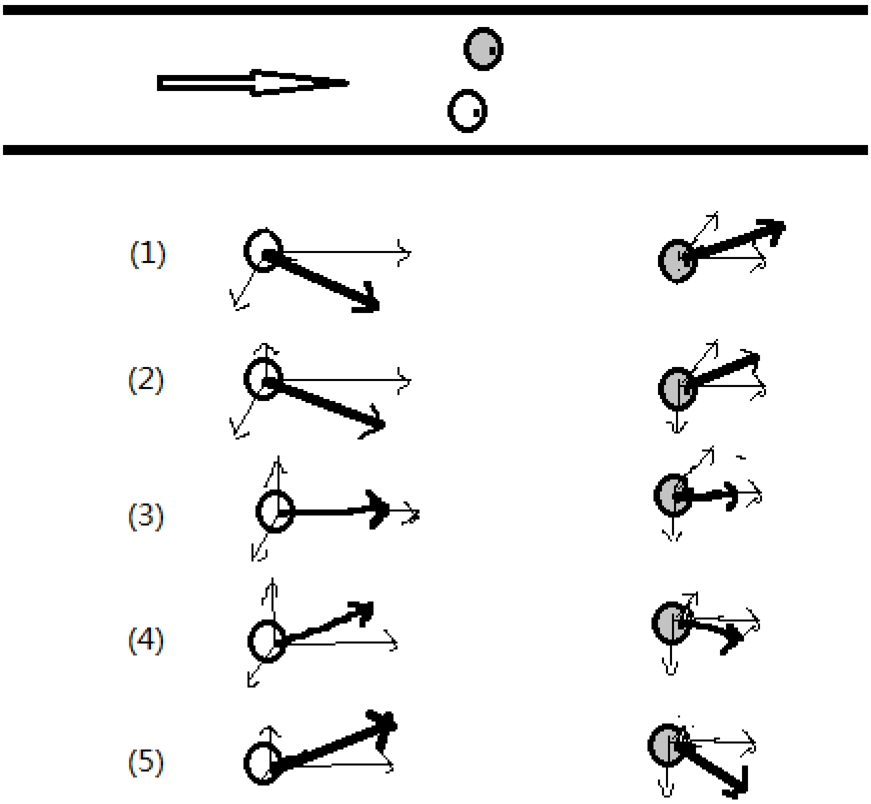


Figure 82 Illustrations on possible effects in phase 2 (thick arrows represent the overall effect)

It can been seen that during the process of getting close and trying to overtake Mr Grey, the total effect on Mr White keeps changing.

Phase 3: After Overtaken

After Mr White passes Mr Grey, the effects of the behaviour rules on him is similar to previous cases. According to the relative positions between the two and the distance between Mr White and the sidewalls, the possible total effects can be divided into five situations and are illustrated in Figure 83.

In situation 1, 2 and 3, all three behaviour rules have effects on Mr White. Situation 1 shows that Mr White moves to southeast (approximately) when he feels a larger repulsive effect from Mr Grey. In situation 2, when Mr White receives a larger repulsive effect from the wall, he moves to northeast (approximately). Situation 3 presents the case that Mr White moves straight forward when the effect from the wall and the effect from Mr Grey are equal.

Situation 4 shows Mr White is far enough from Mr Grey but still has not reached the comfortable position in the corridor. Thus at that point he only feels a repulsive effect from the wall plus the effect which makes him walk towards east end of the corridor. In contrast, situation 5 shows that Mr White has reached a comfortable position in corridor but still feels a repulsive effect from Mr Grey.

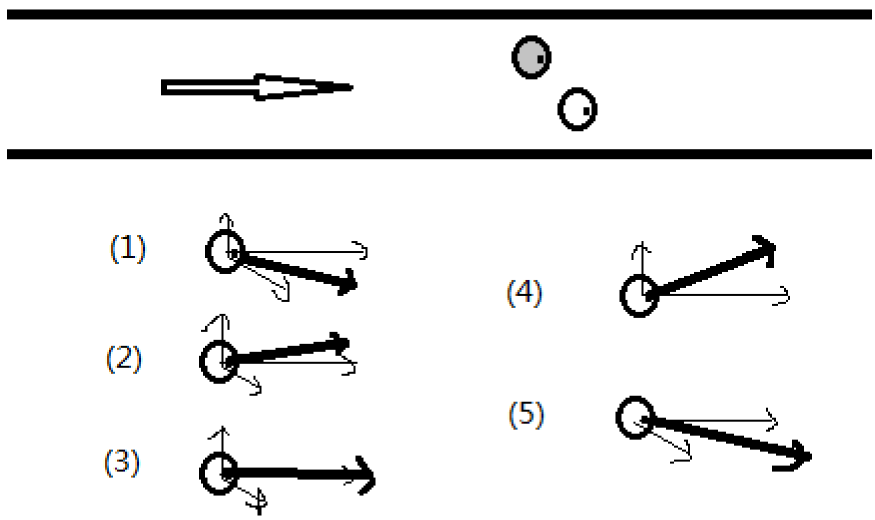


Figure 83 Illustrations on possible effects after the overtaking phase (thick arrows represent the overall effect)

Phase 4: Long after Overtaken -> Reached the Destination

Because Mr White is faster than Mr Grey, the distance between them will increase as time goes by and thus the repulsive effects between them will decrease and finally disappear. The repulsive effects from the wall will keep pushing Mr White away from the wall until he reaches the comfortable position. In the end, only the behaviour rule ‘walk to’ has an effect on Mr White. In this case, his distance from Mr Grey is far enough and he has reached the comfortable position in the corridor.

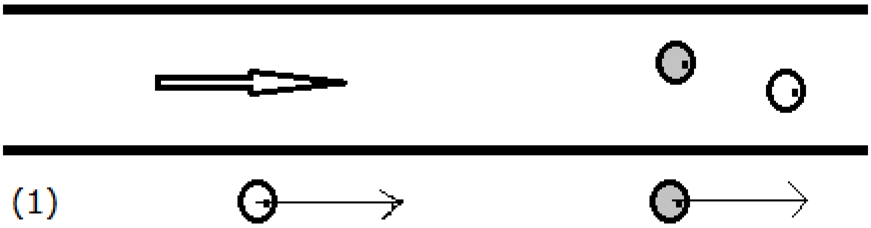


Figure 84 Illustrations on the behaviours long after the overtaken

5. 3. 2. 4 Post-simulation Stage

This example aims to demonstrate how to use proposed crowd model to represent a scenario. Further analysis will not be presented.

5. 4 Summary of the Chapter

This chapter presents the design of the crowd model used in this PhD study. First of all, it introduces the main features, the modelling approaches and the overall structure of the crowd model. Then it describes the model in detail which includes the behaviour representation, the behaviour effect calculation, the agent model design, and the Behaviour Library. Finally, it demonstrates the working process of the crowd model in an overtaking scenario.

Chapter 6 MODEL IMPLEMENTATION

This chapter presents the implementation of the crowd model introduced in the previous chapter. In the first section the simulation engine and the simulation environment is presented. In the second section, the design of the crowd simulation system and the detailed implementation of every aspect of the crowd model is introduced.

6. 1 Introduction to Implementation

6. 1. 1 The Simulation Engine - XNA Framework 4.0

Microsoft XNA Framework is a managed runtime environment for video game development. The XNA Framework consists of a set of managed libraries based on the Microsoft .NET Framework. The first version of the XNA toolset was announced on 24 March 2004 and version 4.0 (based on .NET Framework 4) was released on 16 September 2010. XNA has been released with an integrated development environment (IDE) - XNA Game Studio - which enables game development in Microsoft Visual Studio.

In this study, the implementation of the crowd model utilises the latest version of the XNA - XNA Framework 4.0 (for the justification on this selection, please refer to "*Discussions on Simulation Packages*" section in **Error! Reference source not found.**: **Error! Reference source not found.**). The development has been carried out in Microsoft Visual Studio 2010 with XNA Game Studio 4.0.

In this section, it generically introduces the execution process of a game in XNA, the Game class in the XNA Framework and the lifecycle of a game developed by XNA.

6. 1. 1. 1 Execution Process

In XNA, a running game is usually executed in the following three steps:

- **Initialization/Load** – Sets default and preliminary values to the game, queries and initializes user-based information, loads graphic and non-graphic contents, etc.

- **The Game Loop** – Performs in-game repeating logic and layout calculations and render.
- **Unload/Shutdown** – Saves current state, releases and unloads contents, etc.

This execution process describes a skeleton for any type of game. In XNA framework, it is implemented within the Game class (under the Microsoft.Xna.Framework namespace). An application (i.e. a game) needs to inherit the Game class and override the required methods to add specific game logic.

6. 1. 1. 2 The Methods in the Game Class

The Game class contains 11 public methods (4 of them are inherited from the Object class which can be viewed as the root class in the .NET Framework) and 14 protected methods. These methods cover the various aspects of the listed game process. They are listed below with brief descriptions:

Public Methods

Table 10 The public methods in the Game class in XNA (Microsoft Corporation 2010a)

Name	Description
Dispose	Overloaded. Immediately releases the unmanaged resources used by this object.
Equals	(Inherited from Object.)
Exit	Exits the game.
GetHashCode	(Inherited from Object.)
GetType	(Inherited from Object.)
ResetElapsedTime	Resets the elapsed time counter.
Run	Call this method to initialize the game, begin running the game loop, and start processing events for the game.
RunOneFrame	Run the game through what would happen in a single tick of the

	game clock; this method is designed for debugging only.
SuppressDraw	Prevents calls to Draw until the next Update.
Tick	Updates the game's clock and calls Update and Draw.
ToString	(Inherited from Object.)

Protected Methods

Table 11 The protected methods in the Game class in XNA (Microsoft Corporation 2010a)

Name	Description
BeginDraw	Starts the drawing of a frame. This method is followed by calls to Draw and EndDraw.
BeginRun	Called after all components are initialized but before the first update in the game loop.
Draw	Called when the game determines it is time to draw a frame.
EndDraw	Ends the drawing of a frame. This method is preceeded by calls to Draw and BeginDraw.
EndRun	Called after the game loop has stopped running before exiting.
Finalize	Allows a Game to attempt to free resources and perform other cleanup operations before garbage collection reclaims the Game.
Initialize	Called after the Game and GraphicsDevice are created, but before LoadContent.
LoadContent	Called when graphics resources need to be loaded.
MemberwiseClone	(Inherited from Object.)
OnActivated	Raises the Activated event. Override this method to add code to handle when the game gains focus.
OnDeactivated	Raises the Deactivated event. Override this method to add code to handle when the game loses focus.
OnExiting	Raises an Exiting event. Override this method to add code to handle

	when the game is exiting.
ShowMissingRequirementMessage	This is used to display an error message if there is no suitable graphics device or sound card.
UnloadContent	Called when graphics resources need to be unloaded. Override this method to unload any game-specific graphics resources.
Update	Called when the game has determined that game logic needs to be processed.

Key Methods

In order to implement a game from the Game class, it is not required to override all the methods that have been provided. This section discusses the six key methods that are required to override in order to create a successful running game.

- **Class Constructor**: The Constructor must be implemented when inheriting the Game class. It is used to instantiate and set default values to the required elements. For example, to instantiate the graphics device manager, to define the game frame rate, etc.
- **Initialize**: The Initialize method sets default and preliminary values to the game shell, queries and initializes user-based information, etc.
- **LoadContent**: The LoadContent method loads all graphics and other content required to run the game. For example, LoadContent loads and instantiates graphic sprite batches, background images, sounds, etc.
- **Update**: The Update method is the place where the specific on-going game logic is defined. For example, to calculate current positions, physics, collisions and states; to collect input information from the various input devices; to play audio, etc.
- **Draw**: The Draw method displays the current view of the game. It defines what sprites should be shown and how they are shown.
- **UnloadContent**: The UnloadContent method is used to unload all game content and content managers before the programme closes.

6. 1. 1. 3 Game Lifecycle

The lifecycle of an XNA game is shown in Figure 85. When a game starts the Initialize method is called to allow the game to do any initialization required by the game shell itself. The LoadContent method is called afterwards which allows the game to load all the required content resources as described above.

The Update and Draw methods are called repeatedly by the XNA Framework, not necessarily in sequence, at 60 FPS on a Windows operating system *(Note. the XNA Framework is designed to drop frames automatically in order to keep up with the desired frame rate. There may be cases where Draw is not called even though Update changes the elements to be rendered)*. Typically, all input, game logic, physics, AI, and any other non-graphical processing should be handled in the Update method. All graphical processing and actual drawing of the game should be done in the Draw method.

The UnloadContent method is called when the game closes to allow the game to release loaded resources.

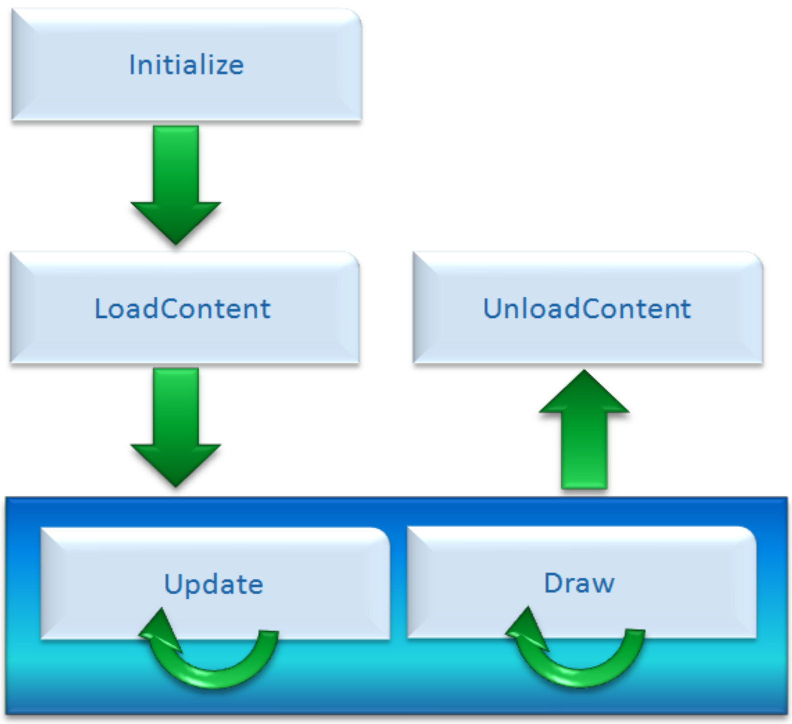


Figure 85 The XNA game lifecycle (Microsoft Corporation 2010b)

6. 1. 2 Graphical Representation

Because this PhD study focuses on crowd modelling and the development of a simulation system primarily serves the purpose of evaluation, the graphical representation of the developed system will be kept simple but needs be able to provide an accurate implementation of the crowd model.

The graphics are represented in 2D. The environment (e.g. rooms, walls, etc.) is represented by simple lines and the agents are represented by small cycles with a dot indication to represent their orientations. Additionally, some information about the simulation is displayed on top of the simulation window.

6. 1. 3 Navigation

Navigation, also called way finding, is the implementation as to how individuals find their way to the decided destinations in the simulation environment.

In this simulation system, the Cell and Portal Graph (CPG) method (Nuria Pelechano et al. 2008) has been adopted. An example of a building layout and its correspondence CPG map is shown in Figure 86 and Figure 87:

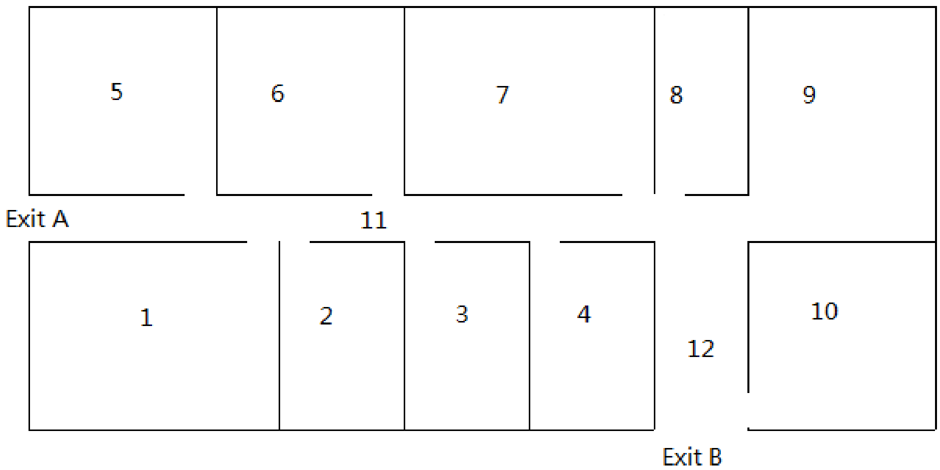


Figure 86 layout of one building

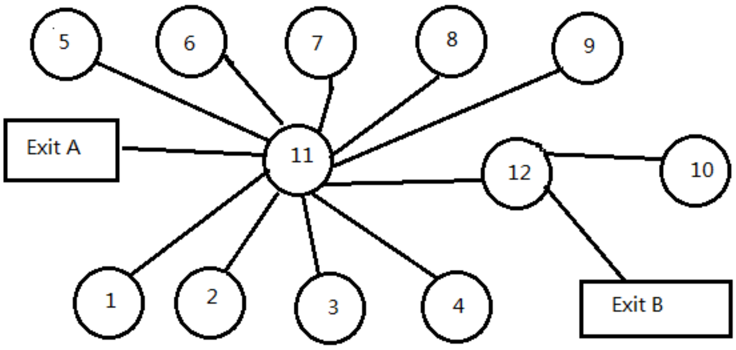


Figure 87 Correspondent CPG for the building in Figure 86

Rooms, corridors and exits are converted into cells. Doors are translated into the links between cells. Agents move from one cell to another until they reach their destination. Movement between any linked cells is straightforward and does not require any further navigation.

6. 2 Detailed Implementation of the Simulation System

6. 2. 1 Structure of the Simulation System

The simulation system is implemented through the XNA framework. The main classes in the simulation system are listed as follows:

- **MyGame**: This is the class inherited from the Game class as it is a requirement of XNA game implementation. It provides a game running process that follows the XNA Game Lifecycle (Figure 85) defined by the XNA framework.
- **Agent**: This is the class which implements the agent model (presented in section 5. 2. 3) and the Behaviour Library (presented in 5. 2. 4).
- **Environment**: This class and its inherited classes present the environment layout and the navigation information.
- **Log**: This class implements the functions that are used to record the simulation information/results.
- **Simulation Configuration**: This class defines the simulation scenario related information.

Figure 88 provides a graphical view of these classes are linked in the system.

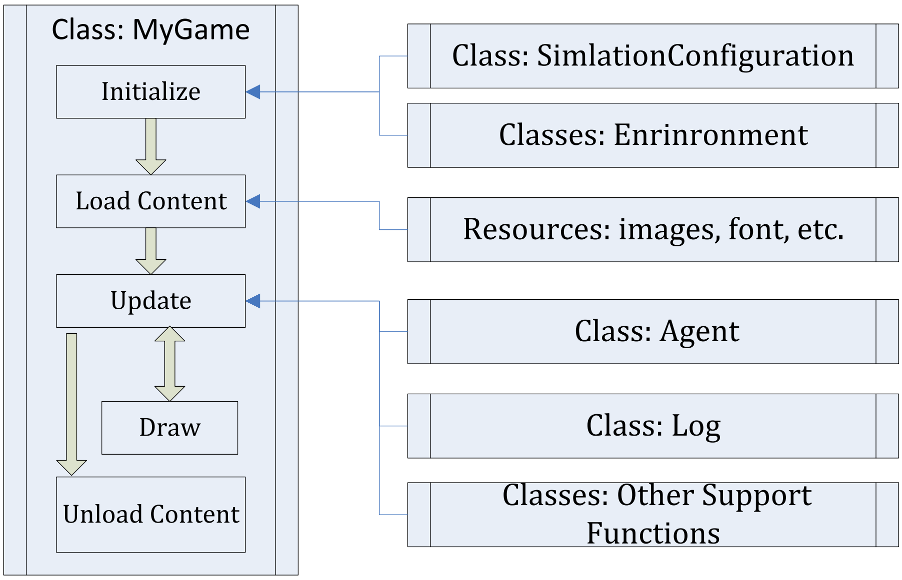


Figure 88 Classes in the simulation system

6. 2. 2 Implementation of the XNA Game Lifecycle

The MyGame class is inherited from "Microsoft.xna.Framework.Game" (a base class for an XNA game defined in the XNA framework). It runs by following the XNA game lifecycle and defines the working process of the simulation system.

The class Constructor, the Initialize() method, the LoadContent() method run in sequence when a simulation is about to start. After the simulation has started, the Update() method and the Draw() method will keep looping until the simulation has been terminated. The UnLoadContent() method is called automatically before the programme exits. The skeleton of a MyGame class is showed below:

```
{
    // Class Constructor
    MyGame() {…}
    // Load per simulation settings
    Initialize(){…}
    // Load the content for drawing
    LoadContent() {…}
    // Release the resources before exit
    UnloadContent(){…}
    // Main crowd simulation logic
```

```
    Update(){…}
    // Display the simulation on the screen
    Draw(){…}
}
```

The detailed function of each method in the MyGame class is presented in the following sections.

✧ **Constructor: MyGame()**

Currently, the MyGame() method is used to initialize the graphics engine of the simulation system.

✧ **Initialize()**

The Initialize() method serves three functions in the simulation system:

- Initialising the simulation settings by using the configuration defined in the SimulationConfiguration class.
- Logging the configuration information into the log file.
- Carrying out the default initialisations required by the XNA framework.

✧ **LoadContent()**

The LoadContent() method has two purposes:

- Creating a SpriteBatch which is used to draw the content on the screen.
- Loading all the graphical resources (e.g. font, images for the environment, agent, etc.) into the simulation system.

✧ **UnloadContent()**

In the implementation of this simulation system, all the contents are managed by the XNA ContentManager. This method does not require further implementation in this research study.

✧ **Update()**

The Update() method has five functions:

- If the simulation has not been started yet, creating the agents as they are defined by the simulation scenario and starting the simulation.

- Updating the relevant simulation scenario information and checking whether the terminating condition of the simulation has been met.
- If the terminating condition has been met, logging the simulation result into the relevant format of files and shutting down the simulation system.
- If the terminating condition has NOT been met, updating the agent status. This update process is defined in the Person class which is the implementation of the Agent Action Engine (details are presented in section 6. 2. 3. 2 below).
- Carrying out the default update actions as required by the XNA framework.

✧ **Draw()**

The Draw() method is used to show the simulation on the screen. Three types of objects are displayed through this function:

- The environment and the background of the simulation venue are drawn in this method.
- The relevant information on the simulation scenario is displayed on the screen, such as elapsed time, total agent amount, etc.)
- All the agents in the simulation system are shown on the screen by this method.

According to the XNA framework document (Microsoft Corporation 2010b), this method may be skipped (known as skip frame) in cases where not enough computer resources can be allocated. Such skips will not affect the agents' actual decision making and movement calculation as they are handled in the Update() method. In other words, the simulation system is always working as the crowd model defines but the graphics representation may be non-contiguous due to the limitation of computer resources.

6. 2. 3 Implementation of the Agent Class

The Person class is the implementation of the agent model defined in the crowd model. It can be divided into two parts:

- The first part is the properties declared in the class. They are the implementation of the agent information (including the agent's attributes, the agent's knowledge and the agent's status) defined in the crowd model.
- The second part is the implementation of the Agent Action Engine which is represented in the ActionEngine() method.

6. 2. 3. 1 Implementation of the Agent's Information

The Agent's Information is implemented as the properties in the Agent class. Each attribute in the Agent's Information is transformed into one or more properties. The following three tables (Table 12, Table 13, and Table 14) list the implementation in detail.

Implementation of the agent's attributes

Table 12 Implementation of the agent's attributes

Attribute in the Agent Model	Property Name in Class	Type (C#)
Position	Position	Vector2
Shape	EntityTexture	Texture2D
Size	BodySize	int
Orientation	Orientation	Vector2
Movement Mode	IsMoving	bool
Default Walking Speed	DefaultSpeed	float
Behaviour Effect Limit (Walking)	WalkingLimit	float
Default Running Speed	DefaultSpeedRun	float
Behaviour Effect Limit (Running)	RunningLimit	float
Movement Speed Adjuster	SpeedAdjuster	float
Sight Range	SightRange	int

Sense Range for Group Behaviour	GroupBehaviourRange	int
Desired Distance from Others	DesiredDistanceFromOthers	int
Minimum Distance from Others	MinimumDistanceFromOthers	int
Desired Distance from Wall	DesiredDistanceFromWall	int
Minimum Distance from Wall	MinimumDistanceFromWall	int
Desired Distance from Obstacles	DesiredDistanceFromObstacles	int
Minimum Distance from Obstacles	MinimumDistanceFromObstacles	int
Leadership	Leadership	int
Willingness to Follow	WillingnessFollow	int
Group Behaviour Modifier	GroupBehaviourModifier	int
Probability of being Affected by POIs	ProbabilityPOI	int
Repulsion Modifier (to Self)	RepulsionModifierself	int
Repulsion Modifier (to Others)	RepulsionModifier	int

Implementation of the agent's knowledge

Table 13 Implementation of the agent's knowledge

Attribute in the Agent Model	Property Name in Class	Type in Class (C#)
Path	Path	List<MapPosition>
Wall	Defined in the Environment Class	
Visual effect area		

Implementation of the agent's status

Table 14 Implementation of the agent's status

Attribute in the Agent Model	Property Name in Class	Type in Class (C#)

Current Position	Position	Vector2
Current Speed	Speed	float
Current Orientation	Orientation	Vector2
Behaviour Mode	CurrentState	State
	previousState	State
	CanMove	bool
Goal	Target	Entity
	NextMovingPosition	MapObject
	PathTargetPoint	Vector2

6. 2. 3. 2 Implementation of the Agent Action Engine

The Agent Action Engine is represented by the ActionEngine() method in the Agent class. During the simulation, the agent calls this method in order to make decisions and take actions in each update loop. The algorithm of the ActionEngine() method is presented below (the C# codes are enclosed in 0):

```
Initialise behaviour and movement related variables for calculation
Retrieve the agent's status in the previous loop
Retrieve the environmental information
Retrieve the crowd information
Decide the active behaviour
Identify the possible passive behaviours
Calculate the behaviour effects of all applicable behaviours
Combine the behaviour effects into one final effect
If the final effect exceeds the agent's maximum movement ability
```

```
   Truncate the final effect to fit the agent's ability
Check whether this effect can result in collisions
If it has potential collisions
   Cancel all the behaviours decided above
   Select the collision avoid behaviour
   Calculate the behaviour effect of collision avoid
   Use it as the final behaviour effect
Update the agent position with the final behaviour effect
Update relevant agent information
If behaviours interact with the surroundings
   Update the state of the surrounding environment
   Update the status of the surrounding crowd
End the update loop
```

6. 2. 4 Implementation of the Behaviour Library

This section presents the implementations of the ten behaviour rules that were introduced in the Behaviour Library. In order to maintain readability for non-programmers, the implementation of each behaviour rule is described in pseudo-code. The actual codes were implemented with C# and are enclosed in 0.

The implementation of the behaviour rules represent the relevant behaviour effect in one update frame. Because all the behaviour rules are used by the agent during its decision making process, the pseudo-codes are written from the agent's point of view.

✧ **Seeking**

```
If target exists
   If has reached the target
      Stop moving
   Else
      Change the orientation to face the target
```

```
      Calculate the base movement distance from the default
walking speed
      Determine the coefficient of the Self Factor
      Determine the coefficient of the Target Factor
      Update the movement distance by applying the above
two coefficients
      Move forward with the updated distance
Else
   Do nothing
```

- **Wandering**

```
Generate a random number in the range of [0, 100]
If the generated number < 5
   Generate a random angle in the range of [-18°, +18°]
   Turn the current orientation at the randomly generated
angle
Else
   Calculate the base movement distance from the default
walking speed
   Determine the coefficient of the Self Factor
   Update the movement distance by applying the above two
coefficients
   Move forward with the updated distance
```

- **Following**

```
If can identify the following target
   If has reached desired position in terms of following
      Stop moving
   Else
      Generate a virtual position which is behind the
following target
      Change the orientation to face the virtual position
```

```
      Calculate the base movement distance from the default walking speed
      Determine the coefficient of the Self Factor
      Determine the coefficient of the Target Factor
      Update the movement distance by applying the above two coefficients
      Move forward with the updated distance
Else
   Do nothing
```

✧ **Keeping a distance from an agent (repulsive effect)**

```
Calculate the distance to the agent
If distance >= the desired distance
   Do nothing
Else
   If distance <= the minimum distance
      Set the coefficient of Distance Factor to 1
   Else
      Set the coefficient of Distance Factor to 1/distance
   Change the orientation to turn the back to the agent
   Calculate the base movement distance from the default walking speed
   Determine the coefficient of the Self Factor
   Determine the coefficient of the Target Factor
   Update the movement distance by applying the above three coefficients
   Move forward with the updated distance
```

✧ **Keeping a distance from a wall (repulsive effect)**

```
Calculate the shortest distance to wall
If distance >= the desired distance
```

```
   Do nothing
Else
   If distance <= the minimum distance
      Set the coefficient of Distance Factor to 1
   Else
      Set the coefficient of Distance Factor to 1/distance
   Change the orientation to turn the back to the wall
   Calculate the base movement distance from the default
walking speed
   Determine the coefficient of the Self Factor
   Determine the coefficient of the Target Factor
   Update the movement distance by applying the above three
coefficients
   Move forward with the updated distance
```

- **Avoiding collision**

```
Get the radius of the obstacle
Calculate the sum of the obstacle radius and the agent's
desired distance to obstacle
Draw a virtual circle that centres at the obstacle
Set the radius of the circle to the sum calculated above
Draw a tangent line from the agent's position to the virtual
circle
Draw a line from the agent's position to the obstacle's
position
Calculate the angle formed by the above two lines
Randomly select turn left or right
Change the orientation to face the obstacle
Further turn the orientation at the angle calculated to left
or right as selected
```

```
Calculate the base movement distance from the default walking speed
Determine the coefficient of the Self Factor
Determine the coefficient of the Target Factor
Update the movement distance by applying the above two coefficients
Move forward with the updated distance
```

- **Walking towards the group**

```
Set the range of the group
Identify all the agents inside the group
Calculate the average position of all the agents (including self) in the group
Change the orientation to face the average position
Calculate the base movement distance from the default walking speed
Determine the coefficient of the Self Factor
Update the movement distance by applying the above coefficient
Move forward with the updated distance
```

- **Aligning direction with the group**

```
Set the range of the group
Identify all the agents inside the group
Calculate the average orientation of all the agents (including self) in the group
Change the orientation to that average orientation
Calculate the base movement distance from the default walking speed
Determine the coefficient of the Self Factor
Update the movement distance by applying the above coefficient
```

```
Move forward with the updated distance
```

✧ **Handling repulsive effect from the crowd**

```
Set the range of the group
Identify all the agents inside the group
For each agent (exclude self) in the group
   Calculate the distance to that agent
      If distance <= the minimum distance
      Set the coefficient of Distance Factor to 1
   Else
      Set the coefficient of Distance Factor to 1/distance
   Change the orientation to turn the back to the agent
   Calculate the base movement distance from the default
walking speed
   Determine the coefficient of the Self Factor
   Determine the coefficient of the Target Factor
   Update the movement distance by applying the above three
coefficients
   Move forward with the updated distance
End Loop
```

✧ **Keeping in a group**

```
Set the range of the group
Identify all the agents inside the group
Apply behaviour rule "Walk toward the Group"
Apply behaviour rule "Align Direction with Group"
```

6. 2. 5 Implementation of the Environment Map

The implementation of the environment map falls within three aspects of the simulation system:

- The environment information

- The graphical resources of the environment
- The navigation of the environment

The Environment Information

The environment information describes what kinds of objects are in the environment and their positional information. It contains three types of information:

- The layout of the environment: such as positional information on the walls, the corridors and the gates.
- Environmental objects: the objects in the simulation environment that can interact with the agents or can affect the agents' behaviours. For example, obstacles, signs, etc.
- Simulation related information: this refers to the descriptive information on the environment. It does not affect the agents' behaviours during the simulation. For example, the venue names, room numbers and other indicators for a better understanding of the specific simulation scenario.

The detailed implementations are explained in the following sections.

➢ The Layout of the Environment

The Environment class contains the layout information of the environment in the simulation system. The class structure is outlined as follows:

```
Class Environment {
   List<Wall> Walls;
   List<Gate> Gates;
   List<Corridor> Corridors;
   List<Room> Rooms;
}
```

Only a wall has a physical existence in the simulation system. The others (i.e. rooms, gates, and corridors) are the spaces that are formed by the presence of walls. Their definitions are listed below.

✧ Wall

A wall is defined by a rectangular shape in the simulation system and is implemented in the Wall class. The Wall class contains the following properties:

- **Size**: This refers to the thickness of the wall.
- **Length**: This refers to the length of the wall.
- **TopLeftPosition**: Represents the top left vertex of the wall (refers to on screen positions).
- **TopRightPosition**: Represents the top right vertex of the wall (refers to on screen positions).
- **BottomLeftPosition**: Represents the bottom left vertex of the wall (refers to on screen positions).
- **BottomRightPostion**: Represents the bottom left vertex of the wall (refers to on screen positions).

✧ **Gate**

A gate is a virtual layout in the simulation system and is not shown as a visible object during the simulation. It is implemented in the Gate class. The Gate class contains the following properties:

- **StartPoint**: Represents one end of the gate.
- **EndPoint**: Represents the other end of the gate.
- **Layout**: Can only be horizontal or vertical.
- **Position**: Represents the centre point of the gate.
- **Width**: the width of the gate.

✧ **Corridor**

A corridor is defined by a rectangular shape in the simulation system and is implemented in the Corridor class. The Corridor class contains the following properties:

- **TopLeftPosition**: Represents the top left vertex of the corridor (refers to on screen positions).
- **TopRightPosition**: Represents the top right vertex of the corridor (refers to on screen positions).

- **BottomLeftPosition**: Represents the bottom left vertex of the corridor (refers to on screen positions).
- **BottomRightPostion**: Represents the bottom left vertex of the corridor (refers to on screen positions).

✧ **Room**

A room is defined by a rectangular shape in the simulation system and is implemented in the Room class. The Room class contains the following properties:

- **TopLeftPosition**: Represents the top left vertex of the room (refers to on screen positions).
- **TopRightPosition**: Represents the top right vertex of the room (refers to on screen positions).
- **BottomLeftPosition**: Represents the bottom left vertex of the room (refers to on screen positions).
- **BottomRightPostion**: Represents the bottom left vertex of the room (refers to on screen positions).
- **AssociatedGates**: represents the gates that are linked with the room.

➢ Environment Objects

Environment objects are represented by the EnvironmenObject class. It contains the following properties:

- **Position**: Represents the position in the simulation environment.
- **Size**: The object is modelled as a circle and the size represents the radius of the circle.
- **Type**: Describes the type of the object.
- **IsVirtual**: Indicates whether the object is virtual or not.

➢ Simulation Related Information

The simulation related information is displayed on the screen to provide a better understanding of the simulation environment and the simulation status. Currently, three types of information can be displayed:

- **The status of the running simulation**: This information is shown on the top position of the simulation window. The status includes the map scale legend, the elapse time, the numbers of the agents in the simulation environment, the average speed of the crowd, and specific information on the scenario.
- **Detailed information on the selected agent**: The simulation system can display detailed information on one selected agent. The information includes the agent's ID, the current behaviour effects, information on the agent's target, current movement speed, and current behaviour state.
- **Environment map related information**: The content of the environment map related information depends on the simulation scenario; for example, in the simulation of a building evacuation it can show information such as room names and numbers, corridors, gates, exits, etc.

The Graphical Resources of the Environment

In the implementation of this simulation system, the graphical resources refer to the pictures that represent the agents, buildings, etc. They are loaded into the simulation system in the LoadContent() method and are used as Textures to draw relevant objects in the Draw() method.

The Navigation of the Environment

As it mentioned in the first section of this chapter, Cell and Portal Graph (CPG) is adopted to represent the map of the simulation and the navigation is based on that. The implementation of the navigation in the simulation environment has the following three steps:

- Prior to the simulation, the CPG will be created for the simulation environment. To simplify the implementation of the route calculation, long corridors in the building will be divided into several sections to form cells in CPG (demonstrated in Figure 89). Each cell in CPG is modelled as a rectangle and the boundary information will be attached to it.

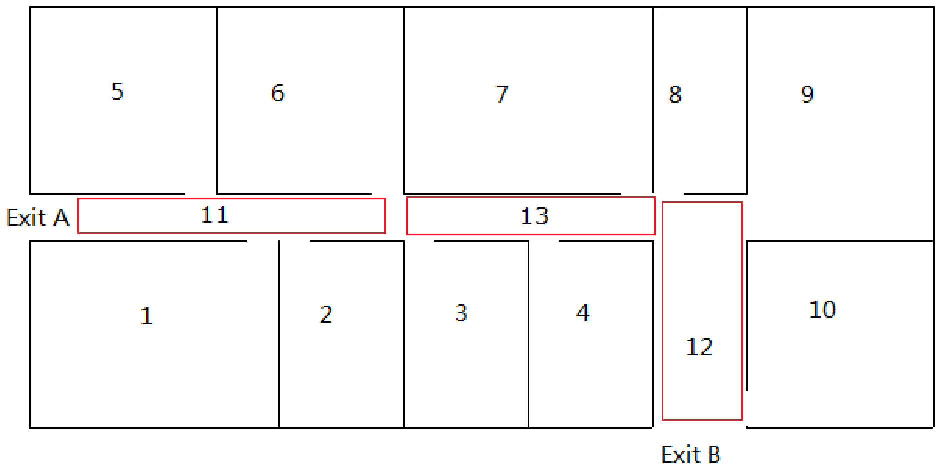


Figure 89 Segmentations of a long corridor

- During the simulation, the agent enquires concerning the route to CPG and suitable route information will be returned. The procedure is shown as follows:
    1. The agent sends its current location and required destination to CPG.
    2. CPG firstly identifies the cell numbers of the agent's location and destination.
    3. CPG calculates the route to the two identified cells in step 3.
    4. CPG returns a route which includes the cells that the agent needs to travel through in sequence.
- The agent creates a path which consists of a list of waypoints (representing the locations of the cells in CPG) and will follow this path for navigation. For example, assuming an evacuation scenario in the building of Figure 89, one agent at cell 5 wants to exit the building via Exit B. CPG will returns a path of cell 5 -> cell 11 -> cell 13 -> cell 12 -> Cell Exit B. The agent will then use this path to perform the evacuation.

6. 2. 6 Implementations of Supporting Functions in the Simulation System

The previous sections introduced the implementations of the simulation system engine and the crowd model. However, the implementations have involvement with many basic and frequently used methods which provide the supporting functions of the system. To have a clear hierarchy for the system and for reuse and further expansion purposes these functions have been implemented in separate classes and are presented in this section (the C# codes are attached in 0).

6. 2. 6. 1 Implementation of Log Functions

The log functions in this simulation system can record the simulation information and results in two types of file format: .txt and .xls.

- The implementation of the .txt format log function uses the namespaces of System.IO.StreamWriter and System.IO.FileStream in the Microsoft .NET framework 4.0.
- The implementation of the .xls format log function utilises one assembly to provide Excel file operation functions. The assembly is called LibXL (http://www.libxl.com/).

6. 2. 6. 2 Implementation of other Supporting Functions

In addition to the log function, there are many other supporting functions that have been implemented in the simulation system. Their method names and functions are listed in the table below.

Table 15 Support functions (excluding log function) in the simulation system

Method Name	Function
DistanceToLine()	Calculates the short distance from a point to a given line
ToRadian()	Converts a measurement unit of the direction from vector format into radian format
ToDirection()	Converts a measurement unit of the direction from radian format into vector format
NormalizeRadian()	Converts a radian value into its equivalent in the range of [0, 2π]
TurnAnAngle()	Turns the direction of a vector anticlockwise at a given angle
GetRadian()	Generates an angle in radian format within of [0, 2π]

GetChance()	Generates a integer from [0,100] randomly
GetRandomInRange()	Generates a integer from a given range randomly
GetRandomDouble()	Generates a number (with fraction) from a given range randomly

6. 3 Summary of the Chapter

This chapter presents the implementation of a simulation system. In the first section, it briefly introduces the Microsoft XNA framework which is used as the game engine in the development of the simulation system as well as the graphical representation and the navigation method in the simulation system. In the second section an overview of the simulation structure is presented first. Then a description of the detailed implementation of the system is presented in five sub-sections: 1) XNA game lifecycle; 2) Agent class; 3) Behaviour Library; 4) Environment map; 5) Supporting Functions.

Chapter 7 MODEL EVALUATIONS: MODEL APPLICATIONS

7. 1 Introduction to Model Evaluations

This study aims to design a configurable crowd model to present crowd heterogeneity and the individuals' interactions. The evaluations of the crowd model (i.e. whether the aim has been achieved) are carried out through a series of simulations. They evaluate the study's crowd model from two aspects: the model applications and the model validation. Descriptions and analysis of these evaluations are presented in this chapter and in the next chapter respectively.

7. 1. 1 Purpose of the Demonstrations of Model Applications

The evaluation of model applications aims to demonstrate the features that the crowd model is designed to present. The features are listed as follows:

- Crowd heterogeneity, i.e. individuals are independent and can be different from each other.
- Crowd behaviours are formed through the combinations of individuals' behaviours.
- Individuals' differences can affect their behaviours thus to influence the crowd behaviours.
- The influence of the environment on crowd behaviours can be represented through simulation.
- Individuals can be configured to have corresponding behaviours to fit in with different scenarios.

7. 2 The Basics of the Simulations

Although the configuration of the simulation environment and the agents are subject to individual scenarios, there are some settings that remain the same or can be used as guidelines in the various simulations in this research study. Before presenting the detailed simulations for evaluation, they are introduced below to keep the thesis concise.

7. 2. 1 Simulation Environment Settings

7. 2. 1. 1 Scale

Simulation setting - 1 : 0.05 (pixel : metre)

In crowd simulation there is a need to scale down the objects of the real world in order to display them on the screen of a monitor. A pixel is the unit of graphical measurement (in resolution) on the screen. In this research study, one pixel on the screen represents five centimetres (0.05 metres) in the real world by default.

With this default scale setting, a screen with 1600 X 1050 resolution can represent a virtual environment up to 80 X 50.25 metres.

7. 2. 1. 2 Time Representation

Simulation setting: real-time simulation

The simulations in this research study are running in real-time which means the time spent in the virtual environment equals the time in the real world, i.e. if a simulation of an emergency evacuation took five minutes on the computer, it means that in the real world the evacuation is considered to take five minutes as well on the premise that crowd simulation can represent reality.

7. 2. 1. 3 Programme Update Interval

Simulation setting: 60 FPS (frames per second)

The human brain and its visual system is considered to handle 10 to 12 separate images per second (Read & Meyer 2000) which means if the images refresh faster than that rate (10-12 FPS) they will be perceived as continuous images to human eyes. In TV and the digital cinema industry the three main frame rates used are 24p, 25p and 30p ("p" refers to frames per second). A typical LCD monitor nowadays usually has a 60 Hz refresh rate.

In this study, the simulation programme updates at 60 FPS which can produce a continuous crowd motion and 60 FPS also is the perfect match for a standard LCD monitor, at which it can provide its best display.

7. 2. 2 Agent's Attributes

7. 2. 2. 1 Body size

Simulation setting: 10 pixels (0.5 metre)

In this crowd model the agents are represented by circles. Because variation in body size is not discussed in this PhD research, the agents in all the simulations have the same body size which is 10 pixels (0.5 metre) in diameter.

Walking speed (a guideline only)

In this research the default walking speed of an agent is 1.20 m/s and the walking speed should not exceed 1.5m/s under normal circumstances. This default setting for the walking speed is based on the following two references:

- Thompson and Marchant (Thompson & Marchant 1995) summarised the data on walking speeds from various studies (listed in Table 16)
- According to Fruin Level of Service (Fruin 1987), the walking speeds of a crowd should take the environment into account, especially the density of the crowd (Table 17).

However, in most of the simulations in this research study, the walking speeds of the agents have usually been tested with different values in order to reveal the potential influence of the walking speeds on crowd behaviours. The default walking speed presented here is mainly used as a guideline.

Table 16 Summary of walking speed from various studies (Thompson & Marchant 1995)

STUDY		WALKING SPEED (m/s)					
		OLD PEOPLE			WALKING SPEED		
		Slow	Normal	Fast	Slow	Normal	Fast
Men	Blanke and Hageman		1.38			1.32	
	Himann et al.		1.21	1.47			
Women	Finley et al.		0.7			0.84	
	Blanke and Hageman		0.32			1.59	
	Ferrandez et al.		0.82	1.08			
	Himann et al.		0.89	1.14			
	Leiper and Craik		0.96	1.15			
	Obrien et al.		0.74	0.97			
Both	Cunningham et al.	1.05	1.33	1.6	1.08	1.39	1.72
	Elble et al.		0.94	1.39		1.18	1.67
	Waters et al.	0.81	1.22	1.5	0.71	1.32	1.76
	Judge et al.		1.06	1.43			

Table 17 Fruin Level of Service (Fruin 1987)

FRUIN LEVEL OF SERVICE	DENSITY (pers/m²)	SPACE (m²/pers)	FLOW RATE (pers/m/s)	AVERAGE SPEED (m/s)
A	<0.31	>3.22	<0.38	1.3
B	0.43–0.31	2.32–3.24	0.38–0.55	1.25
C	0.72–0.43	1.39–2.32	0.55–0.82	1.15
D	1.08–0.72	0.93–1.39	0.82–1.10	1.00
E	2.17–1.08	0.46–0.93	1.10–1.37	0.7
F	>2.17	<0.46	>1.37	

7. 2. 3 Computer Hardware and Software Environment

In this research study, all the simulations were running in a Windows OS environment with a daily working laptop. The environment specifications are introduced below:

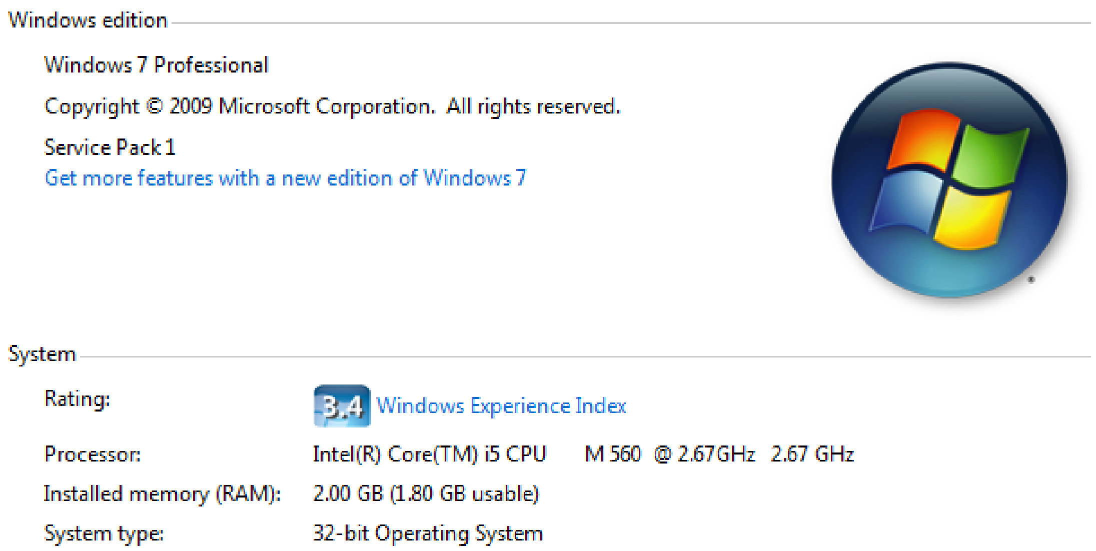


Figure 90 Specification of the computer

7. 3 The Applications of the Crowd Model

In the following sections, three series of simulations are presented:

- A group walking through a corridor
- A crowd exiting a small building (museum)
- Evacuation from a shopping mall

Their presentations are organised with the following structure:

- The introduction to the scenario of that simulation.
- The configurations of simulation settings, environment representation, and agents configurations
- The simulations' results and analysis (which consist of simulation sets with variances).

7. 3. 1 Simulation 1: A Group Walking through a Corridor

7. 3. 1. 1 Scenario

Scenario Description

This scenario describes a group of 24 people walking through a corridor (positioned east to west). The following instructions have been given to the crowd:

- Enter the corridor from the west end and exit from the east end.
- Walk at a normal speed.
- Try to maintain a comfortable distance from others.
- Movement is restrained by the side walls of the corridor.
- Do not collide with others.
- Signals may be given to the crowd to adjust their speeds during the movement.

Environment

The environment is simple and only contains one corridor. The length of corridor is 30 metres and the width is 5 metres. There are no other obstacles inside the corridor.

The Crowd

In this scenario, all the individuals in the crowd are considered to have the same attributes and behavioural preference. In other words, the crowd is homogenous.

Purpose

By altering the personal attributes of the individuals, this series of simulations tries to demonstrate:

- What is the effect on group behaviour if individuals change their comfort distances from others?
- How does the personal desire of maintaining these distances (mentioned in the above question) affect crowd behaviour?
- What is the reaction of the crowd to the environmental effects (speeding up and slowing down)?

7. 3. 1. 2 Simulation Configuration

Simulation Process and Data Capture

➢ **The Start and End of One Simulation Round**

The simulation starts when the agents begin to move from their starting positions. When an agent reaches the east end of the corridor, it exits the corridor. The simulation ends when all agents have exited.

➢ **Data Capture**

- The total time (in seconds) that the whole group uses to walk through the corridor will be captured.
- The crowd formations will be captured in some simulations to demonstrate the differences in group behaviour.

Environment Representation

In the simulation, the corridor is represented by two horizontal walls. Both have a length of 30 metres. The vertical distance between these two walls is 5 metres.

Agent Configuration

➢ **Behaviours**

✧ ***Summary***

The instructions to the crowd can be represented by the corresponding behaviours in the Behaviour Library. They are summarised below:

Table 18 Behaviours' interpretations for walking through a corridor

	Instructions	Simulation Configurations
1	Enter the corridor from west end and exit from east end.	Behaviour Library: Seeking
2	Try to maintain a comfortable distance from others.	Behaviour Library: Handling repulsive effect from nearby crowd

3	Movement is restrained by the side walls of the corridor.	Behaviour Library: Keeping a distance from a wall (repulsive effect)
4	Do not collide with others.	Behaviour Library: Avoiding collision
5	Signals may be given to the crowd to adjust their speeds during the movement.	No specific behaviour rule. Adjusting the base movement speed accordingly.

✧ ***Detailed Analysis***

This section explains the descriptions of the behaviour in the simulation scenario individually and analyse its correspondent behaviour rule in the Behaviour Library.

✦ ***Instruction 1: Seeking***

This describes the goal of the agents in the crowd: to move to the east end of the corridor. The "seeking" behaviour in the Behaviour Library can be used to represent this behaviour. The location of the goal should be a horizontal offset to the agents' current position at the east end of the corridor (exit) which means, without any other affection/interruption, each agent should walk horizontally toward the east end (i.e. via the shortest route).

It is predictable that during the movement, each agent may adjust its vertical position to maintain a comfortable distance from the others. This requires the vertical position of its virtual goal to be updated correspondingly. In other words, the virtual position of the goal should always be a horizontal offset to the east of the agent's current position.

✦ ***Instruction 2: Handling repulsive effect from nearby crowd***

"Handling repulsive effect from nearby crowd" is a behaviour in the Behaviour Library that causes the agents to try to maintain a desired distance from the others during the simulation. This behaviour rule is designed to represent such individual behaviour (for more details see section 5. 2. 4 Behaviour Library: Handling Repulsive Effect from nearby Crowd).

✦ ***Instruction 3: Keeping a distance from a wall (repulsive effect)***

The crowd will be constrained inside the corridor while moving towards the other side. Behaviour "Keeping a distance from a wall (repulsive effect)" is applied to refine the movement of the agents. Due to the constraints of the geometry, no-one can get across the sidewalls (top and bottom) of the corridor.

In order to minimise the effect of the walls and to make them serve as a geometrical boundary, the desired distance from the wall is set to 0.05 metre (1 pixel) for all agents in the simulation. This means that the agents will only receive the repulsive effect from the walls when they get very close to the wall and this effect serves the purpose of not allowing the agents to cross the wall.

✦ ***Instruction 4: Avoiding collision***

Avoid collision is considered as a subconscious action if an agent detects a forthcoming collision in this simulation. The "avoiding collision" behaviour in the Behaviour Library is designed as a passive behaviour to keep the agent out of potential collisions and will be used in this simulation.

✦ ***Instruction 5: Adjusting the base movement speed***

This behaviour does not have a direct mapping to a behaviour in the Behaviour Library. It is an event that will change the strength of agent's desire (i.e. the F_a parameter in the formulas for behaviour effects calculation) and affects the agent's behaviour indirectly. When a signal of speeding up or slowing down is given to an agent, its parameter of F_a will be changed relatively during the simulation.

➢ **Attributes**

Based on the above analysis, the agents' attributes are set to the following values (see Table 19) by default. However, the values of some of the attributes may vary depending on the configuration of each simulation which will be mentioned explicitly in the relevant simulation sets.

Table 19 Agents' attributes in the simulation – a group walking through a corridor

Attribute	Value (In real)	Value (In simulation)

Size	0.5 m	10 pixels
Default Speed	1.5 m/s	0.5 pixel/frame
Maximum Speed	2 m/s	0.67 Pixel/frame
Sight Range	5 m	100 pixels
Group behaviour range	5 m	100 pixels
Desired distance from wall	0.05 m	1 pixel
Minimum distance from wall	0.05 m	1 pixel
Desired distance from others	0.5m	10 pixel
Minimum distance from others	0.05m	1 pixel
Repulsive modifier (to self)	normal	1
Repulsive modifier (to other)	normal	1

➢ **Starting Positions**

The agents are distributed in a "6 × 4" matrix formation at the west end of the corridor. The detailed distributions of the agents are listed as follows:

- The horizontal distance between two agents is 0.3 metre (6 pixels).
- The vertical distance between two agents is 0.3 metre (6 pixels).
- The distance from the top row of the crowd to the topside (north) of the corridor wall is 0.25 metre (5 pixels).
- The distance from the bottom row of the crowd to the bottom side (south) of the corridor wall is 0.25 metre (5 pixels).

Figure 91 Initial crowd positions of the simulation - a group walking through a corridor

7. 3. 1. 3 Simulation Sets and Results

In order to demonstrate how agent's attributes affect crowd behaviour, three sets of simulations were carried out. Each set of simulations focused on one attribute and tried to explore the effect it had on the group behaviour.

All simulations were repeated ten times and the results represented in this thesis are the average result for each simulation.

Set 1 – Different Desired Distances from Others

- **Purpose**

To test whether group behaviour will be affected if the agents have different values of desired distances from others in the simulations.

- **Configuration**

The attribute of "desired distance from others" (affects $D_{desire\ agent}$ in Formula 53) is tested at the following fixed values 0.2, 0.5, 1, 1.5 and 2 metres (corresponding to 4, 10, 20, 30, and 40 pixels) and the following range values [0.5 ,1], [0.5 ,1.5], [0.5 , 2], and [1 , 2] metres (corresponding to [10 , 20], [10 , 30], [10 , 40], [20 , 40] pixels). In the range values, normal distribution is used to generate the random value from that range.

- **Result and analysis**

Various desired distance values were tested through the simulations. (According to the starting positions of the agents, the distances between adjacent agents are all 0.3 metre). Table 20 presents the results for the homogenous crowd: the crowd took more time to pass through the corridor if they wanted to maintain to a larger desired distance from each other. Table 21 reveals that when the crowd is heterogeneous the result times are closer to the higher limit of the desired distance range. In other words, the crowd's overall speed is mainly determined by the people who want to change to a larger desired distance.

It can be concluded that the crowd can achieve the fastest speed if they reach a stable status/formation. The overall movement of the crowd will be slowed down if members try to adjust their relative positions.

Table 20 Results for agents with fixed desired distance values

Desired distance (metre)	0.2	0.5	1	1.5	2
Result (second)	20.0	21.7	22.6	23.2	23.7

Table 21 Results for agents with range desired distance values

Desired distance range (metre)	0.5-1	0.5-1.5	0.5-2	1-2
Result (second)	22.4	22.9	23.2	23.4

Set 2 – Different Levels of Desires to Maintain a Desired Distance

- **Purpose**

To test whether group behaviour will be affected if the agents have different levels of desires to maintain their desired distances in the simulations.

- **Configuration**

In this set of simulation tests the agents' desires to maintain their desired distances from others (affects F_a in Formula 65). Values of F_a were tested at 1, 3 and 5 at four desired distance settings (0.5, 1, 1.5, and 2 meters).

- **Result and analysis**

The crowd becomes even slower when people are in a hurry to adjust their distances. In test simulations, the values of F_a present the agents' desire level to adjust their distance. It can be seen from the results (Table 22) that the quicker (a higher value of F_a) they adjust to their desired distance, the slower the crowd moves. Figure 92 shows that the crowd reached similar formations in the test of F_a = 3 and F_a = 5, but the latter one takes more take in all cases (see Table 22).

Table 22 Agents with different desires to adjust distance

Result(s) F_a = 1	**F_a = 3**	**F_a = 5**	**Desired distance (m)**

21.7	21.8	21.9	0.5
22.6	24.2	24.8	1
23.1	25.1	26.2	1.5
23.7	26.3	27.4	2

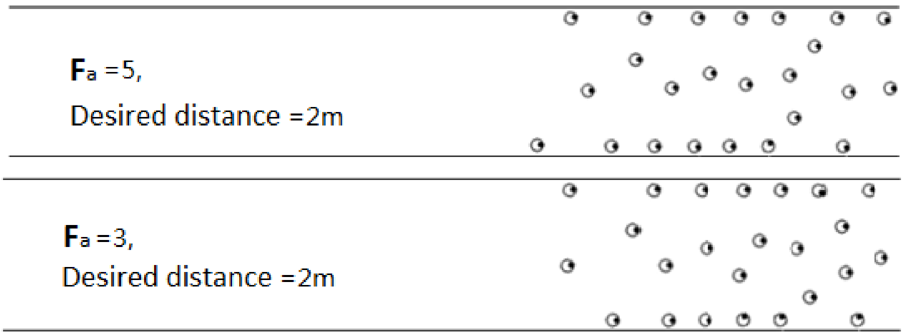


Figure 92 Crowd formations near the end of the corridor

Set 3 – Crowd Sensitivity to Environmental Effects (Speeding-up & Slowing-down)

➢ **Purpose**

To test how group behaviour will be affected if the agents are affected by speed-up and slowing-down effects.

➢ **Configuration**

In the test simulations, agents were triggered to slow down (to 50% of original speed) or speed up (to 200% of original speed) when inside the grey area (Figure 93). Such influence is reflected in the value of E_s (halved or doubled) in all formulas.

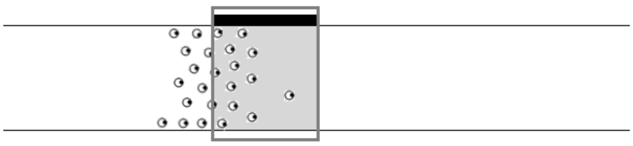


Figure 93 Speed of agents may be affected in grey area

➢ **Result and analysis**

The finding of this simulation is that crowds are more sensitive to the effects of slowing down rather than the effects of speeding up. The results (Figure 94) show that the crowd can be slowed down when a small percentage of members want to slow down.

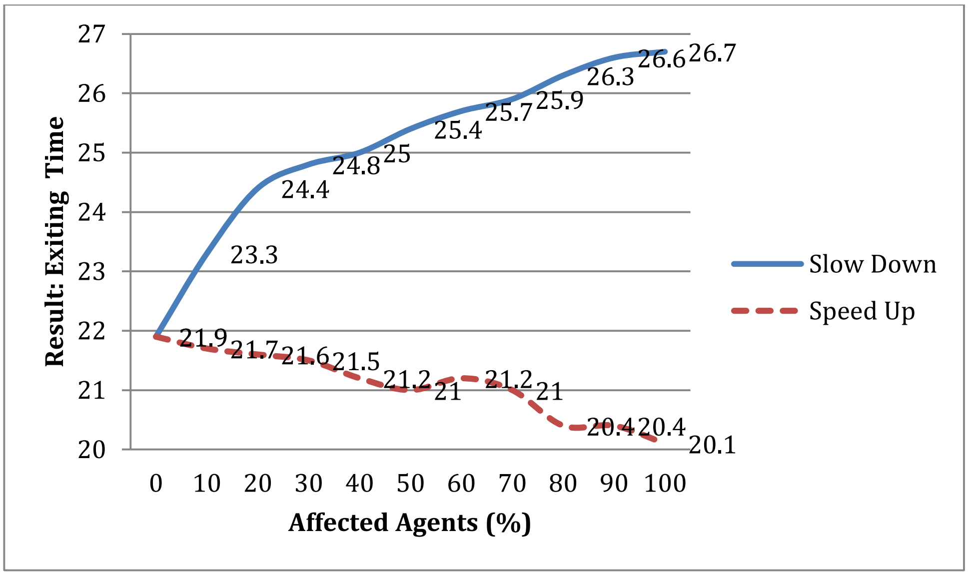


Figure 94 Effects of changing speed

7. 3. 2 Simulation 2: Exiting from a Building

7. 3. 2. 1 Scenario

Scenario Description

This scenario describes visitors evacuating from a museum under a non-emergency circumstance. The following instructions have been given to the crowd:

- Evacuate the museum via your own choice of route.
- Walk at normal speed.
- Try to maintain a comfortable distance from others.
- Movement is restrained by the structure of the museum.

Environment

A rectangular normal size (29m × 13.5m) building is selected to carry out the crowd simulation. The layout of the building is shown in Figure 95. The building contains ten rooms and two exits (Exit A is the emergency exit and Exit B is the main entrance). Each room will contain 10 to 30 visitors as the initial setting for the simulation. The room (No. 10) in the southeast is not for public use so it will not be used in the simulation.

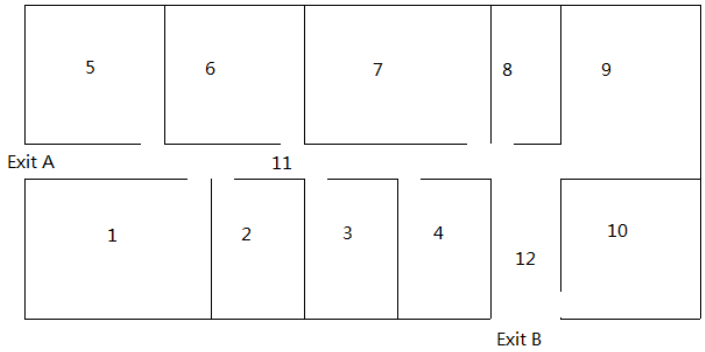


Figure 95 Building plan of the museum

Some key dimensions of the building are listed below:

- The horizontal (from west to east) length of the building is 29 metres.
- The vertical (from north to south) length of the building is 13.5 metres.
- The default width of the corridor is 1.5 metres and so is Exit A.
- The width of the main entrance (Exit B) is 3 metres and the width of all the doors of the rooms is 1 metre.

The Crowd

The individuals in the crowd are considered to have the same attributes and behavioural preferences. In this sense, the crowd is homogenous. However, according to the configurations in different simulation sets, crowd heterogeneity (e.g. agents with different types, various knowledge, etc.) may be introduced.

Purpose

The simulation is designed to test:

- The behaviours of a crowd under different walking speeds.
- The effect of the building layout on crowd behaviour.
- How does crowd distribution affect the overall crowd behaviour?
- The crowd performance by making different choices of route.

7. 3. 2. 2 Simulation Configuration

Simulation Process and Data Capture

➢ **The start and end of one simulation**

When the simulation starts all the visitors begin to evacuate from the building. An agent will exit the building when it reaches the exit (either one). The simulation ends when all the agents have exited. The choice of a route by which to evacuate is based on an agent's knowledge.

➢ **Data capture**

- The total time (in seconds) that the crowd use to exit the building will be captured.
- The crowd formations and the potential queuing phenomenon in the corridor in different simulations will be captured.

Environment Representation

The representation of the museum is identical to Figure 95 introduced in the above section. In simulation, each line represents a wall and the collection of walls forms the geometry of the building. The doors and exits are represented by the gaps between walls.

Because the environment in this simulation is more complicated than the one in the previous simulation, a navigation map is required to calculate the routes for the agents. Figure 96 displays the navigation map for the simulation environment. Each number represents its corresponding region in the geometrical model.

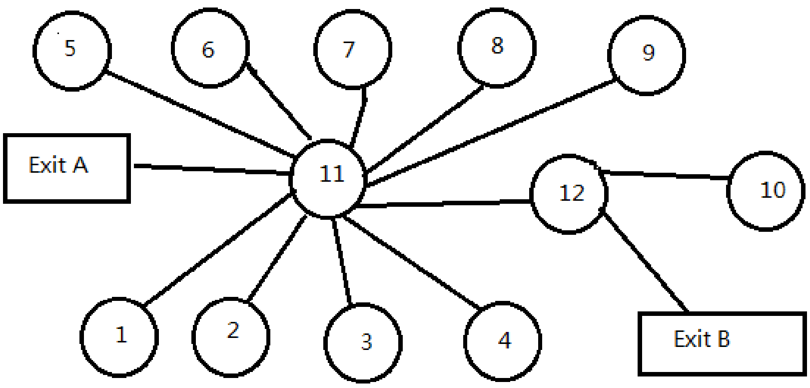


Figure 96 Navigation map for the simulation of exiting from a building

Agent Configuration

➢ **Behaviours**

✧ ***Summary***

The instructions to the crowd can be represented by corresponding behaviours in the Behaviour Library. They are summarised as given below.

Table 23 The agents' behaviour in the simulation of exiting from a building (museum)

	Instructions	Simulation Configurations
1	Evacuate the museum.	Behaviour Library: Seeking
2	Choose own evacuation route.	The waypoints in the evacuation route will be used to set up the targets for "seeking" behaviour.
3	Try to maintain a comfortable distance from others.	Behaviour Library: Handling repulsive effect from nearby crowd
4	Movement is restrained by the structure of the museum.	Behaviour Library: Keeping a distance from a wall (repulsive effect)
5	No specific instruction. Common sense.	Behaviour Library: Avoiding collision

✧ ***Detailed analysis***

✦ ***Instruction 1 & 2: Evacuate the museum with own choice of route.***

In the simulation, all the agents aim to exit the building during the simulation by following a route of self-choice. The behaviour of following such a route can be represented by the behaviour "seeking" in the Behaviour Library. Because the routes usually contains multiple waypoints (e.g. a typical route in this simulation can be: room -> corridor -> exit), the target of "seeking" behaviour will be updated to make sure that each agent is walking in the correct direction.

The choice of the evacuation route is decided by the agent's knowledge of the environment. In this simulation, the decisions are made based on:

- Awareness of the emergency exits.
- The distance to the nearest known exit.

✦ ***Instruction 3: Maintain a comfortable distance from others***

"Handling repulsive effect from nearby crowd" is the behaviour in the Behaviour Library that represents that the agents will try to maintain a desired distance from other agents during the simulation.

✦ ***Instruction 4: Movement is restrained by the structure of the museum***

The crowd will be constrained by the walls while moving towards the exits. The behaviour "Keeping a distance from a wall (repulsive effect)" is applied to refine the movement of agents.

✦ ***Instruction 5: Avoid collision***

Avoid collision is considered as a subconscious action if the agent detects a forthcoming collision in this simulation. A typical situation for this behaviour is when an agent tries to leave a room and go into a corridor. It may need to manoeuvre its position continuously in order to enter the corridor because there will be many other agents trying the same action.

➢ **Attributes**

Based on the above analysis, the agents' attributes are set to the values in Table 24 by default. The values of some of the attributes may vary depending on the simulation which will be discussed in the sections on those particular simulations.

Table 24 Agents' attributes in the simulation - visitors evacuating from a museum

Attribute	**Value (In real world)**	**Value (In simulation)**
Size	0.5 m	10 pixels
Default Speed	1.5 m/s	0.5 pixel/frame

Maximum Speed	2 m/s	0.67 Pixel/frame
Desired distance from wall	0.35m	7 pixel
Minimum distance from wall	0.05 m	1 pixel
Desired distance from others	1 m	20 pixel
Minimum distance from others	0.05m	1 pixel
Repulsive modifier (to self)	normal	1
Repulsive modifier (to other)	normal	1
Repulsive modifier (to wall)	normal	1

- ### Starting Positions

For the default setting, there are 140 agents located in the nine rooms. The distribution of the agents are summarised in Table 25 and shown in Figure 97.

Table 25 Numbers of agents in each room

Room	1	2	3	4	5	6	7	8	9	Total
Amount	20	12	12	12	15	15	20	9	25	140
Formation	4 × 5	4 × 3	4 × 3	4 × 3	3 × 5	3 × 5	4 × 5	3 × 3	5 × 5	

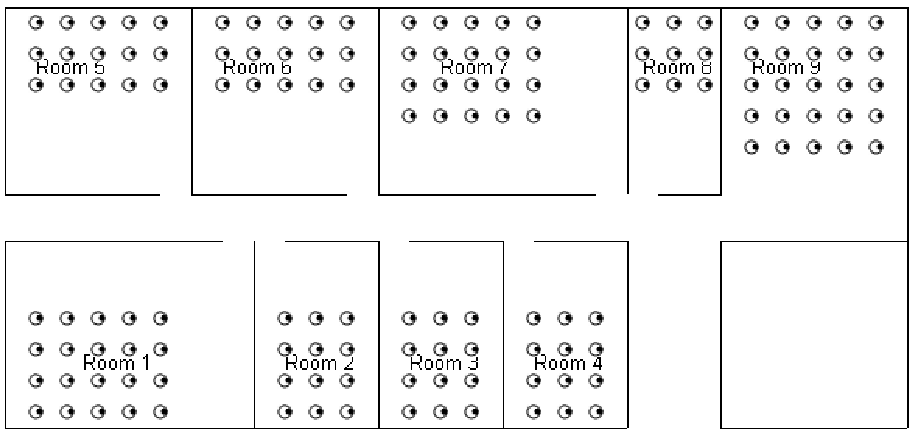


Figure 97 Starting positions of the crowd - simulation of exiting from a building

More specifically, the distributions of the agents in each room are listed in Table 26.

Table 26 Detailed distributions of the agents in each room.

Room	Group formation (Rows × Columns)	Horizontal gaps between the agents (metres)	Vertical gaps between the agents (metres)	Group distance to the left of the room (metres)	Group distance to the top of the room (metres)
1	4 × 5	0.75	0.75	1	1
2	4 × 3	0.75	1	1	1
3	4 × 3	0.75	1	1	1.5
4	4 × 3	0.75	1	1	1
5	3 × 5	0.75	1	1	3.5
6	3 × 5	0.75	0.75	1	2
7	4 × 5	0.75	0.75	1	2
8	3 × 3	0.75	0.75	1	2
9	5 × 5	0.75	0.75	1	2

7. 3. 2. 3 Simulation Sets and Results

In order to demonstrate how agents and environment can affect crowd behaviour, several sets of simulations with carried out with variations.

All the simulations were repeated 50 times and the results represent the average result for the each simulation.

Set 1 – The Relationship between Walking Speed and Exit Time

- **Configuration**

This set tested the effect on exit time by increasing the average walking speed of the agents. The following speeds were tested: 0.9, 1.2, 1.35, 1.5, 1.65, 1.8, 1.95, 2.1, 2.4, and 2.7 m/s.

- **Result and analysis**

The results in Figure 4 show how much time the crowd (using different speeds) needed to exit the building. It can be seen that the slope of the line is quite large when speed is low (below 1.5 m/s) and the slope decreases as the speed increases.

At lower speeds, the graph indicates that evacuation time increased dramatically when speed reduces. However, the evacuation time do not get much improvement in higher speeds. For example, when speed decreases from 2 m/s to 1 m/s (1 m/s decrease), evacuation time increases by about 30 seconds; whereas when speed decreases from 4 m/s to 3 m/s (1 m/s decrease as well), evacuation time only increases by about 5 seconds.

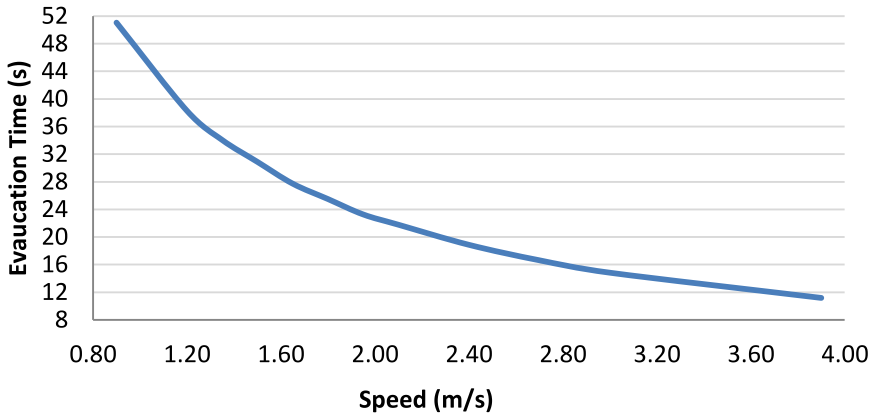


Figure 98 Evacuation time with various speeds (corridor = 1.5m)

Set 2 – Relationship between corridor width and exit time

➢ Configuration

Two values for the corridor width were tested in this set. They were set at 2 metres and 2.5 metres. Each width was tested in all the speeds used in set 1 as well.

➢ Result and analysis

This set of simulations aimed to show the impact of changing the width of the corridor. In this test, the width of corridor was increased to 2 metres (approximately three persons can walk in parallel in such a corridor as was observed in the simulation) and 2.5 metres (approximately four persons can walk in parallel in such a corridor). Figure 99 shows the results giving the evacuation times at different speeds for these two cases (as well as for the original width of the corridor). The curves are very similar although the widths of the corridor are different. Comparing these results with the results from simulation set A, it can be seen (Table 27) that evacuation time can be improved with a wider corridor, but that increasing the width of the corridor has an limited effect on improving evacuation time.

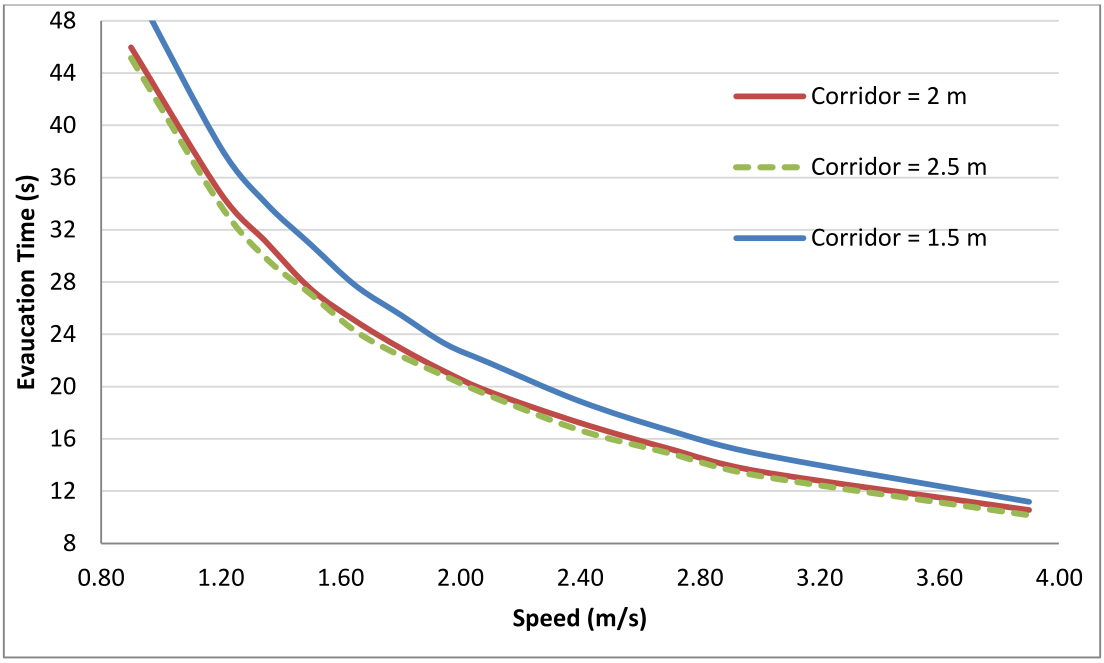


Figure 99 Evacuation time with various speeds (corridor = 1.5m, 2m, and 2.5m)

Table 27 Comparisons of evacuation times with different width of corridors (1.5m, 2m and 2.5m)

Speed (m/s)	0.90	1.20	1.35	1.50	1.65	1.80	1.95	2.10	2.40	2.70	3.00
2.0m : 1.5m	90%	91%	91%	89%	90%	90%	91%	90%	91%	92%	91%
2.5m : 1.5m	88%	88%	88%	88%	87%	88%	89%	88%	88%	89%	89%
2.5m : 2.0m	98%	97%	96%	99%	97%	97%	98%	98%	97%	98%	97%

This experiment also indicates that the effect of the corridor's width on evacuation time is independent to the speed of a crowd. It can be seen that the improvements at different walking speeds are similar in the comparison with a given pair of corridor widths.

Set 3 – Different Crowd Compositions (Effect of Elderly People)

➢ **Configuration**

In this set of simulations, the crowd consisted of a group (twenty) of elderly people and normal people. The older people move more slowly than the others do. The elderly group were tested as starting from both Room 7 and Room 1. The speeds of the elderly people being tested were set at 0.9 and 1.2 m/s

➢ Result and analysis

✧ *Evacuation time increased*

The results are shown in Table 28. It can be seen that, compared to the evacuation time (30.88 seconds) in the case of all people move at 1.5 m/s, the evacuation times in all cases of having a group of elderly people are increased.

Table 28 Evacuation time with elderly people in Room 1 and Room 7

Speed (m/s)	Evacuation time (s)
All: 1.5	30.88
Room 7: 1.2, The other rooms : 1.5	32.73
Room 1: 1.2, The other rooms: 1.5	34.51
Room 7: 0.9, The other rooms: 1.5	39.32
Room 1: 0.9, The other rooms: 1.5	42.49

Additionally, by comparing the two different starting positions (Room 7 and Room 1), one can expect that the evacuation time in the case of elderly people starting in Room 7 should be shorter than the case that they starting in Room 1 because Room 7 is more close to the exit. The results (see Table 28) show that:

- when elderly people walk at 1.2 m/s, the improvement on evacuation time is 34.51 - 32.73 = 1.78 seconds
- when elderly people walk at 0.9 m/s, the improvement on evacuation time is 42.49 - 39.32= 3.17 seconds

However, the improvements are less than expected (based on the theoretical analysis below). In theory, the distance from the gate of Room 1 to Room 7 is 12 metres. The walking times for this distance are 8 seconds at the speed of 1.5 m/s, 10 seconds at 1.2 m/s and 13.3 seconds at 0.9 m/s. If the elderly people start from Room 1, they have to move through the corridor to reach the door of Room 7. Based on the above calculation, they should take 10 - 8 = 2 seconds more if they walk at 1.2

m/s and 13.3 – 8 = 5.3 seconds more than normal people do in order to reach Room 7.

According to the results, it seems that the extra time needed when elderly people are in Room 1 is less than expected. The cause of this situation can be found from the real time simulator. In the case of elderly people starting from Room 7, they will slow down the crowd behind them when entering the corridor, as they are hard to overtake. In this case, they are slowing down the people behind them. Although the people from Room 1, 2, 3, 5, and 6 have a higher speed, they cannot find enough space to overtake the elderly people. On the other hand, a gap also has been observed on the snapshot of from the simulation (see Figure 100). This is because people in Room 8 and Room 9 move faster than the elderly people do.

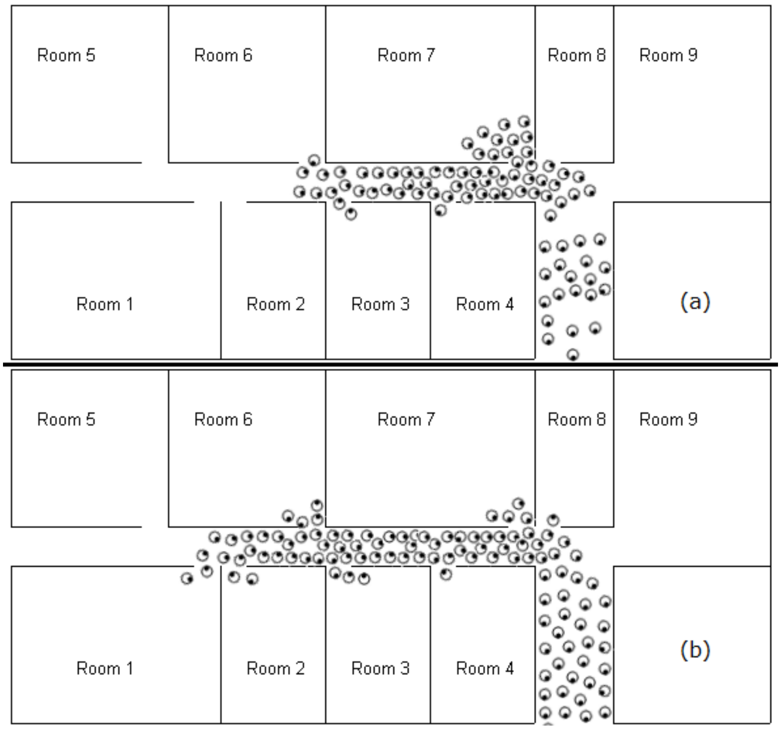


Figure 100 (a) Gap observed in simulation with elderly people (b) simulation with no elderly people

✧ ***Elderly people in room 7 versus normal people in room 7***

Another phenomenon that was observed in the simulation is that, when all the people have the same speed, the ones that are in a room that is closer to an exit usually evacuate quicker. But if these people have a slower speed compared to the others, they will have difficulties in inserting themselves into the crowd flow thus they may have a longer evacuation time than those in the further rooms. From

Figure 100a, it can be observed that when all the rest of the rooms are empty, there are still half the elderly people (in Room 7) waiting to enter the corridor. The right part of Figure 100 is the simulation showing all the people having the same speed. It can be seen that people leave the rooms at a similar rate.

Set 4 – How does the Building Layout (Positions of the Doors) Affect the Evacuation Time?

➢ Configuration

This set of simulations tested the scenario of an alternative layout of the building. The locations of the doors of room 2 and room 7 were changed to the other side of the room as shown in Figure 101 (the left-hand part of the figure shows the original design and the right-hand part shows the alternative design).

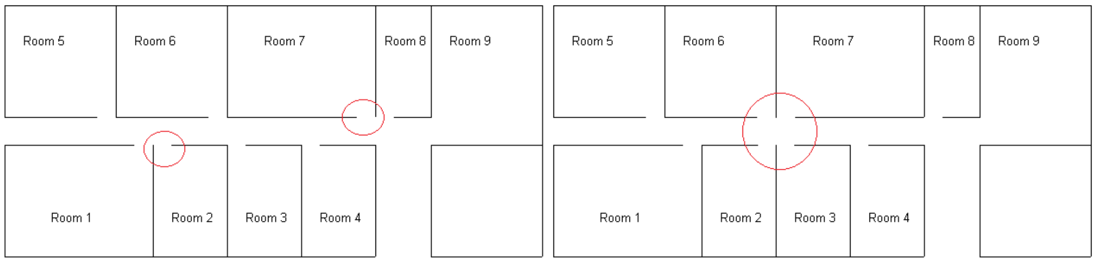


Figure 101 Door positions in original design (left) or and in alternative design (right)

The alternative design puts the doors of rooms 2, 3, 6 and 7 together.

The crowd configurations are set to the default settings. They will use the main entrance (next to Room 4) as the exit.

➢ Result and analysis

By moving the doors of Rooms 2, 3, 6 and 7 closer, congestion (circled in Figure 102) was observed near the four close doors.

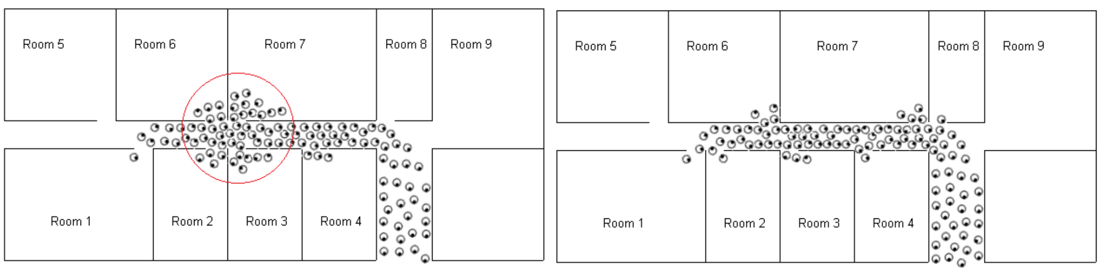


Figure 102 Different building layout designs

Figure 103 shows that the positions of the doors do have an impact on the evacuation time. It takes about 10% more time to exit the building with these alternative door positions. It indicates that doors connecting rooms to a corridor should be distributed separately from each other in order to avoid congestion. If several rooms have their doors close to each other, this could cause congestion because people need to enter the same area of corridor.

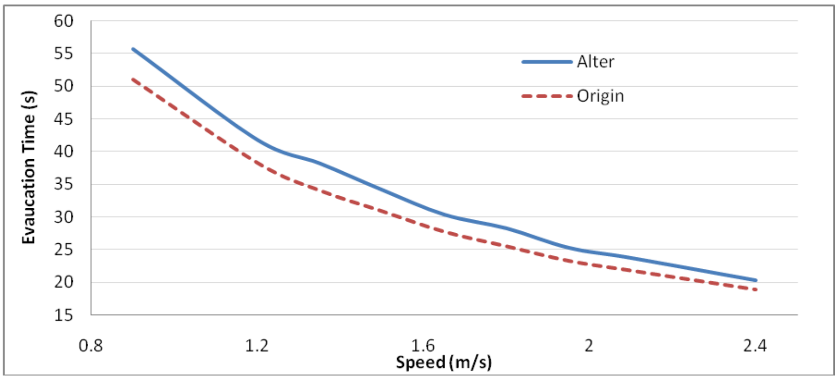


Figure 103 Evacuation times with original layout and alternative layout

Set 5 - Different Exit Routes

➢ **Configuration**

This set of simulations showed the evacuation of people who were given the ability to make use of an emergency exit. It is assumed that people in Rooms 1, 2, 5 and 6 are informed (which can be achieved by providing a guide or a sign) about the location of the emergency exit (which is the west exit of the building).

➢ **Result and analysis**

Figure 104 shows that using an emergency exit route can decrease evacuation times significantly. The evacuation times are around 33% less when compared to using only one exit at various speeds. This is much more efficient than increasing the crowd speed or the width of a corridor. It indicates that a good emergency plan (i.e. guiding the crowd to use a shorter emergency exit route) is crucial in an emergency evacuation.

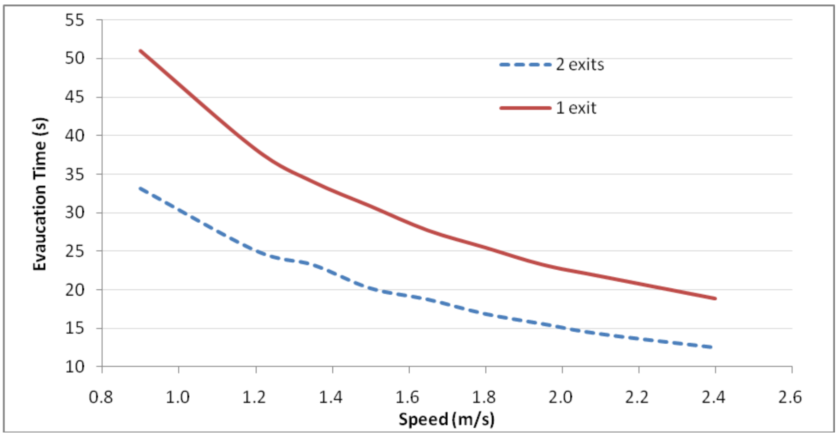


Figure 104 Evacuation times with using different exits

Set 6 - Heterogeneous Crowd with Different Walking Speeds

➢ Configuration

This set of simulations introduced one type of heterogeneity in the crowd. It was considered that the speeds of individuals should have some variations to the default value to reflect a more realistic scenario. In order to represent this, the walking speeds of the agents were randomly assigned at 1.5 ± 10% metres/second, thus the speeds were normally distributed in the range of [1.35, 1.65].

This simulation was repeated five hundred times to collect a large number of records for data analysis.

➢ Result and analysis

The results (Table 29) show that the average time for exiting the building is 31.5 ± 0.02 seconds (confidence level = 95%).

Table 29 Statistics of simulating exiting from a building with speed variations

Mean	31.52229638
Standard Error	0.021076063
Median	31.5333964
Minimum	29.7500595
Maximum	33.0333994

Confidence Level (95.0%)	0.041408758

It can be seen that the ±10% of walking speed does not affect the crowd exit time much. It indicates that, in the situation of this evacuating scenario, the heterogeneity of speed in the crowd does not make much difference on the average result in the statistics.

7. 3. 3 Simulation 3: Evacuation from a Shopping Mall

7. 3. 3. 1 Scenario

Scenario Description

This scenario described the scenario of customers exiting from a small shopping mall. The following instructions were given to the crowd:

- Evacuate the shopping mall with own choice of route.
- Walk at normal speed.
- Try to maintain a comfortable distance from others.
- Movement is restrained by the structure of the shopping mall.

Environment

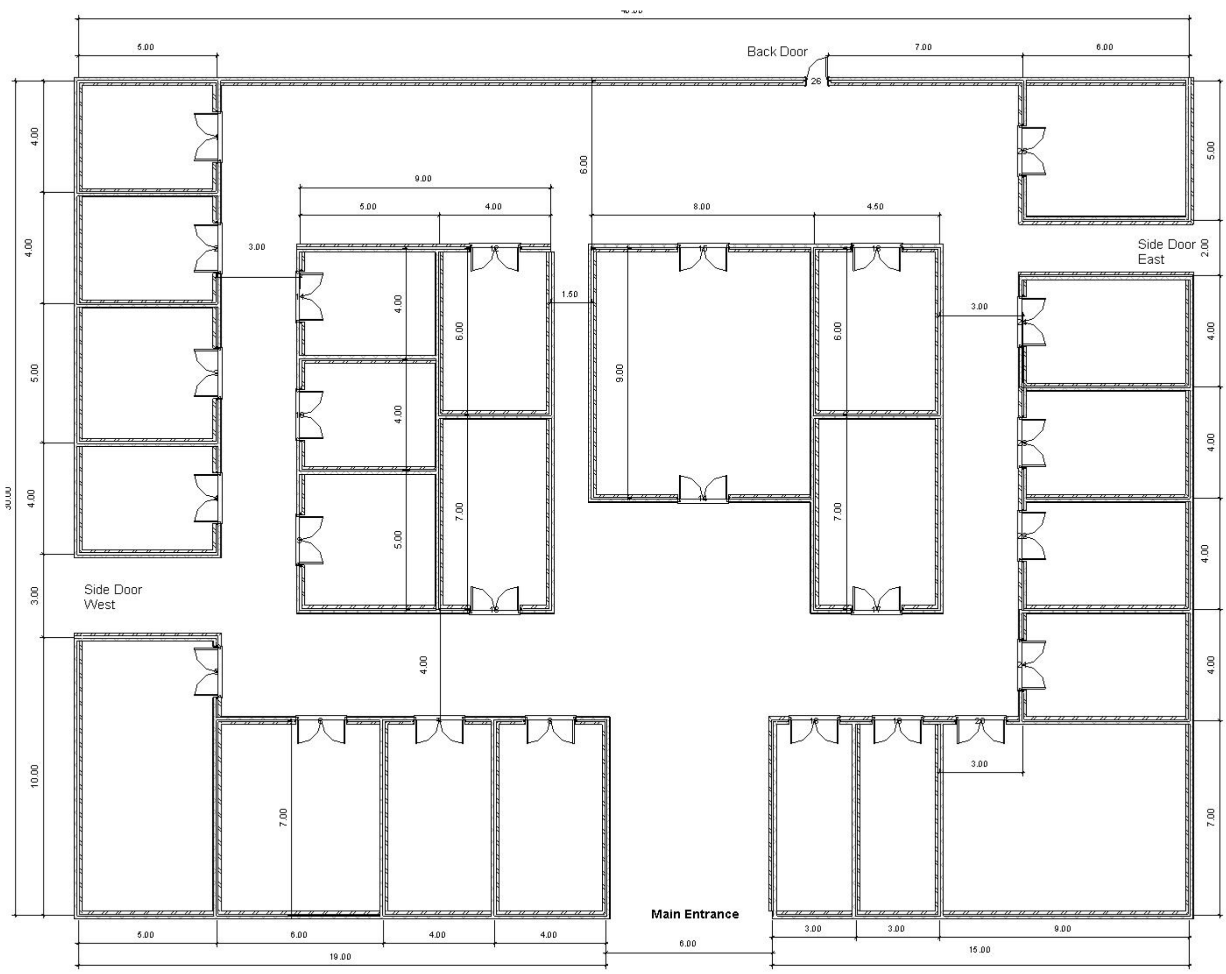


Figure 105 Building plan of the shopping mall

The shopping mall has one level only and it is built in a rectangular shape. The dimension of the shopping mall is 40 X 30 in metres. It has four entrances/exits located on each side of the shopping mall and the main entrance is located on the south of the building.

The Crowd

The individuals in the crowd are considered to have the same attributes and behavioural preferences. In this sense, the crowd is homogenous. However, according to the configurations in different simulation sets, crowd heterogeneity (e.g. agents with different types, various knowledge, etc.) may be introduced.

Purpose

In this section, the scenario of an evacuation from a shopping mall was selected to demonstrate how this crowd model can represent a heterogeneous crowd and can show the effect of different individual decisions as well as the influence of the environment on crowd behaviour.

7. 3. 3. 2 Simulation Configuration

Simulation Process and Data Capture

- **The start and end of one simulation**

The agents will start to exit the shopping mall when the simulation begins. The agents will exit the shopping mall when they reach any of the exits (the main entrance, the side door west and the side door east are used in the simulations). The simulation ends when all agents have exited the shopping mall.

- **Data capture**

  - The total time (in seconds) that the crowd used to exit the shopping mall will be recorded.
  - The crowd formations / queuing phenomenon in the corridor in different simulations will be captured as snapshots.

Environment Representation

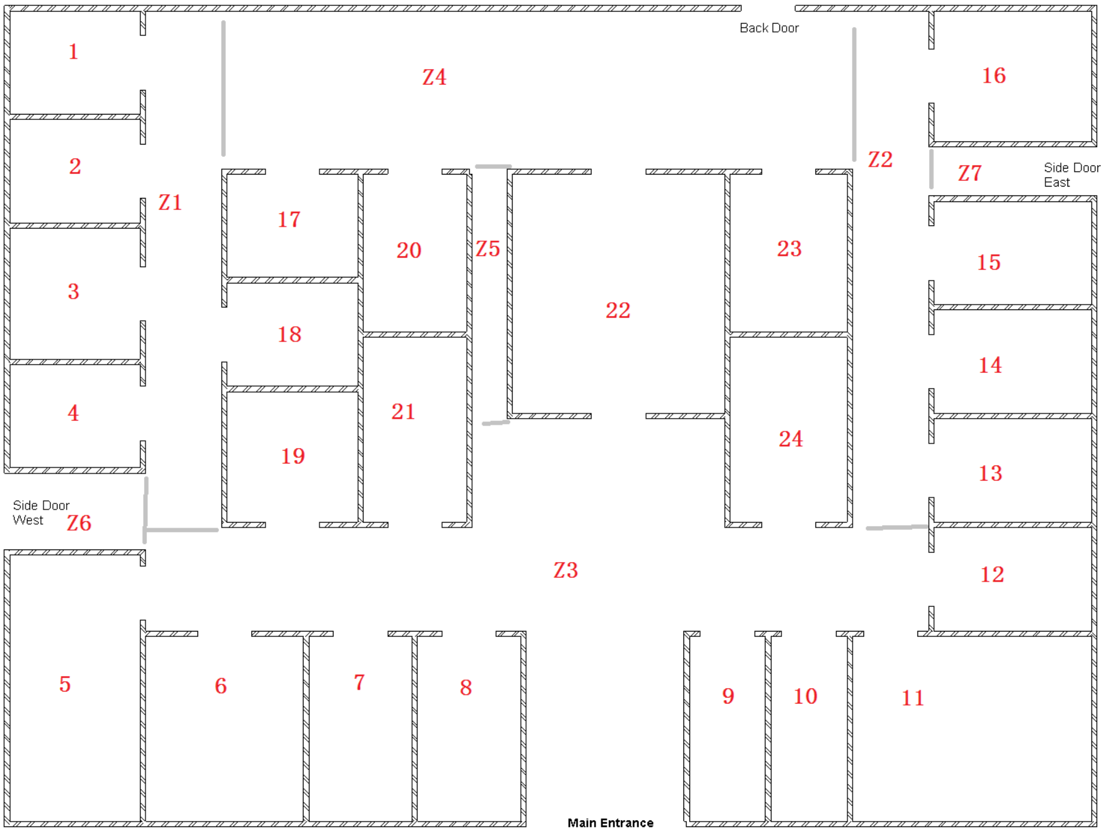


Figure 106 Map of the shopping mall

Agents

➢ Behaviours

✧ *Summary*

The instructions to the crowd can be represented by corresponding behaviours rules the Behaviour Library. They are summarised below.

Table 30 Behaviour configuration for evacuation from a shopping mall

	Instructions	Simulation Configurations
1	Evacuate the shopping mall.	Behaviour Library: Seeking
2	Choose own evacuation route.	The waypoints in the evacuation route will be used to set up the targets for "seeking" behaviour.

3	Try to maintain a comfortable distance from others.	Behaviour Library: Handling repulsive effect from nearby crowd
4	Movement is restrained by the structure of the shopping mall.	Behaviour Library: Keeping a distance from a wall (repulsive effect)
5	No specific instruction. Common sense.	Behaviour Library: Avoiding collision

✧ ***Detailed Analysis***

✦ ***Instruction 1 & 2: Evacuate the shopping mall with own choice of route.***

In the simulation, all the agents aim to exit the building during the simulation by following a route of self-choice. The behaviour of following such route can be represented by behaviour "seeking" in the Behaviour Library. Because the routes usually contain multiple waypoints (a typical route in this simulation can be: room -> corridor -> exit), the target of "seeking" behaviour will be updated to make sure that the agent is walking in the correct direction.

The choice of an evacuation route is decided by the agent's knowledge of the environment. In this simulation, the decisions are made based on:

- Awareness of the emergency exits.
- The distance to the nearest known exit.

✦ ***Instruction 3: Maintain a comfortable distance from others***

"Handling repulsive effect from nearby crowd" is the behaviour in the Behaviour Library to represent the fact that the agents will try to maintain a desired distance from other agents during the simulation.

✦ ***Instruction 4: Movement is restrained by the structure of the shopping mall***

The crowd will be constrained inside the corridors while moving toward the other ends of the corridors. The behaviour "Keeping a distance from a wall (repulsive effect)" is applied to refine the movement of the agents.

✦ ***Instruction 5: Avoiding collision***

Avoiding collision is considered as a subconscious action if the agent detects a forthcoming collision in this simulation. A typical situation for this behaviour is when an agent tries to leave a room to enter into a corridor. It may need to manoeuvre its position continuously in order to enter the corridor because it will be surrounded by other agents.

- **Attributes**

Based on the above analysis, the agents' attributes are set to the values in Table 24 by default. The values of some attributes may vary depending on the simulation. These values will be discussed in the particular simulations.

Table 31 Agents' attributes in the simulation – Evacuation from a shopping mall

Attribute	Value (In real world)	Value (In simulation)
Size	0.5 m	10 pixels
Default Speed	1.5 m/s	0.5 pixel/frame
Maximum Speed	2 m/s	0.67 Pixel/frame
Desired distance from wall	0.3m	6 pixel
Minimum distance from wall	0.05 m	1 pixel
Desired distance from others	0.7 m	14 pixel
Minimum distance from others	0.05m	1 pixel
Repulsive modifier (to self)	normal	1
Repulsive modifier (to other)	normal	1
Repulsive modifier (to wall)	normal	1

- **Starting Positions**

The starting positions of the agents were presented in two cases: normal and full. In the normal-loaded case, there were 364 agents who were all located in the shops (this distribution is shown in Figure 107). In the full-loaded case, there were 650 agents and they were distributed in both the shops and the corridors (this distribution showed in Figure 108).

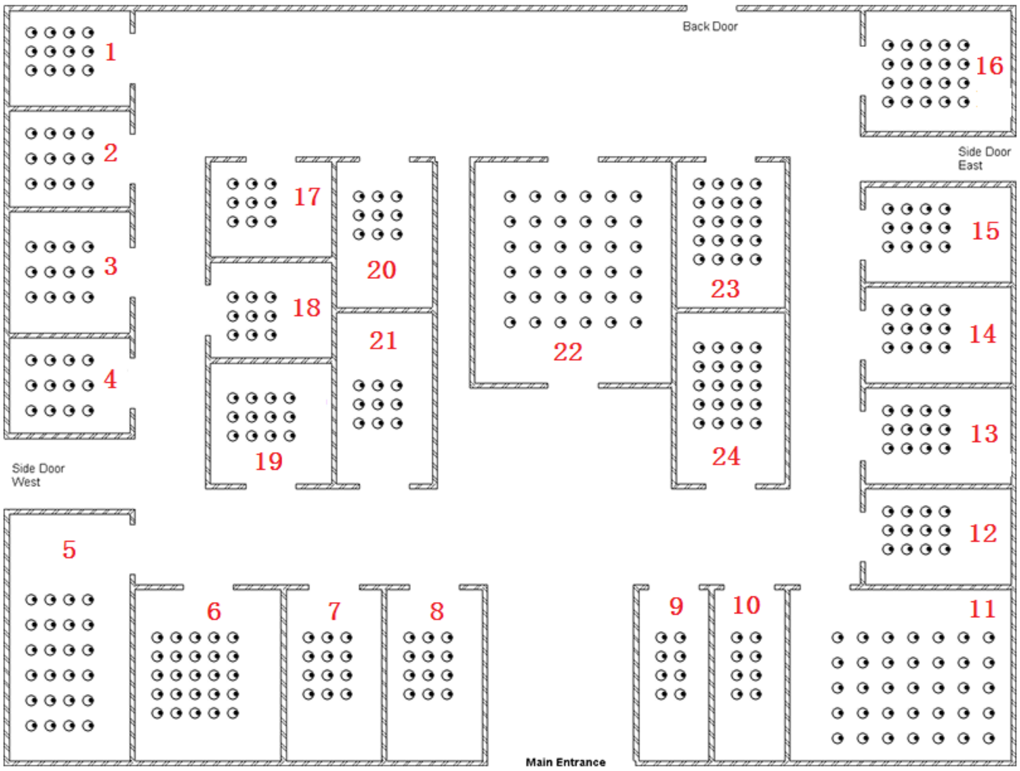


Figure 107 Crowd's initial distribution in the normal-loaded case (shop numbers showed from 1 to 24)

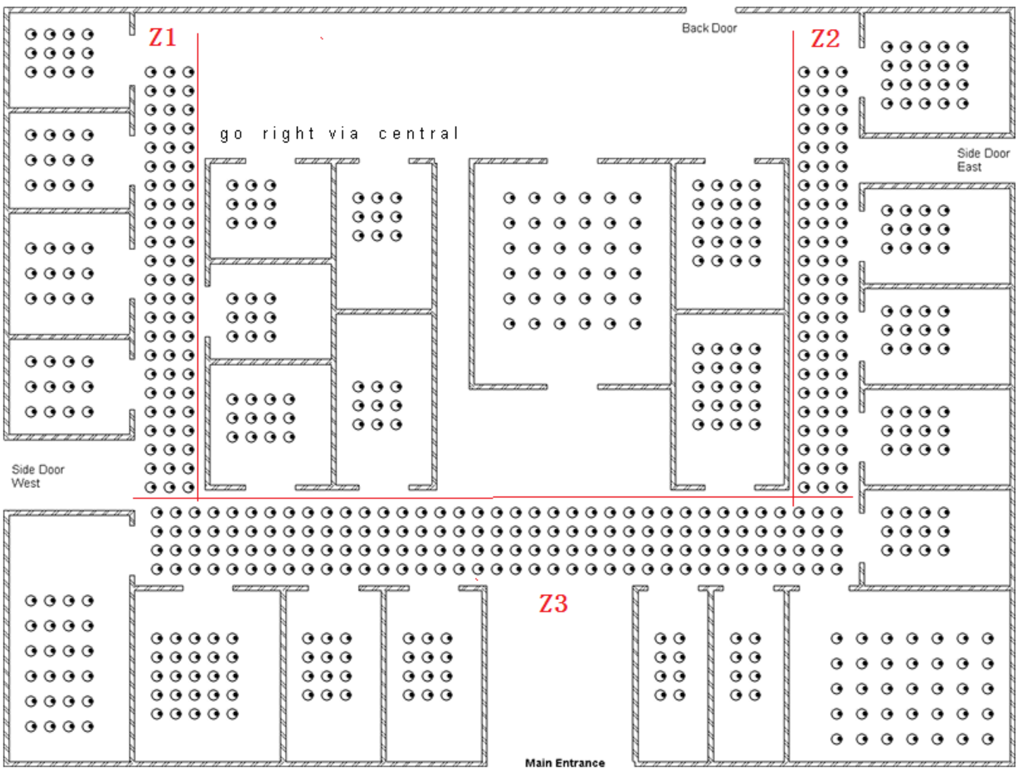


Figure 108 Crowd's initial distribution in the full-loaded case (corridors are divided into three zones: Z1, Z2, Z3)

The starting positions of each agent in the normal-loaded case are listed in Table 32 (the gaps and distances are calculated based on the agents' positions, i.e. the centres of the circles that represent the agents). A total of 364 agents were distributed in 24 shops.

Table 32 Detailed starting positions of the agents in the normal case

Shop	Group formation (Rows X Columns)	Horizontal gaps between the agents (metres)	Vertical gaps between the agents (metres)	Group distance to the left of the shop (metres)	Group distance to the top of the shop (metres)
1	3 X 4	0.75	0.75	1	1
2	3 X 4	0.75	1	1	1
3	3 X 4	0.75	1	1	1.5
4	3 X 4	0.75	1	1	1
5	6 X 4	0.75	1	1	3.5
6	3 X 3	0.75	0.75	1	2
7	3 X 3	0.75	0.75	1	2
8	3 X 4	0.75	0.75	1	2
9	5 X 5	0.75	0.75	1	2
10	3 X 3	0.75	0.75	1	2
11	3 X 3	1	1	2	2
12	4 X 3	0.75	0.75	1	1
13	4 X 3	0.75	0.75	1	1
14	6 X 6	0.75	0.75	1	1
15	5 X 4	0.75	0.75	1	1
16	5 X 4	0.75	0.75	1	1.5
17	4 X 2	0.75	0.75	1	1
18	4 X 2	0.75	0.75	1	1.5
19	4 X 5	0.75	0.75	1	1.5
20	3 X 4	0.75	0.75	1	1.5
21	3 X 4	0.75	0.75	1	3
22	3 X 4	1	1	1.5	1.5

23	3 X 4	0.75	0.75	1	1
24	5 X 7	0.75	0.75	1	1.5

For the full-loaded case, another 286 agents were added into the corridors in addition to numbers in the normal case. The agents located in the shops had the same starting positions. The agents in the corridors were distributed into three zones which are indicated by the red lines in Figure 108. Their distributions are listed in Table 33 (the gaps and distances are calculated based on the agents' positions, i.e. the centres of the circles that represent the agents).

Table 33 Detailed starting positions of the agents in the corridors in the full case

Zone	Group formation (Horizontal X Vertical)	Horizontal gaps between the agents (metres)	Vertical gaps between the agents (metres)	Group distance to the left side of the corridor (metres)	Group distance to the top wall of the shopping mall (metres)
Z1	23 X 3	0.75	0.75	0.75	2.5
Z2	23 X 3	0.75	0.75	0.70	2.5
Z3	4 X 37	0.75	0.75	1	20

7. 3. 3. 3 Simulation Sets and Results

Set 1 – Normal Evacuation Circumstances

➢ **Configuration**

This set tested the evacuation time at various walking speeds of the crowd. It contains three cases with different crowd amounts. Case A only has 2 people in the top shops (i.e. shop 1 and shop 16 in Figure 107) which could provide an evacuation time purely determined by walking speed. Case B is the normal-loaded case with 364 people (distribution illustrated in Figure 107) and Case C is the full-loaded case with 650 people (distribution illustrated in Figure 108). The individuals are designed as homogenous so the results represent overall group behaviour.

➢ Result and analysis

Figure 109 shows that the crowd has a higher density where the crowd turns into the exit from the corridor (see circled areas). Congestion occurred as people from the two directions merged into one direction. It can be seen that some people were pushed into the centre before they could turn and such a phenomenon become more noticeable in large crowds (comparing Cases B and C).

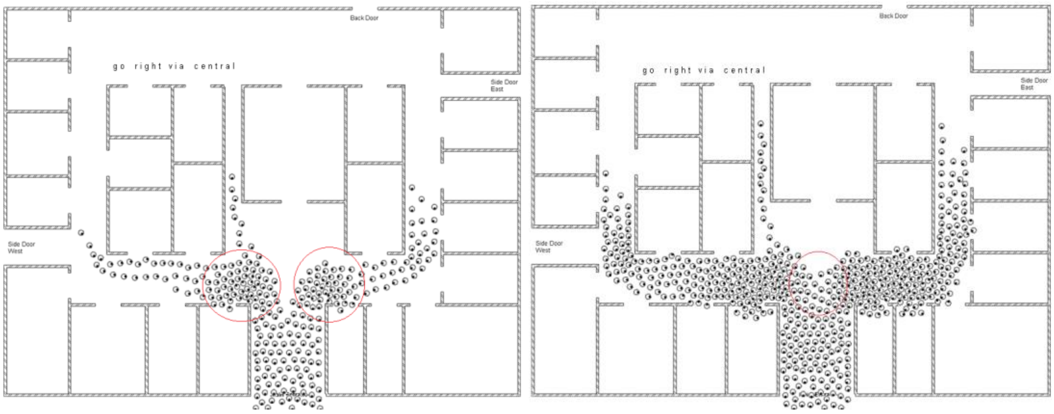


Figure 109 Congestion during the evacuation. (Left) Case B - 364 people; (Right) Case C - 650 people.

The results are shown in Figure 110 (speeds were tested from 0.5 m/s to 2.0 m/s, at a 0.1 m/s interval.). It can be seen that as the total number of the crowd increased, it took more time to evacuate. The slower the walking speed, the larger differences exist in the evacuation times between the two cases. Figure 110 also reveals that despite congestion becoming more serious with larger crowds, the relationship between walking speed and evacuation time remains roughly the same, as all three cases have similar curves.

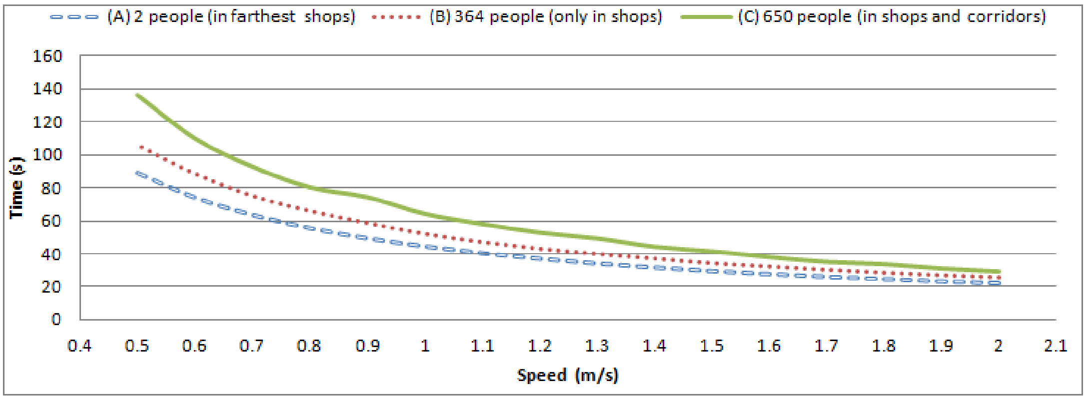


Figure 110 Evacuation times at different walking speeds (3 scenarios)

Set 2 - Evacuation with Elderly People

➢ Configuration

In this set of simulations, the purpose was to test whether individuals with significant differences can make any impact on the evacuation times. Eight elderly people are added into the corridor in addition to the normal-loaded case. These eight people are tested in two cases:

- Case A (Figure 111(a)): elderly people are loosely located in the corridor.
- Case B (Figure 111(b)): elderly people are distributed as a group.

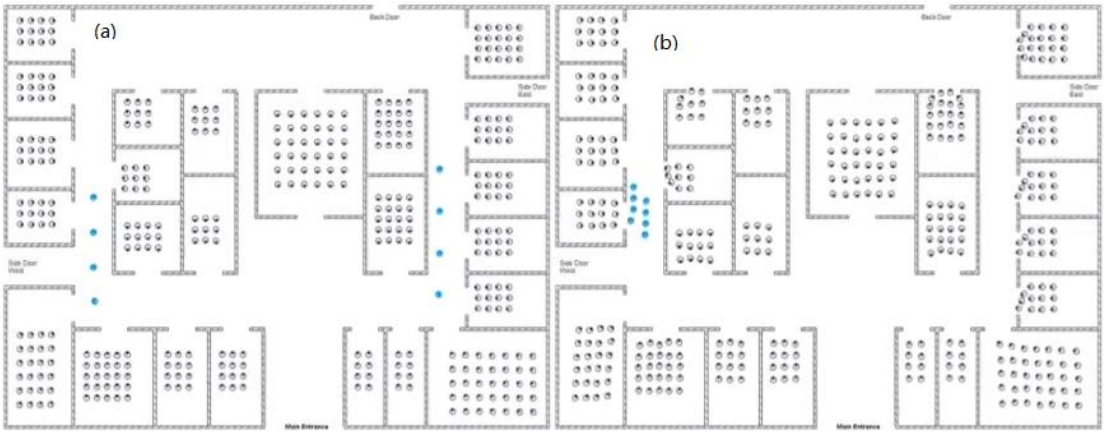


Figure 111 Shopping mall evacuation with elderly people: (a) Case A; (b) Case B

➢ Result and analysis

As a result, for case A, the evacuation time was 42.9 seconds which was 0.9 second more than the normal case (without elderly people). For case B, the evacuation time was 48.2 seconds which was 8.2 seconds more than the normal case. The results revealed that, although same amount of the elderly people was added into the simulation, their effects on others were largely determined by their initial positions. In other words, the layout of environment needs to be taken into account. Figure 112(a) showed that, in case A, the normal people's movements were not affected much as they could overtake the elderly people one by one easily. But in case B, because the elderly people started as a group, their slow movement actually blocked the others and reduced the efficient width of the corridor. It can be observed in Figure 112(b), the evacuation rate for the left side was slower than the right side due to the blockage effect of the elderly group. The results suggested that individuals' effects on crowd behaviour were environment dependant.

Figure 112 Positions of the elderly people during the evacuation: (a) Case; (b) Case

Set 3 - Different Exit Routes

- **Configuration**

In this set of simulations, the effects of agents making different decisions were tested: the agents in the left corridor might randomly use the west exit or the main entrance (50% chance of each). These agents are highlighted in blue in the simulation (see Figure 113).

- **Result and analysis**

During the evacuation process, clear differences were observed. The agents choosing the closer exit (which was the west exit) did evacuate much sooner than the others. Figure 113(b) shows at around 20 seconds, the agents who chose the west exit have already evacuated while other agents queuing at the main entrance.

Surprisingly, the evacuation time was only improved by 0.2 second in this case than in the case of using only one exit. The reason that the overall evacuation time did not improve much was because the remaining crowd were still queuing at the south main entrance while the west exit was cleared. The result suggested that better decisions by individuals (choosing the closer exit in this case) cannot always be reflected in the final result (overall the scenario still had a similar evacuation time).

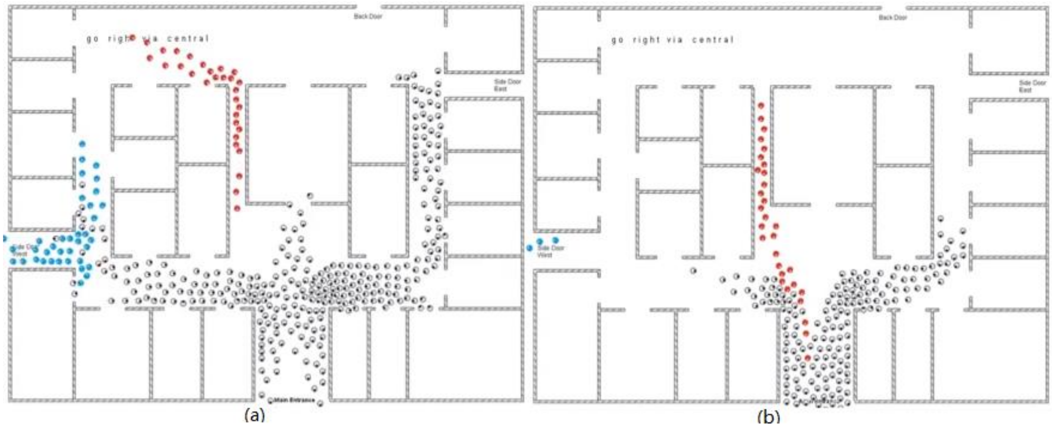


Figure 113 Simulation of some agents (in blue) make use of the west exit

7. 4 Summary of the Chapter

This chapter firstly introduces the two types of evaluation carried out in this research study. Then it presents the fundamentals of the crowd simulations and the global configurations that have been used across the evaluation simulation series, followed by the main content of this chapter which presents a series of simulations to demonstrate the applications of the crowd model as the first part of the model evaluation. Three scenarios (i.e. walking through a corridor, exiting from a small building, and an evacuation from a shopping mall) were selected and implemented with the proposed crowd model in simulation.

The results from these simulations evaluated the proposed crowd model in the following aspects: 1) the ability to represent crowd heterogeneity and its influences on crowd behaviour; 2) the ability to model complex crowd behaviours through the interactions of simplified individual behaviours; 3) the ability to reflect environmental influences on a crowd and its behaviours.

Chapter 8 MODEL EVALUATIONS: MODEL VALIDATIONS

8. 1 Introduction

This chapter presents the validation of the simulation results that are demonstrated by the model. The validation is achieved through demonstrating that, with proper configurations, the proposed crowd model can represent proved or experimentally observed crowd phenomena in existing studies.

The validation assesses the proposed crowd model in the following aspects:

- Can this model simulate crowd behaviours that have been presented by existing crowd simulation studies?
- Can this model simulate crowd behaviours that have already been proved in experimental observations?
- To what extent will the simulation results be reliable?
- What are the differences between the simulation results and the real data?

In addition, through further simulations of, and discussions on, some crowd behaviours presented during the model validation, the applications of the crowd model have been further demonstrated.

8. 1. 1 Summary of the Simulations in Model Validation

In this chapter, three scenarios have been selected to carry out the simulations:

- Lane formation in a bi-directional crowd flow
- Consensus decision making in small groups
- Consensus decision making in large groups

The first scenario has been presented in many studies (both empirical and simulate) while the latter two have only been observed in real-life experiments and have not been presented in crowd simulations yet.

8. 2 Simulation 4: Lane Formation in a Bi-directional Crowd Flow

8. 2. 1 Scenario Descriptions

This phenomenon describes that pedestrians tend to move in the same lane when walking in the same direction. It has been observed that spontaneous formation of unidirectional lanes will be formed in bi-directional pedestrian flows in a contained environment (e.g. the crowd coming from both sides of a corridor or a street). A demonstration of this phenomenon is showed in Figure 114.

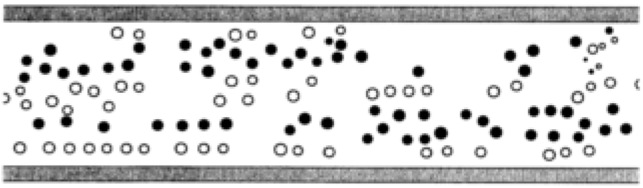


Figure 114 Spontaneous lane formation in a bi-directional crowd flow (Dirk Helbing & Peter Molnar 1995)

(For more detailed descriptions of this phenomenon, please refer to section 2. 6. 2 in the Literature Review chapter)

8. 2. 2 Simulation Configuration

8. 2. 2. 1 Simulation Process and Data Capture

The Start and the End of one simulation

The simulation starts with an empty corridor. After the simulation starts, the agents will enter the corridor from either side randomly and walk through the corridor. The agents will exit the corridor when reaching the other side of the corridor. The simulation does not have an end. The randomly generated agents keep entering the corridor at a steady rate.

Data Capture

The aim of this simulation was to observe the lane formation in the counter crowd flow. Because the agents are walking in two opposite directions, the counter flow of the crowd can be observed soon after the simulation starts. According to the existing studies mentioned above, spontaneous formation of unidirectional lanes will be formed in the bi-directional pedestrian flows.

Screen snapshots will be captured to demonstrate the crowd walking spontaneously in lanes.

8. 2. 2. 2 Environment Representation

The dimension of the corridor used in the simulation was 45 X 10 metres. It is an empty corridor with no obstacle present.

8. 2. 2. 3 Agent Configuration

Behaviours

✧ ***Summary***

Because the lane formation in the counter crowd flow is formed spontaneously, no explicit instructions should be given to the crowd to move in lane. The behaviours of the individuals in the crowd only consist of walking normally and keeping a distance from others and the walls. The behaviours that are represented by the corresponding behaviour rules in the Behaviour Library are listed below.

Table 34 Agents' behaviours while walking bi-directionally in a corridor

	Behaviours	**Simulation Configurations**
1	Walk towards the other side of the corridor	**Behaviour Library**: Seeking
2	Try to maintain a comfortable distance from others	**Behaviour Library**: Handling repulsive effect from nearby crowd
3	Movement is restrained by the structure of the corridor	**Behaviour Library**: Keeping a distance from a wall (repulsive effect)
4	Enter the corridor from either side	The agents will be generated at either

		side of the corridor
5	Exit the corridor when reaching the other side	The agents reaching the other side of the corridor will be removed from the simulation environment

✧ ***Detailed Analysis***

This section explains the descriptions of the behaviour in the simulation scenario individually and analyses its correspondent behaviour rule in the Behaviour Library.

✦ ***Behaviour 1: Walk towards the other side of the corridor***

After entering the corridor, an agent's aim is to walk to the other side of the corridor without any special behaviour. The "seeking" behaviour in the Behaviour Library can be used to achieve this aim. The location of the goal should be a horizontal offset to the agent's current position at the other side of the corridor, which means, without any other interruption, one agent should walk horizontally toward the other side (using the shortest route).

It is predictable that, during the movement, the agent may adjust its vertical position to maintain a comfortable distance from others. This requires that the vertical position of its virtual goal should be updated correspondingly. In other words, the virtual position of the goal should always be a horizontal offset to the right/left of the agent's current position in this simulation.

✦ ***Behaviour 2: Try to maintain a comfortable distance from others***

"The repulsive effect from crowd" is the behaviour in the Behaviour Library that represents the fact that agents will try to maintain a desired distance from others during the simulation. This behaviour rule is designed to represent such individual behaviour (see more details in 5. 2. 4 Behaviour Library: Handling Repulsive Effect from nearby Crowd).

✦ ***Behaviour 3: Movement is restrained by the structure of the corridor***

The crowd will be constrained inside the corridor while moving towards the other side. The behaviour "Keeping a distance from a wall (repulsive effect)" is applied to refine the movement of the agents. Due to the constraint of the geometry, no-one can get across the sidewalls (top and bottom) of the corridor.

In order to minimise the effect of the walls and make them serve as a geometrical boundary, the desired distance from a wall is set to 0.05 metre (1 pixel) for all agent in the simulation. This means that the agent will only receive the repulsive effect from the wall when it get very close to the wall and this effect serves the purpose of not allowing the agent to cross the wall.

✦ ***Behaviour 4: Enter the corridor from either side***

The agents will enter the corridor from either side in this scenario. In the simulation, the agents will be randomly generated as appearing at either side of the corridor (thus having an equal chance to appear at each side).

✦ ***Behaviour 5: Exiting the corridor when reaching the other side***

The agents will exit the corridor if they have reached the other side. In the simulation, the agents will be removed from the simulation environment once they have walked beyond the boundary of the corridor.

Attributes

Table 35 Agents' attributes in simulation – walking in bi-directional crowd flows.

Attribute	Value (In real world)	Value (In simulation)
Size	0.5 m	10 pixels
Default Speed	1.5 m/s	0.5 pixel/frame
Maximum Speed	2 m/s	0.67 Pixel/frame
Desired distance from wall	0.3m	6 pixel
Minimum distance from wall	0.05 m	1 pixel
Desired distance from others	0.7 m	14 pixel

Minimum distance from others	0.05m	1 pixel
Repulsive modifier (to self)	normal	1
Repulsive modifier (to other)	normal	1
Repulsive modifier (to wall)	normal	1

Starting Positions

There are no agents distributed in the corridor prior to the start of the simulation. The agents will be randomly generated at both ends of the corridor.

8. 2. 3 Result and Analysis

Figure 115 shows the snapshot of the simulation (about 400 agents in the corridor after reaches the stable state) using the above configuration. The top part of the figure shows the stage when the crowd from two direction are about to encounter. The bottom part of the figure show lane formation has been formed and such a formation has reached a stable state.

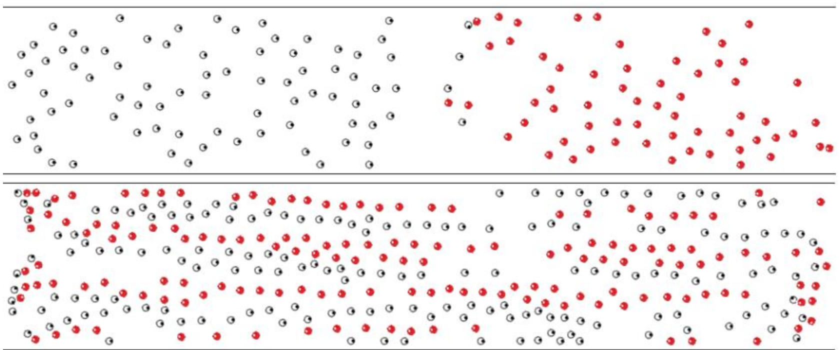


Figure 115 Spontaneous lane formation observed by using the study's crowd model

The success in representing the spontaneous lane formation without explicit behaviour configuration in the proposed model is considered as a preliminary validation of the model in this PhD study. More in depth validation and analysis on the proposed crowd model will be presented in the following two sections which simulate the crowd phenomenon in leadership and consensus decision making.

8. 3 Simulation 5: Leadership and Consensus Decision Making (in Small Groups)

8. 3. 1 Background

Leadership and consensus decision making have been found in many animal groups, such as honey bees (Seeley & Buhrman 1999; Chittka et al. 2003; Seeley & Visscher 2004), fishes (Bumann et al. 1997; S. Reebs 2000; S. G. Reebs 2001), and monkeys (Leca 2003). It refers to the phenomena that a group with different aims can eventually reach a consensus decision which allows them to remain together and reaches a destination of the choice of those with strong leaderships.

Dyer et al. have performed a series of experiments (J. R. G. Dyer et al. 2008; J. R. G. Dyer et al. 2009) on consensus decision making on human groups. Their studies showed similar findings to animal groups, i.e. the minority can lead the group effectively and the importance of the positions of the informed individuals in small size human groups. But the findings on large size groups were described as anecdotal due to "*the logistical difficulties*" (J. R. G. Dyer et al. 2009), i.e. insufficient experiment samples.

The aim of this simulation is to evaluate the reliability of a crowd simulation model in this study by reproducing Dyer et al.'s (2009) experiments. Additionally, the simulation also aims to reveal further findings via numerous simulation data (much more samples than the original experiments) and demonstrate the applications of the study's crowd model.

8. 3. 2 The Original Experiment

(The experiment described below was reported in Dyer et al.'s paper (J. R. G. Dyer et al. 2009)*. In order to keep brevity, the descriptions and quotations relating to Dyer et al.'s experiment may not explicitly refer to their paper in this section. This experiment was marked as experiment 2 in their study.)*

8. 3. 2. 1 Scenario Description

The original experiment aimed to investigate leadership, consensus decision making and collective behaviour in humans (in small groups). It was took place between January 2006 and March 2007 at the University of Leeds (England) and the University of Wales at Bangor and was implemented by Dyer et al. (J. R. G. Dyer et al. 2009).

The process of the experiment can be summarized as a group of ten people starting walking from the centre of a circle trying to reach the destinations which were determined by their roles in the experiment. They were all required to walk in one group while trying to reaching their destinations (i.e. the participants had to maintain in one group). There were two roles given to the participants:

- **Informed individuals**: Two of the participants in the group were told to seek a target which was located on the periphery of the circle during the experiment.
- **Uninformed individuals**: The rest of the eight participants were not given any target and were told to walk randomly.

The experiment ended when the group reached any position on the periphery of the circle. The time and accuracy in reaching the target on the periphery of the circle were recorded as the results.

8. 3. 2. 2 Experiment Venue

The experiment venue (showed in Figure 116) was designed as follows: *"a circular arena with a 10m diameter was marked on the floor and cards labelled 1–16 were spaced equally around its perimeter. A circle with a diameter of 2m was marked out in the centre of the first circle with the letters A–H spaced equally around its perimeter"* (I and J are in the centre of the inner circle).

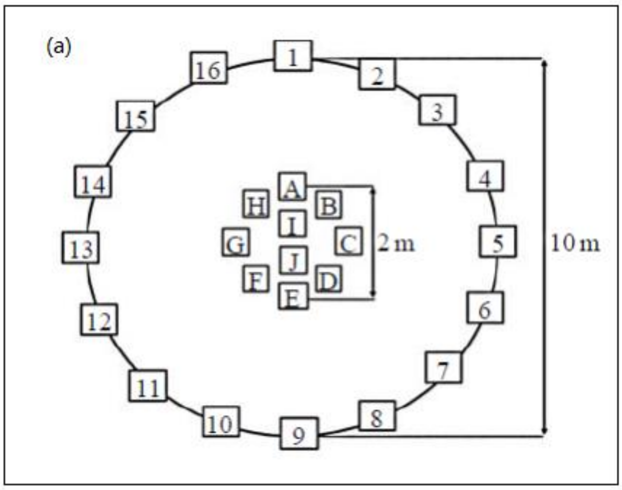


Figure 116 Layout of the experiment venue: small group of 10 (J. R. G. Dyer et al. 2009)

8. 3. 2. 3 Participants in the Experiment

"Participants were undergraduate students. In total, 15 mixed-sex groups of ten individuals were used for testing. All experiments were carried out double-blind in that both the participants and the individuals who measured the response variables were not aware of the purpose of the experiment."

8. 3. 2. 4 Detailed Description of the Experiment

Instructions of the Experiment

The experiment design and process have been introduced briefly above. The following list quotes the instructions given in the experiment for the record.

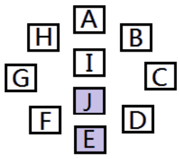
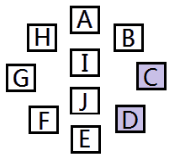
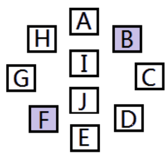
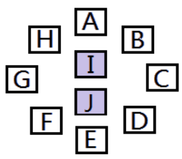
- The starting positions of participants were described as *"Individuals were asked to stand on a letter (A–J)"*.
- Their initial orientations were decided by "*randomly facing a number from the outer circle"* in order *"to avoid any bias due to initial direction of locomotion"*.
- A common instruction was given to all the participants: *"when we tell you to begin you should start walking at a normal speed and do not stop before being told to do so. You can walk anywhere inside or outside the circle but you have to stay within an arm's length of another individual and you should not talk or gesture to each other."*

- In addition, to create informed and uninformed individuals, the following instructions were given respectively: to the informed individuals, *"Go to number X, without leaving the group"*; to the uninformed individuals, *"stay with the group"*.
- The experiment ends when the group reaches the periphery (outer circle). The arrival time and the number arriving there were recorded for further analysis.

Illustrations of the Positions of the Informed Individuals

The experiment was carried out with four treatments. Each treatment had different starting positions for the informed individuals (demonstrated in Table 36).

Table 36 Positions of informed individuals in the experiment on leadership and consensus decision making in small groups.

No.	Treatment Name	Positions of Informed Individuals (Highlighted)
1	**Mixed Treatment** **(J, E)**	
2	**Close Treatment** **(C, D)**	
3	**Far Treatment** **(B, F)**	
4	**2 Core Treatment** **(I, J)**	

8. 3. 2. 5 Results and Findings of the Original Experiment

Results

The results of the experiment were analysed with generalised liner mixed models (GLMMs) to identify whether significant differences existed between the treatments.

- **Arrival Time**

The "arrival time" represented the time that the group used to reach the periphery of the circle. It was reported that there were significant differences between the treatments (based on the experiments using 15 groups). In detail:

- "Mixed treatment (J, E)" used less time than "far treatment (B, F)" and "2 core treatment (I, J)", which had significant differences.
- No other significant differences were found between the treatments.

- **Arrival Accuracy**

The "arrival accuracy" represented the successful rate of the group reaching the target which was given to the "informed individuals". It was reported that "mixed treatment" had significant differences from all the other treatments. The "mixed treatment" also had the best accuracy among the treatments. Apart from this, no other significant differences were found between the groups.

Original Discussions on the Results

Dyer et al. considered the reason why the "mixed treatment" could provide the best result was because the mixed types of the leader in the group. On the one hand, one informed individual on the periphery was more mobile thus could quickly align with the target. On the other hand, the other informed individual in the centre of the group could influence most of the uninformed individuals through his/her movement towards the target.

8. 3. 3 Reproducing the Experiment through Simulation

8. 3. 3. 1 Simulation Description

The simulation was designed to reproduce Dyer et al.'s experiment (introduced above) in a virtual environment. The simulation environment, the agents and their behaviours were based on the descriptions of the venue, participants and the instructions in the experiment.

In the simulation, each treatment was repeated 1600 times. For each treatment, each target number was tested 100 times.

8. 3. 3. 2 Purposes

The simulation serves two purposes:

- The first one is to validate the crowd model in this PhD study by comparing the simulation results to the experimental results.
- The second one is, through the numerous simulation data, to further analyse the existing findings and explore more findings.

8. 3. 4 Simulation Configuration

8. 3. 4. 1 Simulation Process and Data Capture

The Start and the End of one Simulation

The simulation started with 10 agents located in the inner circle (showed in Figure 116 letter A-J). The simulation ends when the group of the agents reach the periphery of the outer circle.

Data Capture

The simulation ran 1600 times for each treatment. For each simulation the following information was captured:

- The starting positions of the informed individuals (i.e. which treatment)
- The target number

- The number of individuals arrived at the destination
- The time used to reach the periphery

8. 3. 4. 2 Environment Representation

The simulation environment is identical to the experiment venue. There is no geometrical constraint or obstacle in the environment. The target numbers were displayed on the periphery of the outer circle for visual indication. They were modelled as virtual targets in the simulation environment.

8. 3. 4. 3 Agent Configuration

Behaviours

- **Summary**

From the experiment instructions, the following behaviour rules and modelling configurations for the simulation were identified:

Table 37 Behaviour configuration for the simulation – leadership and consensus decision making in small groups

Experiment Instructions	Behaviour in Behaviour Library
You can go anywhere	Wandering
Go to number X	Seeking
Without leaving (Stay within) the group	Keeping in a group
Stay within an arm's length of another individual	Keeping a distance from nearby crowd
Do not talk or gesture to each other	N/A (No information exchange)
Avoid collision (no explicit description)	Handling repulsive effect from nearby crowd

➢ Details

The following section provides detailed descriptions as to how the instructions have been interpreted in the simulation.

✧ ***Instruction 1: Can go anywhere -> Wandering***

The instruction "you can go anywhere" indicates that the participants can move freely during the experiment. The "Wandering" behaviour from the Behaviour Library can be used to represent this instruction. In the study's simulations, Δt is 0.167s as the simulation programme has a update rate of 60 frame/second and ɵ is set to 0.5. In each time frame, the probability to change an angle is set to 5%. This would create a smooth wandering trajectory.

✧ ***Instruction 2: Go to number X -> Seeking target***

The instruction "go to number X" informed the participants to move to a target position. This can be directly linked to the "Seeking" behaviour in the model. Under this behaviour, an individual walks directly towards the position of number X.

✧ ***Instruction 3: Without leaving (Stay within) the group -> Keeping in a group***

The instruction "stay within the group" has been given to all the participants. As there is no detailed descriptions about how to keep in the group, it is considered that this behaviour should have two effects according to the literature: (a) a cohesion effect that make one individual move to the average position of nearby individuals (Reynolds 1987); (b) an alignment effect that adjusts an individual's walking direction towards the average heading of neighbours (Reynolds 1987; Couzin et al. 2005). Thus the "keeping in a group" behaviour defined in this crowd model is used.

✧ ***Instruction 4: Stay within an arm's length of another individual -> Keep certain distance with others***

The instruction "stay within an arm's length of another individual" does not produce any behaviour but serves as a threshold that triggers the behaviour "keeping in a group". Once a participant finds himself/herself is out of range with others, he/she will perform the "keep in group" behaviour to return to the group. Otherwise, he/she will carry on his/her default behaviour as either informed or uninformed individuals.

Although a standard arm's length was demonstrated to the participants in Dyer et al.'s experiments, they did not specify what length was demonstrated. In addition, no data could be found on what was used as the standard arm's length in the literature. Considering that the participants in the experiments were mainly undergraduate, 0.7m was used as the standard arm's length in the simulation.

✧ ***Instruction 5: Do not talk or gesture to each other -> No information exchange***

In order to minimize the effect of active information transformation, communications between participants were not allowed in the original experiments and the participants are told only to move under the instructions given to them. Such rules can be achieved in the simulation without additional settings as the agents only do what they are told to do. Therefore, no special configuration is required to interpret this instruction.

✧ ***Instruction 6: Avoid collision ->Repulsive effect from others***

The repulsive effect helps individuals adjust their positions while walking and avoid collisions. Although this behaviour cannot be found in the experiment instructions explicitly, it can be treated as the subconscious behaviour of the participants and this is considered as a standard behaviour in the study's crowd model. In addition, Couzin et al. (2005) used a similar mechanism to maintain the distance between individuals in modelling the consensus decision making behaviour in an animal group, which is comparable to this study's simulations concerning humans. Adding this behaviour to the simulation is believed to be reasonable.

Attributes

➢ Walking speed

In the experiment, the participants were instructed to walk at normal speed. However, "*Normal walking speed was not defined but was demonstrated to the participants*" (J. R. G. Dyer et al. 2008) in the experiments. In the simulation 0.4 ± 10% m/s was used as the default walking speed. By taking into account the crowd density in this scenario and the crowd walk in the normal condition, this value of crowd walk speed is supported by Sakuma et al.'s study (2005). Other speeds have been tested in order to find out the effect of changing the walking speed; in those cases, a ±10% variance was applied as well.

Because the participants were required to walk at normal speed, the agents' maximum speed was set to the same value as its default speed.

➢ Summary

Based on the above analysis, the agents' attributes were set as follows:

Table 38 Agents' attributes for Simulation – Leadership and consensus decision making in small groups

Attribute	Value (In real world)	Value (In simulation)
Size	0.5 m	10 pixels
Default Speed	0.4 m/s	0.13 pixels / frame
Maximum Speed	0.4 m/s	0.13 pixels / frame
Sight Range	5 m	100 pixels
Group behaviour range	5 m	100 pixels
Desired distance from others	0.7 m	14 pixel
Minimum distance from others	0.05m	1 pixel
Repulsive modifier (to self)	normal	1
Repulsive modifier (to other)	normal	1

Starting Positions

The agents are located at the positions of the letters A – J. The positions of the informed individuals for each treatment have been illustrated in Table 36.

➢ **Initial Orientation**

In the experiments, the orientation of each individual was randomly chosen as facing a number from 1 to 16 (representing the targets at the outer circle) in order to avoid any bias in the starting direction of the locomotion.

In the simulation, this instruction was improved to assign each agent a randomly starting direction from an angle of 0^o to 360^o.

8. 3. 5 Simulation Results and Analysis

(The simulations' results were processed by Microsoft Excel 2007 Analysis ToolPak to generate the graphical report. The statistical analysis was processed by SPSS (v20)).

8. 3. 5. 1 Overall Results and Comparison

The overall results compare the arrival accuracy and the arrival time between the four treatments.

Arrival Accuracy

➢ **Simulation Result**

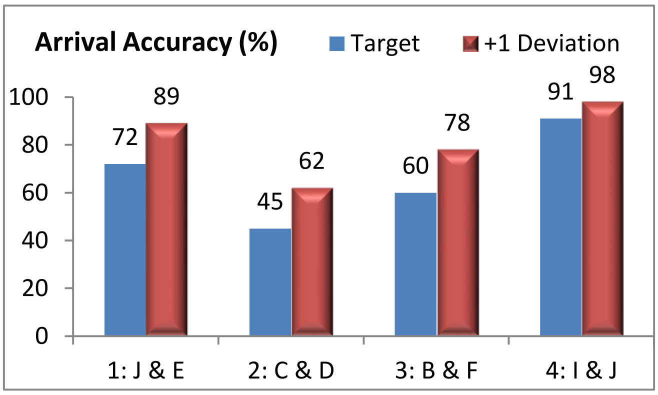


Figure 117: Accuracy of the agents reaching the target number

The order of arrival accuracy from high to low is: treatment 4 > 1 > 3 > 2. If arrived at the adjacent numbers of the target number (e.g. from target number 2, number 1 and 3 were the adjacent numbers) was also counted, the +1 deviation arrival accuracy remains in the same order. By considering the starting positions of the informed people in each treatment, it can be found that when the core positions (I&J)

are occupied by the informed individuals, the group has the highest accuracy in reaching the target. This conclusion is in line with the findings by Dyer et al and is also supported by Leca's finding (2003): informed individuals in the core position can influence the uninformed individuals and are more likely to lead the group movement.

One-way ANOVA test on the arrival accuracy shows that significant differences exist (F(3, 6396) = 310.95, p < .000). The post-hoc tests (Tukey HSD) show that the arrival accuracies of the treatments are significantly different to each other (the detailed statistical report can be found in Appendix.1).

- **Dyer et al.'s Results**

"Groups with informed individuals in core and peripheral positions deviated from their targets significantly less than groups in all other treatments. There were no other significant differences between the treatments."

The order of the arrival accuracy of the four treatments could not be drawn from the original experiment results as not all the treatments had significant difference. A direct comparison is not possible.

Arrival Time

- **Simulation Result**

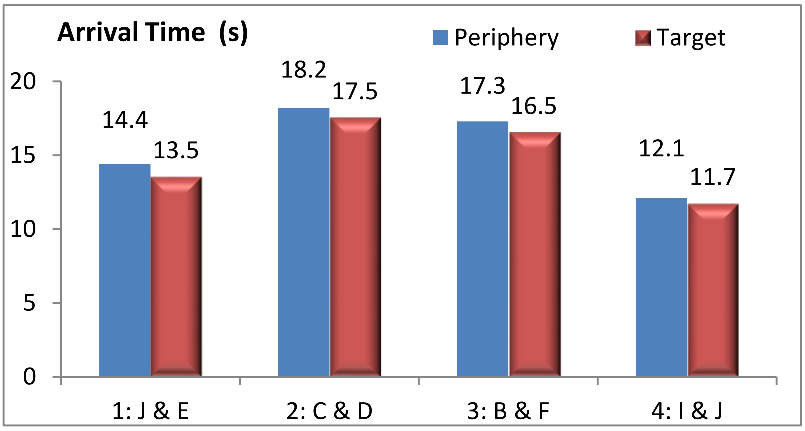


Figure 118: Arrival time. Periphery means any number on the outer circle including the target.

The order of the arrival time (from short to long) is: treatment 4 > 1 > 3 > 2 which has the same order as the arrival accuracy. It reveals that higher accuracy can help the group reach the target quicker. When the group is more likely to make a consensus decision on the same direction (higher accuracy), they spend less time on moving in random directions, thus can reach the target quicker. Another finding concerning arrival time is that when the group arrived at the target successfully, it took less time than arriving at the periphery. This is considered consistent with the finding at the beginning of this paragraph as, if the group are not heading to the target number, the informed individual will try to resist moving in the wrong direction which results in a longer time to reach the periphery.

In addition, by considering arrival accuracy and time together, it is found that higher arrival accuracy can result in a shorter arrival time.

One-way ANOVA test on the arrival accuracy shows that significant differences exist ($F(3, 6396) = 262.57$, $p < .000$). The post-hoc tests (Tukey HSD) show the arrival times of the treatments are significantly different to each other (the detailed statistical report can be found in Appendix.1).

- **Dyer et al.'s Results**

"Groups with one informed individual starting in the core and one on the group periphery reached the perimeter in significantly less time than groups with two core leaders and groups with two leaders on opposite sides of the edge. There were no other significant differences between treatments in time to periphery."

Again, the order of the arrival time of the four treatments cannot be drawn from the original experiment results as not all the treatments were significantly different. A direct comparison is not possible.

Comparison to Dyer et al.'s Findings

Dyer et al.'s experiment indicated that treatment 1 (J&E, median time about 14s) spent less time to arrive at the periphery than treatment 3 (B&F, median time about 24s) and treatment 4 (I&J, median time about 21s) but could not find statistically a difference between the other treatments based on its experimental samples (15 groups). The original experiment found that treatment 1 (J&E) has much less deviation on arriving at the target than all the other three treatments, However, the small sample size prevented Dyer et al. from analysing the data further.

This study's simulation results indicate that when the informed individuals are located at the core positions, the group has better accuracy and a smaller arrival time. It suggests that informed individuals can influence the group more when they start at core positions than at peripheral positions. This finding was reported in Dyer et al.'s experiment's findings and are also supported by Leca's (2003) research results.

There is one difference between the study's simulation results and Dyer et al.'s experiment results. In the study's simulations, treatment 4 (I&J) has better accuracy and arrival time than treatment 1 (J&E). The reason of this difference could be the issues - *"the informed individual in the core position was constrained in terms of mobility and needed some time to find the target while the peripheral positions are easier to move and align with the target"* (J. R. G. Dyer et al. 2009) was not considered in this simulation. Because the informed individual was designed to know the target position from the beginning and no specific constraint rule was applied to the core position, the constraints of the core position on target seeking and movement have not been explicitly modelled in the simulation.

8. 3. 5. 2 Further Analysis

As the simulation repeated each treatment 1600 times (6400 in total), it provided a larger sample size than Dyer et al.'s 15 groups. With these numerous data, it is possible to generate distribution histograms and have an in-depth analysis of the relationships between starting positions and different target numbers.

Distribution of Arrival Time

The histograms (Figure 119) show that the arrival times to the periphery have a Gaussian distribution *(For the reason of better visibility, this histogram only shows the arrival times that are less than 60 seconds. Within 60 seconds, it contains 99.75%, 99.31%, 99.50%, 99.88% data for each treatment respectively).*

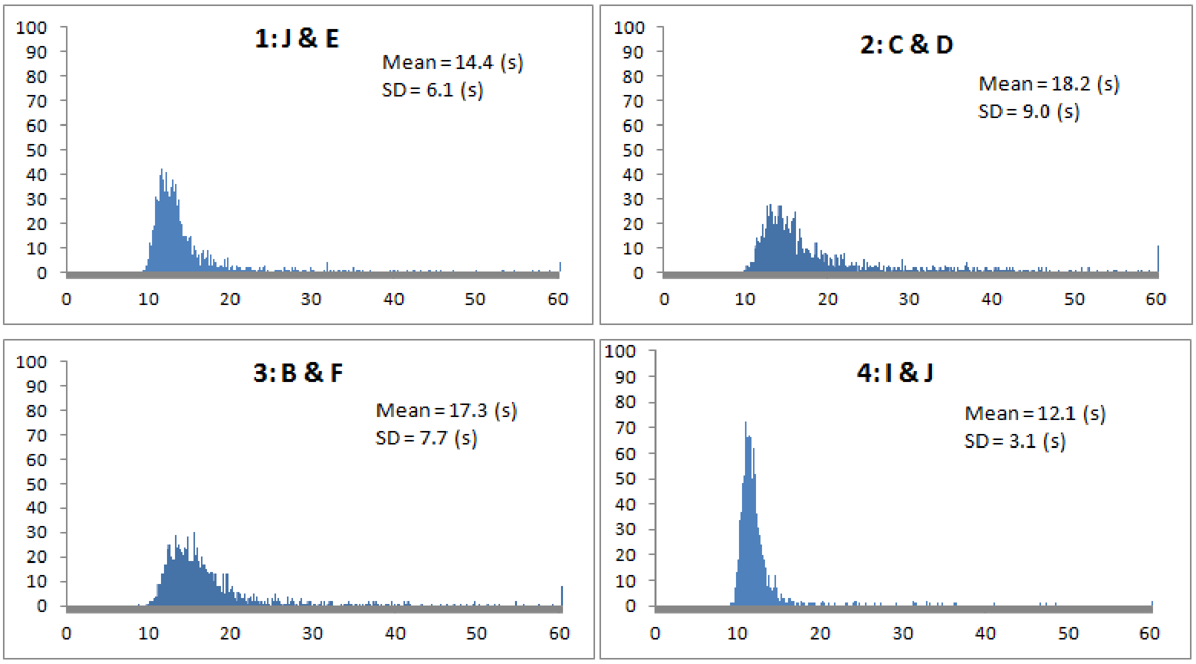


Figure 119 Histogram of arrival time (in seconds) (Y-axis represents the frequency)

It can be seen that treatment 4 (I&J) has the smallest SD (standard deviation) which means, in this treatment, the arrival times are more likely to be at the mean time: 12.1 seconds. Treatment 2 (C&D) and treatment 3 (B&F) have a quite large SD, which indicates the arrival times in these two treatments are distributed over quite a large range. As these distributions are the overall results of all the sixteen target numbers, one can infer that different positions of informed individuals in each treatment do affect the time to arrive at the periphery (in the case of different target numbers).

The reason could be, for different starting positions, their distances to the target numbers are not equal. For example, in treatment 2 (starting positions at C&D), the informed agents' distance to target number 6 is about 4 meters while their distance to target number 14 is about 6 meters. This explains why treatment 4 (I&J) has the narrowest distribution and treatment 2 (C&D) has the widest. For positions I&J, all the target numbers have the same distance. Positions C&D have the most significant distance variance to the target numbers among the four treatments.

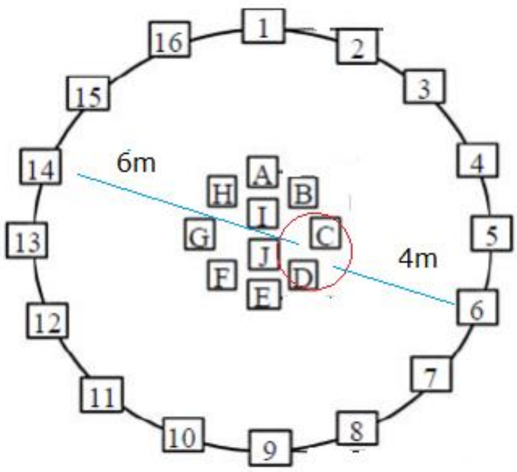


Figure 120 Difference in distance from informed agents to target numbers for some starting positions

Detailed Arrival Accuracy of Each Treatment

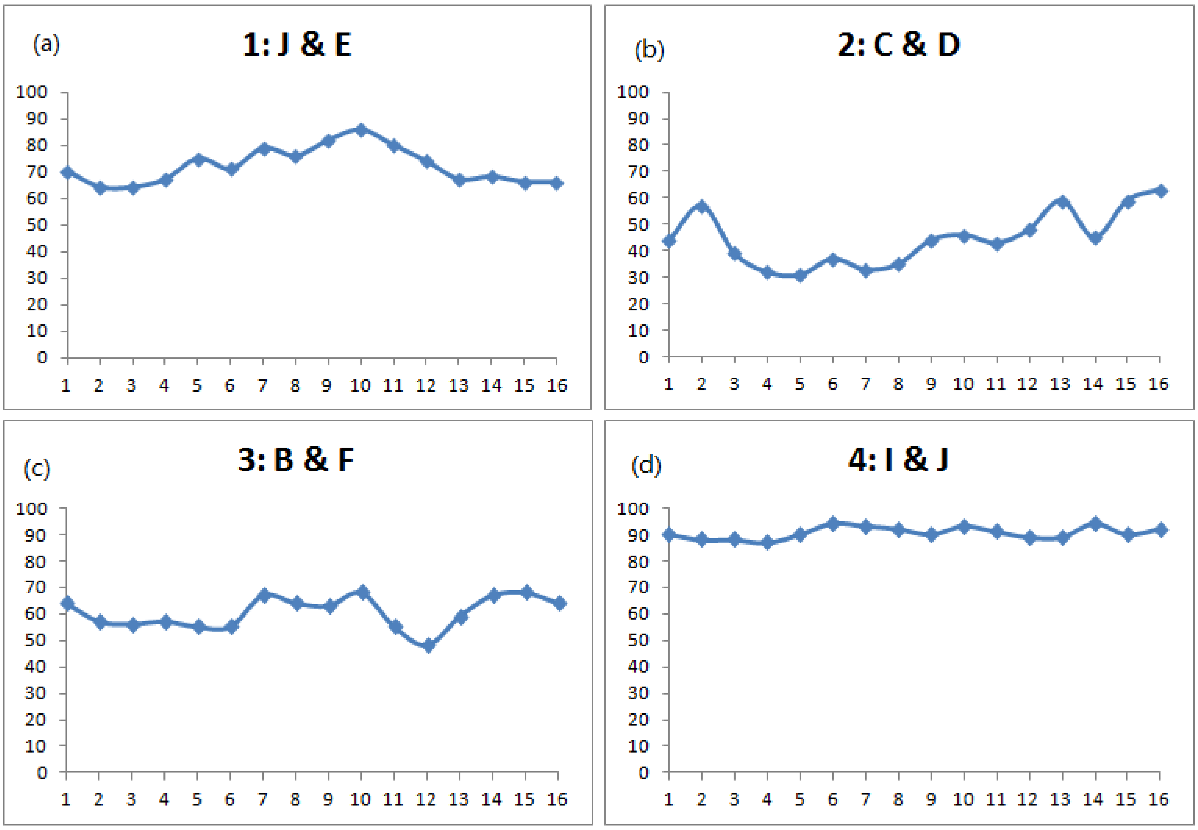


Figure 121 Accuracy (Y-axis) of reaching the target number (X-axis)

Figure 121 shows the group accuracy in reaching the target number is also affected by the positions of the informed individuals. It is no surprise to see there is not much difference between the arrival accuracies at the target numbers in treatment 4 (I&J) because the two informed individuals started at the centre positions. Treatment 2 (C&D) has the lowest accuracy because the two informed individuals were located on same side of the group next to each other which lowers their influence (Leca 2003; J. R. G. Dyer et al. 2009) on the whole group.

In addition, it can be found that if there is an informed individual in the core position, the other informed individual at the periphery can increase the group arrival accuracy to the target numbers that are closest to him/her (Figure 121(a)). If there is no informed individual in the centre, the informed individual actually lowers the chance for the group to reach the target numbers which are closest to him/her (Figure 121(b)&(c)).

Effect of Peripheral Informed People on Arrival Time and Accuracy

Continuing the analysis on the cases (Figure 122 and Figure 123) with only peripheral informed individuals, it can be found that when the informed individuals are located on the periphery of the group, they find it more difficult to guide the group to the target number that is closest to them. Figure 122(a) shows that, in treatment 2, the group took more time to reach the periphery for the target numbers 5, 6, 7, 8 which are actually closer to the starting positions (C & D) of the informed persons. A similar situation has also been found in treatment 3 (Figure 122(b)). The times to arrive at the target numbers 2, 3, 10, 11 are slightly longer than the others. Figure 123 indicates that the peripheral informed individuals also have a negative effect on arrival accuracy. The group arrived at the target numbers that are closer to the informed positions with a lower accuracy.

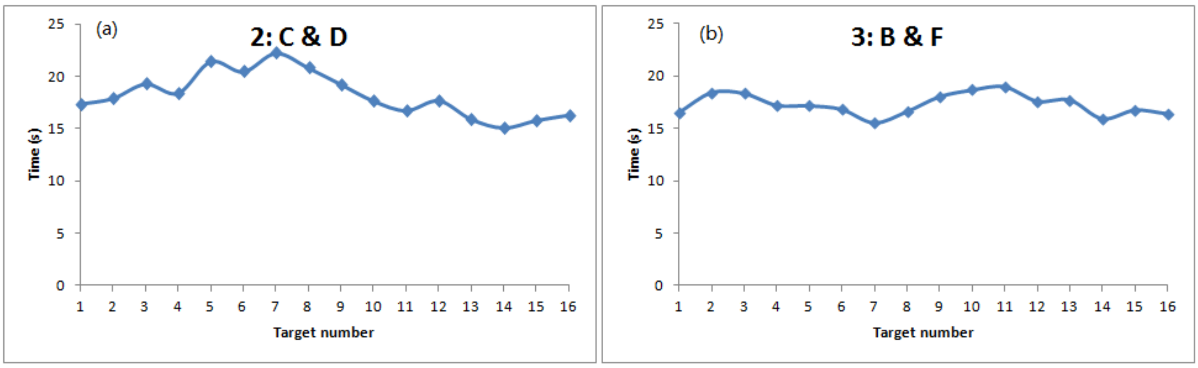


Figure 122: Arrival time to periphery for each target number

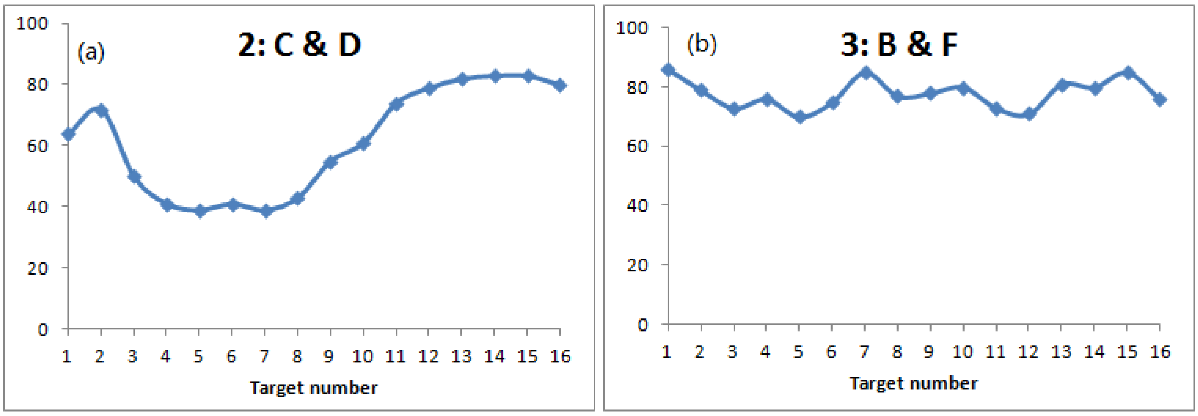


Figure 123 Arrival accuracy for each target number (+1 deviation)

Figure 123 provides further evidence that higher accuracy can result in a shorter arrival time. The shapes of the curves (for a prominent contrast, +1 deviation accuracy is used) are contrast with the arrival time in Figure 122. This finding is consistent with the conclusion drawn in the previous section about the overall result.

Peripheral informed positions show a negative effect on arrival time and accuracy. It is also noticed that this only happened when there is no core informed position present. In treatment 1 (J&E), with one informed individual in the core position J, the peripheral informed position E showed a positive effect. The arrival accuracy does increase around target number 10 (Figure 122(a)). The study's simulation results indicate the initial core informed individual is crucial to the group behaviour.

8. 3. 5. 3 Experiments to Explore the Relationship between Speed and Accuracy

Dyer et al. (2008) claimed that they did not find any relationship between arrival time (showing the speed of making a consensus decision) and accuracy; such a situation was found in ant colonies (N.R. Franks et al. 2003) where a trade-off between accuracy and speed has been observed. The reason for this was considered as possibly due to the small sample size.

However, it was found in this simulation that the treatment with the higher accuracy actually resulted in a faster arrival which appears to be different from the two scenarios mentioned above. But one can notice that, in Dyer et al.'s experiment and in this simulation, the consensus decision was happening at the same walking speed (as the default speed was demonstrated to the participants). So this is not similar to the case which Franks et al. (2003) mentioned: the group sacrificing accuracy to reach a quicker consensus decision. The corresponding question to ask here should be: If the quicker consensus decision is achieved via the individuals moving faster, can such a trade-off between default walking speed and arrival accuracy be observed?

To investigate the relationship between speed and accuracy, treatment 1 (J&E) was tested at various speeds (0.3, 0.4, 0.55, 0.8, 1.0, and 1.25 m/s). Each speed is simulated 1600 times. The results (Figure 124 & Figure 125) show that the arrival time decreases when the walking speed increases. Such relationship is not linear. The arrival time changed more dramatically at lower speeds (below 0.5 m/s) than at higher speeds (above 1 m/s).

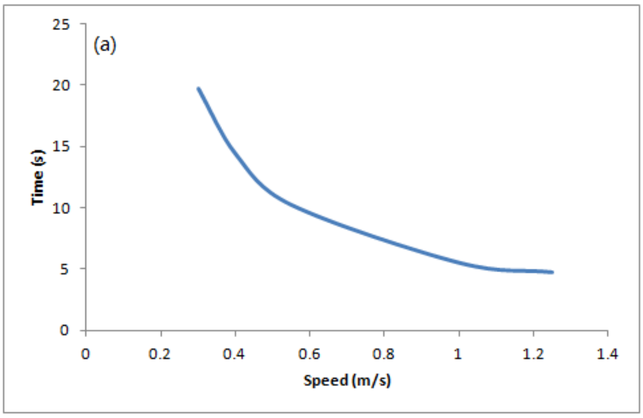


Figure 124 Simulation on leadership and consensus decision making: arrival time at various speeds

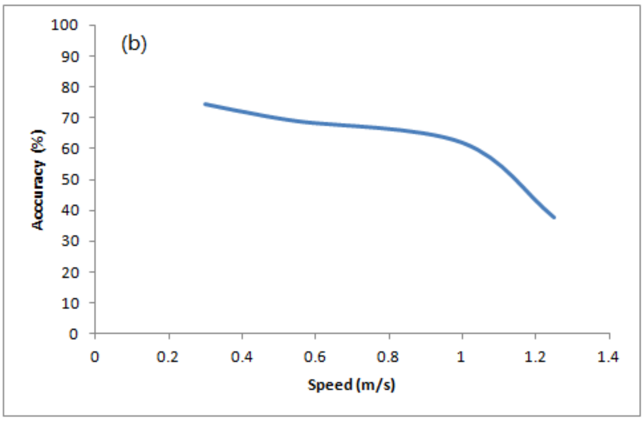


Figure 125 Simulation on leadership and consensus decision making: arrival accuracy at various speeds

These results indicate that, when the group move at a higher speed and has less time to reach a consensus decision, the time to reach the periphery and the arrival accuracy have both been decreased at different rates. An accuracy trade-off with higher walking speeds was observed.

8. 4 Simulation 6: Leadership and Consensus Decision Making2 (in Large Groups)

8. 4. 1 The Original Experiment

(This experiment was reported in the Dyer et al.'s paper (J. R. G. Dyer et al. 2009)*. In order to keep brevity, the descriptions and quotations relating to Dyer et al.'s experiment may not explicitly refer to their paper in this section. This experiment was marked as experiment 3 in their study.)*

8. 4. 1. 1 Scenario Description

The original experiment aimed to investigate leadership, consensus decision making and collective behaviour in humans (in large groups). It was similar to the experiment introduced in the above section but on a larger scale. A group with 200 people was tested to reach the target on the periphery on a fifty metres (in diameter) circle. The experiments took place on 4 March 2007 in Cologne (Germany) and 5 May 2007 in Freiburg (Germany).

The process of the experiment could be summarized as a group of ten people starting walking from the centre of a circle to try to reach the destinations which were determined by their roles in the experiment. They were all required to walk in a group while trying to reaching their destinations (However, in contrast to the experiment in small groups introduced above in section 8. 3 , the participants were allowed to break into small sub-groups as they found it was inevitable in the groups).

There were two roles given to the participants:

- **Informed individuals**: Two of the participants in the group were told to seek to a target which was located on the periphery of the circle during the experiment.

- **Uninformed individuals**: The rest of the eight participants were not given any target and were told to walk randomly.

The experiment ended when the group reached any position on the periphery of the circle. The time and accuracy of reaching the target on the periphery of the circle were recorded as the results.

8. 4. 1. 2 Experiment Venue

The outer circle (diameter = 50m) was the area in which all the participants were told to stay. The participants started from the inner circle (diameter = 12m).

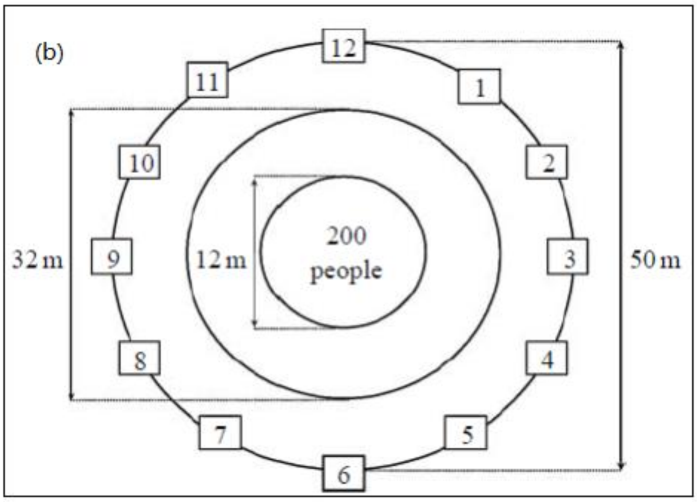


Figure 126 Experiment arena – leadership and consensus decision making (large group)

8. 4. 1. 3 Participants of the Experiment

"Participants were volunteers between the age of 18 and 70 of both sexes who had answered TV or radio advertisements asking for participants for a swarm experiment (no further information on the nature of the experiment was given until the experiment was finished)."

8. 4. 1. 4 Detailed Description of the Experiment

The experiment design and process have been introduced briefly above. The following list quotes the instructions given in the experiment for the record.

- The experiment had 200 participants and they were asked to stand freely in the inner circle (Figure 126) before the start.

- The participants were told to move as per the following description: *"when we tell you to begin you should start walking at a normal speed and do not stop before being told to do so. You can walk anywhere inside or outside the circle but you have to stay within an arm's length of another individual and you should not talk or gesture to each other."*
- In addition, "Slips of paper gave one of two different behavioural rules, one for uninformed individuals and one for informed individuals. Behavioural rule 1 gave instructions to simply 'stay with the group', resulting in uninformed individuals. Behavioural rule 2 gave instructions to 'Go to number 9, without leaving the group' creating informed individuals."
- The informed individuals were randomly chosen so they should be randomly distributed in the group.
- The experiment ended when the group reached the periphery (outer circle). The arrival time and the number of those who arrived were recorded for further analysis.

The percentages of informed individuals were tested at 2.5%, 5%, and 10% for once each.

8. 4. 1. 5 Results and Findings of the Original Experiment

The experiment results showed that, with a minority of informed individuals, the whole uninformed crowd group can be guided to the informed position. When 10% of the group were informed, the whole crowd can successfully reach the target position without being split into sub-groups. When 5% of the group were informed, the crowd split into several groups. One sub-group could reach the target with 90% of the population. The experiment with 2.5% informed individuals only resulted in a sub-group of 5% of the population.

These finds were described as anecdotal evidence as only one group was tested for each informed percentage.

8. 4. 2 Reproducing the Experiment through Simulation

8. 4. 2. 1 Simulation Description

The simulation was designed to reproduce Dyer et al.'s experiment on large groups (introduced above) in a virtual environment. The simulation environment, agents and their behaviours were based on the descriptions of the venue, the participants and the instructions in the experiment.

The simulation tested various informed individuals' compositions (i.e. different informed percentages). For each case, the simulation was run 100 times.

8. 4. 2. 2 Purpose

The simulation serves three purposes:

- The first one is to validate the crowd model in this PhD study by comparing the simulation results to the experiment results.
- The second one is to investigate whether the findings described as anecdotal in the original experiment persist in the large number of simulations.
- The third one is, through the numerous simulation data, to further analyse the results and explore more findings that could not be drawn from the experiment results.

8. 4. 3 Simulation Configuration

8. 4. 3. 1 Simulation Process and Data Capture

The Start and the End of one Simulation

The simulation started with a group of 200 agents located in the inner circle (Figure 126). The simulation ended when a group of the agents reached the periphery of the outer circle (it did not matter whether they reached the target or not). In the case of the group splitting up, the simulation ended when the first sub-group reached the periphery of the outer circle.

Data Capture

For each round of the simulation, the following information was captured:

- Percentage of informed individuals
- Time used to reach the periphery
- Whether the target has been reached
- The amount of the individuals in the first group that reached the periphery

8. 4. 3. 2 Environment Representation

The simulation environment is identical to the experiment venue. There is no geometry constraint or obstacle in the environment. A target indicator will be displayed on the periphery of the outer circle (i.e. at the position of number 9) for visual indication. All the circles were not visually displayed.

8. 4. 3. 3 Agent Configuration

Behaviours

- **Summary**

From the experiment instructions, the following behaviour rules and modelling configurations for the simulation have been identified:

Table 39 Behaviour configuration – leadership and consensus decision making in large groups

Experiment Instructions	Behaviour in Behaviour Library
You can go anywhere	Wandering
Go to number 9	Seeking
Without leaving (Stay with) the group	Keeping in a group
Stay within an arm's length of another individual	Keeping a distance from nearby crowd

Do not talk or gesture to each other	N/A (No information exchange)
Avoid collision (no explicit description)	Handling repulsive effect from nearby crowd

➢ **Details**

Because the instructions are similar to the experiment on small groups, the analysis on the instructions and their corresponding behaviours can be referred to in the discussions on agent behaviours' configuration in of section of "*Simulation 5: Leadership and Consensus Decision Making (in Small Groups)*".

Attributes

The agents' attributes' settings are the same as those in the section of "*Simulation 5: Leadership and Consensus Decision Making (in Small Groups)*". They are listed below for quick reference purpose.

Table 40 Agents' attributes in simulation – leadership and consensus decision making in large groups

Attribute	Value (In real world)	Value (In simulation)
Size	0.5 m	10 pixels
Default Speed	0.4 m/s	0.13 pixels / frame
Maximum Speed	0.4 m/s	0.13 pixels / frame
Sight Range	5 m	100 pixels
Group behaviour range	5 m	100 pixels
Desired distance from others	0.7 m	14 pixel
Minimum distance from others	0.05m	1 pixel
Repulsive modifier (to self)	normal	1

Repulsive modifier (to other)	normal	1

Starting Position

The agents were distributed in a matrix formation (Figure 127) occupying the approximate area of the inner circle in the experiment. The informed individuals were randomly selected from the group and their locations varied in every round of the simulation.

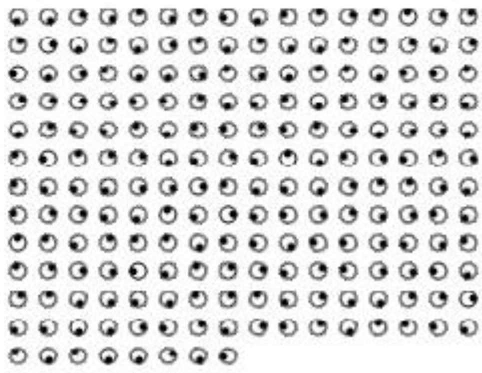


Figure 127 Initial formation of the agents in the simulation of the large group

In the simulation, each agent's initial orientation was randomly chosen from an angle of 0^{o} to 360^{o}. It was considered that this implementation can avoid the bias caused by the initial direction of locomotion (although no such instruction was mentioned in Dyer et al.'s experiment on a large group, a similar instruction was given in the experiment on the small group to create randomised initial orientations).

8. 4. 4 Simulation Results and Analysis

8. 4. 4. 1 Overall Result

Arrival Accuracy

- **Simulation Result**

The simulation was tested on four proportions of informed individuals: 2.5%, 5%, 10%, and 15%. Each set of configurations was repeated for 100 times. It can be seen

(Figure 128) that the group can achieve a reasonable arrival accuracy when the proportion of informed individuals was more than 10%.

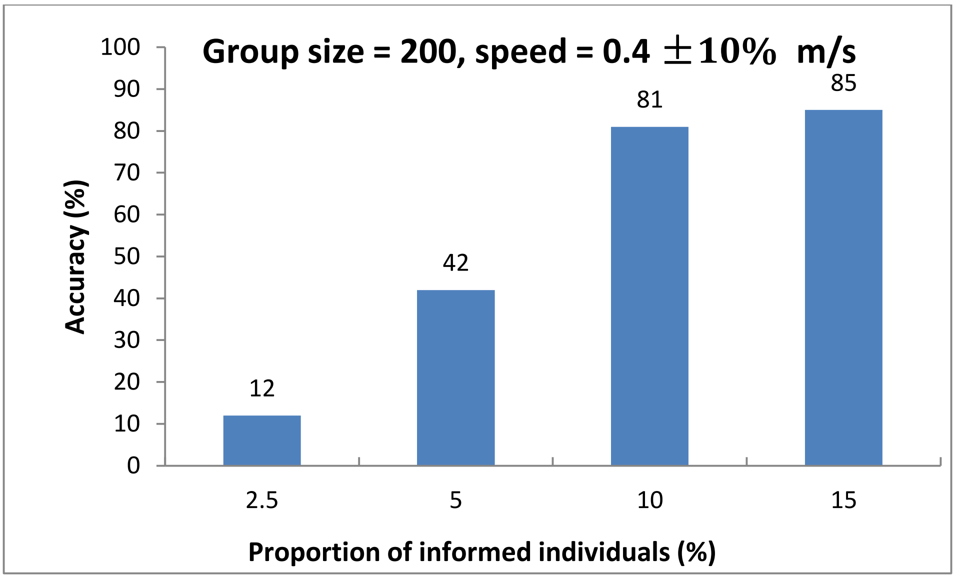


Figure 128 Successful rate of reaching the target with various percentages of informed agents

➢ Dyer et al.'s Results

In Dyer et al.'s experiment, each proportion of informed individuals was only tested once. Such data are considered as non-comparable to the simulation results.

Group Split

During the simulation, the group splitting phenomenon was not observed in any proportion of the informed individuals. However, it was reported that such a phenomenon was observed in Dyer et al.'s experiment where the informed individuals had a low percentage (less than 5%).

Because Dyer et al. claimed this finding was anecdotal due to the sample size, it remained unclear what the issues were that caused this group splitting behaviour happen. To further investigate such a phenomenon, a series of additional simulations were carried out and the results are discussed in a later section (8. 4. 5. 1).

Arrival Time

The following table lists the arrival times in both Dyer et al.'s experiment and in the simulation.

Table 41 Arrival time comparison – leadership and consensus decision making in large groups

Informed percentage	Dyer et al.'s experiment (s)	Simulation Average(s)
2.5%	222	75
5%	250	70
10%	75	58
15%	N/A	54

It can be seen that the arrival times in the simulation are quite different from those in the Dyer et al.'s experiment. The following reasons are considered to cause the differences.

- The walking speed is considered as the one main factor that can influence the arrival time. However, the walking speed of the participants cannot be identified in Dyer et al.'s experiment. In the simulation, the walking speed of agents is based on Sakuma et al.'s study (2005) which may not represent the experiment situation.
- The simulation only uses simplified rules to represent the individuals' behaviour which has been proved quite effective in the simulation of small groups. The mobility constraints of individuals in a large group could have more influence on the group walking speed thus result in a longer arrival time.
- The result of the experiment was for one round of tests only. It, therefore, has not much meaning in a statistical sense.

Agents' Walking Speed

In Dyer et al.'s study, it did not mention whether the participants' walking speeds had influence on the group behaviour. To further study the influence of the walking speed on the group behaviour, a series of simulations were carried out and further details on this can be seen in a later section (8. 4. 5. 2).

8. 4. 5 Further Simulations with Variations

8. 4. 5. 1 Simulation with different Group Behaviour Ranges

In Dyer et al.'s experiment, the participants were required to keep in the group but the group split phenomenon was still observed when the proportions of the informed individuals were low (In contrast, in the experiment with the small groups, the participants were able to keep within a whole group following the instruction not to split).

Although Dyer et al. did not undertake further discussion on the difference (i.e. the group split phenomenon in the experiments with small and large groups), it is worth further analysing why this phenomenon existed, especially if such a phenomenon had not been observed in the initial simulation?

Analysis on the Possible Issues

The instructions in the two experiments were identical so the simulation used the same configuration to model the agents in both cases. However, this approach may introduce some inaccurate interpretations on the participants' keep in group behaviour.

In the small group, one individual can easily align his/her movement with the whole group. However, in the large group, as 200 people would occupy a relatively large area, it is reasonable for one individual on one side of the group not to be aware of the individuals on the other side of group. When he/she coordinates his/her movement in the group, he/she will not take the individuals on the far side into account i.e. the keep in group behaviour is actually based on a subset of the people nearby. Therefore, for a large group, it is possible that when the people on one side decide to go in one direction were as the ones on other side want to go in the opposite direction, this could result in a group split phenomenon. In this case, all the individuals are still following the experiment instructions as they are all in groups although they end up in split-up groups.

In the simulation, the attribute - group behaviour range defines how far one agent can react to the other agents. In other words, this attributes decides what is a group for the agent. The agent only coordinates its behaviour within this range to maintain its position within group. However, there is no reference to suggest what value for the range should be set for the simulation. In the previous simulation of small groups, the range of 5 metres was given arbitrarily and seemed to provide the appropriate group behaviour. The same configuration has been used in this simulation, which seems to be the issue that influences group splitting behaviour.

The Splitting Behaviour with Various Group Behaviour Ranges Settings

To investigate the relationship between group splitting behaviour and the range of group behaviour attributes, a series of simulations were carried out with a fixed proportion (10%) of informed individuals at various group behaviour ranges (*simulations with each observed range were repeated 40 times*).

Table 42 Simulation results with a 200 agents' group (10% informed)

Group Range	Result
2.5m	In 95% of the simulations, the crowd reached the target as a whole group. In the rest 5% had a sub-group reaching the target with an average 70% proportion of the crowd.
5m	Crowd reached the target as a whole group in all simulations.
7.5m	Crowd reached the target as a whole group in all simulations.

8. 4. 5. 2 Simulation at different Walking Speeds

Simulation Configuration

In order to find out the relationships between speed, arrival time and numbers of informed people, a series of simulations were tested with agents at different walking speeds. A total of 9 sets of simulations were tested with the walking speed set at 0.4 m/s, 0.8 m/s and 1.2 m/s and with the informed individual percentages of 5%, 10% and 15% respectively.

(For each treatment, the simulation was repeated 100 times)

Result

Figure 129(a) shows that the arrival time has approximately a linear relationship with the informed numbers of people. Figure 129(b) shows that the amount of informed individuals could increase the accuracy of reaching the target and such an effect will reach a fairly effective rate when the informed percentage reaches 10% of the group. This result is similar to Dyer et al.'s (2009) experiment. In addition, this is also in line with the simulations in the animal group (Couzin et al. 2005) and the leader behaviour in evacuation study (Nuria Pelechano & Norman I Badler 2006).

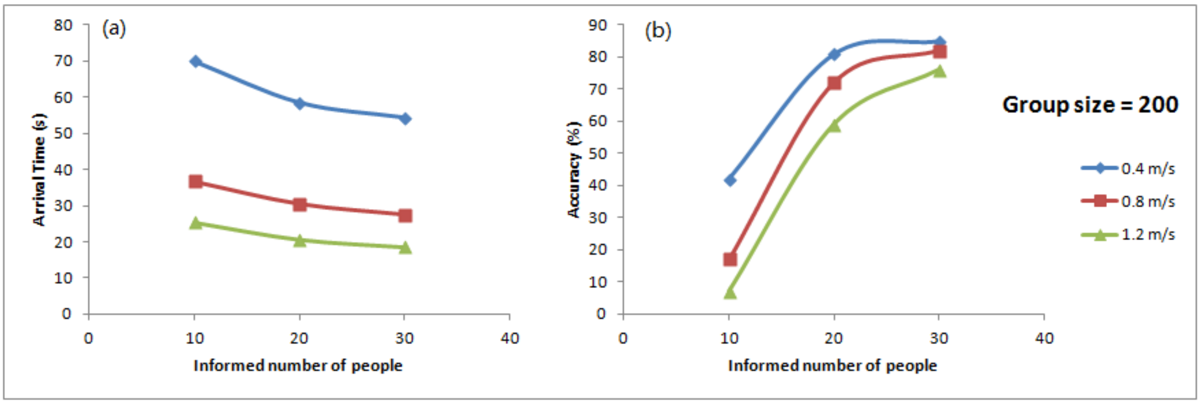


Figure 129 Arrival time and accuracy changed with the various informed numbers of people

Figure 130(a) shows that the arrival time changes with speed. Compared to Figure 124(a), it can be found the relationship between the arrival time and the walking speed in a large group is similar to that of the small group. Figure 130 also indicates that the relationship between the arrival accuracy and the walking speed seems linked to the informed percentage of the group as well. Although the arrival accuracy drops when the walking speed increases, the higher percentage of informed people, the less the accuracy decreases. By comparing Figure 124(b) and Figure 130(b), one can see that, in the large group, the arrival accuracy is affected less by the increasing walking speed. In the case of 2 informed individuals within 10 people (equal to 20% being informed), the accuracy dropped from 75% to 45% when the walking speed increased from 0.4 m/s to 1.2 m/s. In the case of having 15% informed people in a group of 200, the accuracy only changed from 85% to 80%.

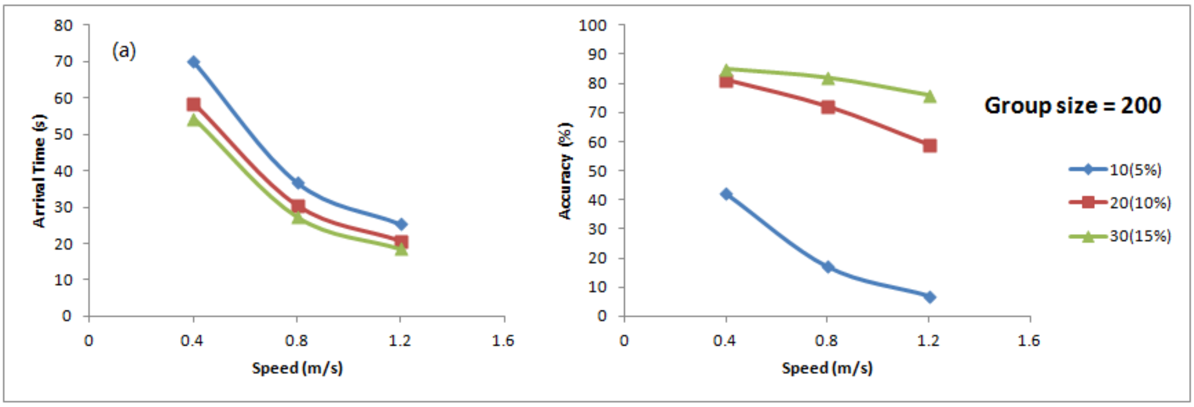


Figure 130 Arrival time and accuracy at various walking speeds

8. 5 Summary of the Chapter

This chapter introduces the second aspect of the evaluation of the proposed model which validates the crowd model by comparing the simulation results to existing studies or to real life experiments in three selected scenarios. The first simulation represents the scenario of simultaneous lane formation in a bi-directional crowd walking flow. The second and the third simulations reproduce the experiments on leadership and consensus decisions in small and large groups (J. R. G. Dyer et al. 2009). The analysis and discussions on the results show that the crowd model can provide similar and reliable simulations on the known crowd behaviours. Furthermore, in the latter two simulations, additional configurations were tested and additional findings were reported which further demonstrates the applications of the crowd model and how it can support real-life studies.

Chapter 9 CONCLUSION

This final chapter provides a summary of this PhD research study. It begins with a review of the whole thesis. Then, an assessment is presented to evaluate the achievements of this study's research objectives in the second section, followed by a section that states the contributions made by this research study. The last section discusses the suggestions for future work.

9. 1 Summary of the Thesis

This thesis presented an innovative research study on the subject of crowd behaviour modelling and simulation. The whole thesis consisted of three parts:

✧ **Chapter 1: A General Introduction to the Research**

The first chapter laid out the foundations of this research study by presenting the research context and the research motivation. In order to carry out the study, a research methodology, including a literature review, software prototyping and case studies, was introduced. The research aim was to develop and implement a crowd model to simulate and analyse crowd movement which provides the flexibility to configure individual behaviours (increase heterogeneity) and the ability to represent the interactions between individuals. Six research objectives were identified to complete this research aim. At the end of Chapter 1, the structure of the thesis and the content of each chapter were introduced.

✧ **Chapters 2 to 7: A Detailed Description of the Main Research Work**

The middle part of the thesis contained a description of the main research work. It first justified and introduced the three research methods adopted in this study (i.e. the literature review and software prototyping). Then, a literature review was conducted with the focus on the three areas : 1) identifying the key elements and research requirements in crowd modelling; 2) providing background knowledge on crowd modelling and a comprehensive review of model design, simulation applications and technologies so that the appropriate approaches for this study could be selected; 3) surveying the crowd behaviours that have been presented in simulations and in real-life studies which could be used in the evaluation stage of this study.

Chapter 4 presented the detailed design of the crowd model which was the key part of this study. It proposed a crowd model that combined force-based modelling and agent-based modelling to take into account crowd heterogeneity in the behaviour effects' representation. A generic formula with seven parameters to present different behaviour effects was proposed at the bottom level of the crowd model. An agent model which described the individuals' attributes, knowledge, status and their decision making processes was presented at the top level. A Behaviour Library was introduced to link those two levels.

In the implementation stage, a prototype of the simulation system was developed based on the proposed crowd model. Chapter 6 firstly introduced the design of the simulation environment and the simulation engine foundation - the Microsoft XNA framework. Secondly, the detailed implementation of the crowd model was presented.

The evaluation of the crowd model was presented to complete this study. Chapter 7demonstrated the model applications through three selected scenarios. Chapter 8tested the validity and reliability of the crowd model and further three simulations were conducted to reproduce the crowd behaviours both from existing crowd simulation studies and real-life experiments.

✧ **Chapter 8: A Conclusion of the Research Assessment, Contribution, and Future Work**

In this, final chapter of the thesis, the research aim of the study is assessed via reviewing the achievements of the research objectives. Then the research contributions are introduced. Finally, the future work will be suggested to complete this thesis.

9. 2 Research Assessment

This PhD research study aimed to develop and implement a crowd model which provides the flexibility to configure individual behaviours (i.e. increasing heterogeneity) and the ability to represent the interactions between individuals in order to simulate and analyse crowd movement. In order to assess whether the research aim has been achieved, the research objectives are used as the basic criteria for the assessment.

- ✧ **Objective 1: To identify the key element(s) and research need(s) in crowd modelling and simulation.**

This objective was addressed via the critical literature reviews of the existing studies of crowd modelling and simulation. The achievement of this objective was delivered in two aspects. The first aspect was to identify the design of the crowd model and the key elements in crowd simulation through reviews on the components, purposes, benefits and limitations of crowd modelling and simulation which was presented in section 2. 1 . The second aspect related to identifying the gaps in the literature which suggested a need to design a crowd model that featured crowd heterogeneity and individual behaviours with a generic approach. This was presented in section 2. 7 .

- ✧ **Objective 2: To review crowd modelling approaches, crowd models, simulation applications, crowd behaviours, model design technologies, and simulation software in the context of crowd simulation.**

This objective was achieved through the comprehensive literature reviews described in chapter 3. The crowd modelling approaches were categorised into macroscopic and microscopic in this study with a focus on the latter. Five sub-categories of the microscopic crowd models were presented and the typical studies and models in each sub-category were critically reviewed. Then, the studies on simulation applications and crowd behaviours were discussed. Finally, the implementation of crowd simulation was reviewed via three aspects: simulation software, simulation packages, and the navigation representation.

- ✧ **Objective 3: To define a unified method of representing individual behaviours by taking into account crowd heterogeneity.**

The achievement of this objective was delivered by the establishment of a foundation for the crowd model in this study. The foundation laid on was built by? representing different behaviour effects on individuals' movement through a unified formula. The formula contained seven key generic parameters and their values were determined both by the types of the behaviour (primary) and the individual personal attributes (secondary).

- **Objective 4: To design a crowd model that can represent human behaviours and the complex effects of these behaviours on movements.**

This objective was achieved through the presentation of the crowd model in Chapter 4. A generic crowd model that combined force-based modelling and agent-based modelling was introduced. The model was described in detail in three aspects: 1) the behaviour effect representation and its calculation; 2) the agent model; 3) the Behaviour Library. The crowd model workflow section explained how human behaviours were represented and how those behaviours affect the agents' movements. In Chapter 7, the positive evaluation results from the three selected simulations demonstrated the ability of this model to represent human behaviours and those complex effects on movements.

- **Objective 5: To implement a prototype simulation system for the proposed crowd model.**

The implementation of the crowd simulation system prototype was presented in Chapter 6. The development of the crowd simulation system utilised the Microsoft XNA framework. And Cell and Portal Graph was adopted as the navigation representation. The implemented system was later used in Chapter 7 and Chapter 8 for evaluation, which confirmed the completion of this objective.

- **Objective 6: To evaluate and validate the crowd model with a series of simulations.**

The evaluation and validation of the proposed crowd model were carried out through a series of simulations by using the implemented crowd simulation system. Six selected scenarios and their simulations were presented in Chapter 7 and Chapter 8. As a conclusion, the analysis and discussions on the simulation results showed that the crowd model could present validated and reliable simulations on crowd behaviours. It also indicated that crowd heterogeneity did have influences on crowd movement.

To sum up, with the successful delivery of the six objectives identified in the research, this PhD study is concluded as having achieved its research aim fully.

9. 3 Contribution to Knowledge

Alongside the achievement of the research aim and objectives, the contributions of this PhD research can be described in four aspects:

- It presented individuals' behaviours as quantitative effects on their movement and proposed a unified formula to calculate these effects;
- This research considered that an individual's movement should take multiple behaviours into account. Thus, a universal mechanism to calculate those combinations of behaviour effects was proposed.
- It introduced a Behaviour Library in which generic behaviours were defined;
- The proposed crowd model was designed with a structure with four loosely-coupled modules.

The first contribution to knowledge is to interpret individuals' movement-related behaviours in the manner of quantitative effects, and proposes a unified formula (Formula 39) to calculate these effects. Although there are existing studies attempting to provide methods to calculate such behaviour effects, none of them is able to provide a universal representation of different behaviour effects. In this study, a behaviour that can result in a positional change of an individual is defined as a movement-related behaviour. This type of behaviour is interpreted as an effect on the individual. And as a result of that effect, the position of individual is changed. This study has identified seven generic parameters which determine the effects of these behaviours, and proposes a unified formula to incorporating these parameters for behaviour effect calculations (individuals' attributes and environmental influence are taken into account in determining the values of parameters). This approach to representing effects of different behaviours via a unified formula with seven generic parameters is innovative in the field of crowd modelling thus is considered as a novel contribution to knowledge.

The second contribution is that this research treats each individual's movement as the overall effect of multiple behaviours, and proposes a universal mechanism to combine the effects of these behaviours. In the real world, when an individual takes action to move, the movement is usually influenced by multiple behaviours at the same time (e.g., an individual has his/her own desire but he/she is also influenced by nearby crowd or environmental objects). In this study, these multiple behaviour effects are firstly calculated via applying the unified formula (Formula 39) and then are combined into one final effect to determine the movement of the individual. Because these effects are represented by forms of vectors, their combination can be achieved via basic vector operations. In existing studies of multiple behaviours, a weighting factor is required for each behaviour to reflect individual preferences. When introducing a new behaviour, the balance of these weighting factors needs to be reconsidered. However, in this study's model, no additional weighting factor for behaviour is required because such weightings have already been considered via the parameters in the behaviour effect calculation formulas (which are derived from the unified formula for each behaviour). This simplified behaviour effect combination mechanism introduces a novel approach in crowd modelling.

The third contribution of this PhD study is to build a Behaviour Library where generic behaviours are defined. Individuals' behaviours in real-life are interpreted and represented via the behaviours in the Behaviour Library or their combinations. The Behaviour Library, on the one hand, presents behaviour effect calculations for behaviours via the derivations of the unified formula (Formula 39). On the other hand, it describes how the agent information and the environmental issues affect the values of the parameters in those formulas from a generic perspective. Such approach introduces the concept of configuring a generic crowd model into specific scenarios. For a specific scenario, only extra rules need to be identified in order to provide scenario-specific behaviour selections, while the crowd heterogeneity and variances in behaviours can be achieved via assigning different attributes to agents. As the working mechanisms of behaviours are already defined in the Behaviour Library, this approach provides the flexibility to configure a generic crowd model in order to fit in with different scenarios.

The fourth contribution from this research study is to design the crowd model with a structure of loosely coupled modules. The proposed crowd model consists of four modules: Agent Action Engine (which describes the decision making process), Behaviour Library (where generic behaviours are defined), Agent Information (refers to a collection of agents' attributes, status, and knowledge), and Simulation World (which provides information of all the objects in the model). Each module is designed to serve its specific purpose and its detailed implementation is independent to other modules. Because they are working in a loosely coupling manner, each module can be expanded or modified separately as long as it provides the designed functions. Such approach enables future studies can focus on certain aspect of the crowd model and is considered as a novel attempt.

9. 4 Future Work

To complete this thesis, a few suggestions for further research are made. The following topics have been selected from many possibilities and are considered as having the most potential for future expansion:

- **Overlapping Positions of the Agents**

The positions of the agents are modelled not to overlap with each other in this research study for two reasons: 1) to simplify the design of the crowd model and its implementation; 2) the crowd model does not aim to deal with overcrowded scenarios where the agent's occupied area may be small than its usual body size. In the proposed crowd model, the agents can have zero distance from each other but cannot overlap each other if a collision happens. In the case of such a situation likely to occur, they will simply stop moving or manoeuvre to spaces to avoid the collision/overlapping. However, it is realised that collisions and overlapping behaviour could produce a more realistic simulation and some studies have taken this into account (e.g. the Social Force model (D. Helbing et al. 2000)). Future research can consider integrating collisions/overlapping behaviour.

- **The Constraint of Acceleration**

In this crowd model, acceleration was not considered in the continuous updating period of the agent's movement which means that the speed of an agent can change to any value and its orientation can be turned into any direction in the next update period. Such a design is based on the assumption that the agent can adjust its speed and walking direction within the update period. Although during simulations of the evaluation, no unrealistic movement was observed and the simulation results were in-line with existing studies, more practical rules of acceleration could be introduced in future research, i.e. the agent may need several update periods to reach certain speed rather than one if such acceleration cannot be completed within one update period.

✧ **Integration of Artificial Intelligence**

In this study, the Agent Action Engine has provided a guideline to the agent decision making process. The rationale for decision making is identified during the simulation configuration for each scenario individually. Future research can focus on integrating artificial intelligence into the decision making process in the crowd model. For example, cognitive theory could be introduced to the agent's perception.

✧ **Expanding the Behaviour Library**

Currently, the Behaviour Library consists of ten behaviour rules which have provided adequate combinations to represent all the behaviours in the evaluation simulations. However, it is possible and ideal that more generic behaviour rules should be identified through various scenarios and case studies in further research.

REFERENCE

Aguiar, F.H.M., 2010. *Crowd Simulation Applied to Emergency and Evacuation Situations*. FEUP, Universidade do Porto.

Averill, J.D.. et al., 2005. *Occupant Behavior, Egress, and Emergency Communication. Federal Building and Fire Safety Investigation of the World Trade Center Disaster (NIST NCSTAR 1-7)*, Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Occupant+Behavior,+Egress,+and+Emergency+Communication.#4 [Accessed June 15, 2012].

Bai, Y. et al., 2008. Agent-Based Spatial Query Optimization in a Grid Environment. *2008 International Conference on Computer Science and Software Engineering*, pp.611–614. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4722693.

Bandini, S., Manzoni, S. & Vizzari, G., 2002. Situated cellular agents: A model to simulate crowding dynamics. *IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS E SERIES D*, 87(3), pp.669–676. Available at: http://www.lintar.disco.unimib.it/space/Persone/Dott.+Giuseppe+Vizzari/Giuseppe+Vizzari+(english)/BanManViz-IEICE.pdf.

Bandini, Stefania, Federici, M.L. & Vizzari, Giuseppe, 2007. Situated Cellular Agents Approach To Crowd Modeling and Simulation. *Cybernetics and Systems*, 38(7), pp.729–753. Available at: http://www.informaworld.com/openurl?genre=article&doi=10.1080/01969720701534141&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3 [Accessed October 11, 2010].

Bandini, Stefania, Manzoni, Sara & Vizzari, G., 2009. Agent based modeling and simulation: an informatics perspective. *Journal of Artificial Societies and Social Simulation*, 12(4), p.4. Available at: http://jasss.soc.surrey.ac.uk/12/4/4.html [Accessed January 12, 2011].

Bell, J., 2010. *Doing your research project : a guide for first-time researchers in education, health and social science*, Maidenhead: McGraw-Hill Open University Press.

Blaikie, N., 2007. *Approaches to Social Enquiry* Second Edi., Polity Press.

Blue, V.J. & Adler, J.L., 2001. Cellular automata microsimulation for modeling bi-directional pedestrian walkways. *Transportation Research Part B: Methodological*, 35(3), pp.293–312. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0191261599000521.

Boehm, B., 1986. A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), pp.14–24. Available at: http://portal.acm.org/citation.cfm?doid=12944.12948.

Le Bon, G., 1895. *The Crowd: A Study of the Popular Mind*,

Bonabeau, E., 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3), p.7280. Available at:

http://www.pnas.org/content/99/suppl.3/7280.short [Accessed March 26, 2011].

Bourner et al., 1996. *Research methods : guidance for postgraduates* T. Greenfield, ed., J. Wiley.

Bratman, M.E., 1987. *Intention, Plans, and Practical Reason.*, CSLI Publications.

Braun, A. et al., 2003. Modeling individual behaviors in crowd simulation. In *16th International Conference on Computer Animation and Social Agents*. Published by the IEEE Computer Society, pp. 16–21. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1199317 [Accessed March 7, 2013].

Bumann, D., Krause, J. & Rubenstein, D., 1997. Mortality Risk of Spatial Positions in Animal Groups: the Danger of Being in the Front. *Behaviour*, 134(13), pp.1063–1076. Available at: http://openurl.ingenta.com/content/xref?genre=article&issn=0005-7959&volume=134&issue=13&spage=1063 [Accessed September 29, 2011].

Burstedde, C. et al., 2001. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3–4), pp.507–525. Available at: http://www.sciencedirect.com/science/article/pii/S0378437101001418.

Chalk, A., 2013. Microsoft Confirms the End of XNA. *The Escapist*. Available at: http://www.escapistmagazine.com/news/view/121843-Microsoft-Confirms-the-End-of-XNA [Accessed August 2, 2013].

Cheng, W. et al., 2008. A modified particle swarm optimization-based human behavior modeling for emergency evacuation simulation system. In *Information and Automation, 2008. ICIA 2008. International Conference on*. IEEE, pp. 23–28. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4607962 [Accessed July 26, 2011].

Chenney, S., 2004. Flow tiles. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '04*. New York, New York, USA: ACM Press, p. 233. Available at: http://portal.acm.org/citation.cfm?doid=1028523.1028553 [Accessed March 28, 2013].

Chittka, L. et al., 2003. Psychophysics: bees trade off foraging speed for accuracy. *Nature*, 424(6947), p.388. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12879057 [Accessed September 29, 2011].

Cho, K. et al., 2008. BDI Model-Based Crowd Simulation H. Prendinger, J. Lester, & M. Ishizuka, eds. *Simulation*, 5208, pp.364–371. Available at: http://www.springerlink.com/content/tjm49p6665280j70.

Chu, C.-Y., 2009. A Computer Model for Selecting Facility Evacuation Design Using Cellular Automata. *Computer-Aided Civil and Infrastructure Engineering*, 24(8), pp.608–622. Available at: http://dx.doi.org/10.1111/j.1467-8667.2009.00619.x.

Colombo, R.M. & Rosini, M.D., 2005. Pedestrian flows and non-classical shocks. *Mathematical methods in the applied sciences*, 28(13), pp.1553–1568. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.6594&rep=rep1&type=pdf [Accessed July 28, 2011].

Comte, A., 1988. *Introduction to Positive Philosophy*, Hackett Pub Co.

Courage, C. & Baxter, K., 2005. *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools and Techniques*, Morgan Kaufmann.

Couzin, I.D. et al., 2005. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025), pp.513–6. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15690039.

D'Ambrogio, a. et al., 2009. Simulation model building of traffic intersections. *Simulation Modelling Practice and Theory*, 17(4), pp.625–640. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1569190X08002153 [Accessed December 10, 2010].

Davidsson, P., 2001. Multi agent based simulation: beyond social simulation. *Multi-Agent-Based Simulation*, pp.141–155. Available at: http://www.springerlink.com/index/tgx3cjt6r3apy44e.pdf [Accessed March 26, 2011].

Defense Science Board, 1987. *Report of the Defense Science Board Task Force on Military Software*,

Dena Taylor(University of Toronto), THE LITERATURE REVIEW : A FEW TIPS ON CONDUCTING IT. Available at: http://www.writing.utoronto.ca/advice/specific-types-of-writing/literature-review.

Denzin, N.K. & Lincoln, Y.S., 2000. *Handbook of Qualitative Research*, SAGE Publications, Inc.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), pp.269–271. Available at: http://link.springer.com/10.1007/BF01386390 [Accessed March 23, 2013].

Dijkstra, J., 2008. An Agent Architecture for Visualizing Simulated Human Behavior to Support the Assessment of Design Performance. *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, pp.808–813. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5172729.

Dijkstra, J., Timmermans, H. & Jessurun, A., 2000. A multi-agent cellular automata system for visualising simulated pedestrian activity. In *Proceedings on the 4th International Conference on Cellular Automata for Research and Industry*. pp. 29–36. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.25.7825&rep=rep1&type=pdf.

Drury, J., Cocking, C. & Reicher, Steve, 2009. Everyone for themselves? A comparative study of crowd solidarity among emergency survivors. *The British* *journal of social psychology / the British Psychological Society*, 48(Pt 3), pp.487–506. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18789185 [Accessed July 6, 2010].

Dyer, J.R.G. et al., 2008. Consensus decision making in human crowds. *Animal Behaviour*, 75(2), pp.461–470. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0003347207003764 [Accessed August 28, 2011].

Dyer, J.R.G. et al., 2009. Leadership, consensus decision making and collective behaviour in humans. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364(1518), pp.781–9. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2689712&tool=pmcentrez&rendertype=abstract [Accessed July 13, 2010].

Ebihara, M., Ohtsuki, A. & Iwaki, H., 1992. A Model for Simulating Human Behavior During Emergency Evacuation Based on Classificatory Reasoning and Certainty Value Handling. *Computer-Aided Civil and Infrastructure Engineering*, 7(1), pp.63–71. Available at: http://dx.doi.org/10.1111/j.1467-8667.1992.tb00417.x [Accessed May 9, 2013].

Epic Games Inc., 2009. Unreal Engine Features. *Epic Games Inc.* Available at: http://www.unrealengine.com/en/features/ [Accessed August 1, 2013].

Fang, W., Yang, Lizhong & Fan, W., 2003. Simulation of bi-direction pedestrian movement using a cellular automata model. *Physica A: Statistical Mechanics and its Applications*, 321(3–4), pp.633–640. Available at: http://www.sciencedirect.com/science/article/pii/S0378437102017326.

Faria, J.J. et al., 2010. Leadership and social information use in human crowds. *Animal Behaviour*, 79(4), pp.895–901. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0003347210000047 [Accessed August 24, 2011].

Flyvbjerg, B., 2006. Five Misunderstandings About Case-Study Research. *Qualitative Inquiry*, 12(2), pp.219–245. Available at: http://qix.sagepub.com/cgi/doi/10.1177/1077800405284363 [Accessed June 8, 2011].

Franks, N.R. et al., 2003. Speed versus accuracy in collective decision making. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1532), p.2457. Available at: http://rspb.royalsocietypublishing.org/content/270/1532/2457.short [Accessed October 6, 2011].

Fruin, J.J., 1987. *Pedestrian Planning and Design* 2nd ed., Mobile AL, Elevator World.

Gardner, M., 1970. Mathematical games: The fantastic combinations of John Conway's new solitaire game "Life". *Scientific American*, 223(4), pp.120–123. Available at: http://www.autzones.com/din6000/textes/semaine04/Gardner (1970)-Games.pdf [Accessed July 26, 2011].

George, M.P., Ingrand, F.F. & Georgey, M.P., 1989. Decision-Making in an Embedded Reasoning System. In *IJCAI*. Citeseer. Available at:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.7047&rep=rep1&type=pdf [Accessed July 26, 2011].

Georgoudas, I., Sirakoulis, G. & Andreadis, I., 2006. A Simulation Tool for Modelling Pedestrian Dynamics during Evacuation of Large Areas. In I. Maglogiannis, K. Karpouzis, & M. Bramer, eds. *Artificial Intelligence Applications and Innovations*. Springer US, pp. 618–626. Available at: http://dx.doi.org/10.1007/0-387-34224-9_72.

Hamagami, T. & Hirata, H., 2003. Method of crowd simulation by using multiagent on cellular automata. *IEEE/WIC International Conference on Intelligent Agent Technology, 2003. IAT 2003.*, pp.46–52. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1241047.

Heïgeas, L. et al., 2003. A physically-based particle model of emergent crowd behaviors. In *International Conference Graphicon 2003*. Citeseer, pp. 5–10. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.98.8963&rep=rep1&type=pdf [Accessed December 9, 2012].

Helbing, D et al., 2003. Lattice gas simulation of experimentally studied evacuation dynamics. *Physical Review E*. Available at: http://pre.aps.org/abstract/PRE/v67/i6/e067101 [Accessed March 8, 2013].

Helbing, D et al., 2002. Simulation of pedestrian crowds in normal and evacuation situations. In S. Schreckenberg, M and Sharma, ed. *PEDESTRIAN AND EVACUATION DYNAMICS*. pp. 21–58.

Helbing, D, Keltsch, J. & Molnár, P., 1997. Modelling the evolution of human trail systems. *Nature*, 388, pp.47–50.

Helbing, D., Farkas, I. & Vicsek, T., 2000. Simulating dynamical features of escape panic. *Nature*, 407(6803), pp.487–490. Available at: http://papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID948129_code653694.pdf?abstractid=947685&mirid=1.

HELBING, D., SCHONHOF, M. & STARK, H.-U., 2005. How individuals learn to take turns: emergence of alternating cooperation in a congestion game and the prisoner's dilemma ¨. *Advances in Complex Systems*, 8(1), pp.87–116.

Helbing, Dirk & Molnar, Peter, 1995. Social force model for pedestrian dynamics. *Physical review E*, 51(5), p.4282. Available at: http://pre.aps.org/abstract/PRE/v51/i5/p4282_1 [Accessed March 23, 2011].

Heliövaara, S. et al., 2012. Counterflow model for agent-based simulation of crowd dynamics. *Building and Environment*, 48, pp.89–100. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0360132311002630 [Accessed July 13, 2012].

Henderson, L.F., 1971. The statistics of crowd fluids. *Nature*, 229(5284), pp.381–3. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16059256 [Accessed July 26, 2011].

Hoes, P. et al., 2009. User behavior in whole building simulation. *Energy and Buildings*, 41(3), pp.295–302. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0378778808002168.

Hogg, M., 2009. Conway's Game of Life - An Illustrated Guide. Available at: http://www.michael-hogg.co.uk/game_of_life.php [Accessed February 15, 2013].

Honderich, T. ed., 1995. *The Oxford Companion to Philosophy*, Oxford University Press. Available at: http://www.informaworld.com/openurl?genre=article&doi=10.1080/01445340701300429&magic=crossref.

Hoogendoorn, S., 2003. Pedestrian travel behavior modeling. *Conference on Travel Behavior Research*. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.2244&rep=rep1&type=pdf [Accessed July 26, 2011].

Hughes, R.L., 2000. The flow of large crowds of pedestrians. *Mathematics and Computers in Simulation*, 53(4-6), pp.367–370. Available at: http://cat.inist.fr/?aModele=afficheN&cpsidt=813431.

Jian, L., Lizhong, Y. & Daoliang, Z., 2005. Simulation of bi-direction pedestrian movement in corridor. *Physica A: Statistical Mechanics and its Applications*, 354, pp.619–628. Available at: http://linkinghub.elsevier.com/retrieve/pii/S037843710500227X [Accessed August 19, 2012].

Johansson, A., Helbing, Dirk & Shukla, P.K., 2007. Specification of the Social Force Pedestrian Model By Evolutionary Adjustment To Video Tracking Data. *Advances in Complex Systems*, 10(supp02), p.271. Available at: http://www.worldscinet.com/acs/10/10supp02/S0219525907001355.html.

Kennedy, J. & Eberhart, R., 1995. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Perth, Australia: IEEE, pp. 1942–1948. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=488968 [Accessed January 27, 2012].

Kirchner, A. & Schadschneider, A., 2002. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, 312(1-2), pp.260–276. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0378437102008579.

Kitchenham, B., Pickard, L. & Pfleeger, S.L., 1995. Case studies for method and tool evaluation. *Software, IEEE*, 12(4), pp.52–62. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=391832 [Accessed June 8, 2011].

Kobes, M., Helsloot, I., De Vries, B., et al., 2010. Building safety and human behaviour in fire: A literature review. *Fire Safety Journal*, 45(1), pp.1–11. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0379711209001167.

Kobes, M., Helsloot, I., Vries, B. De, et al., 2010. Way finding during fire evacuation ; an analysis of unannounced fire drills in a hotel at night. *Building and Environment*, 45(3), pp.537–548. Available at: http://dx.doi.org/10.1016/j.buildenv.2009.07.004.

Kolakowski, L., 1972. *Positivist Philosophy: From Hume to the Vienna Circle (Pelican)*, Penguin Books Ltd.

Kruszewski, P., 2005. A game-based cots system for simulating intelligent 3d agents. *BRIMS'05: Proceedings of the 2005 behavior …*. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+game-based+COTS+system+for+simulating+intelligent+3D+agents#0 [Accessed March 15, 2013].

Kuligowski, E., 2005. Review of 28 Egress Models. *Workshop on Building Occupant Movement During Fire*. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Review+of+28+Egress+Models#2 [Accessed June 8, 2011].

Kuligowski, E.D. & Peacock, R.D., 2005. *A review of building evacuation models*, Available at: http://fire.nist.gov/bfrlpubs/fire05/PDF/f05107.pdf.

Lam, W.H.. et al., 2003. A generalised function for modeling bi-directional flow effects on indoor walkways in Hong Kong. *Transportation Research Part A: Policy and Practice*, 37(9), pp.789–810. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0965856403000582 [Accessed July 19, 2012].

Leca, J., 2003. Distributed leadership in semifree-ranging white-faced capuchin monkeys. *Animal Behaviour*, 66(6), pp.1045–1052. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0003347203922767 [Accessed August 12, 2011].

Lee, R. & Hughes, R., 2007. Minimisation of the risk of trampling in a crowd. *Mathematics and Computers in Simulation*, 74(1), pp.29–37. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0378475406002138 [Accessed December 16, 2011].

Leggett, R., 2004. Real Time Crowd Simulation: A Review. , (December), pp.1–12. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.116.9167&rep=rep1&type=pdf [Accessed December 10, 2012].

Lerner, A., Chrysanthou, Y. & Cohen-Or, D., 2006. Efficient cells-and-portals partitioning. *Computer Animation and Virtual Worlds*, 17(1), pp.21–40. Available at: http://dx.doi.org/10.1002/cav.70.

Lin, Z., Hsu, H. & Wang, F., 2005. Intention Scheduling for BDI Agent Systems. In *29th Annual International Computer Software and Applications Conference (COMPSAC'05)*. Ieee, pp. 133–140. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1510010.

Liu, M. & Lo, S.M., 2011. The quantitative investigation on people's pre-evacuation behavior under fire. *Automation in Construction*, 20(5), pp.620–628. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0926580510002128 [Accessed July 26, 2011].

Lo, S et al., 2008. A simulation model for studying the implementation of performance-based fire safety design in buildings. *Automation in Construction*, 17(7), pp.852–863. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0926580508000319 [Accessed September 9, 2010].

LOVAS, G., 1994. Modeling and simulation of pedestrian traffic flow. *Transportation Research Part B: Methodological*, 28(6), pp.429–443. Available at: http://linkinghub.elsevier.com/retrieve/pii/0191261594900132 [Accessed July 26, 2011].

Luo, L. et al., 2008. Agent-based human behavior modeling for crowd simulation. *Computer Animation and Virtual Worlds*, 19(3-4), pp.271–281. Available at: http://onlinelibrary.wiley.com/doi/10.1002/cav.238/abstract [Accessed September 5, 2011].

Macal, C.M. & North, M.J., 2007. Agent-based modeling and simulation: Desktop ABMS. *2007 Winter Simulation Conference*, pp.95–106. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4419592.

McKenzie, F.D. et al., 2006. Crowd Federate architecture and API design. *US Patent App. 11/*, 19. Available at: http://www.google.com/patents?hl=en&lr=&vid=USPATAPP11406805&id=rayXAAAAEBAJ&oi=fnd&dq=Crowd+Federate+Architecture+and+API+Design&printsec=abstract.

Microsoft Corporation, 2010a. Game Methods. Available at: http://msdn.microsoft.com/en-us/library/Microsoft.Xna.Framework.Game_methods(v=xnagamestudio.40).aspx [Accessed July 9, 2013].

Microsoft Corporation, 2010b. Intro To Game.

Milazzo, J. et al., 1998. Effect of Pedestrians on Capacity of Signalized Intersections. *Transportation Research Record*, 1646(1), pp.37–46. Available at: http://trb.metapress.com/index/V571327727LW0135.pdf [Accessed July 26, 2011].

Moradi, F., Ayani, R. & Mahmood, I., 2008. An Agent-Based Environment for Simulation Model Composition. *2008 22nd Workshop on Principles of Advanced and Distributed Simulation*, pp.175–184. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4545345 [Accessed August 30, 2010].

Moussaïd, M. et al., 2009. Experimental study of the behavioural mechanisms underlying self-organization in human crowds. *Proceedings. Biological sciences / The Royal Society*, 276(1668), pp.2755–62. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2839952&tool=pmcentrez&rendertype=abstract.

Moussaïd, M., Helbing, Dirk & Theraulaz, G., 2011. How simple rules determine pedestrian behavior and crowd disasters. *PNAS*, pp.1–5.

Musse, S.R., 2001. Behavioral Modeling of Virtual Human Actors. *RITA*, 8(2), pp.125–141. Available at: http://wotan.liu.edu/docis/lib/tian/rclis/dbl/ritari/(2001)8:2<125:BMOVHA>/www.inf.ufrgs.br/~revista/docs/rita08/rita_v8_n2_p125a141.pdf [Accessed June 8, 2011].

Musse, S.R. & Thalmann, D, 1997. A model of human crowd behavior: Group inter-relationship and collision detection analysis. In *Computer Animation and Simulation*. Citeseer, pp. 39–51. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.7049&rep=rep1&type=pdf [Accessed June 8, 2011].

Musse, Soraia Raupp et al., 2005. Groups and crowd simulation. *ACM SIGGRAPH 2005 Courses on SIGGRAPH 05*, p.2. Available at: http://portal.acm.org/citation.cfm?doid=1198555.1198672.

Nagai, R., Fukamachi, M. & Nagatani, T, 2005. Experiment and simulation for counterflow of people going on all fours. *Physica A: Statistical Mechanics and …*. Available at: http://www.sciencedirect.com/science/article/pii/S0378437105003936 [Accessed March 8, 2013].

Naumann, J.D. & Jenkins, A.M., 1982. Prototyping: The New Paradigm for Systems Development. *MIS Quarterly*, 6(3), pp.29–44. Available at: http://www.jstor.org/stable/248654.

Von Neumann, J. & Burks, A.W., 1966. *Theory of self-reproducing automata*, University of Illinois Press.

Ng, M., Park, J. & Waller, S.T., 2010. A Hybrid Bilevel Model for the Optimal Shelter Assignment in Emergency Evacuations. *Computer-Aided Civil and Infrastructure Engineering*, 25(8), pp.547–556. Available at: http://dx.doi.org/10.1111/j.1467-8667.2010.00669.x.

Oxford Dictionaries, 2013a. “Behaviour”. *Oxford University Press*. Available at: http://oxforddictionaries.com/definition/english/behaviour [Accessed March 7, 2013].

Oxford Dictionaries, 2013b. “crowd”. *Oxford University Press*. Available at: http://oxforddictionaries.com/definition/english/crowd [Accessed March 7, 2013].

Oxford Dictionaries, 2013c. “Heterogeneous”. *Oxford University Press*. Available at: http://oxforddictionaries.com/definition/english/Heterogeneous [Accessed March 7, 2013].

Oxford Dictionaries, 2013d. “Homogeneous”. *Oxford University Press*. Available at: http://oxforddictionaries.com/definition/english/Homogeneous [Accessed March 7, 2013].

Oxford Dictionaries, 2013e. “individual”. *Oxford University Press*. Available at: http://oxforddictionaries.com/definition/english/individual [Accessed March 7, 2013].

Pan, X et al., 2006. Human and social behavior in computational modeling and analysis of egress. *Automation in Construction*, 15(4), pp.448–461. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0926580505000737.

Pan, Xiaoshan et al., 2007. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai & Society*, 22(2), pp.113–132. Available at: http://www.springerlink.com/index/10.1007/s00146-007-0126-1.

Parisi, D. & Dorso, C.O., 2007. Morphological and dynamical aspects of the room evacuation process. *Physica A: Statistical Mechanics and its Applications*, 385(1), pp.343–355. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0378437107007200 [Accessed March 31, 2011].

Pelechano, N, Allbeck, J M & Badler, N I, 2007. Controlling Individual Agents in High-Density Crowd Simulation. *Social Forces*, 1.

Pelechano, Nuria, Allbeck, Jan M. & Badler, Norman I., 2008. Virtual Crowds: Methods, Simulation, and Control. *Synthesis Lectures on Computer Graphics and Animation*, 3(1), pp.1–176. Available at: http://www.morganclaypool.com/doi/abs/10.2200/S00123ED1V01Y200808CGR008 [Accessed March 23, 2011].

Pelechano, Nuria & Badler, Norman I, 2006. Modeling crowd and trained leader behavior during building evacuation. *IEEE computer graphics and applications*, 26(6), pp.80–86. Available at: http://www.ncbi.nlm.nih.gov/pubmed/17120916.

Pelechano, Nuria & Malkawi, a, 2008. Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. *Automation in Construction*, 17(4), pp.377–385. Available at: http://dx.doi.org/10.1016/j.autcon.2007.06.005 [Accessed December 10, 2010].

Pelechano, Núria & Malkawi, A., 2007. Comparison of Crowd Simulation for Building Evacuation and an Alternative Approach. In *Proceedings of the Tenth International Building Performance Simulation Association IBPSA*. pp. 1514–1521.

Perez, G.J. et al., 2002. Streaming, disruptive interference and power-law behavior in the exit dynamics of confined pedestrians. *Physica A: Statistical Mechanics and its Applications*, 312(3-4), pp.609–618. Available at: http://www.sciencedirect.com/science/article/pii/S0378437102009871 [Accessed July 26, 2011].

Pettre, J., Laumond, J. & Thalmann, D, 2005. A Navigation Graph for Real-Time Crowd Animation on Multilayered and Uneven Terrain. Available at: http://vrlab.epfl.ch/~jpettre/publis/05vcrowdpettre.pdf.

Pradeep, 2013. Microsoft Confirms End Of XNA/DirectX Development. *Microsoft-News.com*. Available at: http://microsoft-news.com/microsoft-confirms-end-of-xnadirectx-development/ [Accessed August 2, 2013].

Presagis USA Inc., 2009. AI.implant - product overview. Available at: http://www.presagis.com/products_services/products/modeling-simulation/simulation/aiimplant/#overview [Accessed March 15, 2013].

Qiu, F. & Hu, X., 2010. Modeling group structures in pedestrian crowd simulation. *Simulation Modelling Practice and Theory*, 18(2), pp.190–205. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1569190X09001555 [Accessed August 9, 2011].

Quest 3D, 2009. Quest 3D - Product Info. Available at: http://support.quest3d.com/index.php?title=Reference_Manual/Getting_Started/Introduction_and_overview [Accessed August 20, 2013].

Read, P. & Meyer, M.-P., 2000. *Restoration of motion picture film*, Oxford; Boston: Butterworth-Heinemann. Available at: http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=204209.

Reebs, S., 2000. Can a minority of informed leaders determine the foraging movements of a fish shoal? *Animal behaviour*, 59(2), pp.403–409. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10675263 [Accessed June 15, 2011].

Reebs, S.G., 2001. Influence of Body Size on Leadership in Shoals of Golden Shiners, Notemigonus Crysoleucas. *Behaviour*, 138(7), pp.797–809. Available at: http://www.catchword.com/cgi-bin/cgi?body=linker&ini=xref&reqdoi=10.1163/156853901753172656 [Accessed September 29, 2011].

Reynolds, C.W., 1987. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. ACM, pp. 25–34. Available at: http://portal.acm.org/citation.cfm?id=37406 [Accessed March 23, 2011].

Reynolds, C.W., 1999. Steering behaviors for autonomous characters. In *Game Developers Conference. http://www. red3d. com/ …*. Citeseer. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Steering+Behaviors+For+Autonomous+Characters#0 [Accessed March 7, 2013].

Robson, C., 2002. *Real world research : a resource for social scientists and practitioner-researchers*, Oxford [u.a.]: Blackwell.

Rossmann, J., Hempe, N. & Tietjen, P., 2009. A flexible model for real-time crowd simulation. *2009 IEEE International Conference on Systems, Man and Cybernetics*, (October), pp.2085–2090. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5346308.

Royce, W., 1970. Managing the development of large software systems. *Proceedings of IEEE Wescon*, (August), pp.1–9. Available at: http://personal.stevens.edu/~pkorfiat/SET/Research/5_006_Manage_dev_Large_SW_Sys.pdf.

Sakuma, T., Mukai, T. & Kuriyama, S., 2005. Psychological model for animating crowded pedestrians. *Computer Animation and Virtual Worlds*, 16(3-4), pp.343–351. Available at: http://doi.wiley.com/10.1002/cav.105 [Accessed August 18, 2011].

Santos, G. & Aguirre, B.E., 2004. A critical review of emergency evacuation simulation models. In *In Proceedings of the NIST Workshop on Building OccupantMovement during Fire Emergencies.* Disaster Research Center, pp. 25–50. Available at: http://dspace.udel.edu:8080/dspace/handle/19716/299 [Accessed July 25, 2013].

Schultz, M., Lehmann, S. & Fricke, H., 2007. A discrete microscopic model for pedestrian dynamics to manage emergency situations in airport terminals. In N. Waldau et al., eds. *Pedestrian and Evacuation Dynamics 2005*. Springer Berlin Heidelberg, pp. 369–375. Available at: http://dx.doi.org/10.1007/978-3-540-47064-9_35.

Seeley, T.D. & Buhrman, S.C., 1999. Group decision making in swarms of honey bees. *Behavioral Ecology and Sociobiology*, (45), pp.19–31. Available at: http://www.springerlink.com/index/X7HUAX26PJTBRF8A.pdf [Accessed September 29, 2011].

Seeley, T.D. & Visscher, P.K., 2004. Group decision making in nest-site selection by honey bees. *Apidologie*, (35), pp.101–116. Available at: http://www.blackburnbeekeepers.com/HOW BEES SWARM.pdf [Accessed September 29, 2011].

Seidel, T. et al., 2008. An agent-based approach to self-organized production. *Swarm Intelligence*, pp.219–252. Available at: http://www.springerlink.com/index/k11803632lg45425.pdf [Accessed September 5, 2011].

Shen, W., Shen, Q. & Sun, Q., 2012. Building Information Modeling-based user activity simulation and evaluation method for improving designer–user communications. *Automation in Construction*, 21(1), pp.148–160. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0926580511001117 [Accessed June 27, 2013].

Shendarkar, a et al., 2008. Crowd simulation for emergency response using BDI agents based on immersive virtual reality. *Simulation Modelling Practice and Theory*, 16(9), pp.1415–1429. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1569190X08001433.

Simpson, D., 2004. Behaviour based motion simulation for fire evacuation procedures. *Proceedings Theory and Practice of Computer Graphics, 2004.*, 2004, pp.112–118. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1314460.

Singh, M.P., 1998. Semantical considerations on intention dynamics for BDI agents. *Journal of Experimental & Theoretical Artificial Intelligence*, 10(4), pp.551–564. Available at: http://www.informaworld.com/openurl?genre=article&doi=10.1080/095281398146752&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3.

Song, Weiguo et al., 2006. Evacuation behaviors at exit in CA model with force essentials: A comparison with social force model. *Physica A: Statistical and Theoretical Physics*, 371(2), pp.658–666. Available at:

http://linkinghub.elsevier.com/retrieve/pii/S0378437106003633 [Accessed November 23, 2010].

Stern, E. & Richardson, H., 2005. Behavioural modelling of road users: current research and future needs. *Transport Reviews*, 25(2), pp.159–180. Available at: http://www.informaworld.com/openurl?genre=article&doi=10.1080/01441640 42000313638&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3 [Accessed July 21, 2010].

Stoot, C. & Reicher, Stephen, 1998. Crowd action as intergroup process: introducing the police perspective. *European Journal of Social Psychology*, 28(4), pp.509–529. Available at: http://cat.inist.fr/?aModele=afficheN&cpsidt=2331352 [Accessed June 8, 2011].

Sud, A. et al., 2008. Real-time path planning in dynamic virtual environments using multiagent navigation graphs. *IEEE transactions on visualization and computer graphics*, 14(3), pp.526–38. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18369262.

Sung, M., Gleicher, M. & Chenney, S., 2004. Scalable behaviors for crowd simulation. *Computer Graphics Forum*, 23(3), pp.519–528. Available at: http://doi.wiley.com/10.1111/j.1467-8659.2004.00783.x.

Tajima, Y & Nagatani, T, 2002. Clogging transition of pedestrian flow in T-shaped channel. *Physica A: Statistical Mechanics and its Applications*. Available at: http://www.sciencedirect.com/science/article/pii/S0378437101004241 [Accessed March 8, 2013].

Tajima, Yusuke, Takimoto, Kouhei & Nagatani, Takashi, 2002. Pattern formation and jamming transition in pedestrian counter ow. , 313, pp.709–723.

Tang, F. & Zhang, X., 2008. A GIS-Based 3D Simulation for Occupant Evacuation in a Building. *Tsinghua Science & Technology*, 13(October), pp.58–64. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1007021408701275 [Accessed March 23, 2011].

Thompson, P.A. & Marchant, E.W., 1995. Testing and application of the computer model “SIMULEX”. *Fire Safety Journal*, 24(2), pp.149–166. Available at: http://linkinghub.elsevier.com/retrieve/pii/037971129500020T [Accessed September 26, 2012].

Torrens, D.P.M., 2007. Behavioral Intelligence for Geospatial Agents in Urban Environments. *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT’07)*, pp.63–66. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4407261 [Accessed November 14, 2010].

Turner, R.H. & Killian., L.M., 1957. *Collective behavior*, Englewood Cliffs, N.J .: Prentice-Hall.

Varas, a. et al., 2007. Cellular automaton model for evacuation process with obstacles. *Physica A: Statistical Mechanics and its Applications*, 382(2), pp.631–642. Available at:

http://linkinghub.elsevier.com/retrieve/pii/S0378437107003676 [Accessed February 27, 2013].

Vassalos, G., 2004. *Research project 490 phase 1: the effects of ship motion on the evacuation process, Critical Review of Data Available as Input to Evacuation Simulation Tools*,

Villamil, M.B. et al., A Model for Grouping Virtual Individuals Based on Social Behaviors. , pp.1–8.

Vizzari, Giuseppe, Pizzi, G. & Da Silva, F.S.C., 2008. A framework for execution and 3D visualization of situated cellular agent based crowd simulations. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, p.5. Available at: http://portal.acm.org/citation.cfm?doid=1363686.1363692.

Wang, Z. et al., 2012. Team-moving effect in bi-direction pedestrian flow. *Physica A: Statistical Mechanics and its Applications*, 391(11), pp.3119–3128. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0378437112000155 [Accessed August 19, 2012].

Wikipedia, 2013. Conway's Game of Life. Available at: http://en.wikipedia.org/wiki/Conway's_Game_of_Life [Accessed February 15, 2013].

Wolfram, S., 2002. History of cellular automata. In *A new kind of science*. Wolfram Media, Inc., p. 876. Available at: http://www.wolframscience.com/reference/notes/876b [Accessed November 23, 2011].

Wolfram, S., 1983. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3), pp.601–644. Available at: http://link.aps.org/doi/10.1103/RevModPhys.55.601 [Accessed December 5, 2011].

Wolfram, S., 1986. *Theory and Applications of Cellular Automata*, World Scientific.

Wright, S., 1987. *Crowds And Riots*, Sage Publications.

Yu, YF & Song, WG, 2007. Cellular automaton simulation of pedestrian counter flow considering the surrounding environment. *Physical Review E*, 75(4), p.046112. Available at: http://pre.aps.org/abstract/PRE/v75/i4/e046112 [Accessed July 26, 2011].

Yuan, J. et al., 2009. Integrated network approach of evacuation simulation for large complex buildings. *Fire Safety Journal*, 44(2), pp.266–275. Available at: http://linkinghub.elsevier.com/retrieve/pii/S037971120800091X.

Yuan, W. & Tan, K.H., 2007. An evacuation model using cellular automata. *Physica A: Statistical Mechanics and its Applications*, 384(2), pp.549–566. Available at: http://www.sciencedirect.com/science/article/pii/S0378437107006085.

Yue, H. et al., 2010. Study on bi-direction pedestrian flow using cellular automata simulation. *Physica A: Statistical Mechanics and its Applications*, 389(3), pp.527–539. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0378437109007882 [Accessed August 19, 2012].

Zhang, H. & Huang, S.Y., 2006. A General Framework for Parallel BDI Agents. *Computer Engineering*.

Zhang, Q., Zhao, G. & Liu, J., 2009. Performance-Based Design for Large Crowd Venue Control Using a Multi-Agent Model. *Tsinghua Science & Technology*, 14(3), pp.352–359. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1007021409700513 [Accessed December 10, 2010].

Zhang, T.I., Kendall, E. & Jiang, H., 2003. A Software Engineering Process for BDI Agnet-based Systems. In *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03)*.

Zhao, D., Yang, LZ & Li, J., 2006. Exit dynamics of occupant evacuation in an emergency. *Physica A: Statistical Mechanics and its Applications*, 363(2), pp.501–511. Available at: http://www.sciencedirect.com/science/article/pii/S0378437105008447 [Accessed July 26, 2011].

Zheng, M. et al., 2002. A model describing collective behaviors of pedestrians with various personalities in danger situations. In L. Wang et al., eds. *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.* Nanyang Technol. Univ, pp. 2083–2087. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1199043 [Accessed June 8, 2011].

Zheng, X., Zhong, T. & Liu, M., 2009. Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3), pp.437–445. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0360132308000577.

Zhou, S. et al., 2010. Crowd modeling and simulation technologies. *ACM Transactions on Modeling and Computer Simulation*, 20(4), pp.1–35. Available at: http://portal.acm.org/citation.cfm?doid=1842722.1842725 [Accessed June 27, 2011].

Zoumpoulaki, A., Avradinis, N. & Vosinakis, S., 2010. A multi-agent simulation framework for emergency evacuations incorporating personality and emotions. *Artificial Intelligence: Theories, …*, pp.1–6. Available at: http://www.springerlink.com/index/lj26871378x41638.pdf [Accessed March 4, 2013].

APPENDIX.1 SIMULATION RESULTS

Raw data of the simulations of Dyer et al.'s experiments (i.e. simulation 5 and simulation 6) are provided. Due to the nature of these data are mainly numeric and the large amounts of them, they are provided in an electronic manner.

They can be found in the CD which is part of the submission of this thesis.

Alternatively, they can be downloaded from the link below:

https://drive.google.com/folderview?id=0B9MyscUcRB-cNVcxcFE5S2ZSbkE&usp=sharing

APPENDIX.2 CODES OF IMPLEMENTATION

The simulation system of the crowd model was written in C# and was developed in Microsoft Visual Studio 2010.

The source codes are provided as a Visual Studio 2010 solution (zipped) which is in the CD as part of the submission for this thesis.

Alternatively, it can be downloaded from the link below:

https://drive.google.com/folderview?id=0B9MyscUcRB-cNVcxcFE5S2ZSbkE&usp=sharing

APPENDIX.3 VIDEOS OF SOME SIMULAITONS

During the evaluation of the crowd model, some videos were recorded for demonstration purpose. They can be found via the following three methods:

1. In the CD as part of the submission for this thesis.
2. Watch live at Youtube:
   http://www.youtube.com/channel/UCVrwG6zJSqSOuiC-BW8eU6Q/videos?view=1&feature=guide
3. Download from the link below:
   https://drive.google.com/folderview?id=0B9MyscUcRB-cNVcxcFE5S2ZSbkE&usp=sharing

APPENDIX.4 LIST OF PUBLICATIONS

- Sun, Q & Wu S. [under review]. A configurable agent-based crowd model with generic behaviour effect representation mechanism. Computer-Aided Civil and Infrastructure Engineering
- Wu, S & Sun, Q. [under review]. Computer simulation of leadership, consensus decision making and collective behaviour in humans. PLoS one.
- Shen W., Shen Q., Sun Q., 2012. Building Information Modeling-based user activity simulation and evaluation method for improving designer-user communications, Automation in Construction. Vol 21, pp 148-60.
- Sun, Q. & Wu, S., 2011. A Crowd Movement and Behaviour Observe Tool with Configurable Individual Agents to Support Building Layout Design and Navigation Plan. CIB 2011 w078/w102.
- Sun, Q. & Wu, S., 2011. A Crowd Model with Multiple Individual Parameters to Represent Individual Behaviour in Crowd Simulation. ISARC 2011.
- Sun, Q. & Wu, S., 2009. A Configurable Approach to Behavioural Modelling and Crowd Simulation. 9th UK-CARE AGM 2009.