The Extraction and Usage of Patterns from Video Data to Support Multi-Agent Based Simulation

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy by

Muhammad Tufail

January 2017

Dedication

To my parents

Acknowledgements

All praise be to Allah, the Most Merciful and the Most Gracious, for giving me the strength and power to overcome all the challenges that came my way while conducting my Ph.D study. I am thankful to him for blessing me with those people who encouraged me all the time, so that my efforts would be fruitful by the end of my Ph.D journey.

I am deeply thankful to my Supervisor, Professor Frans Coenen, who has given me a chance to work under his kind supervision and supported me with his extremely useful assistance and guidance throughout my Ph.D research. He was always supportive and cooperative; and always willing to listen, discuss and suggest research ideas. I learned, through his extraordinary experience, how to be a good researcher, an intellectual person, and above all to be scientist. He was the most perfect example for me, he inspired me and built up my experience to make me the person that I am today. I feel pleasure and honour to have been supervised by him.

I would also like to thank the following. Prof. Jane Hurst, Head of the Mammalian Behaviour and Evolution Institute of Integrative Biology at The University of Liverpool, who provided valuable support by recording the videos used in this research work and for her expertise and advise. I am thankful to Dr. Emmanuel Agiriga, Dr. Humayun Bakht, Dr. Kashif Kifayat, Dr Zahid, Dr. Muhammad Zaman and Saqib Jamshed Rind, who provided me with significant assistance. Thanks also to my fellow PhDs, in particular: Abdullah, Alsuhaibani, Ghazi, Fatima, Latifa, Joe, Kaka and Cheng Qiao. I am also grateful to the staff at the Department of Computer Science for providing the facilities I needed for my PhD study and helping in numerous other ways.

I am thankful to the Government of Khyber Pakhtunkhwa (KPK) Pakistan, who provided me with a Ph.D scholarship and I would like to thank all of my colleagues at Government Khushal Khan Khattak Degree College Akora Khattak, especially Prof. Shakeel who has been in contact with me throughout my Ph.D and encouraged me towards the completion of this thesis.

It is very difficult for me to find words to express my gratitude to my parents, they not only provided my endless care, love and support, but they have also encouraged me to be a self-confident person and contribute to society. I would like to thank all

my nephews, nieces, sisters and brothers, especially my elder brother Akhtar Munir, without whose support and encouragement this work would not have been started.

It will not be fair to complete this acknowledgement without mentioning gratitude to my children and my wife, who supported me all the time, provided too much care and scarifies in term of money and time. Due to this research I have used up time that I should have spent with my wife and my children, and I apologies to them for this.

Last but not the least this acknowledgement would not be complete if I did not express my sincere gratitude to my teacher, Jan Pervaiz, who (when I was 12 years old) said “I wish, you shall be a Ph.D Doctor in the future”; I hope he will be proud of me.

Abstract

The research work presented in this thesis is directed at addressing the knowledge acquisition bottleneck frequently encountered in computer simulation. The central idea is to extract the required knowledge from video data and use this to drive a computer simulation instead of the more conventional approach of interviewing domain experts and somehow encapsulating this knowledge in a manner whereby it can be used in the context of computer simulation. More specifically the idea presented in this thesis is to extract object location information from video data and then to mine this information to identify Movement Patterns (MPs) and then to utalise these MPs in the context of computer simulation.

To act as a focus for the work rodent behaviour simulation was considered. Partly because video data concerning rodent behaviour was relatively easy to obtain and partly because there is a genuine need to achieve a better understanding of rodent behaviour. This is especially the case in the context of crop damage.

There are a variety of computer simulation frameworks. One that naturally lends itself to rodent simulation is Multi Agent Based Simulation (MABS) whereby the objects to be simulated (rodents) are encapsulated in terms of software agents.

In more detail the work presented is directed at a number of research issues in the context of the above: (i) mechanisms to identify a moving object in video data and extracting associated location information, (ii) the mining of MPs from the extracted location information, (iii) the representation of MPs in such a way that they are compatible with computer simulation frameworks especially MABS frameworks and (iv) mechanisms where by MPs can be utilized and interacted with so as to drive a MABS. Overall two types of mechanisms are considered, Absolute and Relative.

The operation of rodent MABSs, driven using the proposed MP concept, is fully illustrated in the context of different categories of scenarios. The evaluation of the proposed MP driven MABSs was conducted by comparing real world scenarios to parallel simulated scenarios. The results presented in the thesis demonstrated that the proposed mechanisms for extracting locations, and consequently mining MPs, from video data to drive a MABS provides a useful approach to effective computer simulation that will have wide ranging benefits.

Contents

Dedication **i**

Acknowledgements **ii**

Abstract **iv**

Content **v**

List of Figures **ix**

List of Tables **xii**

List of Algorithms **xiv**

Notations **xv**

1 Introduction **1**
- 1.1 Overview and Motivation 1
- 1.2 Research Question 3
- 1.3 Contribution 4
- 1.4 Research Methodology 5
- 1.5 Publications 6
- 1.6 Thesis Structure 8
- 1.7 Summary 9

2 Background and Related work **10**
- 2.1 Introduction 10
- 2.2 Simulation 10
  - 2.2.1 Advantages of Simulation 11
  - 2.2.2 Application of Simulation 13
  - 2.2.3 Simulation Evaluation 14
  - 2.2.4 Multi Agent Based Simulation (MABS) 14
  - 2.2.5 Mechanisms for Representing Agents in MABS 15
  - 2.2.6 Data Acquisition 17
- 2.3 Knowledge Extraction From Video Data 17
  - 2.3.1 Video Data 18
  - 2.3.2 Object Detection 19

2.3.3 Object Tracking 22
2.3.3.1 Prediction of Object Next Locations 24
2.3.3.2 Multiple Object Tracking 25
2.3.4 Rodent Detection and Tracking in Video Data 27
2.4 Movement Pattern and Video Mining 29
2.4.1 Mining Movement Patterns 29
2.4.2 Video Mining 30
2.5 Application Domain 31
2.5.1 Animal behaviour simulation 31
2.5.2 Rodent movement behaviour simulation 32
2.6 Summary 33

3 Environment Modelling and Data Acquisition 34
3.1 Introduction 34
3.2 Environment Objects and Areas 36
3.3 Grid Representation 37
3.4 Relationship Between Agents 38
3.5 Movement Vectors 40
3.6 Location Descriptor 40
3.7 Scenario Categorisation 42
3.7.1 Simple Scenarios 42
3.7.2 Complex Scenarios 43
3.7.3 Two Rodents Scenarios 44
3.8 Video Data Analysis and Location Extraction 44
3.9 Summary 49

4 Pattern Mining Framework 51
4.1 Introduction 51
4.2 Formalism 52
4.3 Absolute Movement Patterns 53
4.4 Relative Movement Patterns (RMPs) 56
4.5 Advantages of relative mechanism over absolute mechanism 57
4.6 States 59
4.7 Movement Pattern Mining 62
4.8 Summary 65

5 MABS Operation Single and Two Rodent Scenarios 66
5.1 Introduction 66
5.2 Movement Pattern selection 67
5.2.1 Valid Accessible MP 70
5.2.2 Random Weighted Probabilistic Movement Pattern Selection . . . 72
5.2.3 Relaxation of MP Selection Criteria 75
5.3 Simulation Implementation 77
5.3.1 Environment Definition 77
5.3.2 Agent Definition 78
5.3.3 Simulation Time And Sample Time 78
5.3.4 Simulation Interface and Visualisation 81

5.4 Summary 84

6 Evaluation of MABS using MPs 85

6.1 Introduction 85
6.2 Category 1: Simple Scenarios 87
6.2.1 Comparison of the Usage of Absolute Movement Patterns with Real Life Experimentation (Simple scenarios) 89
6.2.2 Comparison of the Usage of Relative Movement Patterns with Real Life Experimentation (Simple scenarios) 90
6.2.3 Comparison of Frequency whereby nest sites, wall and open space locations are visited (Simple scenarios). 93
6.2.4 Comparison of Absolute versus Relative Movement Patterns (Simple scenarios) 94
6.2.5 Comparison of Different Path Lengths (Simple scenarios) 96
6.3 Category 2: Complex Scenarios 97
6.3.1 Comparison of the Usage of Absolute Movement Patterns with Real Life Experimentation (Complex scenarios) 98
6.3.2 Comparison of the Usage of Relative Movement Patterns with Real Life Experimentation (complex scenarios) 100
6.3.3 Comparison of Frequency with which Divisions were Visited (Complex Scenarios) 102
6.3.4 Comparison of Frequency of Divisions Transits (Complex Scenarios) 103
6.3.5 Comparison of Absolute versus Relative Movement Patterns (Complex Scenarios) 104
6.3.6 Usage of Relative Movement Patterns With Respect to Alternative Category 2 Scenarios 105
6.4 Category Study 3: Two Rodents Scenarios 106
6.4.1 Comparison of the Usage of Absolute Movement Patterns with Real Life Experimentation (Two Rodent Scenarios) 107
6.4.2 Comparison of the Usage of Relative Movement Patterns with Real Life Experimentation (Two Rodent Scenarios) 110
6.4.3 Comparison of Absolute versus Relative Movement Patterns, With and Without States (Two Rodent Scenarios) 110
6.4.4 Usage of Relative Movement Patterns With Respect to Alternative Category 3 Scenarios 111
6.5 Summary 111

7 Conclusion and Future Research Works 114

7.1 Introduction 114
7.2 Summary 114
7.3 Main Findings and Contribution 116
7.4 Future Work 120

Reference 124

A Location descriptors for three categories of scenarios 138

B Tables for Two Rodents Scenarios Video 1 and 2, Using Absolute Mechanism 142

C Tables for Two Rodent Scenario Videos 1 and 2, using Relative Mechanism 149

D Supporting Letter From Domain Expert 171

Illustrations

List of Figures

1.1 Rodent behaviour MABS vision 7

2.1 The main themes of the thesis: Simulation, Knowledge Extraction From Video Data and Pattern Mining 11
2.2 Illustration of the split and merge phenomena 26
2.3 Illustration of new object, lost and found objects and noise 26
2.4 Tendency of tourists to visit a set of places A to E in a particular sequence A → B → C → D → E within specified duration [40] 30

3.1 Grid Structure Linearised using Location IDs 38
3.2 Zone representation 39
3.3 Simple environment grid with Ground Types 41
3.4 Complex environment grid with Ground Types 41
3.5 Stills for Simple scenario 43
3.6 Stills for Complex scenarios 44
3.7 Stills from rodent video featuring two mice 45
3.8 Block Diagram for Video Data Acquisition Software (VDAS) 46
3.9 Background subtraction process 46
3.10 Flow chart for VDAS system 48

4.1 Example environment with absolute addresses in the context of the Simple Scenario category 54
4.2 Example environment with absolute addresses in the context of the Complex Scenario category 55
4.3 Example environment with absolute addresses in the context of the Two-rodent scenario category 55
4.4 Example environment with ground types in the context of the Simple Scenario category 56
4.5 Example environment with ground types in the context of the Complex Scenario category 57
4.6 Example environment with ground types in the context of the Two-rodent Scenario category 58
4.7 State Graph 60
4.8 Arcs of agents for defining the “follow” state 62

5.1 Components of envisioned MP driven MABS with respect to rodent simulation 67
5.2 Overview of movement pattern selection process 69
5.3 Illustrations of invalid MPs 72

5.4 Selection of a Movement Pattern based on Random Weighted Probability using the set of MPs given in Table 5.2 75
5.5 Example of the rotation of a relative location descriptor: (a) 0° rotation, (b) 90° rotation, (c) 180° rotation and (d) 270° rotation. 77
5.6 User Interface for Different Categories of Scenarios 83

6.1 Adopted Corroborative Evaluation Strategy ("Closing the loop") 86
6.2 Comparison of Absolute Movement Pattern Simulation data with Real Life data in the context of the Simple scenario category and in terms of the number of standard grid cells visited 88
6.3 Comparison of Absolute Movement Pattern Simulation data with Real Life data in the context of the Simple scenario category and in terms of the number of coarse block grid cells visited 89
6.4 Comparison of video and simulation block counts where data was split 50:50 90
6.5 Comparison of Simulation Movement with Real Movement using the relative representation for Simple scenarios 92
6.6 Comparison of nest box location counts obtained using video data versus those obtained using absolute and relative movement pattern simulations . . 93
6.7 Comparison of wall location counts obtained using video data versus those obtained using absolute and relative movement pattern simulations 94
6.8 Comparison of open space locations counts obtained using video data versus those obtained using absolute and relative movement pattern simulations . . 94
6.9 Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for left division of the example Complex scenario 99
6.10 Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for middle division of the example Complex scenario . . 99
6.11 Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for right division of the example Complex scenario . . . 99
6.12 Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for tunnel 1 of the example Complex scenario 100
6.13 Comparison of cells counts in simulation data with real life data, using the absolute mechanism, for tunnel 2 of the example Complex scenario 100
6.14 Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for the left division of the example Complex scenario 101
6.15 Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for the middle division of the example Complex scenario 101
6.16 Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for the right division of the example Complex scenario 102

6.17 Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for tunnel 1 of the example Complex scenario . 102
6.18 Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for tunnel 2 of the example Complex scenario . 103
6.19 Comparison of divisions counts in video data, with the simulation data, using absolute and relative movement patterns, and a Complex scenario 103
6.20 Comparison of division to division transit counts in real life video data with simulation data using absolute and relative movement patterns and the example complex scenario 104
6.21 Comparison of division counts using simulation data split 50:50 105
6.22 Case study: Usage of Relative Movement Patterns With Respect to an Alternative Category 2 Scenario. 106
6.23 Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns without state and (ii) video 1 data; for the example scenario Two rodent category 108
6.24 Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns without state and (ii) video 2 data; for the example scenario Two rodent category 108
6.25 Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns with state and (ii) video 1 data; for the example scenario Two rodent category 108
6.26 Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns without state and (ii) video 2 data; for the example scenario Two rodent category 109
6.27 Comparison of location counts in MPs (extracted from video 1) driven simulation data with video 2 data not used for MP extraction 109
6.28 Comparison of location counts in MPs (extracted from video 2) driven simulation data with video 1 data not used for MP extraction 110
6.29 Usage of Relative Movement Patterns With Respect to Alternative Category 3 Scenarios (3 mice), for Video 1 112
6.30 Usage of Relative Movement Patterns With Respect to Alternative Category 3 Scenarios (alternative obstruction arrangement), for Video 2 112

7.1 Additional Video Data for additional scenarios : (a) Maze, and (b) O-Box . . 121
7.2 Some suggested alternative configuration for relative location descriptors. . . 122

List of Tables

4.1 Example location information extracted video data using $i = 25$ 63

4.2 An example set of location IDs in a form of sequences using the location information from Table 4.1 63

5.1 Fragment of example MP database with respect to illustration given in Figure 5.3(c) 73

5.2 An example set of legal candidate MPs with their calculated probability values 74

5.3 Mechanisms for Alternative MP Selection 75

5.4 Mutual similarities among states 75

5.5 Location relationship percentages with respect to current location for 1000 locations using a range of values for T_{samp} and a simple category scenario where the observed entities are rats 79

5.6 Location relationship percentages with respect to current location for 1000 locations using a range of values for T_{samp} and complex and two-rodent category scenarios where the observed entities are mice 80

6.1 Statistical significance calculation 95

6.2 Statistical Significance testing results. 96

6.3 Average Error results using different path lengths 97

6.4 Comparison of AMPs versus RMPs in Complex Scenarios 105

6.5 Summary of simulation average error results 111

6.6 Summary of average error results with respect to the three different scenario categories (best results highlighted in bold font) 113

A.1 The complete set of location descriptors for the environment considered with respect to the Simple Scenario category 138

A.2 The complete set of location descriptors for the environment considered with respect to the Complex Scenario category 139

A.3 The complete set of location descriptors for the environment considered with respect to the Two Rodents Scenario category 140

B.1 Average absolute difference of two agents from video and simulation data using absolute mechanism without states for Two Rodents Scenarios Video-1 142

B.2 Average absolute difference of two agents from video and simulation data using absolute mechanism without states for Two Rodents Scenarios Video-2 143

B.3 Average absolute difference of two agents from video and simulation data using absolute mechanism with states for Two Rodents Scenarios Video-1 . . 145

B.4 Average absolute difference of two agents from video and simulation data using absolute mechanism with states for Two Rodents Scenarios Video-2 . . 146

C.1 Recorded results for Video 1 and Simulation using relative mechanism without states 149

C.2 Recorded results for Video 2 and Simulation using relative mechanism without states 152
C.3 Recorded results for Video 1 and Simulation using relative mechanism with states 154
C.4 Recorded results for Video 2 and Simulation using relative mechanism with states 162

List of Algorithms

1 Algorithm for Video Data Acquisition Software process 47
2 Algorithm for States 61
3 Mining MPs from Location DataBase 64
4 Pseudo Code illustrating the high level operation of an MP directed MABS 68
5 Selecting a valid and legal MP from a set of MPs 71
6 Calculation of Weighted Random Probability for Selecting a Next MP from a legal set of MPs P 74
7 Algorithm for finding simulation path points between two locations 82

Notations

The following notations and abbreviations are found throughout this thesis:

AMP	Absolute Movement Pattern
AVI	Audio Video Interleave
BDI	Belief Desire Intention
BLOB	Binary Large OBject
CA	Cellular Automaton
FSM	Finite State Machine
HD	High Definition
IBM	Individual Based Modelling
LtoL	Left to Left
LtoR	Left to Right
LtoM	Left to Middle
LBPs	Local Binary Patterns
MABS	Multi-Agent Based Simulation
MP	Movement Pattern
MPEG	Motion Pictures Expert Group
MtoL	Middle to Left
MtoM	Middle to Middle
MtoR	Middle to Right
.mov	Quick time MOVie
Open CV	Open source Computer Vision
RtoL	Right to Left
RtoM	Right to Middle
RtoR	Right to Right
RMP	Relative Movement Pattern
ROI	Region Of Interest
.rm	Real Media
SD	Standard Definition
VB	Visual Basic
VHFs	Viral Haemorrhagic Fevers
VC++	Visual C++
VDAS	Video DAta Acquisition Software

WMV	Windows Media Video
$d \times d$	grid cell size
$n \times n$	grid size
$d_{pz} \times d_{pz}$	Proximity Zone size
$d_{iz} \times d_{iz}$	Intrusion Zone size
T_{samp}	Sample Time Interval
T_{sim}	Simulation Time Interval

Chapter 1

Introduction

1.1 Overview and Motivation

Computer simulation is an important tool with respect to many application domains. Example domains can be found in: the Auto mobile industry [90], manufacturing and materials handling [106], health care [157] and military logistics [26]. In [90] simulation was used to establish and improve processes for assembling engines, processes which involve the integration of large numbers of components. In [106] simulation was used to analyse the design, operation and development of manufacturing processes. In [157] simulation was used to enhance the self-efficacy of physicians and nurses when using electronic medical records. In [26] it was used to overcome a wide range of military and related problems. The application domain considered in this thesis is that of animal behaviour analysis, more specifically rodent (rats and mice) behaviour analysis.

The motivation for rodent behaviour analysis is as follows:

1. Rodents are frequent carriers of viruses affecting both humans and livestock.
2. Rodents cause substantial amounts of crop damage to the world's food supplies.
3. Rodents cause significant and expensive damage to the built environment.

Rodents are vectors for pathogens causing many zoonotic and livestock diseases. For example, rodents are considered to be the main reservoir for viruses causing Viral Haemorrhagic Fevers (VHFs) diseases such as Ebola [32]. The multimammate rat, cotton rat, deer mouse, house mouse, and other field rodents are all examples of reservoir hosts. VHF diseases, and similar, can be controlled by discouraging rodents from entering or living in population areas. Rodent behaviour simulation can provide a better understanding of how to provide such control mechanisms.

Costs worldwide due to rodent damage to food supplies and the environment run to many billions of pounds. The extent of food loss is of particular current concern for global food security. For example, rodent infestation currently causes losses of 10-20% in cereal crops/food stores, and sometimes considerably more during outbreaks [141].

To illustrate the impact of this, models suggest that a 5% reduction in rodent losses to the cereal harvest in undernourished countries could feed one third of all undernourished people worldwide at current population levels [98]. Further, long term population projections indicate that global food production will need to increase by more than 40% by 2030 and 70% by 2050 compared to 2005-2007 levels [162]. Pressure to increase crop intensity will reduce fallow periods and create ideal conditions for near continuous rodent reproduction; this is predicted to lead to an increased frequency of serious rodent outbreaks [141]. Currently, to prevent crop damage by rodents, the world relies heavily on the use of rodenticides for rodent control, particularly anticoagulants ($> 90\%$ of rodenticides used in Europe). These are persistent, toxic to mammals and birds and cause substantial environmental damage [158], leading to increased restrictions on their usage in the EU but with no alternatives currently available. There is also growing public concern about the humaneness of most current rodenticides [97].

In the context of the built environment, and especially domestic premises, rodents cause significant damage to electrical cabling and the fabric of buildings. Areas favoured by mice are food storage and preparation areas such as kitchens and pantries, and warm areas such as airing cupboards, sub floor areas, enclosed pipes, baths and loft areas. Mice are very good climbers and typically obtain access to buildings via faulty drainage or service ducts, or by means of structural defects such as broken vents and damaged building fabric [11].

Given the above there is thus an urgent requirement to achieve more effective and more humane rodent control and reduce the use of rodenticides that harm the environment. This in turn necessitates a better understanding of rodent behaviour, such as that which can be facilitated using rodent simulations of the form envisioned in this thesis.

Although the work presented in this thesis is directed at rodent behaviour simulation to support rodent behaviour study, the techniques presented and evaluated are equally applicable to other domains. For example the way that people move round station forecourts and airport concourses. In general, computer simulation of real life scenarios offers a number of advantages [1, 9] over alternative forms of analysis as follows:

1. It is controllable in a manner that allows detailed investigation of parameters and repeat runs (difficult to achieve in the context of real life experimentation).
2. It is non-intrusive; it does not require participation from individuals.
3. It is cost effective (real life experimentation can be very costly).
4. It can be applied in areas where real life experimentation would be hazardous (in other words, it is safe).

The main disadvantage of simulation is (clearly) that we are not using real data. For simulation to be as effective as possible it is essential that the model used is as "realistic" as possible; the accuracy of the data on which a simulation model is based is thus of primary importance. This data can be collected in a variety of ways. The most straightforward, and the most frequently used, is simply to adopt a process of observation followed by "hand coding" of the observed behaviour [1–3]. There are two significant disadvantages of using such an observational approach: (i) it is resource intensive (time consuming) and (ii) it is prone to human error. To address these disadvantages the central idea proposed in this thesis is to extract data in an automated manner from video information in such a way that it is compatible with a desired simulation framework. More specifically to extract *Movement Patterns* (MPs) from video data. A number of different kinds of movement pattern are considered in this thesis, at a high level these can be categorised as being either referenced in absolute terms or in relative terms.

From the literature we can identify a number of frameworks whereby automated simulation can be achieved [1, 3, 34, 149]. One such framework is the Multi-Agent Based Simulation (MABS) framework, a framework based on the concept of multi agent technology. A framework founded on the idea that the entities that feature in a simulation can be expressed as agents. These agents then exist in some multi-agent platform, where they interact with each other via the mechanism provided by the chosen platform. MABS are particularly appropriate for modelling scenarios that involve people or animals, because they can be represented as individual agents within the environment in which they are intended to operate [1, 41, 80]. The main four characteristics of software agents which are as follows. (i) Responsive behaviour, (ii) Pro-active behaviour, (iii) Social behaviour, and (iv) Flexible behaviour are considered in the context of proposed MABS. The work presented in this thesis assumes usage of a MABS framework founded on the movement patterns.

The rest of this introductory chapter is organised as follows. Section 1.2 presents the research question and associated research issues. Section 1.3 lists the contributions of the thesis. Section 1.4 presents the research methodology adopted to provide a solution to the identified research question and issues. Section 1.5 reviews the publications to date that have arisen out of this thesis and Section 1.6 presents the structure of the remainder of this thesis. The chapter is concluded in Section 1.7 with a chapter summary.

1.2 Research Question

From the foregoing the research presented in this thesis is directed at how best to avoid hand coding of MABS by automatically processing video data of scenarios to be simulated. Thus the research question to be addressed can be formulated as follows:

> **"What are the most appropriate mechanisms that can be adopted so that some form of machine learning can be applied to video data describing animal (human) behaviour so as to extract sufficient information to populate and drive a MABS framework of some description?"**

The provision of an answer to the above research question requires the resolution of a number of associated research issues as follows:

1. *What is the most appropriate mechanism for extracting data from video?*
2. *Given that data can be successfully extracted from video, how can this best be translated to form the input to a simulation framework (more specifically, a MABS framework).*
3. *How best can a MABS model operate given the nature of the input data?*
4. *What is the most appropriate mechanism for evaluating simulators that operate using data extracted from video?*

The overriding requirement for addressing the above is that we wish to provide a simulation that is as realistic as possible (and consequently as useful as possible).

1.3 Contribution

This thesis makes a number of contributions to the domain of computer simulation, especially in the context of MABS. These can be categorised as being either technical contributions or application contributions. The technical contributions are as follows:

1. **Video Analysis Software.** Software, the Video Data Acquisition Software (VADS), for processing video data that incorporates a technique for identifying and tracking moving objects in such data. The proposed software can automatically track moving objects in video data and store captured location information in a database format.
2. **Absolute mechanism for mining movement patterns from video data.** A mechanism for representing agent (object) movements using the concept of "absolute" movement patterns referenced to some environment origin.
3. **Relative mechanism for mining movement patterns from video data.** A mechanism for representing agent (object) movements using the concept of "relative" movement patterns referenced in terms of the neighbourhood of a given agent.
4. **Movement pattern based simulation.** A mechanism whereby movement patterns, extracted from video data, can be used to drive computer simulations.
5. **A mechanism for agents to have a "Memory".** A mechanism whereby agents have "memory" in the sense that they have a planned route they wish to follow. Using this concept agents are also prevented from "pinging" back and forth between two locations in what may be considered to be an unrealistic (unnatural) manner.

6. **A mechanism for evaluating the quality of video data based computer simulations.** A process for comparing real scenarios with computer simulated scenarios by comparing the real video data with “video” extracted from the simulations (a process referred to as “closing the loop”).

The consequent application contributions are then as follows:

1. **A Rodent movement behaviour MABS.** A mechanism whereby behaviourists can study rodent behaviour (movement) in a simulated environment.
2. **A generic MABS mechanism.** A mechanism which can be applied in the context of other behaviour studies such as fire exit simulation and behaviour at rail terminal/station forecourts or airport concourses.

The following methodologies have been used in this research work.

1. **An environment modelling** A mechanism used to represent environments in which the moving objects (agents in a MABS) of interest exist, using a uniform grid representation comprised of cells enumerated in such a way so as to facilitate ready translation across the space. It is acknowledged that grid representations are well established, however, the novel contribution is how such representations have been adopted to represent the environments of interest. Note that, the nature of the proposed environment representation has a strong influence on the nature of the proposed MPS.
2. **State Representation.** A mechanism for representing the relationship between pairs of moving objects (agents) using the concept of states, and an associated *state graph*, to support the selection of MPs. It is to be mentioned, that the concept of states in its self is not novel, however, the representation of states to be able to utilize in MP selection process can be regarded as a novel approach.

1.4 Research Methodology

To achieve the desired research goals (specifically, how best to avoid hand coding for MABS) the adopted research methodology was founded on the idea of considering a sequence of “rodent in a box” scenarios of increasing complexity: (i) Simple scenarios, (ii) Complex scenarios and (iii) Multiple rodents scenarios. In each case the scenario would be run using real rodents and the movement captured by video. The scenarios would be run at the University of Liverpool’s Leehurst Veterinary centre. The basic idea was to suspend a video camera over a box (measuring $1.2m \times 1.2m$) containing one or more rodents and possibly a number of other elements (obstructions, food, nest sites, etc.)

What was clear prior to commencement of the work presented in this thesis was that some kind of Movement Pattern (MP) had to be extractable from the video data and that these patterns could be expressed in a number of ways. Broadly there were two options, absolute and relative, it was conjectured that both would need to be considered so as to formulate an answer to the research question.

The environment was modelled by imposing a grid mechanism (a tile world) over the area in the video where a rodent may move. MPs were considered first in the context of simple scenarios and, once established, in the context of more sophisticated scenarios. In the case of scenarios involving more than one rodent some mechanism was required to capture the relationship between them. The idea of "states" and "state graphs" suggested itself in this context.

Once some initial data had been captured in the form of MPs, the next task was to investigate how this could be incorporated into a MABS framework. The start point was existing work on MABS and specifically rodent behaviour MABS. To simplify the investigation the initial assumption was that agents would have no "memory" (by the term memory we mean a planned route agents would wish to follow). The idea was to add the concept of memory as the work progressed.

The "criteria for success" was that the MABS, whatever form this might take, operated in as realistic a manner as possible. To measure this the intention was to "close the loop" by, in effect, videoing the simulations, processing them using the proposed video analysis mechanism, and then comparing the "real patterns" with the "simulated patterns".

The overriding vision promoted in this thesis is presented by the block diagram given in Figure 1.1 where the directed edges indicate information flow, the rectangles software processes and the cylinders data stores. The figure includes a number of components: (i) video data held in a Database (DB), (ii) video analysis software, (iii) a Movement Pattern (MP) repository for MPs generated by the video analysis software, (iv) a MP translation process where the identified MPs are transformed into a format that can be used with respect to MABS, and (v) a MABS Platform in which the actual simulations can be run and visualised. The figure also includes a link from the MABS platform back to the DB whereby simulations can be evaluated (thus closing the loop). Each of these components will be discussed in further details later in this thesis.

1.5 Publications

This section presents a brief review of the publications arising out of this thesis. Each is listed in turn together with a short description of its contents and the relevance with respect to this thesis.

1. *Muhammad Tufail, Frans Coenen, Jane Hurst and Tintin Mu. "Extracting Movement Behaviour Pattern from Video Data to Drive Multi-Agent Based Simulation", International Workshop on Multi-Agent Systems and Agent-Based Simulation, Springer International Publishing, 2016*

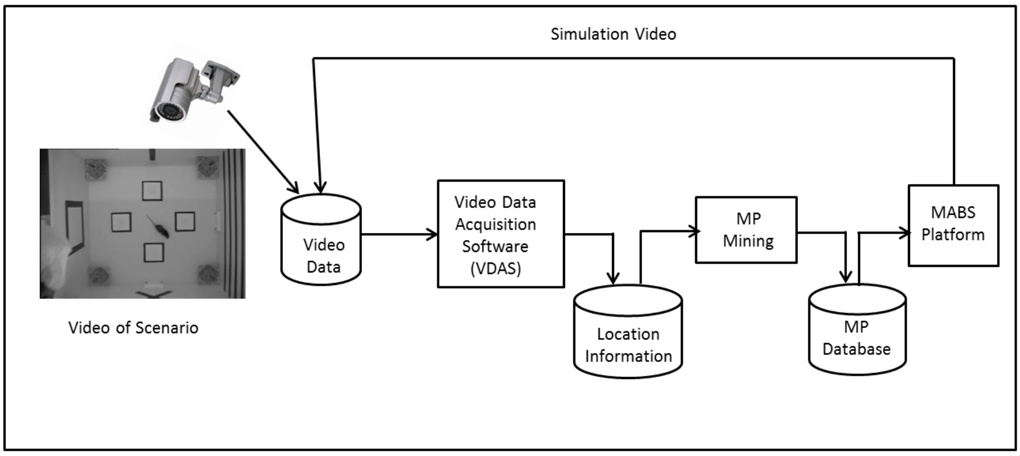


FIGURE 1.1: Rodent behaviour MABS vision

This paper was directed at multiple rodents scenarios whereas the foregoing two papers were directed a single rodent scenarios. An important aspect of the proposed framework presented in this paper was the concept of states. A state defined the relationship between two entities (rodents in our case) in such a way that this could be incorporated into a MABS environment. The concept of "memory" was also introduced in this paper whereby agents could have a planned route and consequently have a "memory". The multiple rodents scenarios considered in this paper are also considered in this thesis. The concepts of states and memory are presented in Chapter 4.

2. *Muhammad Tufail, Frans Coenen, Jane Hurst and Tintin Mu. "Multi Agent Based Simulation Using Movement Patterns Mined from Video Data", Research and Development in Intelligent Systems XXXII, Springer International Publishing, 2016, pages 275–287.*

   This paper followed on from the previous paper and introduced the idea of two categories of movement pattern that needed to be considered: (i) absolute and (ii) relative. The movement patterns presented in this paper had probabilities associated with them which were used to select patterns in a probability driven random manner so as to drive a MABS. When selecting movement patterns only legal patterns could be chosen, those that did not result in a rodent agent passing through or ending up at a location outside of the environment or a blocked location. For evaluation purposes an environment comprised of three separate areas, connected by tunnels, was considered. This introduced a further complication as rodent agents could only move from one area to another using the tunnels connecting the areas. The concept of "divisions" was thus introduced. Evaluation was again conducted by "completing the loop"; the simulations were videoed and these videos were processed in the same way as the original input data. The nature of the identified movement patterns from the simulated data were then compared with the movement patterns generated from the video data. Good levels of comparison were obtained suggesting that realistic simulations were produced using the proposed mechanism. The claimed advantage of relative movement patterns was that they could be used with respect to alternative environments than those from which they were originally extracted. Given that there was no "ground truth" scenario video data available, the claimed advantage was evidenced using an alternative environment as discussed in the case study section in the above paper. The material included in this paper is related to the discussion regarding simple scenarios considered in Chapters 4 and 6 of this thesis.

3. *Muhammad Tufail, Frans Coenen and Tintin Mu. "Mining Movement Patterns from Video Data to Inform Multi-agent Based Simulation", Agents and Data Mining Interaction, Springer International Publishing, LNCS 9145, 2015, pages 38-51.*

   This paper described an approach to mining patterns from video, data which could then be used to support the operation of a mammalian behaviour MABS system. The unique elements of the process described were: (i) the mechanism for representing the videoed scenarios, (ii) the mechanism for capturing and representing movement patterns and (iii) the mechanism for utilising the identified movement patterns in a MABS framework. The operation of the MABS, and consequently the nature of the identified patterns, was evaluated by also applying the process to the simulation data and comparing the frequency of the extracted simulation movement patterns with the frequency of the extracted video data movement patterns. Encouraging "proof of concept" results were produced. The material included in this paper is related to the discussion regarding the simple scenarios considered in Chapters 3 and 6.

1.6 Thesis Structure

This section outlines the structure of the remainder of thesis. Chapter 2 presents the background to the work described with a review of related work along with the application domain. An overview of the proposed environment modelling mechanism and the data acquisition process, for different scenarios (simple scenarios, complex scenarios and two rodents scenarios), is reported on in Chapter 3. The detail of the pattern mining framework, along with the absolute and relative pattern mechanisms, is discussed in Chapter 4. Chapter 5 considers the operation of a MABS framework using the concept of MPs. Chapter 6 reports on the evaluation of the simulations by comparing simulation data with video data. The thesis is concluded with a summary of the main findings, and some suggested fruitful areas for future work, in Chapter 7.

1.7 Summary

In this chapter the research work presented in this thesis has been introduced. The overriding research question and research issues were established, together with their motivation. The main idea of the research presented in this thesis is a mechanism for extracting movement patterns from video data in such a way that they are compatible with a simulation framework, more specifically a MABS framework. The application domain selected as a focus for the work is animal behaviour simulation, especially rodent behaviour simulation. This application was motivated by a desire on behalf of animal (rodent) behaviourists to gain a better understanding of animal behaviour. By using the mechanisms proposed in this thesis it is hoped that rodent simulations can be undertaken that will provide a better understanding of animal movement behaviour. This chapter also discussed the adopted research methodology, including the evaluation of the proposed mechanisms. In the next chapter a review of related work is presented.

Chapter 2

Background and Related work

2.1 Introduction

This chapter introduces the background, related work and application domain to that presented in this thesis. This background work can be categorized according to three established areas of research study as shown in the Figure 2.1: (i) Simulation, (ii) Automated data collection from video data, and (iii) Pattern Mining. The work presented in this thesis sits at the intersection of these three areas.

Each of these three areas are considered in further detail in the remainder of this chapter. Section 2.2 discusses the nature of simulation with a focus on Multi-Agent Based Simulation (MABS). Section 2.3 considers existing work on the automated extraction of knowledge from video, including the process of automatically tracking moving objects in video data. Section 2.4 considers the domain of video mining, the process of extracting useful information from video data; in the context of the work presented in this thesis the main area interest is the extraction of what is referred to as "Movement Patterns" from video data. Section 2.5 then presents a discussion of the application domain that acts as a focus for the work presented in this thesis. The chapter culminates with a summary in Section 2.6

2.2 Simulation

Simulation is concerned with the usage of a model of some real world process or system so as to either: (i) describe and/or analyse the behaviour of this process or system [25, 96, 138], or (ii) so as to gain a better understanding or make predications concerning this process or system [139]. The origins of computer simulation date back to World War II, when Jon Von Neumann and Stanislaw Ulam used simulation to study the behaviour of neutrons as part of the Manhattan Project, the American World War Two effort to develop nuclear weapons [145]. However, in the early days of simulation the models tended to be mathematical, it was not till computers became generally available that computer simulation became wide spread [115], at the same time the concept of simulation became more of a decision support tool than simply a modelling tool.

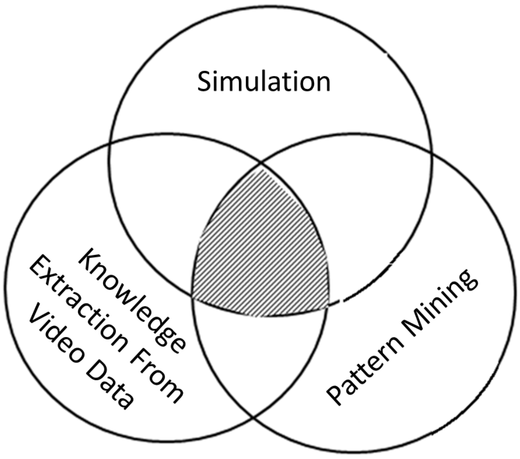


Figure 2.1: The main themes of the thesis: Simulation, Knowledge Extraction From Video Data and Pattern Mining

In this section a review of previous work concerning simulation, with a particular focus on Multi Agent Based Simulation (MABS), is presented. The rest of this section is structured as follows. Sub-section 2.2.1 considers the general advantages offered by simulation. Simulation has been proven to be an effective way of establishing the validity of an idea. Specially in circumstances where either it is not possible to conduct a real life experiment or where it would be costly to do so. Sub-section 2.2.2 then considers a number of application domains where simulation has been fruitfully applied. A particular challenge associated with computer simulation, and one of concern with respect to the work presented in this thesis, is how best to evaluate the operation of a given simulation environment; Sub-section 2.2.3 is therefore directed at ideas concerning the evaluation of simulation frameworks. A common mechanism for realising computer simulations is MABS, Sub-section 2.2.4 is thus directed at the concept of MABS. An issue with respect to MABS is the nature of the framework in which MABS agents (representing the entities within a desired simulation) will operate, this is thus discussed in Sub-section 2.2.5. The section is concluded in Sub-section 2.2.6 with a discussion of data acquisition processes for supporting MABSs.

2.2.1 Advantages of Simulation

In Chapter 1, some general advantages offered by simulation were highlighted; advantages that provide, at least in part, some of the motivation for the work presented in this thesis. In this section these advantages are considered in further details as follows.

1. **Understanding why**: In real world systems certain phenomena occur given a particular set of conditions, where an observer wishes to understand why (or how) this phenomena has occurred. Using real life experimentation it is difficult to reconstruct the particular circumstance, featuring the same set of conditions; whilst simulation, in turn, provides an ideal mechanism whereby this can be achieved.

Simulation allows the reconstruction of scenarios so as to conduct detailed investigations [9, 69].

2. **Exploring possibilities**: Simulations can be run multiple times to explore the effect that the introduction of new policies, theories or methods might have with respect to some domain, without the expense of any real world experimentation [70]. In addition the simulation, once configured, can be run with different parameter sets so as to answer "what if" style questions [9, 83, 96, 133]. In recent times it has become common place to use simulation as a scientific tool for exploring and querying the behaviour of real world systems [44].
3. **Compressing and expanding time**: Simulation allows for the compression and/or expansion of time, hence scenarios can be speeded up or slowed down to support scenario analysis, something not possible in the case of real world experimentation [9, 110, 128].
4. **Cost effectiveness**. Real world experimentation is resource intensive. Simulations allows a system to be examined without the need for the extensive resource required for real world experimentation [119].
5. **Safety**: Simulation allows the investigation of scenarios which may be life threatening and could not normally be investigated otherwise, such as the simulations conducted with respect to the Manhattan Project alluded to above. Simulations provide a safe mechanism for conducting what might otherwise be hazardous experiments [64].
6. **Non Intrusive**: Simulation is non-intrusive [110]; it does not require subjects to be fitted with special devices or implants, or to be marked in some way. For example, in the context of rodent behaviourology, in the 1960s, it was common to conduct real life experiments with groups of rodents and to use "toe clipping" so as to differentiate between subjects, a process that would be unlikely to receive effical approval in modern times.

The disadvantages associated with simulation include:

1. **Simplification**: Due to the complexity of the real world situations that we wish to simulate, simulations typically involve approximations, they therefore may not be as accurate as we might like. Although, it is argued in [139] that the main idea of simulation is to abstract and simplify some real world scenario and that this abstraction and simplification is a positive aspect of simulation.
2. **Simulator development**: A simulation is only as good as the model/mechanism on which it is founded. Substantial resource is frequently required to build a simulation of any degree of sophistication, and even then it still remains, as noted above, an approximation of the real world situation which it is intended to model.

3. **Simulation performance evaluation**: A further disadvantage of simulation is that it is often difficult to be certain that the simulation is working correctly. This is especially the case in the context of simulated experiments that have no direct parallel in real life. Simulation evaluation is considered in further detail in Sub-section 2.2.3.

From the foregoing it is clear that simulation offers significant advantages. Simulations are conducted so as to validate an idea or a proposed solution to a problem, without the need for expensive field trials, filed trials which in many cases are impractical. Accurate simulation implies an accurate understanding of the operational effectiveness of a proposed idea/solution. The most significant disadvantage is the resource (knowledge) required to build a simulator in the first place. This thesis seeks to address this issue by automating the data acquisition process using video data analysis so as to provide for accurate simulations.

2.2.2 Application of Simulation

Given the above, the advantages of simulation are clear. As a consequence, simulation has been widely adopted with respect to many application domains. Some common application domain examples, where computer simulation has been fruitfully applied, are as follows:

1. **Manufacturing**: Manufacturing is the process of making a finished product from raw material using some industrial process [9]. In the context of manufacturing, simulation has been used for simulating: manufacturing processes [12], material transportation [73] and material handling [8, 147].
2. **Automobile Industry**: Simulation has been used in the design and operation of car and truck assembly plants [47, 55] as well as automotive component manufacturing plants [57, 74]. Most of automotive manufacturing, worldwide, currently requires that the operation of any new or modified manufacturing system/process is verified using simulation before the system/process can be approved for installation [9].
3. **Health Care**: In the context of health care, simulation has been particularly directed at surgical technical and procedural skills [48, 81, 131, 132]. For example with respect to: obstetrics [87, 121], cardiology [29, 160], anaesthesia [49, 134], critical care [61, 88] and emergency medicine [127, 143].

The above examples give a flavour of the range of domains where simulation has been applied. There are many more examples. The application domain of interest with respect to the work presented in this thesis is rodent behaviour simulation. An example of previous work where the concept of simulation has been applied to this application domain can be found in [2, 3], where it is proposed that rodent behaviour simulation be conducted using what is referred to as a “behaviour graph”. A behaviour graph comprised of vertices representing states (behaviours) and edges indicating possible state changes (behaviour changes), where state changes occur as a result of an agent (rodent) completing some self appointed task or as a result of some external event. The significance of the work in [2, 3] is that MABS is used as also used with respect to the work presented in this thesis. The concept of MABS is discussed further in Sub-section 2.2.4 below.

2.2.3 Simulation Evaluation

As noted above, one of the further challenges of simulation, in addition to the resource required to construct a simulation, is how to evaluate the performance of a given simulation. The guideline is that any simulation should be as accurate as possible, the question is how to measure this. It is generally acknowledged that this should be done using some form of verification [9, 21, 126]. The term verification in this context means conducting comparisons between simulated outcomes and real world outcomes in the context of identical scenarios [126]. The simulation mechanism proposed in this thesis was evaluated in this manner, that is, the movement behaviours of rodents in simulated contexts was compared with the movement of rodents in parallel real world scenarios. As already noted simulation evaluation becomes more challenging where there is no real world parallel (for example because it would be to dangerous or resource intensive to conduct a similar real world experiment); however, this situation is outside of the scope of this thesis.

2.2.4 Multi Agent Based Simulation (MABS)

From the literature we can identify a number of different kinds of computer simulation, examples include: (i) Equation based simulation, (ii) Monte Carlo Simulation, (iii) Discrete event based simulation and (iv) MABS. The focus for the work presented in this thesis is Multi Agent Based Simulation. MABS is a popular choice for simulation in the case of applications that feature many individuals that operate in an autonomous/independent manner in that each individual can be represented as an agent [33, 168]. A MABS thus consists of a set of interacting agents designed to reproduce the dynamics of some environment to be simulated [130]. In this context an agent is a software entity that behaves in an autonomous manner [92, 167, 168]. Here autonomy means that each agent has control over its own actions and is able to act without external intervention [139].

The main characteristics of a software agent are as follows [139]:

1. **Responsive behaviour**: The agent is able to responds to changes in its environment, in a timely fashion, by monitoring its environment.
2. **Pro-active behaviour**: The agent should be able to demonstrate goal directed behaviour.

3. **Social behaviour**: The agent should be able to interact with other agents and the environment in which it operates.

4. **Flexible behaviour**: The agent should have a range of ways to achieve a given goal and be able to recover from failure.

These are all characteristics that we would like the entities modelled in simulations that feature people or animals, to have.

MABS has been widely applied. Examples can be found in: (i) manufacturing [136], (ii) traffic simulation [58], (iii) financial market analysis [68, 84, 125], (iv) software development [165], (v) agricultural land usage coupled with water management strategies [51], (vi) social networks [24, 67], (vii) crowd dynamics [10, 13, 65], (viii) group learning [144] and (ix) the analysis of police patrol routes [99]. It is also worth noting that there has been a MABS workshop at every Autonomous Agent and Multi Agent System (AAMAS) conference since 1998.

To realise a MABS there are a number of popular mechanisms that can be used with respect to the agents that feature in such simulations. These mechanisms are considered in further detail in Subsection 2.2.5 below. The main challenge of MABS, as in the case of computer simulation in general, is the acquisition of the knowledge required to build the simulators. In [1], this was done by hand and, although good simulations resulted, this approach was found to be very resource intensive. The fundamental idea presented in this thesis is that the necessary knowledge can be derived from video data through a video data analysis process. More specifically the idea is to extract "movement patterns" from video data and utilise these patterns to drive a MABS. Processes for acquiring the knowledge needed to "populate" a MABS are discussed further in sub-section 2.2.6 below.

2.2.5 Mechanisms for Representing Agents in MABS

From the literature we can identify a variety of methods for the realisation of MABS. The most prominent include: (i) Individual Based Modelling (IBM), (ii) Cellular Automaton and (iii) the Belief Desire Intention (BDI) framework. Each is discussed in further details in the remainder of this sub-section.

In IBM the behaviour of agents is individually coded [92]. For example in [78, 139] agent behaviour is specified using a rule-like (declarative) language; decision making is then realized through a process of logical deduction. In [20] IBM was used to decompose complex behaviours into many simple behaviour modules organized into layers. Example applications where IBM has been used to realise MABS include crowd behaviour simulations [65, 116, 148] and animal movement behaviour studies [41, 64, 123]. More specifically, in [41] IBM was used to model animal (movement) behaviour in a pastoral system. In [64] IBM was used to model ant behaviour when choosing nest sites. In [123] the movement behaviour of sheep grazing in a field was simulated using IBM. The advantage of IBM is that it is very easy to implement [63] and produces effective simulation, although on occasion resulting in over simplifications [139]. A further disadvantage of IBM is that it lacks any formal notion of a framework for the simulation, without which some behaviours (concepts) have the potential to be missed.

Cellular Automaton is an alternative approach to the realisation of MABS [139]. The most commonly used cellular automation mechanism is the Finite State Machine (FSM) or some variation of this mechanism. A cellular automaton comprises a set of cells, where each cell represents a *state* [63, 71]. The concept of cellular automaton was first proposed by John Von Neumann in 1940s, when he started to model multiple interacting agents [91, 107]. Generally speaking, cellular automaton is used to describe all possible states an agent can adopt and the triggers for state changes. The central state for a cellular automaton is typically the *idle state*, which is linked, directly or indirectly, to all other states. Examples where cellular automation has been used in the context of MABS can be found in [45, 129]. Cellular automaton, unlike IBM, provides a holistic vision of the operation of a MABS. Additional states and state changes can easily be incorporated, hence it also offers flexibility. Cellular automaton can be readily used to implement large simulations, although as such systems increase in size it becomes increasingly difficult to manage and understand the simulations [116].

According to [15], the BDI concept can be traced back to Aristotle's analysis of how humans (and animals) decide actions. However, in terms of computer simulation, the BDI model can be said to have its origins in practical reasoning [53, 139]. Using the BDI model, and with respect to MABS, each agent has a set of beliefs concerning its *information state.* Desires then represent the goals (objectives), which each agent wishes to attain, whilst the intentions represent an agent's current goals. From the literature, there are many examples where the BDI model has been used in the context of MABS [109, 113, 137, 166]. The BDI model is particularly suited to human behaviour simulation [72, 151, 164]. There are however certain limitations of the BDI model as noted in [35, 53, 144, 151]; namely that the BDI model lacks randomness, it is prescriptive, while randomness is an important aspect of many MABS including rodent behaviour simulation. The BDI model assumes desires to be logically consistent, however this is not usually the case in real world representations of human or animal behaviour [36]. Although it is possible to contrive a level of randomness in the BDI model this is not ideal and it can also then be argued that the incorporation of randomness means that we no longer have a BDI model. For this reason the BDI model was considered unsuitable with respect to the research objectives that this thesis intends to address.

For the purpose of the work presented in this thesis, and given the above, the IBM approach to MABS realisation was adopted because of its simplicity. Note that the main objective of the work presented in this thesis is the idea of extracting data from video to populate animal behaviour MABS, the particular MABS framework to be used is not of primary concern; indeed, it is suggested that the proposed video data simulation population mechanism has corresponding applicability in the context of alternative simuation models.

2.2.6 Data Acquisition

Whatever MABS architecture that is selected, the most challenging task (as noted earlier) is the acquisition of the knowledge (data) required to populate the adopted model [64, 123, 139, 146, 149]. From the literature different methods have been propose concerning this knowledge acquisition process. The usually process is one of observation or interviews with domain experts. For example in [123] a set of behavioural rules were derived through a process of observation, and in [146] surveys of individuals were conducted. In [139] the following observations were made concerning the manual processes of MABS data acquisition:

1. It is difficult to translate data from domain experts (or from observations) into agent's behaviours since it requires extensive domain knowledge.
2. Hypotheses for agent's behaviour based on the domain experts (or derived from laboratory experiments) needs to be very precise.
3. The manual collection of data from real world scenarios, regardless of how this is done, is a time consuming and resource intensive process.

The most significant of the above is that manual data acquisition is a resource intensive process and thus some form of automation is desirable. In [146] the authors briefly discuss the potential for automatically extracting the required data from existing records (documents); in this thesis it is suggested that the automated extraction of knowledge from video data is the solution. Recall that the desire to automate the knowledge acquisition process is also the main motivation for the work presented in this thesis.

2.3 Knowledge Extraction From Video Data

This section is concerned with previous work directed at the automated extraction of information (knowledge) from video data. There are two specific challenges to be overcome:

1. **Object Detection**: The initial identification of the objects of interest.
2. **Tracking**: The continuous tracking of the objects of interest, as the video progresses, so that movement can be recorded.

This section is structured as follows. Sub-section 2.3.1 considers the nature of video data, this is followed in Sub-section 2.3.2 with a discussion of how to detect moving objects (rodents in our case) in video data, the first of the above challenges. Sub-section 2.3.3 is then concerned with the tracking of objects once they have been detected, the second of the above challenges. This is followed by Sub-section 2.3.4 which highlights relevant associated work concerned with rodent detection and tracking in video data.

2.3.1 Video Data

Video data can be conceptualised as comprising a sequence of images, called *frames*, each comprised of a collection of pixels[1]. Image quality depends on many factors, such as: colour accuracy, noise, contrast, distortion and resolution. The latter is a measure of how many pixels an image contains. The size of an image is frequently defined in terms of its width and height measured in the number pixels. If terms of video data a frame (image) typically measures 640×480; this is referred as *Standard Definition* (SD) video data. Frames measuring 1280×720 describe *High Definition* (HD) video data; while frames measuring 1920×1080 are referred as *Full HD*. With respect to the video data used for experimental purposes in this thesis, HD video data was used. Typically we have 25 frames per second.

Videos data is stored in compressed format, so as to reduce the overall file size. A codec (short for Compression and Decompression) is a proforma for: compressing and decompressing video data, interpreting video data and displaying video data on a computer screen. Video data formats can be conceptualised as containers that include the compressed visual and audio streams, the codec and sometimes additional files. The container is what describes the whole structure of the video data. There are a variety of formats in which video can be digitally stored. Common formats include (presented in chronological order):

1. **Audio Video Interleave (.avi)**: The AVI format was developed by Microsoft and was published in November 1992. It can contain both audio and video data. AVI files are able to run under different operating systems like Windows, Macintosh and Linux, and are also supported by popular web browsers. The AVI format can incorporate various forms of compression. The AVI format has been replaced by Microsoft's Windows Media Video (WMV) format, see below, but its significance is that it was the first general use video format.

2. **MP4 or MPEG-4 Format (.mp4)**: The MP4 format was first introduced in 1998. It was developed by the Motion Pictures Expert Group (MPEG) who were responsible for setting industry standards regarding digital audio and video data. The MP4 format is commonly used for sharing video files on the Web. The MP4 video format uses separate compression mechanisms for the audio and video data. The advantage offered is that MP4 file sizes are relatively small while the quality remains high.

---

[1] An image can be conceptualised as an 2D array, or matrix, of pixels (picture element) arranged in rows and columns.

3. **Windows Media Video Format (.wmv)**: The WMV format was developed by Microsoft and was originally designed for internet streaming applications. However, it can cater to more specialized content. Windows Media is a common format for Web usage, but cannot be played on non-Windows computers. Being a Microsoft product, the Windows Media Player is the main application that is used to play WMV files on all Microsoft machines, but there are also WMV players available for free for the Macintosh operating system.

4. **Quick Time Format (.mov)**: The Quick Time format was developed by Apple and is anther common video interchange format. The Quick Time format is compatible with both Windows and Mac platforms, and can be played on a number of types of player (such as Apple's Quick Time player).

5. **Real Media Format (.rm)**: Real Media is a format which was created by Real Networks and contains both audio and video data. The Real Media format is typically used for streaming over the internet. Real Media is compatible with both Mac and Windows platforms. Real player is the most frequently used player to play videos in the Real Media Format.

The format used with respect to the video analysis software presented later in this thesis was MP4.

2.3.2 Object Detection

Given that video data is essentially a sequence of images, it seems natural to address the challenges of object identification and tracking using techniques taken from image processing. Image processing is concerned with manipulating image data with a view to some potential application dependent end goal. Common example applications include image categorisation and image retrieval. Image categorization is concerned with the labelling of images in (say) the context of image banks. Image retrieval is concerned with extracting images from image banks according to some criteria. Image processing is also linked with representing images in an appropriate manner so as to support image categorisation and retrieval; this sometimes involves image segmentation so as to identify particular regions/objects in images.

Object detection is the first step for many video processing (and computer vision) applications including video surveillance and event detection [117]. Object detection is conducted firstly to establish the existence of an object of interest in video data, and secondly to establish the location of such an object. Note that object tracking also requires object detection; object detection is a necessary pre-cursor to object tracking. Object detection can either be conducted with respect to a single frame or as an aggregate with respect to a sequence of frames (the latter so as to reduce the number of false detections).

Video objects, once detected, must be represented in some manner. Objects can be represented by their shape and/or appearance. A variety of different object representations have been proposed. Note that the nature of the adopted object representation influences the nature of the object detection. A number of popular representations are summarised below:

1. As points [153], either as a singe centre point (centroid) or a set of points describing some region in an image representative of the object of interest.
2. As a primitive geometric shape such as a rectangle or an ellipse [28].
3. According to the objects "silhouette", or contour, as defined by the boundary of the object (the region inside the boundary is the silhouette of an object). This type of representation is suitable for complex-non-rigid shapes [171].
4. Using "articulated shape" representations whereby the object is decomposed into "body parts" that are held together with "joints" in the same way that the human body is an articulated object with legs, arms, head and so on.
5. Using what is known as the Binary Large OBject (BLOB) format, which basically defines a group of connected pixels describing what is known as a "binary object".

The representation adopted with respect to the work presented in this thesis was the BLOB representation. According to [104] a BLOB is a region of an image in which some properties are constant or approximately constant; all the points (pixels) in a BLOB can be considered in some sense to be similar to each other. The similarities can be defined in terms of: (i) coherent flow of motion (of the pixels) [77], (ii) colour [62] or (iii) both [19]. Gray scale colour was used with respect to the work presented later in this thesis. The characteristics of BLOBs are presented in [18]. These characteristics in turn can be used to distinguish between BLOBs. These characteristics are as follows:

1. **Area**: The number of pixels included in a BLOB; thus the area of a blob is the number of pixels that the BLOB consist of. This is a useful feature of BLOBs and it is often used to remove BLOBs that are too small or too big for consideration. For example, for detecting people in an image it can be anticipated that the BLOBs will have a certain maximum and minimum size.
2. **Bounding box**: The minimum bounding rectangle (box) that contains a BLOB. The edges of a minimum bounding box can be found by identifying the four pixels with minimum x, minimum y, maximum x and maximum y values. Using these four values, the width and height of the rectangle can be computed. The bounding box idea can be used to define a Region Of Interest (ROI) as sometimes used with respect to the process of object detection and tracking [18] as discussed further in Sub-section 2.3.3 below.

3. **Bounding box ratio**: Defined as the height of the bounding box divided by the width, thus indicating the elongation of the BLOB. The bounding box ratio thus tells something about the shape of a BLOB.

4. **Centre of mass**: The centre of mass of a physical object is the location on the object where you should place your finger in order to balance the object. The centre of mass for a BLOB is a similar concept. It is the average x and y positions of pixels that comprises a given BLOB. According to [18] it is defined as a point, whose x-value is calculated by summing the x-coordinates of all pixels in the BLOB and then dividing by the total number of pixels ((2.1)). Similarly for the y-value, as shown in Equation (2.1). In the equation, N is the number of pixels in the BLOB and x_c and y_c are the coordinates of the centre of mass, while x_i and y_i are the x and y coordinates of the ith pixel.

$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad y_c = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{2.1}$$

As noted earlier in this section, video data can be simply viewed as a sequence of images. During object detection, assuming only one object, each image (frame) can be decomposed into two parts. One part comprising those pixels that correspond to the object of interest and another part that does not correspond to the object of interest. We refer to the object of interest as the *foreground object* (or simply the *foreground*) and the remaining pixels as the background object (or simply the *background*). With respect to tracking applications the foreground comprises moving objects such as cars, ships and people. There are a number of (foreground) object detection methods that can be adopted [104, 117, 170], some related to a specific object representations (see above). Examples include: frame differencing, optical flow and background subtraction. Each of these is briefly described below.

1. In frame differencing the foreground is determined by calculating the difference between two consecutive images by simply subtracting the current frame from the previous frame in a pixel by pixel manner [104]. An alternative is to use three consecutive images instead of two [79].

2. Optical flow is used to describe the motion of a coherent set of pixels (the foreground) between sequences of frame pairs. Examples can be found in [19, 56, 169]. In [19] each foreground pixel was considered independently according to its optical flow between consecutive frame pairs, if the motion was coherent (the same) the pixel is grouped with other similar pixels into a “blob” of pixels that all have a coherent motion. In [56] the edge features of the object of interest were used to determine the flow by considering the length of the edges in consecutive frames. In [169] the motion of parts of the human body were used to identify a coherent, human body, set of pixels. This was done by calculating the optical flow of

the centroid pixel of the individual parts (making up the human body foreground object).

3. Background subtraction is another technique used for the extraction of image foreground objects. In background subtraction [16, 17, 66] object detection is achieved by building a representation of the environment which does not feature the object(s) of interest, called the *background model*, and then finding any changes (deviation) from this model for each incoming frame; any significant deviation is then indicative of an object (the foreground). In order to avoid false foreground detection, resulting from noise and camera jitter, a thresholding technique can be adopted. However, such a threshold is difficult to calculate theoretically [135]; therefore it is usually determined after sampling a number of frames and basing it on the detected maximum pixel difference (even so false foreground detection cannot be entirely avoided). If $B(X, Y)$ is a background model and $F(X, Y)$ is a single frame, where X and Y are the row and column IDs of the pixels, then the difference image $\mathrm{D}(X, Y)$ can be calculated and thresholded using Equation (2.2) [135]. All pixels of the difference image that have value 1 belong to an object, whereas the pixels with value 0 belong to the background and are ignored.

   $$D(x, y) = \begin{cases} 1 & \text{if } |F(x, y) - B(x, y)| \leq \textit{threshold} \\ 0 & \text{if } |F(x, y) - B(x, y)| > \textit{threshold} \end{cases} \tag{2.2}$$

   Background subtraction is a widely used approach for detecting moving objects in video data generated using static cameras. As noted above, background subtraction uses the characteristics of individual pixels or groups of pixels to extract the foreground (moving object) from the background. These characteristics are typically founded on colours and/or edges [17, 120]. In image processing a background model (a frame comprised of pixels) is calculated from n previous frames, where each pixel represents the average value of the previous n frames. Thus the situation, for example, where a sudden change in light effects temporarily the intensity value of a pixel is accounted for.

With respect to the work presented in this thesis the adopted mechanism for object detection (and consequent tracking) was background subtraction. This was chosen because it was easy and simple to implement. As noted above the objects of interest were represented using the BLOB representation described earlier in this sub-section. It is these “blobs” that are further processed for tracking purposes as discussed in the following Sub-section.

2.3.3 Object Tracking

This Sub-section discusses the tracking of a moving object in video data once it has been detected. The issue here is that there might be more than one moving objects in the given video data to be tracked. In computer vision the detection of moving objects and motion based tracking is very important. A second issue is that real time video object tracking is resource intensive. Real time video object tracking is important with respect to many applications such as: activity recognition, traffic monitoring and surveillance, and has become more tractable with the advent of high performance computing capabilities. However, in the context of the work presented in this thesis real time object tracking is not a requirement.

If we have a series of points (locations) describing a moving object, and connect these points, we will have a curve through time. This curve is referred to as the *trajectory* of the object. Video object tracking can thus be defined as the process of plotting the trajectory of an object over a sequence of images [50, 170]. In [104, 112] tracking is defined as the process of following a set of objects that are moving, and assigning a consistent set of labels to the tracked objects in each video frame.

Given the above the process of object tracking can be expressed as follows:

1. Assign IDs to the detected objects (detected using one of the mechanism discussed in Sub-section 2.3.2).
2. Predict each object's future position.
3. In the next frame, match the detected object positions to the predicted positions, using (say) a best matching mechanism, and assigning each previous ID to the most appropriate newly detected object.
4. Repeating from step 2 for the next frame, and so on.

The tracking of objects in video data is complex because of a variety of issues:

1. Loss of information.
2. Image noise.
3. Complex object motion.
4. In some cases the articulated nature of objects.
5. Partial and full object occlusions.

Note that the tracking process is further complicated if existing objects leave the environment and/or new objects arrive into the environment during the time span covered by the video data.

The tracking process can be simplified by applying some constraints on the motion and appearance of the potential objects of interest. For example it could be assumed that the motion of an object should be smooth and with no abrupt changes, or that an object's motion would be with constant velocity. Alternatively, previous information concerning a moving object can be used to simplify the tracking process. For example using the number and size of objects and/or the shape and appearance of objects.

In the context of the above the rest of this section is structured as follows. Mechanisms for the prediction of the future position of an object are first considered and then solutions to the issues associated with object tracking (as itemised above) are discussed.

2.3.3.1 Prediction of Object Next Locations

This Sub-section considers the mechanisms whereby the "next" location of an object can be predicated given a sequence of video frames, by considering the previous locations for the object. Given a single object moving across a collection of video frames, in the absence of noise, this is straight forward. We find the location of the object in the "next" video frame as described above and this is then the next location. In the presence of noise there may be several candidate next locations hence one needs to be selected as the follow on location. Given a situation where multiple objects are being tracked the task becomes more complex, increasing in complexity as the number of objects increases. A further challenge is where objects are occluded or are temporarily lost for technical reasons.

Solutions to the above are typically founded on the usage of object trajectories which are used to predict next locations. The trajectory of objects is maintained as part of the tracking process. Given a set of candidate next locations, the location that is most similar to the predicted location is typically selected. For example by computing the differences, Δx and Δy, between the coordinates of the predicted next location and the candidate next locations and choosing the next location that features the smallest differences. Thus using this technique next locations can be identified.

The whole next location identification process can be made more efficient in terms of processing time, by defining a Region Of Interest (ROI) around the predicted object next location [103, 135] and only considering pixels within this ROI. The size of ROI is defined based on the uncertainty of the predication. The bigger the uncertainty the bigger the ROI should be.

A variety of techniques can be used for prediction purposes. One technique frequently used for predicting the next location of an object is the Kalman filter technique [50, 135, 140]. Kalman filtering uses a series of measurements obtained over time. It follows an iterative process comprised of two steps: (i) the prediction step, and (ii) the update step. In the prediction step the Kalman filter process generates estimates of the current state along with their uncertainties. In the update step the next location is predicted by considering the previous and present locations along with their uncertainty bounds.

There are some advantages and disadvantages, noted in [7, 50, 117, 170], associated with Kalman filtering. The advantages are: (i) it tends to give optimal solutions, (ii) it is efficient in terms of total time elapsed for processing certain frames and (iii) it is able to handle noise [1,2]. The shortcomings are: (i) it is applicable only to single object tracking and (ii) the assumption that the next locations are normally distributed

(Gaussian) which means that Kalman filters will give poor estimations of next locations with respect to applications that do not follow the Gaussian distribution [170]. The limitations of Kalman filtering can be addressed by using particle filtering. Particle filter is efficient for non-linear and non-Gaussian applications and can be used for tracking more than one object. Similar to Kalman filtering, particle filtering also consists of two steps i.e. prediction and update [114]. Particle filtering predicts the next location of a moving object by considering different points of a moving object. These points are known as particles (or samples). Each particle incorporates tests whether how it is likely that the object is at the position where the particle of moving object is. After the particles have been evaluated, weights are assigned to these particles according to how good the particles are. Those particles, predict accurate location of moving object are considered as good, otherwise are bad. Then the good particles are considered while the bad particles are removed through a re-sampling process. The next particle generation then predicts where the object might be. Then this generation is evaluated, and the cycle repeats as long as tracking process is continued. The work presented in this thesis uses the particle filtering method for tracking.

2.3.3.2 Multiple Object Tracking

As noted above, the challenge of object tracking is compounded in the presence of multiple objects. The work presented in this thesis is concerned with the tracking of rodents in video data; there may be more than one rodent in the video data, hence we are interested in multiple object tracking. If the objects being tracked are all significantly different (the attributes of the objects, such as size and colour, are all significantly different) there is little chance of confusion. For example people with different colours of clothing or different coloured cars. Where the objects to be tracked are not significantly different, as in the case of the mouse/rat simulations of interest with respect to work presented in this thesis, the tracking must be conducted in such manner that objects are differentiated throughout the tracking process. For some applications, such as surveillance monitoring, this is very important; in the case of the mouse/rat simulation of interest here this is still important although useful information can still be obtained even though the trajectory of an object is occasionally confused with that of another.

In [103, 170] three multiple object tracking issues were identified as follows:

1. **Split and Merge**: This is where two objects being tracked “merge” together as a single blob for a period of time and then separate again (“split”). The issue is then how to determine which is which. A variation of this is where at commencement of the tracking process a single blob is identified which, after a period of time splits into two. In other words where two blobs have been travelling together as one “merged” blob which then diverge (“split”). The later is less of a challenge as we only need to generate a new ID and attach it to one of the objects while maintaining the original ID for the other object. The split and merge phenomena, both variations, is illustrated in Figure 2.2 (after [103]).

2. **New, lost and found of an object**: This is where, as tracking progresses, a "new" object is detected. If all other objects are still present this is not so much of an issue. What is more challenging is where one (or more) of the original objects being tracked disappears (is lost) and after some time a new object is detected; can the new object be linked to the lost object? An alternative is where an object is not so much lost but the tracking becomes erratic due to noise. The new, lost and found phenomena, all three variations, are illustrated in Figure 2.3 (again, after [103]). Note that with respect to the rodent video used for evaluation purposes with respect to the work presented in this thesis, the scenarios did not feature objects that appeared once tracking had commenced.

3. **Occlusion**: Occlusion occurs when: (i) one part of an object obscures another part of the same object (this is an issue when tracking articulated objects, for example humans, and thus not of relevance with respect to the work presented in this thesis), or (ii) two objects obscure each other, or (iii) an object is obscured by some static feature in the environment. The first is called *self occlusion*, the second *inter object occlusion*. Note that the second is essentially the merge and split phenomena.

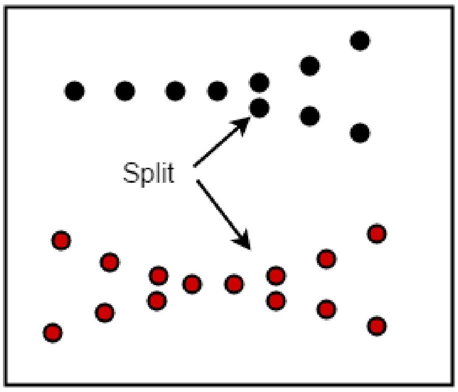


FIGURE 2.2: Illustration of the split and merge phenomena

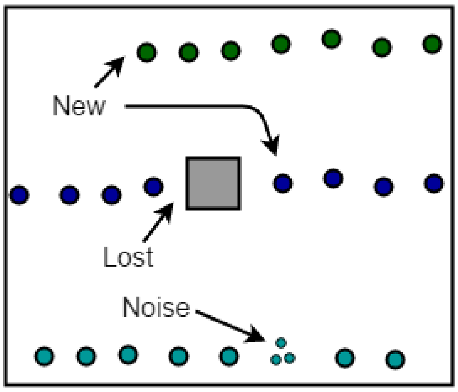


FIGURE 2.3: Illustration of new object, lost and found objects and noise

With respect to the lost and found issue, in the case of single object tracking, it is natural to assume that when the object is "refound" this is the same object (of course this may not always be the case). However, in the case of multiple object tracking, recommencing the tracking is not so straight forward. Although in the context of the rodent tracking scenarios considered with respect to the work reported on in this thesis, the importance of correctly matching objects is less critical than in the case of (say) surveillance applications.

Various mechanisms have been proposed to address the merge and split and occlusion issues identified above [39, 43, 59, 93, 135]. For example in the case of inter object occlusion [93] and [43] propose the usage of knowledge concerning the appearance and position of the objects to resolve the issue. In [59], the idea presented is to use the "silhouettes" of the objects of interest. In the case of [39], what is known as the *optical flow* is used for occlusion resolution and in [135] a "search rectangle" is used. Of course, as noted in [170], the probability of occlusion can be reduced by installing cameras at appropriate positions; however in many cases, such as in the case of the mouse tracking application of interest with respect to this thesis, this is not a practical option (this is discussed further in Chapter 3). In [170] it is noted that *oblique view* cameras are likely to encounter multiple object occlusions. If cameras are installed on airborne devices, so when a birds eye view of a scene is obtained, occlusions between objects on the ground are less likely to occur.

With respect to the research work presented in this thesis, a built-in library was utilized, namely EmguCV[2] consisting of BlobTracking procedure, based on Particle filtering, to offer an effective object tracking. In the case of issues related to splitting and merging, and new lost and found objects, the tracking process was simply continued and the issues resolved "off line".

2.3.4 Rodent Detection and Tracking in Video Data

Following on from the above, this Sub-section considers previous work on rodent tracking in video data, the domain of interest with respect to this thesis. Note that the previous work considered in this Sub-section is concerned with the tracking of rodents in video data and extracting information concerning rodent movement, and not previous work directed at behaviour study (this last is considered in Sub-section 2.3.3). From the literature, reference to rodent tracking can be found in [54, 75, 101, 111, 163]. In [111] a method is described for the detection of mice in video data by marking the back fur of mice with bleach. These "fur patterns" are then used for tracking purpose in the video. However, it is argued that fur marking with bleach may cause the mouse to behave in an unnatural manner [54], it may also be harmful. From a technical perspective, it is also not a full-proof solution as the posture and location of a mouse has the effect of contorting the pattern, which in turn may lead to misidentification.

In [163] the authors describe work directed at automatically detecting and tracking mice in semi-natural environments. The system proposed utilises radio-frequency technology for the identification and tracking of mice; consequently it is referred to as a semi-natural environment. Individual mice were tagged using microchips. Using a radio frequency decoder, the proposed system automatically captures the x-y coordinates associated with individual mice with reference to a prescribed origin in a given enviroment. As a result, what the authors refer to as "behavioural profiles" of individual mice and groups of mice, are captured. A criticism directed at the approach is that the presence of radio frequencies within the environment may affect the normal behaviour of mice, and may also have some side effects on mice nervous systems.

---

[2] EmguCV is a cross platform .Net wrapper for the OpenCV (Open source Computer Vission library), EmguCV allows OpenCV functions to be called from .Net compatible languages such as C#, VB, VC++.

In [75] a "trainable" general purpose automated system is described for the mining of video data with respect to the behavioural analysis of mice in cages. Here a computer software system is trained with labelled example behaviours of interest. The trained system is then used to label behaviours in previously unseen video data. The authors claim good results, especially for mice, compared with behaviour classification by humans. The fact that the system is trained with manually annotated behaviours is a disadvantage as this will be: resource intensive, require recourse to human exerts and be error prone.

In [101] computer vision software was used to analyse Audio Video Interleave (AVI) files to capture the behaviour of mice with respect to a particular challenge known as the Morris Water Maze[3]. The system accepts input video in AVI format and extracts the following parameters: (i) elapsed time till discovery of the platform, (ii) average velocity of mice and (iii) total distance of mice, to support behavioural studies in mice. The authors claim that these parameters are well suited to the behavioural analysis of mice in the context of the Morris Water Maze. However, although such parameters could be used to define some sort of MP, as envisioned in this thesis, it would be not sufficient to support effective simulation.

In [54] a software system called "Mice Profiler" is introduced directed at mice detection and tracking. The system uses geometrical primitives to model and track two mice. The tracking data is then used to extract information concerning the position, orientation, distance and speed of each mouse. The software is used to analyse behavioural states and provides a temporal evaluation. However, the software does not automatically identify the mice in the video, instead, the user is required to mark the mice (in the video) using pairs of virtual marks, a head mark and a body mark, so that the orientation of the individual mice can be discerned. A further disadvantage is that if the software loses the location of a mouse it must be stopped and re-started.

In [54] a software system is proposed to track the position of mice from video using the BLOB representation (see Sub-section 2.3.2) to automatically characterise the social and non-social behaviour of mice. BLOBs are detected using the foreground-background comparison mechanism described previously in Sub-section 2.3.2. Tracking is done using the Particle filtering mechanism described previously in Sub-section 2.3.3.1.

With respect to the work presented in this thesis, rodent tracking and location extraction were conducted using the background subtraction mechanism for object detection as described in Sub-section 2.3.2 and tracking using Particle filtering as described in Sub-section 2.3.3.1. It should also be noted here that none of the above described approaches was directed at knowledge acquisition for rodent simulation.

---

[3] This is a recognised task for studying rodent learning where a rodent is required to find a submerged platform.

2.4 Movement Pattern and Video Mining

A central theme of the work presented in this thesis is the idea of movement pattern mining. This section therefore presents a review of movement patterns mining in various contexts together with a review of previous work concerned with video data mining.

2.4.1 Mining Movement Patterns

Movement can be defined as a change of position (location) of an object over time [5, 40]; moving objects are then entities whose position changes over time. As noted above, the set of points representing a moving objects describe a trajectories or path through space and time. These trajectories feature patterns, Movement Patterns (MPs), which can be extracted (mined) from some data source [40, 100]. MPs can be related to many kinds of object such as humans, animals and vehicles, although in this thesis we are interested in rodents. MPs can consequently be used for all kinds of analysis purposes, for example in [22] bus schedules are considered (see also [85]). In the case of the work presented in this thesis we wish to use MPs to drive rodent behaviour MABS.

MPs thus define a recognizable spatial-temporal relationship with respect to the trajectory associated with a moving object. As such, MPs, as conceived of in this thesis, have parallels with the concepts of Sequential Patterns and Spatio-Temporal Sequential Patterns found in more mainstream data mining [40].

A sequential pattern is an ordered list of items, [4, 6, 22], whereas an MP is essentially an ordered list of locations; an MP can thus be viewed as specialised form of sequential pattern. Sequential pattern mining was first introduced in [4] in the context of transaction databases. Transaction databases contain details of items bought by customers in supermarkets. There are a number of application domains where the concept of sequential patterns have been successfully employed: (i) Engineering [30, 31, 94, 95], (ii) Finance market predication [4, 46, 52, 89] and healthcare [23, 42].

A spatio-temporal sequential pattern is like a “standard” sequential pattern except the items have a temporal and/or geographic reference. In [40] a spatio-temporal sequential pattern is defined as an ordered list of visits to a series of locations and thus consists of a series of connected segments with a start and end point in space and time. An example of a spatio-temporal sequence is shown in Figure 2.4 (taken from [40]), in this case the sequence indicates the tendency of tourists visiting locations A to E in a particular sequence over a particular time period. As such spatio-temporal sequential pattern have some similarity with the MP concept as conceived of in this thesis; although, as will become apparent, the MPs proposed in this thesis are much more sophisticated than the more standard sequential and spatio-temporal sequential patterns referenced in the data mining literature.

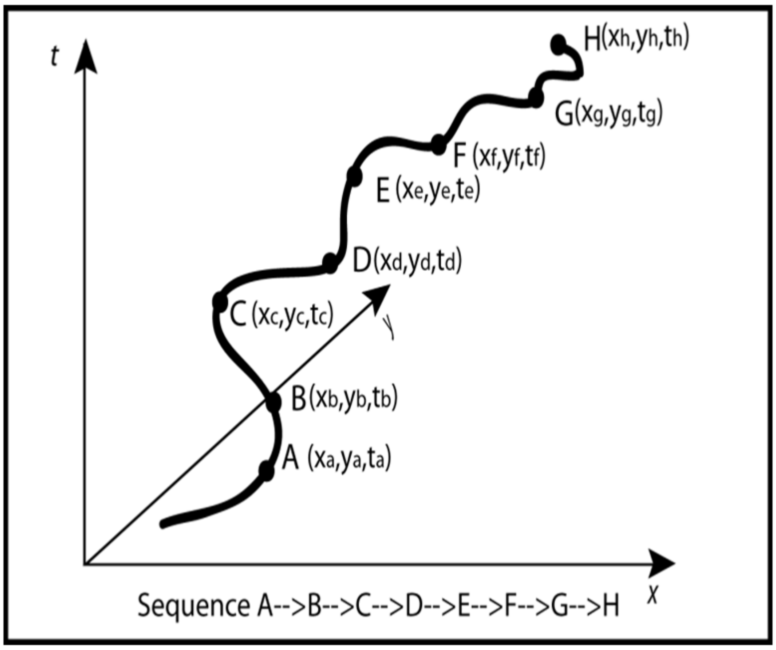


FIGURE 2.4: Tendency of tourists to visit a set of places A to E in a particular sequence A → B → C → D → E within specified duration [40]

2.4.2 Video Mining

Video mining can be defined as the unsupervised discovery of patterns in video data [38, 156]. Video mining can also be viewed as an extension of image mining given that video data can be considered to comprise a sequence of images (frames) [102]. Video mining is very application dependent according to the nature of the features to be mined. To this end we can identify three categories of feature [155]: (i) low level features such as colour, texture and shape of objects, (ii) contents based information such as the spatial temporal position of objects and the spatial temporal relation between objects and (iii) high level semantic information defining what is happing in a video (singing, dancing, sport and so on). In the context the work presented in this thesis we are interested in the second category, contents based information.

Examples of previous work on video mining, related to the work presented in this thesis, can be found in [86, 124, 130]. In [124] video data was mined using a frequent itemset mining technique to detect frequently occurring objects (actors, scenes) in music video. In [86] crowd behaviour analysis from video data was conducted using a clustering technique to extract crowd movement behaviour and individual behaviour. In [130] video data analyses was used to extract customer movement in a commercial area (a clothing store), clustering techniques were again used to group clients with similar trajectories. These clusters then provided the characteristics of client types according to time spent in the different departments within the store, time spent in the changing rooms and so on. The fundamental video mining techniques (such as object detection and tracking as described above) used with respect to these video mining applications have similarities with each other and that proposed in this thesis. However, as already noted, video mining is very much application dependent with each application using a bespoke approach although similar low-level techniques are often used. Video analysis of animal behaviour, and particularly rodent behaviour, is discussed further in Section 2.5 below. The proposed movement pattern mining from video data process is discussed in detail in Section 4.7 of Chapter 4 of this thesis.

2.5 Application Domain

The work described in this thesis is particularly focused on movement behaviour simulation, especially in the context of rodent movement behaviour. Over the last decade, the study of animal behaviour has seen a significant increase in research interest; simulation as a tool to support research on rodent movement behaviour has consequently also received increased attention. This Section considers the rodent movement behaviour application domain in more detail. The section is organized as follows. Sub-section 2.5.1 considers the concept of animal behaviour simulation in general. Sub-section 2.5.2 then considers rodent movement behaviour simulation in particular.

2.5.1 Animal behaviour simulation

The term Animal behaviour describes a set of actions which an animal may perform in response to certain events or to cause the initiation of new events. One particular area of interest is the control of disease spread by animals by studying animal behaviour so that deterrents can be put in place [142]. Animal behaviourologists are interested in different aspects of animal behaviour depending on their domain of interest. We can categorise these aspects according to the particular behaviour trait of interest as follows:

1. **Movement**: Discovering patterns of movement is essential for the understanding of the ecology and life history of animals, often with a view to animal conservation. It is also of particular interest with respect to the rodent application domain that acts as the focus for the work presented in this thesis. An example can be found in [129] where it is reported that isotope markers make it possible to track a diversity of animal species in a variety of habitats.

2. **Sound**: Sound has a direct impact on animal behaviour and is related to object movement irrespective of whether the sound is created by the object itself or by another object in the vicinity. Sound serves to diverted attention and typically causes a change in behaviour (usually movement behaviour) as evidenced by work presented in [122] where it is observed and concluded that elephants dsplay certain behaviour on hearing sound.

3. **Body posture**: Body posture is used with respect to the study of the behaviour of animals in the context of actions such as eating, drinking, grooming and gate. An example can be found in [161] where the body posture of cats was considered.

4. **Odor changes**: Odor (scent) changes are used by animals to orient and navigate their environment. Using odor/scent changes animals are able to determine and move toward targets (such as fresh water or pray). In [154] the authors describe different mechanisms of animal navigation in the context of odor.
5. **Scent markings**: Animals deposit faeces and other forms of scent to mark their territory and indicate their dominance. This is typically done by animals so as to attract a mate. See for example the discussion in [82]. Rodents also use scent to mark out their territory.

The last two of the above can also be viewed as a form of asynchronous communication. It should also be noted that from simplistic patterns of behaviour more sophisticated patterns of behaviour may evolve [35].

From the above categorisation, the category of interest with respect to the work presented in this thesis is movement, specifically movement patterns. Through consideration of certain kinds of movement, one can understand that an animal intends to engage in some particular activity. It is equally important to view animal behaviour as a system of mechanisms and processes by which animals can react to changes or events in their environment, and initiate changes to their environment either directly (by themselves) or indirectly (through other animals). It has also been accepted by animal behaviourologists and social scientists that the study of animal behaviour also provides a good platform for understanding human social issues [14].

Traditional animal behaviour study techniques involved setting up controlled experiments in the wild or by recreating animal habitats on a smaller scale, such as in a laboratory setting, and then observing the way in which the target animals behave. Simulation in turn dispenses with the resource require to set up such experiments. Examples can be found in [1, 3, 76, 108]. In [76] it is argued that in addition to real life experiments and case studies, it is equally important to utilize simulation experiments as an effective tools in validating an idea. It is also notable that in [108] it is argued that automatic video tracking provides a useful mechanism whereby animal behaviour study can be conducted in a reliable and consistent manner. The authors in the above cases were considering beetles, fruit flies, soil insects, parasitic wasps, predatory mites, ticks, and spiders; not rodents. Rodent simulation is considered in the following Sub-section.

2.5.2 Rodent movement behaviour simulation

This Sub-section considers previous work directed at rodent simulation; the application domain used as a focus with respect to the work presented in this thesis. In the context of rodent movement behaviour study several mechanisms have been developed to support "real life" experimentation [64, 110]. The studies considered with respect to this thesis, and used with respect to the evaluation reported on later in this thesis, were studies of the form typically conducted by behaviourologists in laboratory settings. In this setting, and depending on the nature of the study, one or more mice are introduced into a defined environment and their movement behaviour observed. This movement is frequently recorded using a video camera suspended over the environment. The advantage offered by such laboratory experiments is control over the environment; thus different kinds of scenarios can be created for the purpose of a variety of behaviour studies. However, the number of studies that can be conducted in this way is limited due to the resource required; this therefore limits the scope of "what if" style experiments. Again the resource issue can be addressed using simulation.

Examples where computer simulation has been used in the context of rodent behaviour can be found in [2, 3, 54, 75, 101]. In [54] a system is presented comprised of two components, a tracking module and a characterisation module. The first was used to track the position of multiple interacting mice, while the second was utilized to characterise automatically social and non-social behaviour after providing example behaviours. However, this work was more concerned with behaviour study than simulation. Similarly in [75] a software was proposed to analyse video data of cage mice, however in this case the video data required extensive initial manual annotation before the behaviour analyses could take place. In [101] a computer software system was proposed that accepts as input a video file and extracts information, such as escape latency or elapsed time, average velocity and total distance in the context of the behavioural study on rats. In [2, 3] a rodent MABS simulation was proposed that has parallels with the MABS used with respect to the work presented in this thesis. However, the simulation is defined in terms of a hand crafted *behaviour graph* derived through a process of domain expert interviews, a time consuming and error prone process.

2.6 Summary

This chapter has presented the background and application domain to the work presented in this thesis. The chapter divided the previous work into three different sections: (i) Simulation, (ii) Knowledge extraction from video, data and (iii) Pattern Mining. Simulation was discussed along with the advantages offered and the application domains, where simulation has been widely adopted. MABS was discussed in detail along with the mechanisms used to represent agents in MABS. Knowledge extraction from video data was discussed in terms of object detection and tracking. Object representation was also considered in the context video of data knowledge extraction. Pattern mining was reviewed with respect to sequential pattern mining and parallels drawn with respect to movement pattern mining as proposed in this thesis. The chapter culminated with consideration of the rodent behaviour analysis application domain. In the next chapter the proposed approach to environment modelling and data acquisition is presented.

Chapter 3

Environment Modelling and Data Acquisition

3.1 Introduction

A central feature of the work presented in this thesis is the extraction of information from video data, from which to mine *Movement Patterns*, to "feed" multi-agent based simulations. The format of this data extraction is dependent on the nature of the application to be simulated. More specifically the nature of the environments (playing areas) to be modelled [118] has a significance effect on the nature of the video data extraction process (at least with respect to MABS). In other words, the mechanism whereby the environment is modelled dictates the nature of the desired output from the video data extraction process. Thus in this chapter the proposed approach to environment modelling, and consequent data acquisition, is first presented. This is followed by a description of the proposed process for data acquisition from video using a bespoke software system, the Video Data Acquisition Software (VDAS) system.

The environment in which we anticipate our agents will operate will contain objects of various kinds. In the context of rodent behaviour MABS we can imagine obstructions and tunnels; it will also be necessary to distinguish between open areas/spaces and wall proximities, because we know that rodents have a preference for wall locations (a behaviour known as *thigmotaxis*). The environments of interest can be modelled in various ways but the most straightforward is to consider the environment as a grid, in other words as a *tile world* (such as that used in [27] and by others). The idea here is to divide the environment into equal sized grid cells; the entire environment can then be linearised by assigning sequential location IDs to each tile. The reason why we want to do this is that a linearisation of space can be processed more efficiently than a non-linearised space (although as will become clear later in this thesis, this linearisation advantage is only of benefit when *absolute addressing* is used, it breaks down in the context of *relative addressing*). In addition to a sequential ID number, each tile is also allocating a *tile descriptor* or ground type descriptor.

With respect to the rodent behaviour application domain the input to the data acquisition (machine learning) process, as already noted, was video data of rodents moving around some experimental set up. The data to be extracted was then location information, from which movement patterns could later be mined (as discussed in further detail in Chapter 4). Clearly, this location data has to subscribe to the same mechanism as used to model the environments of interest, hence the nature of the environment modelling has to be established first. For the analysis of the video data, software was developed to extract locations and store them in a text file for further processing; the Video Data Acquisition Software (VDAS). This software is discussed in Section 3.8 later in this chapter.

As noted earlier in this thesis, with respect to the presented research, three different categories of scenario were considered: (i) simple scenarios, (ii) complex scenarios and (iii) two rodents scenarios; further detail concerning these scenario categories is provided later in this chapter in Section 3.7.

The movement patterns that we wish to eventually extract (mine) from the location information extracted from the video data are essentially “from-to” patterns. As will become clear later in this thesis, the from and to locations can be described in either absolute terms or in relative terms; thus, at a high level, we have two types of movement pattern that may be extracted from video data and consequently utilised in a MABS. With respect to the work presented in this thesis absolute locations were expressed in terms of grid cell IDs, whilst relative locations were described in terms of a *location descriptor*, a tuple comprising the (ground type) descriptor for a given location and the descriptors of its immediate neighbours. Further detail concerning absolute and relative movement patterns, and the nature of tile and location descriptors, will be presented in Chapter 4.

Given the above this chapter is structured as follows. Section 3.2 considers the different components that may exist in an environment, such as: obstructions, blocked areas, target objects and so on. Details concerning the modelling of the environments of interest, using the proposed grid representations, are then presented in Section 3.3. As will become apparent, using the grid mechanism it is also possible to capture the spatial relationship between pairs of agents (rodents with respect to the application focus of this thesis). The relationship between pairs of agents was considered using the concept of zones; square areas centred over agent locations measuring in terms of a number of grid cells. The relationship between pairs of agents was used to define states and an associated state graph. Further detail concerning the encapsulation of the spatial relationship between pairs of agent is thus presented in Section 3.4. Section 3.5 then introduces the concept of *movement vectors*; the significance of which is with respect to the movement patterns considered in the following chapter. In the following section, Section 3.6, the twin concepts of *tile descriptors* (ground types) and *location descriptors* is explained in further detail. In essence, given a grid environment each grid cell (tile) has a descriptor associated with it which indicates the nature of the object/area to which it belongs; we refer to this as a tile descriptor. Collections of tile descriptors form a location descriptor which in turn is used in the context of relative movement patterns. Section 3.7 is concerned with the modelling of the environments associated with the three different scenario categories of interest with respect to the work presented in this thesis. Section 3.8 then presents the VDAS system developed for the required video data extraction. The chapter is concluded with a summary in Section 3.9.

3.2 Environment Objects and Areas

As noted above, the environments we wish to model are anticipated to comprise collections of objects or regions/areas. In the context of rodent behaviour MABS, the application domain that has acted as a focus for the work presented in this thesis, the following types of object and region/area can be identified (other application domains may feature alternative objects and areas):

Obstruction: Areas that block the movement of agents, for example a solid object located within an environment.

Blocked Area: Areas outside of the environment into which an agent cannot move.

Target Object: An object to which an agent may wish to move; for example a nest site.

Tunnel: An area enclosed between obstructions that serves to connect different parts of a given environment.

Wall Proximity Areas: Wall locations (of interest because of the desire for walls displayed by rodents).

Open Area: Areas which do not subscribe to any of the above.

The distinction between obstructions and blocked areas is that blocked areas are outside of the environment of interest while obstruction are contained within it (the rational for this will become clear later in this chapter).

The purpose of the above categorisation of objects and areas/regions is to distinguish between different parts of an environment, which in turn might have an influence on movement. For example, if we wished to model the grazing behaviour of sheep in an upland region we might identify two different area types: (i) sloping and (ii) flat; because of an intuition that these different ground types will have an influence on the nature of the movement patterns relevant to this application domain. In the case of modelling the behaviour of pedestrians in a railway station we might have the ground types: platform and forecourt; because of an intuition that pedestrians will behave differently in these areas.

3.3 Grid Representation

In the introduction to this chapter it was established that a grid representation, a tile world, was adopted with which to represent the environments of interest. The x-y location represented by a grid cell was deemed to be the centre of the grid cell. For reference purposes each grid cell was given a sequential grid number, an *address* or grid cell ID, as shown in Figure 3.1. Note that in the figure the origin is in the top-left hand corner. With respect to discussions concerning environments presented in this thesis the origin is always assumed to be in the top-left corner. Environments are also considered to feature an orientation expressed in terms of the cardinal and inter-cardinal directions. Thus, with reference to Figure 3.1, North is “up” the page, and so on. We also refer to environments as having top and bottom, and left and right, sides with the obvious meanings. Whatever the case grid mechanisms of this form offer the following advantages:

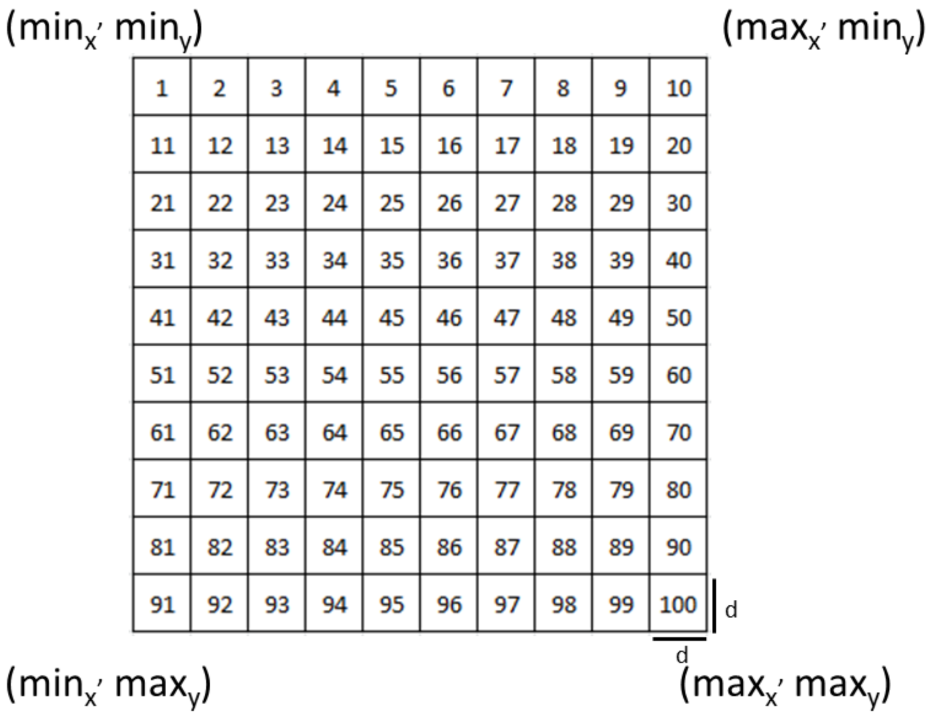
1. They support the linearisation of the space which provides for processing efficiency (at least in the case of absolute representations).
2. Following on from (1) movements can be expressed as a single integer value which defines a 2D vector, measured from the origin of the space, that encapsulates both distance and direction (this is discussed further in Section 3.5 below).
3. It provides a foundation for the representation of movement patterns as will become clear later in this chapter.

Environments are conceptualised as being comprised of an $n \times m$ bounding box where n and m are numbers of cells. Some of the environment considered may be irregular in shape, so some of the cells in a given $n \times m$ bounding box may be outside of the environment, hence the concept of “blocked areas”. With reference to Figure 3.1 the values for n and m may be calculated using Equations 3.1 and 3.2.

$$n = \lceil \frac{max_y - min_y}{d} \rceil \tag{3.1}$$

$$m = \lceil \frac{max_x - min_x}{d} \rceil \tag{3.2}$$

Where d is the size of a grid cell measured in term of pixels. Each grid cell measures $d \times d$ pixels as also shown in Figure 3.1. The selected value for d depends on the nature of the entities that agents are intended to model and the nature of the simulation application. We wish our agents, whatever they represent, to fit into a single grid square. Thus if our agents represent humans the value for d will be larger than if our agents represent rodents. In the case of our rodent MABS d will be larger for rats than for mice.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

FIGURE 3.1: Grid Structure Linearised using Location IDs

3.4 Relationship Between Agents

Given a scenario with more than one agent, the agents will be in some spatial relationship with each other. We wish to capture the nature of this relationship during the proposed video mining and encapsulate it within the context of the movement pattern concept as envisioned in this thesis. To this end the usage of a set of *zones* is proposed with which to define the relationship between pairs of agents. These in turn are used to define the "states" of individual agents. The relationship between potential states is then defined using what is referred to in this thesis as a "state graph". A zone is a square area centred over an agent location (the *reference agent*), measuring $z \times z$ where z is a number of grid cells. Each agent is surrounded by three concentric zones: (i) the *intrusion zone*, (ii) the *proximity zone* and (iii) the *ignore zone*. The intrusion zone describes the area where two agents can be said to be meeting. The proximity zone describes the area where two agents can be said to be "concerned" about each others presence. The ignore zone describes the area where two agents can be said to be "unconcerned" of each others presence. The intrusion zone takes precedence over the proximity zone, which in turn takes precedence over the ignore zone.

With respect to the rodent MABS used as a focus for the work presented in this thesis the intrusion zone measured 3×3 grid cells, the proximity zone measured 7×7 grid cells, and the ignore zone was anything else. These dimensions were derived through

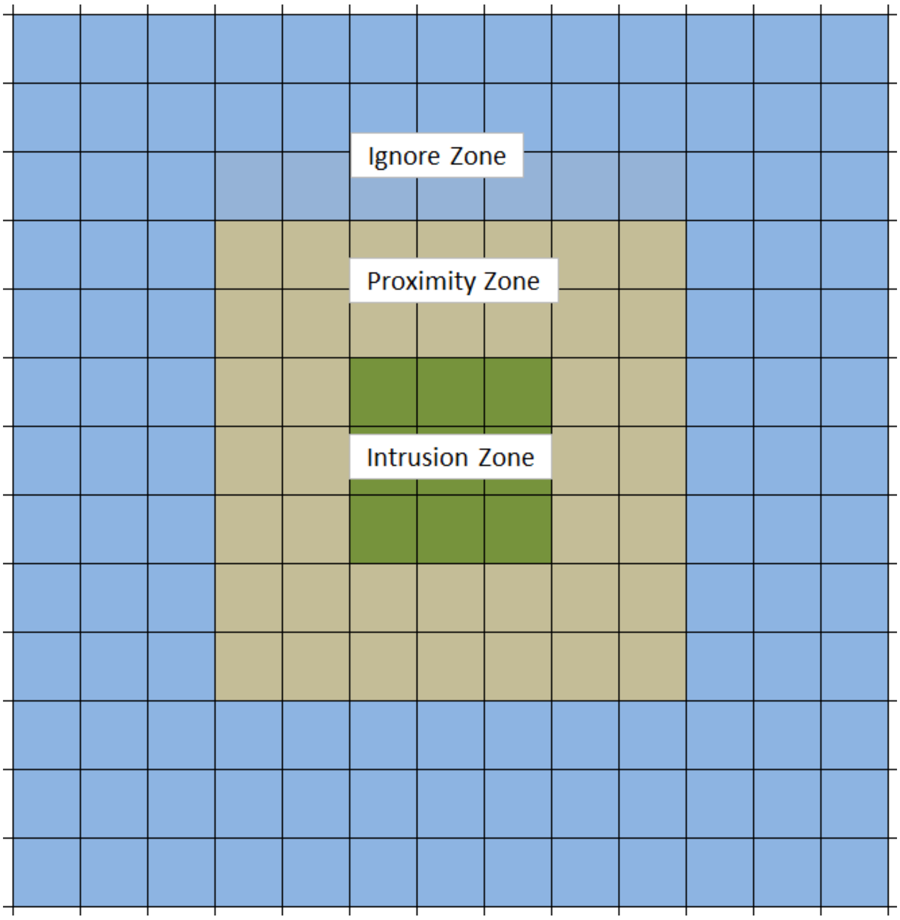


Figure 3.2: Zone representation

consultation with domain experts[1]. Interestingly, discussion with the domain experts indicated that the size of these zones is dependent on the nature of the rodents under investigation; rats prefer larger intrusion zones than mice. Conveniently rodent size is captured by the adopted cell size, therefore the above ratios will hold for both rats and mice. Whatever the case, the work presented in this thesis is concerned with the concept of zones as a means of expressing the relationship between pairs of rodents, not the specific sizes of such zones. Figure 3.2 illustrates the zone concept in the context of the considered rodent MABS. Each zone is centred over the reference agent of interest for the purpose of recording the relationship between the reference agent and some other agent in the environment. Square zones were considered simply for ease of computation. It is acknowledged that circular zones defined by a radius might have been more appropriate, or square/rectangular zones referenced to the direction in which an agent is facing; these alternative mechanisms are considered as an appropriate topics for future work as discussed in Chapter 7.

---

[1] Prof. Jane Hurst from, the William Prescott professor of Animal science and the director of the Mammalian Behaviour and Evaluation Institute of Integrative Biology at the University of Liverpool Leahurst campus.

3.5 Movement Vectors

The concept of movement vectors was introduced in the introduction to this chapter, they are an integral part of MPs. Depending on whether absolute or a relative movement patterns are considered (as discussed further in the following chapter, Chapter 4), movement vectors can be described in terms of either: a single constant integer number (absolute mechanism) or $\langle x, y \rangle$ components (relative mechanism).

Returning to Figure 3.1, in terms of the absolute mechanism, the distance between two cells can be captured by a constant, k, which can be added to a particular cell ID to "move" to a next location. For example, with reference to Figure 3.1, to move one cell to the north of cell number 45, we use $k = -10$ (number of columns) and to move one cell to the south-west we use $k = 9$ (number of columns - 1)[2]. The constant k when used in this context is referred to as a *movement vector*; capturing both distance and direction.

In case of the relative movement pattern mechanism absolute addressing is not used, hence movement vectors are represented in terms of a tuple of the form $\langle x, y \rangle$, where x is the number of columns between the *from location* and the *to location*, and y is the number of rows between the from and the to locations. For example to move one cell to the north the movement vector would be expressed as $\langle -1, 0 \rangle$, -1 rows from the current location and 0 columns. Similarly to move one cell to the north-west the movement vector would be expressed as $\langle -1, -1 \rangle$, -1 rows from the current location and -1 columns.

3.6 Location Descriptor

The relative movement pattern mechanism makes used of tile descriptors. Earlier in this chapter it was noted that environments comprise objects and/or areas; the different categories of object/area were listed in Section 3.2. Objects and areas comprise collections of grid cells. Each grid cell has a tile descriptor (or ground type) associated with it indicating the nature of the object/area type to which it belongs. More specifically each grid cell (address) has a "ground type" associated with it taken from a set of ground types $D = \{d_1, d_2, \dots\}$.

The nature of ground types depend on the application domain of interest. In the case of the rodent simulations of interest with respect to the work presented in this thesis the set of ground types D was $\{b, g, n, o, t, s, w, -\}$ defined as follows:

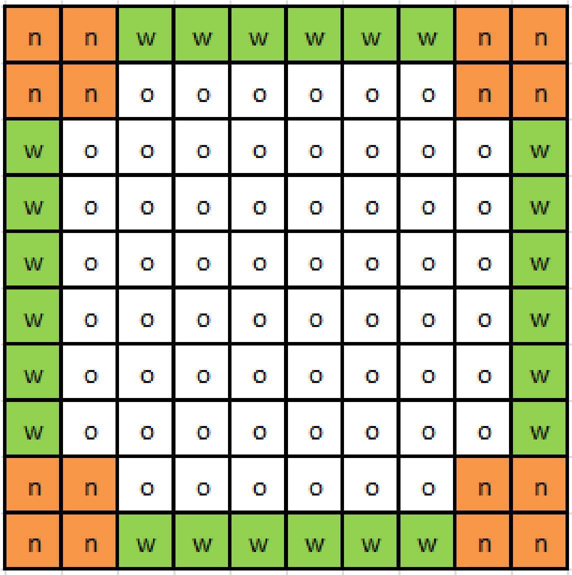
b Obstructed area.

g Gate location (only used in the context of entrances and exists to tunnels).

i Water source.

---

[2] It is to be noted that, cells at the boundaries of grid environment, can not be fill with the value of k.



n	n	w	w	w	w	w	w	n	n
n	n	o	o	o	o	o	o	n	n
w	o	o	o	o	o	o	o	o	w
w	o	o	o	o	o	o	o	o	w
w	o	o	o	o	o	o	o	o	w
w	o	o	o	o	o	o	o	o	w
w	o	o	o	o	o	o	o	o	w
w	o	o	o	o	o	o	o	o	w
n	n	o	o	o	o	o	o	n	n
n	n	w	w	w	w	w	w	n	n

FIGURE 3.3: Simple environment grid with Ground Types

w	w	w	w	w	b	b	w	i	i	i	w	b	b	w	w	w	w	w
w	o	o	o	w	b	b	w	i	i	i	w	b	b	w	o	o	o	w
w	o	o	o	w	b	b	w	o	o	o	w	b	b	w	o	o	o	w
w	o	o	o	g	t	t	g	o	o	o	g	t	t	g	o	o	o	w
w	o	o	o	w	b	b	w	o	o	o	w	b	b	w	o	o	o	w
n	n	n	o	w	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	o	w	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	w	w	b	b	w	n	n	n	w	b	b	w	w	n	n	n

FIGURE 3.4: Complex environment grid with Ground Types

n Nest location.

o Open space (the default descriptor used where no other descriptor is applicable).

t Tunnel location.

w Wall location.

$-$ Blocked area, a location outside of the environment bounding box under/consideration.

The usage of the $-$ ground type (blocked area) will be explained shortly.

Not every available ground type is used for every scenario. In terms of the categories of scenario considered in this thesis, in the case of the Simple scenario category $D = \{n, o, w, -\}$, while in the case of the Complex category $D = \{b, g, I, n, o, t, w, -\}$ and the Two-rodent category $D = \{b, n, o, w, -\}$. Example of Simple and Complex category scenarios, with Ground Types, are shown in Figure 3.3 and 3.4. Note that in the later case there are three connected *divisions* which are labelled: "left" (L), "middle" (M) and "right" (R). Note also that the scenario features two tunnels connecting the left and right portions; the tunnels are used to connect adjacent divisions, if a rodent agent wants to move from one division to another it has to pass through two gate cells/locations.

A location descriptor S, in its basic form, comprises a set of ground types (tile descriptors) associated with the set of cells encompassed by 3×3 grid square centred over the current location. The ground types are ordered using a left to right, top to bottom, sequencing; thus $S = \langle \lambda_1, \lambda_2, \ldots, \lambda_9 \rangle$ where λ_1 to λ_9 take values from D. Therefore a location descriptor comprises the ground types of the current location and its immediate eight neighbours. In some cases we also wish to include a division identifier, thus $L = \langle S, a \rangle$, where S is the set of ground type descriptors and a is a division identifier taken from the set A. In the case of the work presented in this thesis the latter is used with respect to complex scenarios (in the case of the complex scenario shown in Figure 3.4 the set A comprises the set $\{L, M, R\}$).

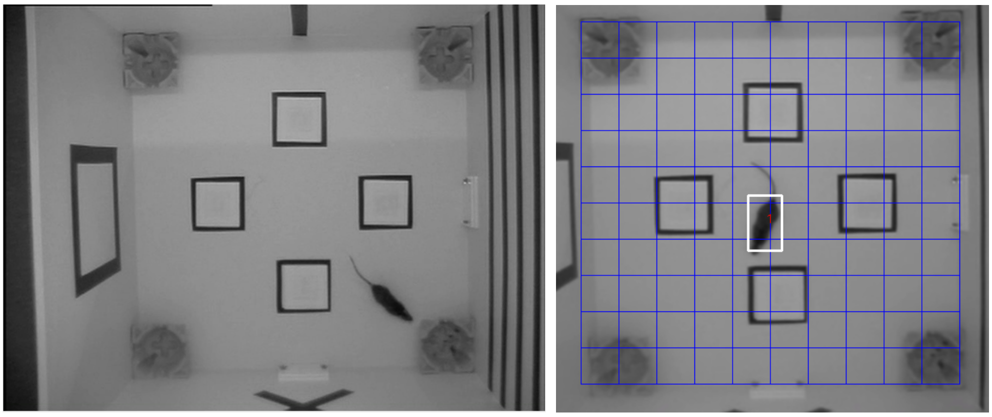
In some cases, where the location of interest is at an edge or corner, the 3×3 grid square centred over the current location will overlap with space outside of the bounding box surrounding the environment. In this case the $-$ ground type descriptor, introduced above, will be used for locations outside of the bounding box. Thus with reference to Figure 3.3 the location descriptor for the top-left corner tile will be $\langle -, -, -, -, n, n, -, n, n \rangle$ and the tile immediately to the right will be $\langle -, -, -, n, n, w, n, n, o \rangle$.

3.7 Scenario Categorisation

As already noted earlier in this thesis, the application domain to which the work described was directed was rodent behaviour. The basic idea was to run and record a sequence of "rodent in a box" experiments, of increasing sophistication, by suspending a video camera over the box. The boxes used measuring $1.2 \times 1.2\text{m}^2$. Such laboratory experiments are typically conducted by rodent behaviourologists. The experiments comprise one or more rats/mice and possible a number of other elements (obstructions, food, nest sites and so on). To this end the following three categories of scenario, already introduced, were considered: (i) Simple, (ii) Complex and (iii) Two-rodent. Video data concerning experiments subscribing to each of these categories was collected. Some further detail concerning each of these categories is given in the following three subsections.

3.7.1 Simple Scenarios

With respect to the Simple scenario category, Figure 3.5(a) shows a still taken from the video data collected for this category. From the figure it can be seen that this scenario includes four "nest boxes", located at each corner of the playing area. It also includes some markings on the base of the box, these were ignored for the purpose of



(a) Still from rodent video data featuring a Simple scenario

(b) Still from Figure 3.5(a) with 10×10 grid superimposed over it (Simple scenario)

FIGURE 3.5: Stills for Simple scenario

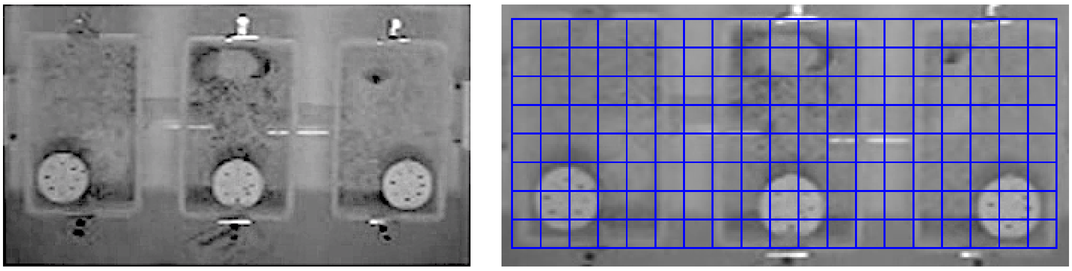
video data extraction as considered in this thesis (they are used by behaviourologists in the context of experiments using scent markings). For the Simple scenario video collection laboratory rats were used, hence a 10×10 grid was used with respect to the simple scenarios, as shown in Figure 3.5(b) which shows the same video still but with a grid superimposed. The same grid was shown in schematic form in Figure 3.3. From Figure 3.5(b) (and by referring back to Figure 3.3) it can seen that the nest boxes at each corner covered approximately a $2 \times 2 = 4$ configuration of grid cells. As the box measures 120×120cm the gird cell size d in this case was $120/10 = 12\text{cm}^3$. A grid cell thus measured 12cm^2, which is approximately equal to the size of a laboratory rat minus the tail.

3.7.2 Complex Scenarios

The complex scenario category featured scenarios that comprised interconnected divisions joined by tunnels. Again this type of experiment is frequently used by behaviourologists. Complex scenarios thus feature “blocked areas”; the areas between divisions which a rodent will be unable to access. The specific examples for which video data was obtained also featured water supplies and nest boxes. An example still from the video data collected with respect to complex scenarios is given in Figure 3.6(a). In this case the depicted experiment comprised three equal sized divisions connected by two tunnels. Inspection of the figure also indicates: (i) three enclosed “nests” (the circular objects with air holes) and (ii) a water source (at the top of the middle box in Figure 3.6(a)). In the figure, although difficult to see, the mouse is located in the upper half of the right-hand division. To move from one division to another a rodent (a mouse in this case) must use one of the tunnels.

---

[3] This equates to a 50×50 pixels grid cell size with respect to the video data used for evaluation purposes in this thesis.



(a) Still from rodent video data featuring a Complex scenario comprising three interconnected boxes

(b) Still from Figure 3.6(a) with 19 × 8 grid superimposed over it (Complex scenario)

Figure 3.6: Stills for Complex scenarios

In this case a 19 × 8 grid, to covere the entire 120 × 120cm box space, was superimposed over the environment of interest to give a grid cell size d of 6 × 6cm, which is approximately equal to the size of laboratory mouse as shown in Figure 3.6(b)[4]. The same grid was shown in schematic form in Figure 3.4. In the case of the Complex scenarios considered in this thesis the experiments featured laboratory mice rather than laboratory rats, laboratory mice are smaller than laboratory rats hence the smaller grid size.

3.7.3 Two Rodents Scenarios

For the two rodents scenario category the environments used to produce the required video data covered the entire 120 × 120cm box space. Two stills from the video data collected are given in Figures 3.7(a) and 3.7(b). In the figures two rodents (mice) can be clearly identified. Note that the two stills feature slightly different scenarios in that the obstructions are not in the same place in both cases. The circular objects are nests that have one opening where a mouse can enter and exit. The boxes also feature side panels but these are not of significance with respect to the desired video analysis reported on here. In this case a 14 × 14 grid was superimposed over the video data as shown in Figure 3.7(c). The resulting grid cell size d will then be $120/14 = 8.5$cm[5] enough to capture a single mouse agent.

3.8 Video Data Analysis and Location Extraction

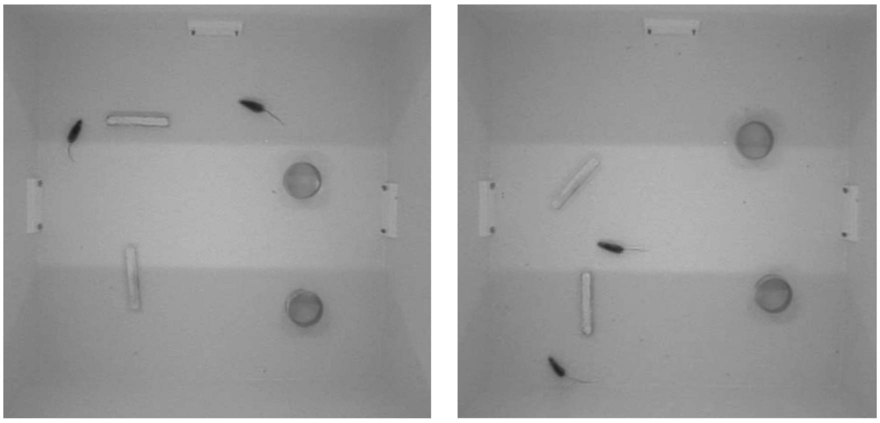
This section considers the software developed for extracting data from video, the Video Data Acquisition Software (VDAS)[6]. In general terms Video Data Acquisition (VDA)

---

[4] A grid cell size of $d = 6$cm equates to 15 pixels with respect to the video used for evaluation purposes in this thesis.

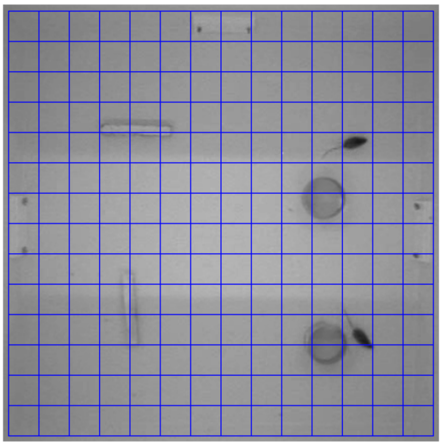
[5] A grid cell size of $d = 8.5$cm equates to a 30 × 30 pixels grid cell size with respect to the video used for evaluation purposes in this thesis.

[6] This software is based on the *BlobTrackingModule* available at:
https://nsl.cs.usc.edu/enl/trunk/aqua/OpenCV-2.3.1/doc/vidsurv/Blob_Tracking_Modules.doc,
https://nsl.cs.usc.edu/enl/trunk/aqua/OpenCV-2.3.1/doc/vidsurv/Blob_Tracking_Tests.doc.



(a) Still from rodent video data featuring a Two rodent scenario, Example 1

(b) Still from rodent video data featuring a Two rodent scenario, Example 2



(c) Still from rodent video data with 14 × 14 grid (Two rodent scenario)

FIGURE 3.7: Stills from rodent video featuring two mice

is the process of extracting information from video using appropriate software. With respect to the work presented in this thesis the information we wish to extract is rodent location information. A high-level block diagram of the adopted VDAS system is presented in Figure 3.8. From the figure it can be seen that VDAS takes video data as input, in MP4 format (see Sub-section 2.3.1 of Chapter 2), and produces location data as output; the later obtained using a location data extraction process. The location data is stored in a set of sets $L = \{L_1, L_2, \ldots, L_n\}$, where n is the expected number of objects (rodents) of interest. Note that the VDAS system described above was developed in C# and used Ms SQL server 2008 for data storage.

So as to identify the rodent objects of interest in the video data background subtraction was adopted as described in Sub-section 2.3.2 in Chapter 2. From Chapter 2 it will be recalled that the idea is to compare a video frame which is known not to feature an objects of interest, referred to as the *background image* (img_b), with a current frame img_i. The two images are compared pixel by corresponding pixel and a *foreground image* (img_f) produced (Equation 3.3). If two corresponding pixels in img_b and img_i

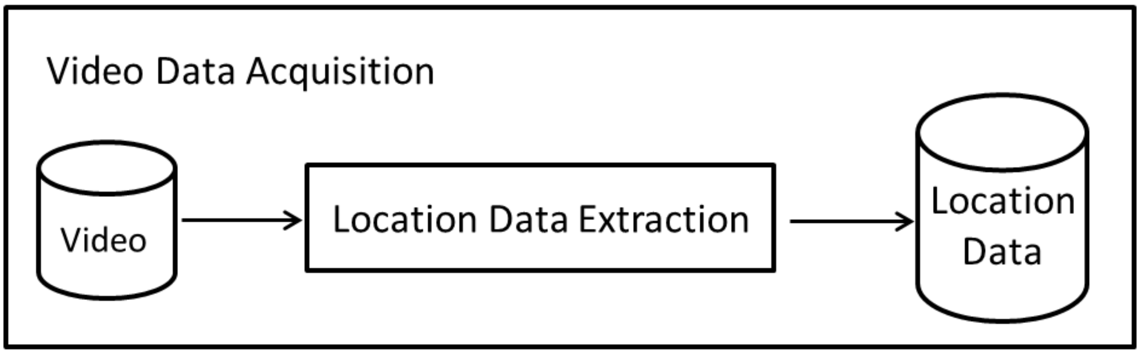


Figure 3.8: Block Diagram for Video Data Acquisition Software (VDAS)

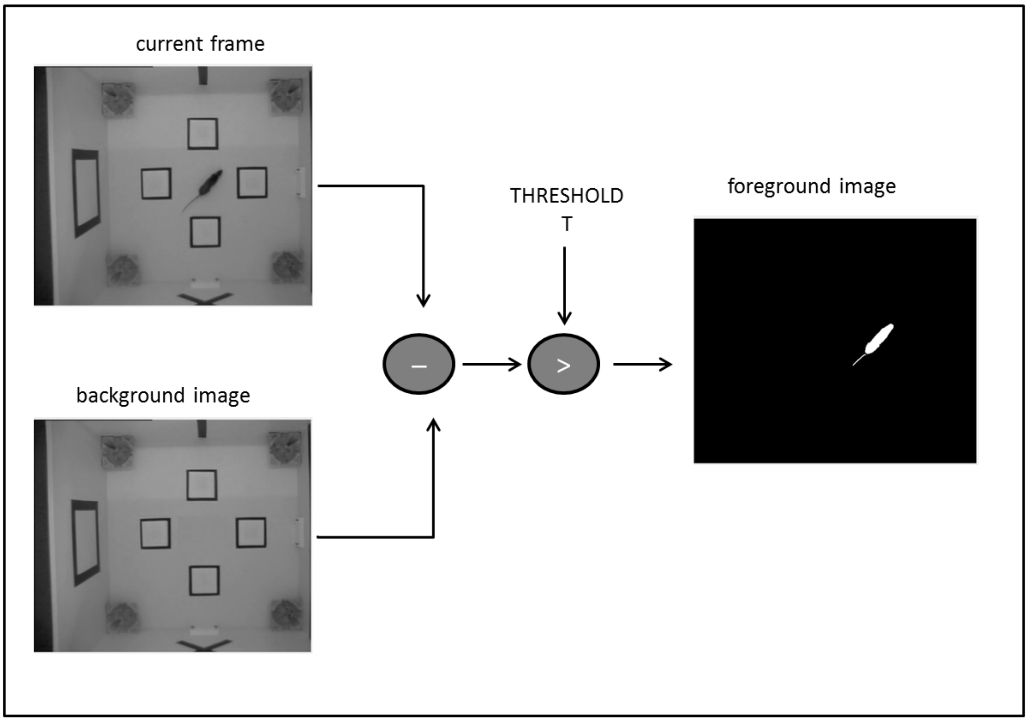


Figure 3.9: Background subtraction process

have the same colour value (within some level of tolerance expressed by a threshold T) the corresponding pixel colour value in img_f is set to 0 (black), otherwise it is set to 1 (white). On completion the img_f will thus feature white blobs indicating the objects we are interested in. Noise can be removed by pruning blobs below a minimum size. The remaining blobs can then be segmented and stored. With respect to the VDAS software the location of a blob is the geometric centre of a given blob, expressed in terms of x-y pixel coordinates referenced to some origin and translated into a cell ID number. The background subtraction process is illustrated in Figure 3.9

$$img_f = img_i - img_b \tag{3.3}$$

The VDAS location data extraction process is illustrated by the flow diagram presented in Figure 3.10. The associated pseudo code also for VDAS is presented in Algorithm 1. With reference to Algorithm 1 (and the flow chart given in Figure 3.10) the input is a MP4 video file comprised of a sequence of frames $\{f_0, f_1, \ldots, f_k\}$ and a sample interval s. The output is a set of sets of locations $L = \{L_1, L_2, \ldots, L_n\}$ where n is the number of objects of interest. The process commences (line 4) by superimposing a grid over the initial video frame. The nature of this grid is defined by the user. The user interface included with VDAS allows the user to superimpose a grid over the video by selecting the grid corner locations and specifying a value for d. Each cell in the grid is defined using absolute addressing as described earlier in this chapter and illustrated in Figure 3.1[7].

Return to Algorithm 1, in line 5 of the algorithm a background image, img_b is generated. To this end it should be noted that the video data, in all cases, initially does not feature any rodents, these are introduced once the video has started. This was done specifically so that a background image can be obtained for use with respect the background subtraction process. The remainder of the video is process using a loop (lines 7 to 16). Every s frames a new sample image (img_i) is obtained (line 8). This is then compared to the background image using the *backgroundSubtrcation* function (line 9) and a foreground image (img_f) produced. The foreground image is then processed, using the function *blobExtraction*, and a set of zero, one or more "blobs" extracted and stored in the set *Blobs* (line 10)[8]. This set is then processed (lines 11 to 14) and for each $\text{blob}_i \in Blob$ a corresponding cell number is identified. This is appended to the set L_i associated with the current object.

Algorithm 1 Algorithm for Video Data Acquisition Software process

```
 1: INPUT: Video File in WMV format comprising a set of frames $\{f_0, f_1, \ldots, f_k\}$
 2: sample intervals s
 3: OUTPUT: $L$ = Set of sets of Locations ($\{L_1, L_2, \ldots, L_n\}$)
 4: initialiseGrid()                                    ▷ Superimpose grid
 5: $\text{img}_b$ =extractBackground($f_0$)                 ▷ Capture frame from video
 6: i = s
 7: while (Not EOF) do                ▷ If Not End Of File repeat the loop
 8:     $\text{img}_i$ = extractFrameImage(i)
 9:     $\text{img}_f$ = backgroungSubtraction ($\text{img}_b$, $\text{img}_i$)
10:     Blobs = blobExtraction($\text{img}_f$)
11:     for (j ← 1 to |Blobs|) do
12:         cellNo = findCell_ID ($\text{blob}_j$)
13:         $\text{L}_j$ = ($\text{L}_j \cup$ cellNo)
14:     end for
15:     i ← i + s
16: end while
17: Exit                              ▷ If End Of File reaches exit program
```

---

[7] The selected grid size d used for extracting location information from the video data should be the same as that used to later mine movement patterns as described in Chapter 4.

[8] A blob in this context is thus a group of connected white pixels.

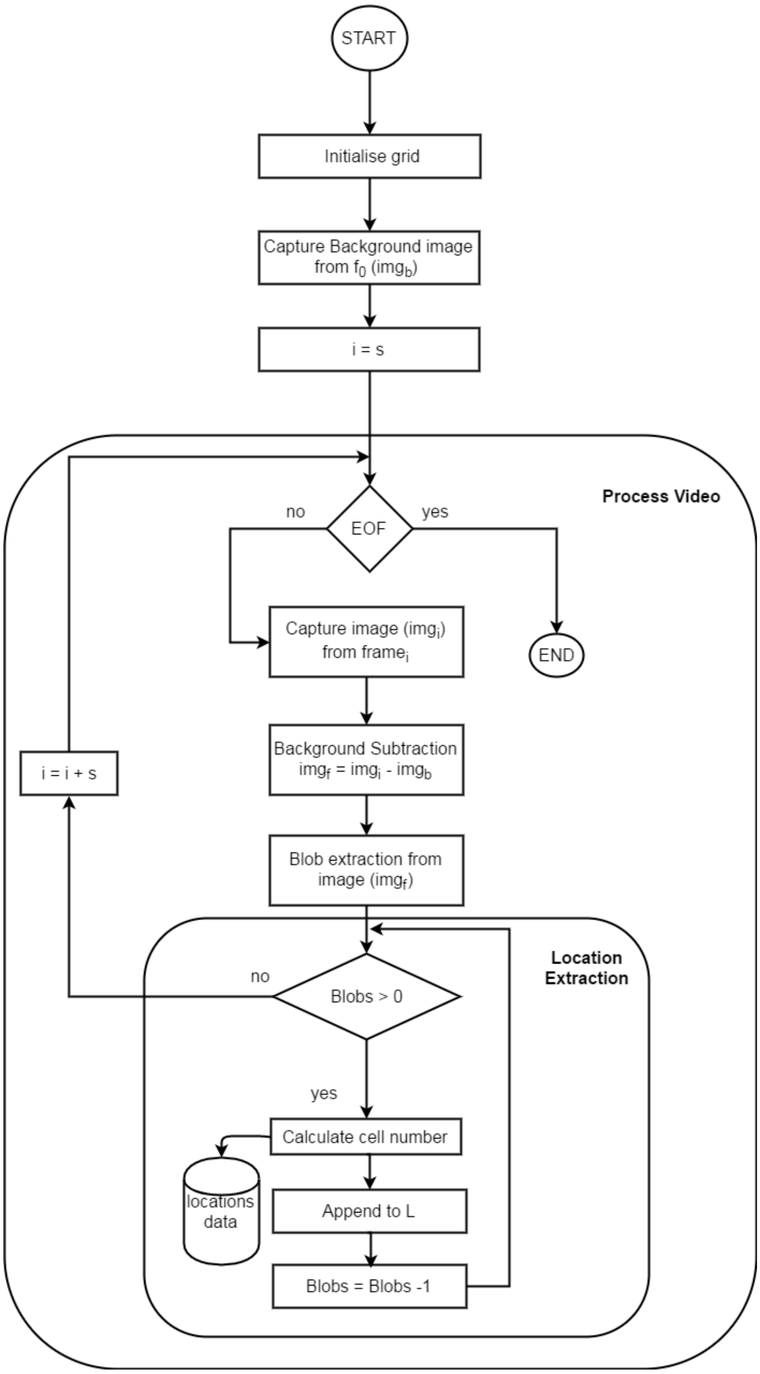


FIGURE 3.10: Flow chart for VDAS system

Where the number of expected objects is greater than one ($N > 1$) the appropriate location set L_i is identified by examining the most recent previously recorded locations in each set L_i and predicting next locations by "dead reckoning"[9]. By comparing the current location with each predicted locations, the new location can be added to the location set for the object where the predicted location is closest to the current location. It may be the case, where two or more objects of interest are close to each other, that a location is allocated to the wrong location set; this is not considered to be an issue as we wish to extract movement patterns from the location data rather than track individual objects. Video object tracking in concerned with the accurate following of objects in video data; with respect to the work presented in this thesis, the interest is in identifying MPs rather than accurate tracking of objects. On occasion an object we wish to track may be lost, typically when it enters a nest box, in which case a *null* location will be recorded and appended to L_i.

3.9 Summary

In this chapter environmental modelling and the location data acquisition process were discussed. It was noted that the way in which an environment is modelled impinges on the requirements for the location data acquisition. It was also noted that the environments of interest may contain a variety of objects such as: obstructions, blocked areas, open areas, tunnels and wall proximity areas. It was further noted that agent locations could be expressed in absolute terms or relative terms. Agents are anticipated to interact with the environment objects. A grid representation (tile world) was adopted as the most appropriate mechanism for modelling environments. Using the grid representation environments were represented using equal sized grid cells. A grid cell numbering system was used so that the environments of interest could be linearised. The relationship between entities, as in the case of the two-rodent scenario category, where captured using the concept of zones. Three zones were identified: (i) intrusion zone, (ii) proximity zone and (iii) ignore zone. The mechanism for expressing movement vectors using the grid representation was then considered; a movement vector captures, distance, direction and speed (distance travelled over a prescribed time interval). Note that the stopped condition is considered to be a special type of movement with speed of zero. It was noted that the nature of movement vectors using the absolute mechanism is different to their nature using the relative mechanism. The relative mechanism was founded on the concept of location descriptors made up of the ground types associated with a 3×3 sub grid surrounding a current location of interest, linearised from top-left to bottom-right. Three different categories of scenario are considered in this thesis: (i) Simple, (ii) Complex and (iii) Two-rodent; each category was detailed in the chapter. For the extraction of location information from the video data a software system was presented, the Video Data Acquisition Software (VDAS) system, that tracks objects of

---

[9] The phrase dead reckoning, commonly used in navigation, means predicting a next location assuming current velocity and direction is maintained.

interest and extracts associated locations using a given sample interval. The next chapter describes the process whereby movement pattern are mined from the location data extracted from the training videos using the VDAS system.

Chapter 4

Pattern Mining Framework

4.1 Introduction

This chapter details the nature and structure of Movement Patterns (MPs). Recall that MPs describing the movement of a subject (rodents in the case of the focus of this thesis) from time t to $t+i$. Recall also that the environment in which entities of interest operate, the "playing area" in the context of MABS operation, was encapsulated in the form of a grid across which subjects could be tracked (as described in Section 3.3 in Chapter 3).

As discussed previously in Chapter 3, two mechanisms were considered for representing MPs: (i) absolute and (ii) relative. The distinction between the two, as the terminology suggests, is that in the first case locations are recorded relative to the origin of the environment while in the second the location is recorded relative to the local surroundings. Absolute locations are therefore expressed in terms of a specific address (a unique number), while when using relative patterns the locations are represented using descriptors. The significance is that absolute patterns can only be used with respect to simulations that feature the same environment as that from which the patterns were mined, while relative patterns are more versatile and can be used for a variety of simulations. However, relative descriptors are more complex. In this chapter both mechanisms are considered in further detail in the context of MPs. It should also be noted here that the extracted MPs provided a "knowledge base" with which to drive the desired MABS. The operation of this MABS is presented in the following chapter, Chapter 5.

The remainder of this chapter is organised as follows, Section 4.2 introduces the generic structure and formalism of the MP concept. Sections 4.3 and 4.4 then consider the two MP representations mechanisms, absolute and relative, respectively. The following section, Section 4.5, then discuses the advantages offered by the relative mechanism over the absolute. This is followed by Section 4.6, which discusses (and demonstrates) the concept of states as incorporated into MPs. Section 4.7 describes the process of extracting MPs from the location information extracted from video data using VDAS. The chapter is summarised in Section 4.8.

4.2 Formalism

This section presents a formal definition of the concept of MPs. The fundamental structure of an MP is that of a tuple of the form:

$$MP = \langle F, S, v, Path \rangle$$

Where: (i) F is the "From" location (where the movement represented by the MP starts); (ii) S is a collection of zero, one or more states describing the spatial relationship between agents featured in a scenario; (iii) v is a *movement vector* (as introduced in Section 3.5 in the previous chapter); and (iv) $Path$ is the *path*, encapsulated by the MP, which an agent needs to follow to get to the "To" location. The nature of these elements is discussed in further detail in the remainder of this section.

The From location (F) is the start location in the grid environment from where the movement described by a MP commences. The variable F is itself a tuple of the form:

$$\langle loc\_ID, div\_label \rangle$$

where: (i) where loc_ID is a location identifier and (ii) div_label is a *division label.* The format of the variable loc_ID is dependent on whether we are considering absolute MPs or relative MPs. In the first case it will simply be a grid cell number, in the second case it will be a location descriptor. Recall that location descriptors were described in Chapter 3, Section 3.6. The div_label variable is used where the environment is divided into a number of *divisions* as in the case of scenarios that fall into the Complex Scenario category; one of the three categories of scenario considered in this thesis. Where appropriate it is drawn from a set of labels $\{d_1, d_2, \dots\}$. If a scenario features no divisions the variable is simply omitted.

Each state s in S defines the relative relationship between two agents using a set of labels, {"ignore", "closeby","meeting", "follow"} defined using a set of concentric zones as explained previously in chapter 3. With respect to the work presented in this thesis a number of different states are defined, details are given later in this chapter in Section 4.6. An MP can feature zero, one or more states depending on how many agents feature in a scenario. If we have n agents then $S = \{s_1, s_2, \dots, s_{n-1}\}$, we are not interested on how an agent relates to itself hence $n - 1$. In case of multiple mice scenario, the set of state will be going longer, thus it is suggested to use count of instances for different states. If there is only one agent, then $S = \emptyset$. Note that a MP defines movement in terms of a single entity.

The element v of the MP tuple, as already noted, is a movement vector. The concept of movement vectors was discussed previously in Chapter 3. The value for v can be expressed as a single number v or as a coordinate pair $\langle x, y \rangle$ depending on whether we are using absolute or relative movement patterns. Recall that in the absolute case the movement vectors are applicable only in the context of environments that feature a particular size (length and width), the nature of absolute movement vectors will be different for different environments, not the case with respect to relative movement vectors (although consequently they are expressed using two values rather than one). The value of v when applied to an agent's current location indicates the "To" location associated with the MP.

The fourth component of the MP tuple, $Path$, as already noted, indicates the "route" that the MP prescribes whereby an agent adopting the MP can get from its From location to the indicated To location (indicated by v). The number of elements in $Path$ ($|path|$) depends on how far we wish to "look ahead". With respect to the evaluation and case studies presented later in this thesis $|path| = 5$ was used [1]. Using $|path| > 1$ means that our rodent agents have a "memory", they have a planned route they wish to follow. The elements of $Path$ are all movement vectors. Thus using absolute movement patterns, where $|path| > 1$, we have a sequence of movement vectors of the form $\{v_1, v_2, \ldots, v_{|path|}\}$ were $v_{|path|}$ indicates the end location (the same location as indicated by the variable v included in all movement patterns) and the remaining vectors indicate locations at intermediate locations, we refer to these locations as *waypoints*. In the case of relative movement patterns where $|path| > 1$, we have a sequence of movement vectors of the form $\{\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_{|path|}, y_{|path|}\rangle\}$.

4.3 Absolute Movement Patterns

Absolute Movement Patterns (AMPs) are derived from absolute location data. As already established, absolute locations are recorded relative to the origin of the environment. More specifically the grid numbering introduced in the previous chapter. These numbers were thus used to define the From location F in a given MP tuple. Similarly the value for v, and the contents of the set $Path$ were defined in terms of absolute values from the origin. Some example AMPs, in terms of the three categories of scenario considered in this thesis, are presented below.

In the case of the Simple Scenario category the scenarios featured only one agent (rodent) and no divisions, thus MPs in this case had the form:

$$\langle\langle loc\_ID\rangle, v, Path\rangle$$

An example is given in Figure 4.1. In the figure the orange colour indicates nest location, green indicates a wall proximity area and white indicates open space. Figure 4.1 features a 10×10 environment with cells numbered using absolute addressing and a set of rodent locations at times t_0 to t_5. In the figure (and similar figures used later in this chapter) the solid disc represents the location at t_0, the open double edged disc represents the location at t_5 and the intervening open discs the locations in between at times t_1, t_2, t_3 and t_4. The edges connecting the discs are simply intended to indicate the sequence of location visits, not the actual paths followed. Assuming $|Path| = 5$, we would extract an AMP of the form:

---

[1] The reason why $|path| = 5$ is explained in Chapter 6.

$$\langle\langle 2\rangle, 18, \{41, 49, 6, -8, -70\}\rangle$$

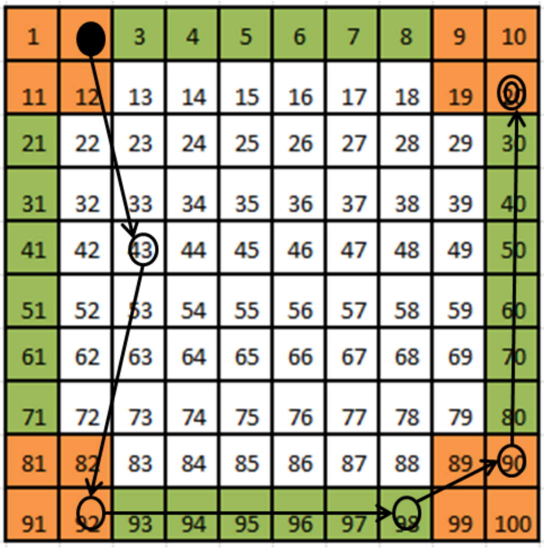


FIGURE 4.1: Example environment with absolute addresses in the context of the Simple Scenario category

In case of the complex scenario category the scenarios of interest featured a number of divisions connected by tunnels, but again only one agent (rodent). Thus the structure of the AMPs was:

$$\langle\langle loc\_ID, div\_label\rangle, v, Path\rangle$$

An example is given in Figure 4.2 which features a complex scenario with three divisions, connected by two tunnels, and a set of rodent locations at times t_0 to t_5. In the figure different colours are assigned to indicate different areas and objects within the environment as follows: (i) green indicates wall proximity, (ii) yellow indicates blocked areas, (iii) white indicates open space, (iv) blue indicates tunnel area, (v) red indicates gate cells, (vi) brown nest locations and (vii) gray water points. Assuming $|Path| = 5$, we would extract an AMP of the form :

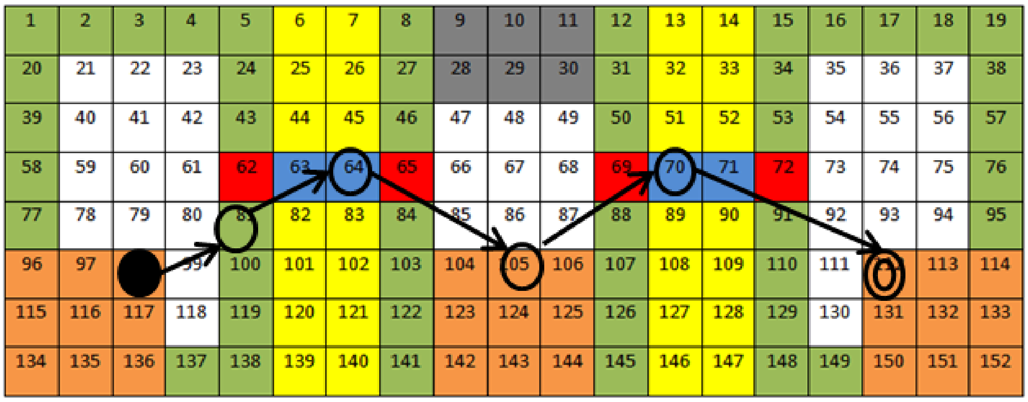
$$\langle\langle 98, \text{“}L\text{”}\rangle, 14, \{-17, -17, 41, -35, 42\}\rangle$$

Note that in this case the adopted set of division labels is $\{L, M, R\}$ indicating the left, middle and right divisions respectively. The adopted absolute addressing is as indicated by the cell numbering shown in the figure.

Finally, in case of the Two rodent Scenario category the scenarios will feature more than one agent (rodent). Consequently AMPs were structured as follows:

$$\langle\langle loc\_ID, div\_label\rangle, S, v, Path\rangle$$

Figure 4.3 shows an example Two-rodent scenario featuring two agents (rodents) A and B and a set of six locations for each agent at times t_0 to t_5. Again, in the figure the

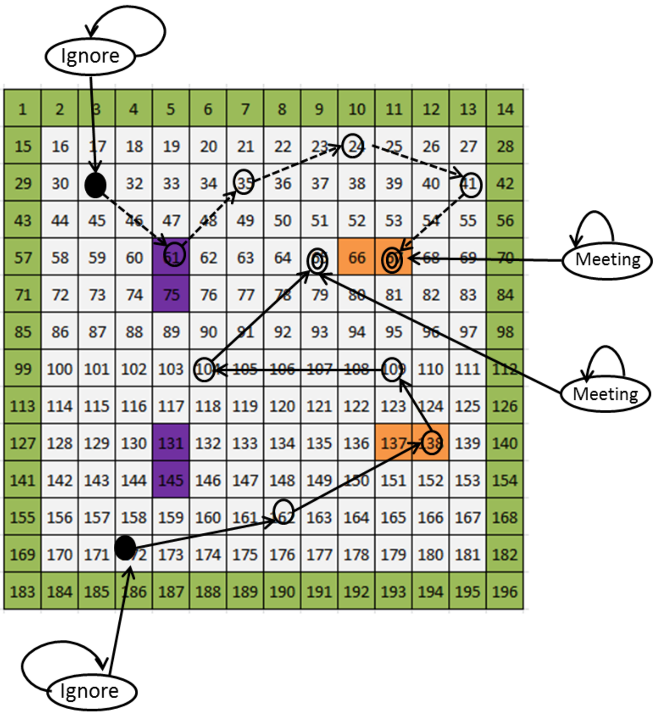


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76
77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133
134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152

FIGURE 4.2: Example environment with absolute addresses in the context of the Complex Scenario category

colours indicate different areas and objects, for example orange indicates a nest site and purple indicates an obstruction object. The AMPs associated with agents A and B respectively, assuming $|Path| = 5$, will then be :

$$\langle\langle 31\rangle, \{Ignore\}, 36, \{30, -26, -11, 17, 26\}\rangle$$
$$\langle\langle 172\rangle, \{Ignore\}, -42, \{-10, -24, -29, -5, -39\}\rangle$$



1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126
127	128	129	130	131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150	151	152	153	154
155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192	193	194	195	196

FIGURE 4.3: Example environment with absolute addresses in the context of the Two-rodent scenario category

4.4 Relative Movement Patterns (RMPs)

Using Relative Movemet Patterns (RMP) the idea was to use location descriptors whereby a location is described by its location ground type and that of its immediate neighbours as described in Chapter 3. The important thing to note here is that the relative mechanism will work regardless of the nature of the grid size or the playing area size. Some example RMPs, in terms of the three categories of scenarios considered in this thesis, are presented in the remainder of this section.

In the case of the Simple Scenario category, as in the case of AMPs, RMPs will have the following structure:

$$\langle\langle loc\_ID, div\_label\rangle, \text{v}, Path\rangle$$

Figure 4.4 shows the same scenario as shown in Figure 4.1 but with ground types instead of cell numberings. The associated RMP in this case, assuming $|Path| = 5$, will be:

$$\langle\langle nn - nn - - - -\rangle, (-2, 0), \{(-3, -6), (-5, -2), (-1, 3), (1, 4), (6, 1)\}\rangle$$

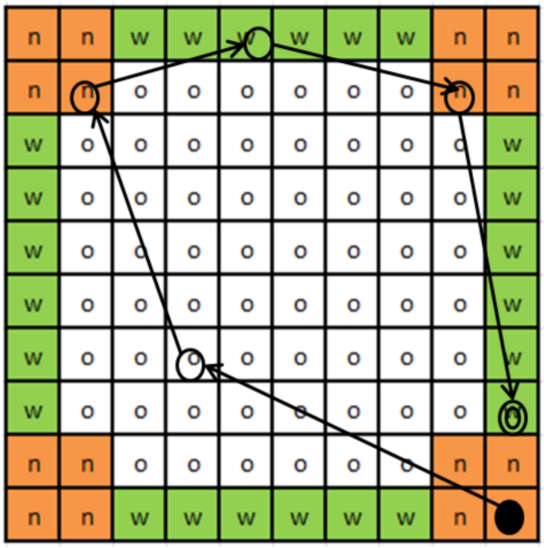


FIGURE 4.4: Example environment with ground types in the context of the Simple Scenario category

In the context of the Simple scenario considered for evaluation purposes with respect to the work presented in this thesis the 45 different potential descriptors are listed in Appendix A (Table A.1). Thus, considering the descriptor for the grid cell at the top left hand corner of Figure 4.4, which in turn represents a nest location (see Figure 3.3 in Chapter 3), this is described by the descriptor $\langle - - - - n\ n - n\ n\rangle$ indicating that five of the neighbourhood locations are outside of the playing area.

In the case of the Complex Scenario category the RMPs will again have the same format as in the case of AMPs:

$$\langle\langle loc\_ID, div\_label\rangle, \text{v}, Path\rangle$$

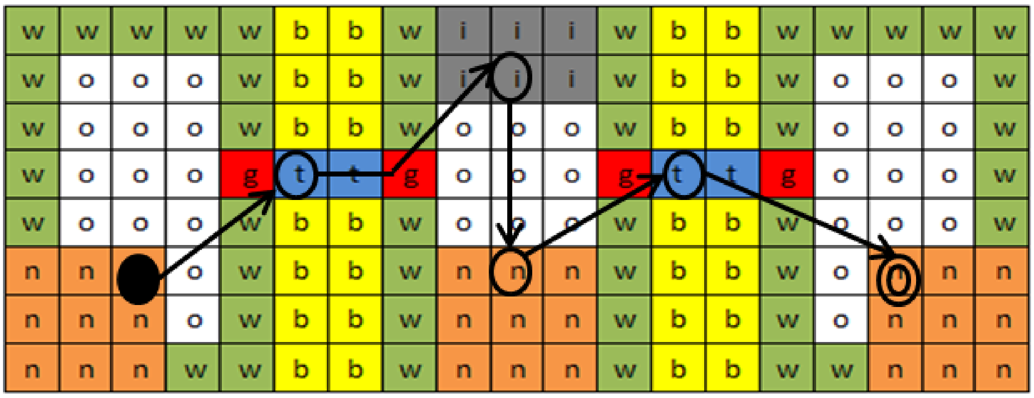


FIGURE 4.5: Example environment with ground types in the context of the Complex Scenario category

Figure 4.5 shows the same configuration as in Figure 4.2 but with ground type identifiers instead of cell numbers. The associated RMP, again assuming $|Path| = 5$, would be:

$$\langle\langle ooonnonno, \text{“}L\text{”}\rangle, (0, 14), \{(-2, 3), (-2, 4), (4, 0), (-2, 3), (2, 4)\}\rangle$$

In the context of the Complex scenario category simulations considered for evaluation purposes with respect to the work described in this thesis the 119 different potential relative descriptors as listed in Appendix A (Table A.2). The identified set of location ground types labels in this case is $L = \{b, g, i, n, o, w, t, -\}$ as discussed previously in Chapter 3. Note also that with respect to Figure 4.5 the environment features three divisions: L (left), M (Middle) and R (Right).

Finally, in case of the Two-rodent scenario category RMPs will have the form:

$$\langle\langle loc\_ID, div\_label\rangle, S,\text{v}, Path\rangle$$

Figure 4.6 shows the same example Two-rodent scenario as shown in Figure 4.3 but with ground type labels. In the Figure 4.6, RMPs have been shown along with the states. The RMPs associated with the two agents will then be:

$$\langle\langle wwwoooooo\rangle, \{Ignore\}, (8, -7), \{(3, -1), (-3, -3), (1, -6), (4, 1), (2, 3)\}\rangle$$
$$\langle\langle ooooooooo\rangle, \{Ignore\}, (-1, 3), \{(3, 2), (0, 6), (-3, 1), (1, -4), (-1, -3)\}\rangle$$

4.5 Advantages of relative mechanism over absolute mechanism

This section considers the advantages and disadvantages of the two representations for movement patterns. Starting with absolute movement patterns the main advantage with respect to relative movement patterns is that the absolute addressing mechanism used with respect to absolute MPs is succinct and easy to apply.

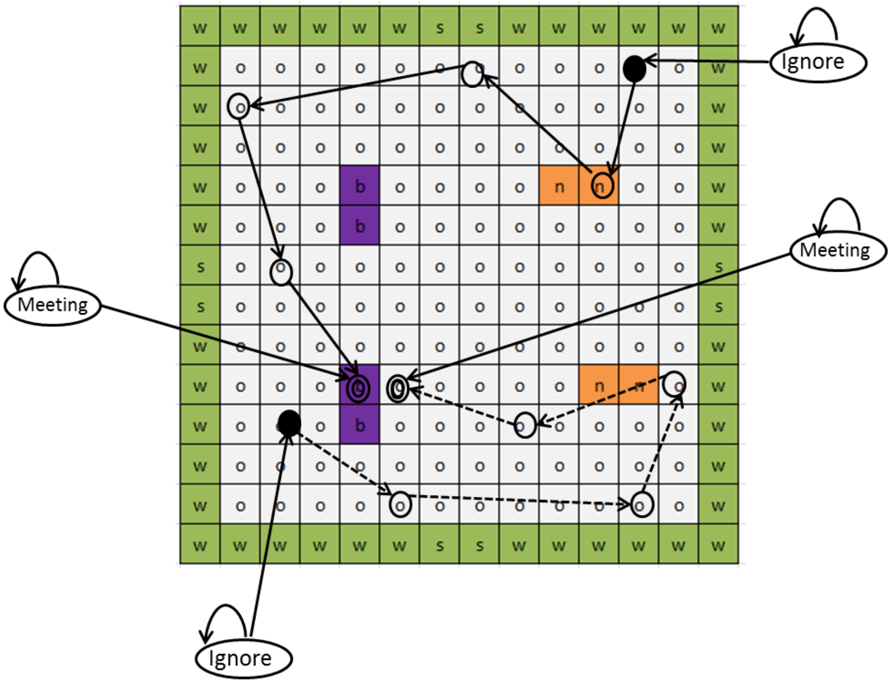


FIGURE 4.6: Example environment with ground types in the context of the Two-rodent Scenario category

With respect to relative MPs the advantages are firstly that relative MPs are more generic than absolute MPs and consequently have wider applicability. Absolute MPs are only applicable to the specific environment with respect to which they were defined. Thus with respect to simulations absolute patterns can only be used in the context of identical environments. If we wish to conduct simulations using a smaller/larger environment we would have to collect further appropriate raw video data and repeat the movement pattern mining process. As the collection of raw video data is both time consuming and resource intensive this is not desirable. Also the idea of computer simulation is that the simulation is as generic as possible so that users can run a variety of scenarios and conduct "what if" style experiments. Secondly using the relative representation for MPs fewer location patterns are typically required than the number of cell labels; thus the relative representation, in addition to being more generic, also offers some simplification. However, a disadvantage of relative MPs, with respect to work presented in this thesis, is that the relative descriptors used are currently rotation invariant, for future work the intention is to derive descriptors that are rotation variant in which case the number of required descriptors will be decreased.

In summary the advantages of using relative patterns over absolute patterns are as follows:

1. Fewer location patterns are typically required than the number of cells in the grid.
2. They are more generic than absolute patterns.

3. They can be used for larger/smaller gird sizes (environments) and to generate them.
4. They are potentially rotatable and have a mirroring property.

To illustrate point 3, the reader might consider the relative patterns generated using (say) a 10×10 grid, and then equally be used in the context of a 20×20 grid or a 30×30 grid.

4.6 States

An important aspect of movement patterns in the context of the Two-rodent scenario category is the concept of states. As noted previously, a state defines the relationship between two entities (mice in our case) in such a way that this can be incorporated into a MABS environment. We have four different states {"ignore", "closeby","meeting", "follow"} arranged in a state graph as illustrated in the Figure 4.7. In the figure the nodes represent the four individual states and the directed edges indicate the follow-on state for each individual state. The figure shows the possible state changes; every state is accessible from the "closeby" state. A state changes from "ignore" to "meeting" or follow can only occur via the "closeby" state. Note also that states can follow on from themselves (they self reference). The state names used are indicative of the relationship between the two entities of interest. Note also that states are defined on the basis of "zones" as discussed in Chapter 3 and the relative direction θ of one from the other. More specifically the states are defined as follows.

Ignore: Describes the situation where the two entities are so far apart that their mutual presence is not considered to impact on one another. The ignore state is defined in terms of a $d_{pz} \times d_{pz}$ "proximity zone" defined about each entity's location where d_{pz} is measured in terms of a number of grid squares. If two entities are outside of one another's proximity zones they are said to be in an "ignore state". In the case of the mouse behaviour MABS of interest with respect to this thesis, $d_{pz} = 7$ grid squares are chosen. The definition of proximity zones is of course application dependent and has to be set in consultation with domain experts. A simpler mechanism for defining the ignore state is to say it is the default state used where none of the other proposed states are applicable.

Meeting: The meeting state describes the situation where two entities are in close intrusion zone. Intrusion zone is define in terms of a $d_{iz} \times d_{iz}$ "intrusion zone" (not the same as an proximity zone) defined about each entity's location where d_{iz} is again measured in terms of a number of grid squares. A good heuristic for determining d_{ip} is $d_{ip} = \frac{d_{pz}-1}{2}$. In the case of the mouse behaviour MABS presented in this thesis, $d_{iz} = 3$ thus adopted. If two entities are within one another's proximity zones, regardless of whether they are moving or stopped, they are considered to be in a "meeting" state.

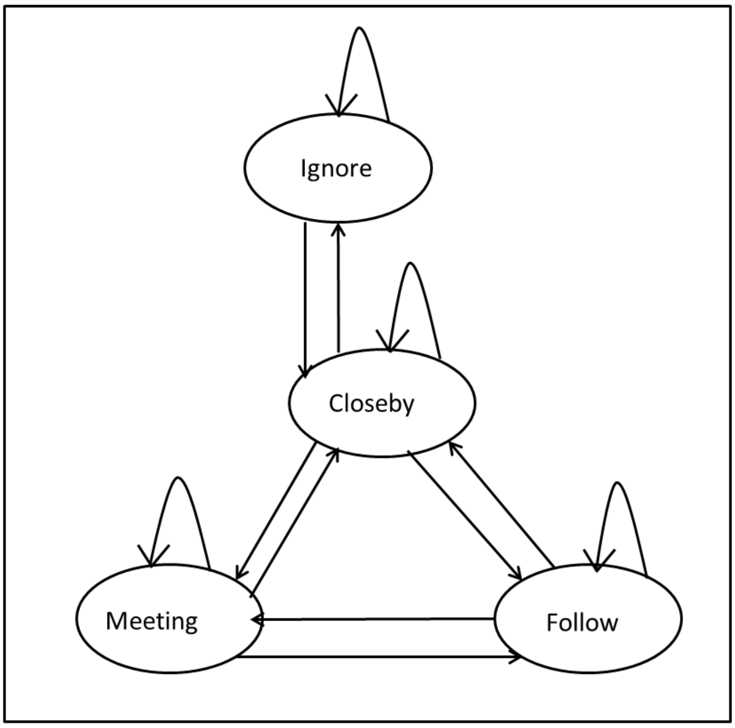


Figure 4.7: State Graph

Closeby: Two mice are in a closeby state if they are within each other's proximity zones but not within each others intrusion zones and do not subscribe to the definition for the following state (see below).

Follow state: Describes the situation where two mice are in a "closeby" situation and both are moving in approximately the same direction so that one can be said to be following the other. The following situation was found to be a relatively frequent occurrence with respect to the sample video data which was used to both drive and evaluate the work presented in this thesis, hence a specific follow state was defined. The following state exists, given a current agent m_1, and a second agent m_2, if:

1. m_1 and m_2 are in a "Closeby relationship".
2. The bearing (α) of m_2 from m_1 at time t_1 is equal to the direction of travel of m_1 within a tolerance of θ,
3. The direction of travel of m_2 is equal to direction of travel of m_1, within a tolerance of ω.

Note that for the "Follow" state to be applicable all of the above conditions should exist. The first condition expresses the requirement that both agents should be in a "Closeby" relationship, the second condition that one of the agents should be approximately in front of the other, and the third condition that the two agents should be travelling in approximately the same direction.

Algorithm 2 Algorithm for States

```
 1: INPUT:
 2:  location ($l_1$) and direction ($h_1$) of agent $m_1$
 3:  location ($l_2$) and direction ($h_2$) of agent $m_2$
 4: D = Proximity zone expressed as a set of locations about $m_1$
 5: E = Intrusion zone expressed as a set of locations about $m_1$
 6: OUTPUT: State for $m_1$
 7: c = $Threshold_1$
 8: k = $Threshold_2$
 9: if ($l_1 \in E$) then
10:     state ← "Meeting"
11: else if ($l_1 \in D$) then
12:     $\alpha \leftarrow$ bearing ($l_1$, $l_2$)
13:     $\theta \leftarrow$ arc defined by ($\alpha + c$ and $\alpha - c$)
14:     $\omega \leftarrow$ arc defined by ($\alpha + k$ and $\alpha - k$)
15:     if ($h_1 \in \theta$ and $h_2 \in \omega$ ) then
16:         state ← "Follow"
17:     else
18:         state ← "Closeby"
19:     end if
20: else
21:     state ← "Ignore"
22: end if
```

The algorithm for determining the state of a rodent entity is given by the pseudo code presented in Algorithm 2. The pseudo code describes the process for finding the appropriate state label for a given agent. The input is the locations and directions for a pair of agents m_1 and m_2; that is, location l_1 and direction h_1 for agent m_1 and the same for agent m_2. The algorithm also takes as input a proximity zone (D) and an intrusion zone (E), these zones are defined in terms of sets of locations about agent m_1. The output is one of the four identified states in the set S. In line 9 the algorithm checks whether location l_1 is located within the intrusion zone, if so the state is "meeting". In line 11, the algorithm checks whether location l_2 is within the proximity zone and that direction of m_1 is within θ and direction of m_2 is within ω so if the state is "follow"; otherwise the state will be "closeby". If none of the condition are met the default "ignore" state is selected.

Figure 4.8 illustrates the "follow" state concept. The figure features an agent m_1 at location l_1 shown as a black coloured circle, and an agent m_2 at location l_2 shown as a red coloured circle at time t_1. The bearing α of agent m_2 is found from locations l_1 and l_2. Arc θ is defined by (α+ c) and (α− c) while arc ω is defined by $(\alpha + k)$ and $(\alpha - k)$[2].

---

[2] For the purpose of the evaluation presented later in this thesis $c = 15°$ and $k = 45°$ was used.

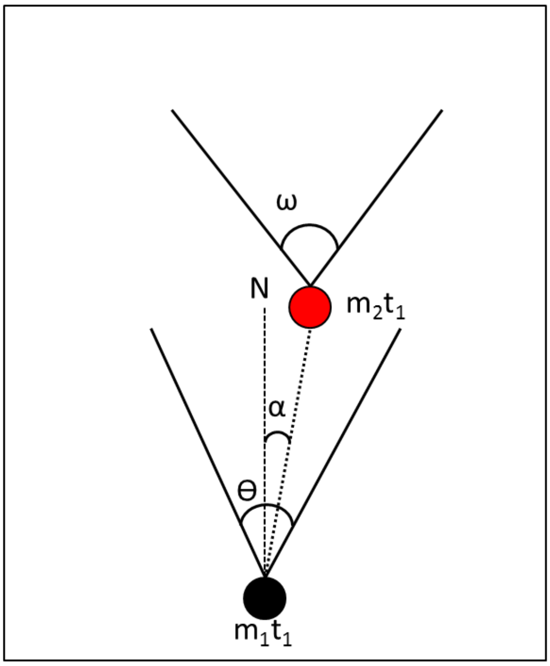


FIGURE 4.8: Arcs of agents for defining the "follow" state

4.7 Movement Pattern Mining

This section describes the process whereby MPs are mined from the location information extracted from video data using a VDAS system, as described in the forgoing section. The location information is in the form of location IDs each associated with a frame number t. The interval between locations is some number of video frames i. Thus if we have a location recorded at time t the next location will be recorded at time $t + i$, and so on. The nature of i will depend on the simulation application domain under consideration. In the case of the rodent simulations considered in this thesis, $i = 25$ was used with respect to scenarios featuring rats and $i = 5$ with respect to scenarios featuring mice. Location information was stored in a text file format as shown in Table 4.1. In the table the first column lists the time stamp (frame number) and the second the associated location ID. Note also that with respect to the example shown in the Table $i = 25$ was used.

The pseudo code for the MP mining is given in Algorithm 3. The algorithm takes as input: (i) a set of sets of locations, $L = \{L_i, L_2, \ldots, L_n\}$, where n is the number of objects (agents) being tacked; (ii) a desired path size k; (iii) a set of agents ($A = \{A_1, A_2, \ldots, A_n\}$); and (iv) the desired representation type, *repType*, either: absolute or relative. The output is a set of MPs, M, for later use.

The first step is to generate a set of sets of location sequences, $Q = \{Q_1, Q_2, \ldots, Q_n\}$, where each sequence is of length k. Table 4.2 shows the sequences that will be extracted from the location data example given in Table 4.1 assuming $k = 5$.

The remainder of the algorithm comprises two loops (lines 2-18 and lines 20-26). The first loop is responsible for mining the set of sets S to extract all the MPs contained

Table 4.1: Example location information extracted video data using $i = 25$

Time	Location ID
25	1
50	2
75	4
100	7
125	8
150	19
175	80
200	90
...	...

Table 4.2: An example set of location IDs in a form of sequences using the location information from Table 4.1

Loc_t	Loc_{t+i}	Loc_{t+2i}	Loc_{t+3i}	Loc_{t+4i}
1	2	4	7	8
2	4	7	8	19
4	7	8	19	80
7	8	19	80	90
8	19	80	90	90
...	...	...	...	...

there in, while the second loop is responsible for determining the frequency with which each MP occurs and generating the output set M. Recall that MPs are of the form:

$$\text{MP} = \langle \text{F, S, v, Path} \rangle.$$

where: F is the from location, S is the set of states (each state indicating the relationship between the current agent and another agent in the scenario), v is the movement vector linking the from location to the desired "To" location, and $Path$ is the path (of length k) that the current agent should follow to get to the To location.

In the first loop the value for F is instantiated on lines 4 to 8, S on lines 9 to 12, v on line 13 and $Path$ on line 14. The algorithm features 4 functions: (i) *genDescriptor* (line 7), (ii) *calcState* (line 10), (iii) *calcVector* (line 13) and (iv) *calcPath* (line 14), each is discussed in further detail below.

1. *genDescriptor*: This function is used to generate a descriptor of a current location where the relative representation is used (descriptor generation was discussed in Chapter 3, Section 3.6).

2. *calcState*: This function calculates the state describing the relationship between the current agent, agent A_i, and another agent A_j in the scenario, by considering the first two locations in the sequence table of each agent. Note that the calcState function is only called when considering a multiple mice scenario. The state concept was discussed in detail in Section 4.6; states are calculated according to Algorithm 2 also given in Section 4.6.

Algorithm 3 Mining MPs from Location DataBase

INPUT:
$L = \{L_1, L_2, \ldots, L_n\}$, a set of sets of Locations (as shown in Table 4.1)
k, path size
$A = \{A_1, A_2, \ldots, A_n\}$, a set of agents
$repType$, the desired representation type (absolute or relative)

OUTPUT: M, a set of MPs with counts

```
$Q$ = A set of sets of location sequences of the form $Q = \{Q_1, Q_2, \ldots Q_n\}$
for ($i = 1 \rightarrow |A|$) do
    for ($j = 1 \rightarrow |L_i|$ in steps of $k$) do
        if (repType == absolute) then
            $F = Q_{i,j}(0)$
        else
            $F = genDescriptor(Q_{i,j}(0))$
        end if
        for ($a = 1 \rightarrow |A|$ and $a \neq i$) do                ▷ if more than one rodents
            $state = calcState(L_{i,j}(0), L_{i,j}(1), L_{a,j}(0), L_{a,j}(1))$
            State = State ∪ state
        end for
        $v = calcVector(repType, L_{i,j}(0), L_{i,j}(k))$
        $Path = calcPath(repType, \{L_{i,j}(0), \ldots, L_{i,j}(k)\})$
    end for
    $mp = \langle F, State, v, Path \rangle$
    $MP = MP \cup mp$
end for
$M = \emptyset$
for ($i = 1 \rightarrow |MP|$) do
    if ($mp_i \in M$) then
        Increment count
    else
        $M = M \cup \langle mp_i, 1 \rangle$
    end if
end for
Return $M$
```

3. *calcVector*: This function is used to calculate the movement vector v by subtracting the first location of the current sequence, $Q_{i,j}(0)$, from the last location in the current sequence $L_{i,j}(k)$. In case of the absolute mechanism, locations are represented in an integer format, while in case of the relative mechanism locations are represented in x, y component format. This is why the function needs *relTyp* as an argument. The mechanism for finding movement vectors was discussed in Chapter 3, Section 3.5.

4. *calcPath*: This function calculates the path element of a MP (*Path*). The function takes the current sequence, S_{i_j}, as an input and calculates the path part for the MP. Recall that a path is defined in terms of a set of points (waypoints) obtained by

subtracting each current location ID from the previous location ID of the sequence. In the case of the absolute mechanism location IDs are integer numbers, but in the case of the relative mechanism location IDs are representing in term of x, y components. Thus the function needs $relTyp$ as one of its arguments. The concept of paths was discussed in Section 4.2.

In the second loop (lines 19-26) the set-off MPs is refined so that we have a set M comprising tuples of the form $\langle mp, count\rangle$. These counts are used with respect to the MP selection strategy used to drive MABSs as described in the following chapter. The algorithm terminates (line 27) by returning the set M. The generated MPs are stored in a movement pattern database so that each MP has its count stored within it.

4.8 Summary

This chapter has presented the formalism for the Movement Pattern (MP) concept. Essentially a MP is a four part tuple that takes the form $\langle F, S, v, Path\rangle$. Two different types of mechanisms were considered: (i) absolute and (ii) relative. Both mechanisms were also considered in terms of the three scenario categories considered in this thesis. At the end of the chapter the idea of states, used in connection with the two-rodent scenario category was presented. States defines the relationship between two entities (rodents in our case). In the following chapter mechanisms whereby MABS can operate using the concept of movement patterns, as described in this chapter, are explored and discussed.

Chapter 5

MABS Operation Single and Two Rodent Scenarios

5.1 Introduction

This chapter describes how Movement Patterns (MPs), as presented in the previous chapter, Chapter 4, can be utilised in a Multi-Agent Based Simulation (MABS) framework. In other words, how MPs can be used to drive a (rodent behaviour) MABS. A block diagram outlining the components of the envisioned MP driven MABS, with respect to rodent simulation, is presented in Figure 5.1. In the figure the part of the framework in which the agents operate, rodents in our case, is conceptualized in the form of cloud in which agents and other objects within the simulation exist. In the figure a number of example objects have been included, namely nests, tunnels and divisions (we can think of others). The cloud has an interface, having two parts: a user interface and a visualisation interface. The user interface allows users to interact with the MABS, while the visualization interface allows uses to observe a simulation as it progresses. On the edge of the cloud (in Figure 5.1) there is also a simulation controller, which interfaces with a database of MPs and provides a mechanism whereby agents can select MPs to be implement by them. How this database is used is the central theme of this chapter. For analysis purposes, as the simulation progresses, the simulator controller collects agent locations and stores them in a location database; this also feeds the visualization software.

In the real world scenario (video data), it has been observed that, mice can group together at any location (e.g. nest location etc.) therefore collision might occur. In small population, collisions in real world are smoothly resolved and mice reach the desired destinations. Therefore, neither collision was considered in this research work nor a specific procedure was implemented for it, though it could be an interesting aspect in future work studying larger groups (in which collisions could be more often and result in changing the destinations). Algorithm 4 presents some high level pseudo code describing the operation of the proposed MP driven MABS. As it is difficult to know in advance,

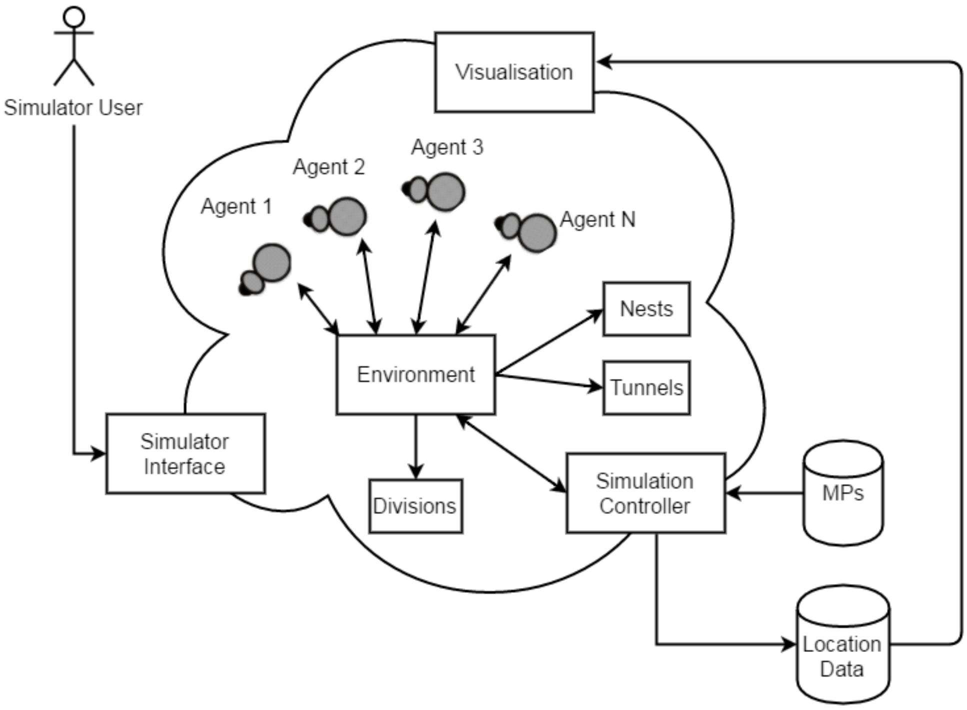


FIGURE 5.1: Components of envisioned MP driven MABS with respect to rodent simulation

that, what will be the next location of other mouse agent at the time of MP selection. Thus to prevent mouse agent to go to a location, where already another mouse agent approaching, is not considered in the algorithm. The algorithm takes as input: (i) a set of agents $A = \{a_1, a_2, a_3, \ldots, a_n\}$ where n is the number of agents in the simulation, (ii) an environment and (iii) a database of MPs. Output from the algorithm is a simulation that can be observed using the visualization interface (see Figure 5.1). The algorithm commences by first "setting-up" the simulation environment, this is discussed further in Sub-Section 5.3.1 below. The algorithm then enters a loop (lines 2 to 10) and processes each agent in A. If an agent has a path which it is following (line 4), the location of the agent is updated as indicated by the path. Otherwise (line 7) the agent selects a (new) MP.

The rest of this chapter is structured as follows. Section 5.2 presents an overview for the process of selecting MPs. This is followed in Section 5.3 with details concerning the implementation of the MP driven MABS framework presented in Figure 5.1. The chapter is concluded, in Section 5.4, with a summary and a "look ahead".

5.2 Movement Pattern selection

The fundamental idea underpinning the envisioned MABS is that the simulations are driven using MPs extracted (learnt) from video data. The conjectured advantage is that the resulting simulations will be more realistic and that MP generation will be much less resource intensive than in the case where agent behaviour is hand coded. On

Algorithm 4 Pseudo Code illustrating the high level operation of an MP directed MABS

INPUT: Set of Agents $A = \{a_1, a_2, a_3, \ldots, a_n\}$, Environment, DB of MPs
OUTPUT: A visualisation of the simulation

```
Environment set up
while (true) do
    for ($i = 1 \rightarrow |A|$) do
        if ($A_i.hasPath$) then
            update location for $A_i$ according to path
        else
            select new MP for $A_i$ (Algorithm 5)
        end if
    end for
end while
```

start up each agent needs to select an appropriate MP, implement it, and when the implementation is complete select a follow on MP, and so. The process was given by the pseudo code in Algorithm 4.

The idea is that the MP selection process is embedded in the MABS simulation controller (see Figure 5.1). The process for selecting MPs is illustrated in block diagram form in Figure 5.2, where the directed edges indicate process flow. With reference to Figure 5.2 the process starts by extracting all MPs, from the MP database, that feature the given agent's current location (the From location F in the context of the MPs) and, in the case of scenarios featuring more than one agent, the contents of the current agents set of states S describing its relationship with other agents that exists in the scenario. If there is no MP available these constrains are relaxed (right-hand branch of the block diagram) so that alternative, but similar, From locations are considered (this is discussed further in Sub-section 5.2.3 below). When the relative representation is adopted the next step is to prune those MPs which are not "legal". This is not necessary where the absolute mechanisms is adopted because absolute MPs will always be legal as they only operate with environments identical to those for which they are generated, not the case using relative patters but as a consequence such patterns may not always be legal. For a potential MP to be legal, the indicated To location, and any intervening way points, must be both *valid* and *accessible*. What is meant by the terms valid and accessible in this context is discussed further in Sub-section 5.2.1 below. If there is no MP available after pruning the set of MPs for the given From locations, the constrains for the From location are again relaxed in the same manner as in the case where no suitable MPs were found in the first place. After finding a set of available MPs, one MP is selected using a random weighted probability mechanism; this is discussed in further detail in Sub-section 5.2.2.

Algorithm 5 illustrates the MP selection process in further detail. The algorithm takes three inputs: (i) the current from location l_f for the agent under consideration, (ii) a database of MPs, D, and (iii) a set B of cell location IDs representing blocked

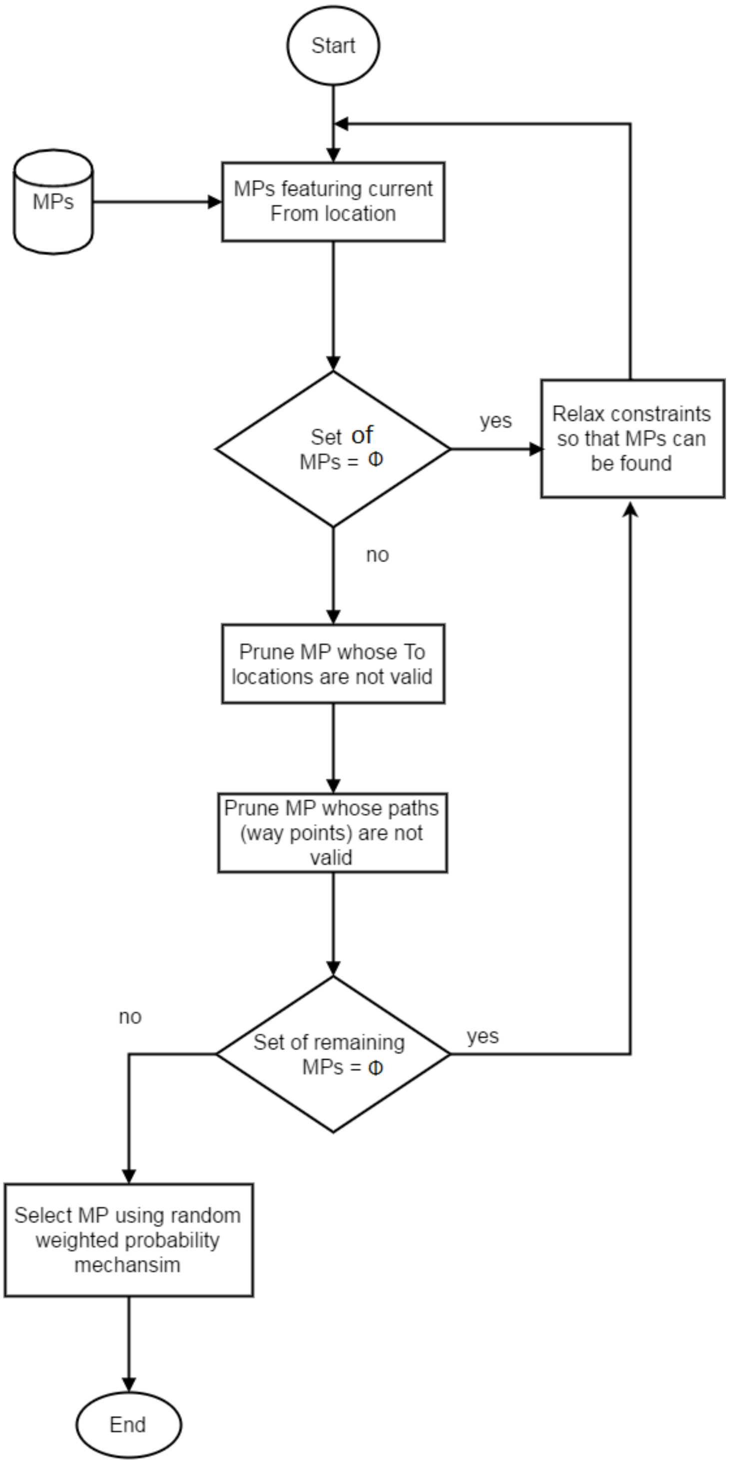


Figure 5.2: Overview of movement pattern selection process

areas. Output from the algorithm is a selected MP s. The algorithm consists of two loops one nested inside the other. The algorithm commences (line 2) by generating a set P of all MPs in D that feature the from location l_f. If the set P is empty then the alternative selection strategy is adopted (as discussed further in Sub-section 5.2.2), otherwise the algorithm loops through the set P (lines 6-18) and for each $p_i \in P$ extracts the To location l_t (line 7); this is calculated using the movement vector included in all MPs. The location l_t is then checked to determine whether it is "valid and accessible" using the function VALIDLOCATION(*cell*,*B*) (line 26). If l_t is not a valid location the associated MP, p_i, is removed from P and the next location in P processed. Otherwise the way points making up the $Path$ in p_i are checked for validity and accessibility (lines 11-16). The function VALIDLOCATION (*cell*,*B*) is again used for this purpose (line 13). If at the end of this process the set P is empty then the alternative selection strategy is again adopted. Otherwise weighted random probability selection is applied to the set P (line 23) and the selected MP s returned (line 24).

The function VALIDLOCATION (*cell*,*B*) in Algorithm 5 takes as input a given location ID and a set of blocked cell IDs. It also knows about the size of the environment. The function returns *true* if *cell* is valid and accessible, and $false$ otherwise. The function checks whether *cell* is in B, and if so returns $false$, and whether *cell* is outside the "playing area" in which case it also returns $false$, like wise the function checks (line 29) that there should be at least one MP in the MP database for the current *cell*. If both of the conditions mentioned in the function are true, function returns $true$, otherwise returns $false$.

5.2.1 Valid Accessible MP

As noted above, the selection of a MP to be implemented by an agent is done from a set of potential MPs that have been pruned so that MPs that are not "legal" have been removed. This subsection considers the legality of MPs in further detail. Note that this is of significance with respect to relative MPs only. To be legal a MP must be both: (i) valid and (ii) accessible. A MP is said to be valid if the indicated final To location, and the associated way points along the path to that To location, are inside the "playing area" and are not blocked or obstructed locations (ground types $-$ and b). A MP is said to be accessible (features accessibility) if:

1. The To location and associated way points can all be accessed from the agents current location.
2. The To location is also a From location in a least one MP in the MP database.

The significance of the later is so as to ensure that an agent will not get "stuck" at a location where it cannot move out of.

The above is illustrated in Figure 5.3 which shows three different situations involving a particular complex category scenario using relative MPs. The figure illustrates three

Algorithm 5 Selecting a valid and legal MP from a set of MPs

INPUT:

from location l_f

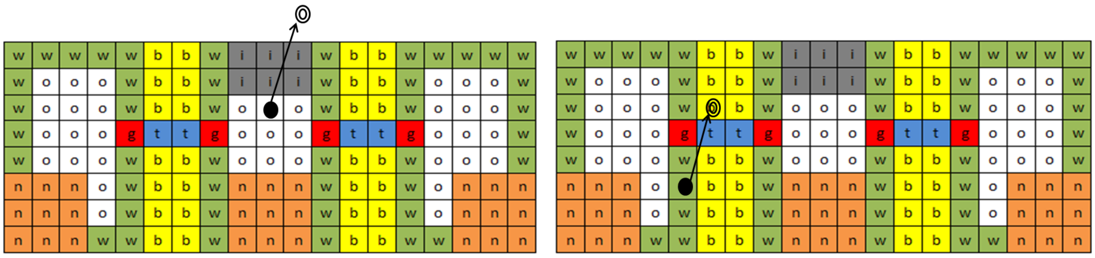
D a database of MPs

B = Set of cell identifiers representing blocked areas

OUTPUT:

s = a selected MP

```
main()
P = set of MPs $\{p_1, p_2, \dots\}$ that feature the From location $l_f$
if ($P == \phi$) then
    adopt alternative selection
else
    for ($i = 1 \to |P|$) do
        $l_t$ = To location extracted from $p_i$
        if (not validLocation($l_t$,$B$)) then
            remove $p_i$ from $P$
        else
            for ($j = 0 \to |P_i.Path|$) do
                cell = grid ID for waypoint $j$ in $P_i.Path_j$
                if (not validLocation(cell,$B$)) then
                    remove $p_i$ from $P$ and break
                end if
            end for
        end if
    end for
end if
if ($P == \phi$) then
    adopt alternative selection
else
    $s$ = random weighted probability selection applied to $P$
    return $s$
end if
function validLocation(cellID,$B$)
    if ($cellID \in B$ OR $cellID$ outside grid) then
        return false
    else if (cell features as a From location in at least one MP in $D$) then
        return true
    else
        return false
    end if
end function
```

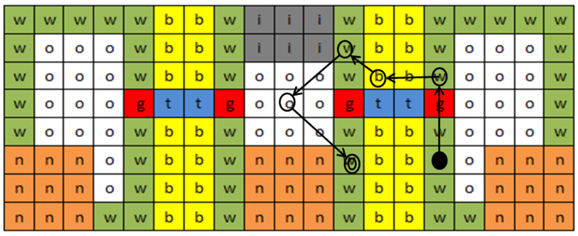


w	w	w	w	w	b	b	w	i	i	i	w	b	b	w	w	w	w	w
w	o	o	o	w	b	b	w	i	i	i	w	b	b	w	o	o	o	w
w	o	o	o	w	b	b	w	o	●	o	w	b	b	w	o	o	o	w
w	o	o	o	g	t	t	g	o	o	o	g	t	t	g	o	o	o	w
w	o	o	o	w	b	b	w	o	o	o	w	b	b	w	o	o	o	w
n	n	n	o	w	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	o	w	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	w	w	b	b	w	n	n	n	w	b	b	w	w	n	n	n

(a) Next location outside of environment

w	w	w	w	w	b	b	w	i	i	i	w	b	b	w	w	w	w	w
w	o	o	o	w	b	b	w	i	i	i	w	b	b	w	o	o	o	w
w	o	o	o	w	◎	b	w	o	o	o	w	b	b	w	o	o	o	w
w	o	o	o	g	t	t	g	o	o	o	g	t	t	g	o	o	o	w
w	o	o	o	w	b	b	w	o	o	o	w	b	b	w	o	o	o	w
n	n	n	o	●	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	o	w	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	w	w	b	b	w	n	n	n	w	b	b	w	w	n	n	n

(b) Next location is a cell located in a blocked area



w	w	w	w	w	b	b	w	i	i	i	w	b	b	w	w	w	w	w
w	o	o	o	w	b	b	w	i	i	i	w	b	b	w	o	o	o	w
w	o	o	o	w	b	b	w	o	o	o	w	b	b	w	o	o	o	w
w	o	o	o	g	t	t	g	o	o	o	g	t	t	g	o	o	o	w
w	o	o	o	w	b	b	w	o	o	o	w	b	b	w	o	o	o	w
n	n	n	o	w	b	b	w	n	n	n	w	b	b	●	o	n	n	n
n	n	n	o	w	b	b	w	n	n	n	w	b	b	w	o	n	n	n
n	n	n	w	w	b	b	w	n	n	n	w	b	b	w	w	n	n	n

(c) Invalid Path, way point in blocked area

FIGURE 5.3: Illustrations of invalid MPs

different conditions under which a MP can be rejected. The filled disk in each case indicates a mouse agent's current From location, while the double edged disk indicates a To location. Figure 5.3(c) includes a number of way points, while in the first two figures $|Path| = 1$.

Considering Figure 5.3(a) first, the From location is represented by the descriptor $\langle iiiooooooo, M\rangle$ and the movement vector v associated with a candidate MP is $(1, -3)$. Applying v to the From location (1 grid cell east and 3 grid cells north) we get a location which is outside the playing area. Hence the candidate MP is deemed to be illegal.

Moving on to Figure 5.3(b) the from location is represented by the descriptor ⟨ owbowbowb, L⟩. The movement vector v associated with the candidate MP under consideration in this case is again assumed to be $(1, 3)$. The To location in this case is in a blocked area so the candidate MP is again considered to be illegal.

Considering Figure 5.3(c) the From location descriptor is $\langle bwobwobwo, R\rangle$ and the movement vector associated with the candidate MP is assumed to be $(-3, 0)$ which gives a To location which is valid (within the playing area and not in a blocked area). However, in this case the candidate MP, taken from a MP database of the form shown in Table 5.1, the candidate MP has a path associated with it of the form: {(0,-3), (-2,0), (-1,-1), (-2, 2), (2, 2)}. The four intermediate way points are indicated in Figure 5.3(c). Inspection of the figure indicates that the second way point is located in a blocked area. Thus in this case the MP is also rejected on the grounds that it is invalid.

5.2.2 Random Weighted Probabilistic Movement Pattern Selection

From the foregoing as part of the MP selection process, given a set P of two or more legal MPs, all featuring a particular From location, an agent needs to select one. As

Table 5.1: Fragment of example MP database with respect to illustration given in Figure 5.3(c)

From Loc. (f)	Mov. Vector (v)	wp$_1$	wp$_2$	wp$_3$	wp$_4$	wp$_5$
$\langle bwobwobwo, R\rangle$	**(-3, 0)**	**(0, -3)**	**(-2, 0)**	**(-1, -1)**	**(-2, 2)**	**(2, 2)**
$\langle bwobwobwo, R\rangle$	(-2, -8)	(-1, 0)	(-1, -1)	(0, -3)	(0, -1)	(0, -3)
$\langle bwobwobwo, R\rangle$	(-2, -6)	(-2, 0)	(0, -3)	(1, -2)	(0, -1)	(-1, 0)
$\langle bwobwobwo, R\rangle$	(-2, -6)	(-1, 0)	(-1, -2)	(0, -2)	(1, -1)	(-1, -1)
$\langle bwobwobwo, R\rangle$	(-2, -6)	(-1, 0)	(-1, -1)	(0, -3)	(0, -1)	(0, -1)
$\langle bwobwobwo, R\rangle$	(-1, -5)	(-1, 0)	(-1, -1)	(0, -2)	(0, -1)	(1, -1)
$\langle bwobwobwo, R\rangle$	(-1, -5)	(-1, 0)	(-1, 0)	(0, -3)	(1, -2)	(0, 0)
$\langle bwobwobwo, R\rangle$	(0, -6)	(-1, 0)	(-1, -1)	(1, -3)	(0, -1)	(1, -1)
...	...	...	...	...	...	...

also noted in the foregoing, this is done using a random weighted probability mechanism. The weighed probability element is included to reflect the occurrence count of individual MPs, some are more likely (popular) than others. The random element is included so that when the same simulation is run several times the agents will not always behave in exactly the same manner, as would be expected given a real life situation. The process for determining weighted random probabilities is presented in this sub-section.

The weighted probability p_i associated with each MP in the set of movement patterns P is given by the occurrence count c_i in the MP database, normalised by dividing by the sum of all the occurrence counts of the MPs in P (Equation 5.1). Thus for each $p_i \in P$ we have a probability value p_i such that $0.0 \leq p_i \leq 1.0$ and the sum of all the p_i values equates to 1 ($\sum_{i=0}^{i=|P|} p_i = 1.0$).

$$p_i = \frac{c_i}{\sum_{j=0}^{j=|P|} c_j} \tag{5.1}$$

The random selection element can best be described by considering the probabilities to be sequentially arranged along a number line from 0.0 to 1.0 such that the MP associated with each probability is represented by a sub-range of the number line. In this manner the more probable MPs will be represented by larger sub-ranges. A random number is then generated between 0.0 and 1.0 and the MP indicated by the sub-range in which the number falls selected.

The weighted random probability selection process is illustrated by the pseudo code given in Algorithm 6. The algorithm takes as input a set of valid MPs (P). The output is a selected MP s from the set P. The algorithm commences by determining a $totalCount$ value in the loop (from line 3 to 5) . A random number r is then generated (line 6), which is used for the selection of an MP. In the second loop (lines 7 to 13) a variable n, incremented on each iteration, is used to select a MP s.

The process may be illustrated as follows. After the legality of a set of candidate MPs, has been checked we will be left with a set MPs from which one has to be selected. For example we might be left with the set of AMPs (Absolute MPs) listed in Table 5.2

Algorithm 6 Calculation of Weighted Random Probability for Selecting a Next MP from a legal set of MPs P

INPUT:
P = set of MPs
OUTPUT:
a selected MP from P

```
totalCount = 0
n = 0
for ($i = 0 \rightarrow |P|$) do
    totalCount = totalCount + $P_i$.count
end for
r = randomNumberGenerator()
for ($j = 0 \rightarrow |P|$) do
    if (r $\leq$ n + ($\frac{P_i.count}{totalCount}$)) then
        return $P_i$
    else
        n = n + ($\frac{P_i.count}{totalCount}$)
    end if
end for
```

Table 5.2: An example set of legal candidate MPs with their calculated probability values

No	From Loc. (f)	Movement Vector (v)	Path						
			wp_1	wp_2	wp_3	wp_4	wp_5	Count (c)	Prob.(p)
1	100	-32	-19	0	-17	2	2	5	**0.04**
2	100	-31	-19	-18	2	2	2	5	**0.04**
3	100	-31	-19	-18	3	2	1	8	**0.07**
4	100	-29	-19	-17	3	2	2	8	**0.07**
5	100	-28	-19	-17	2	3	3	10	**0.09**
6	100	-29	-19	-18	3	2	3	13	**0.11**
7	100	-30	-19	-18	3	2	2	14	**0.12**
8	100	-29	-19	-18	3	3	2	15	**0.13**
9	100	-29	-19	-17	2	2	3	17	**0.14**
10	100	-28	-19	-17	3	2	3	22	**0.19**

with respect to a From location with an absolute address of 100 (second column in the table). Note that, with respect to Table 5.2, the pattern numbers have been included simply to facilitate discussion. The third column in the table gives the movement vector v associated with each MP and the following five columns the way points assuming $|Path| = 5$. The second to last column contains the occurrence count for each MP. The probability p for each MP is then calculated as described above. This is given in the last column of the table. These probability values are then arranged along a number line as shown in Figure 5.4. If the value for the random number r is 0.14, MP number 3 will be selected, if $r = 0.7$ MP number 9 would be selected, and so on.

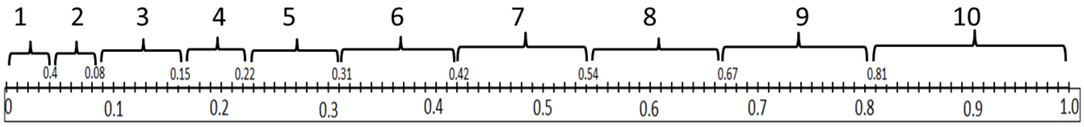


Figure 5.4: Selection of a Movement Pattern based on Random Weighted Probability using the set of MPs given in Table 5.2

5.2.3 Relaxation of MP Selection Criteria

As noted above, there is the potential, after all candidate MPs have been checked, that there are no MPs available from which a selection can be made. Although in practice this situation did not occur except in the two rodent scenario category. A contingency for this event needed to be included in the proposed MP driven MABS framework. This Sub-section considers the process whereby alternative MP selection can take place.

Table 5.3: Mechanisms for Alternative MP Selection

Mechanisms for alternative MP selection	Mechanisms applicability		Categories of scenarios		
	Relative	Absolute	Simple	Complex	Two rodents
State Replacement	✓	✓	×	×	✓
Descriptor Rotation	✓	×	✓	✓	✓
Mirroring	✓	×	✓	✓	✓
Nearest Location	✓	✓	✓	✓	✓

Table 5.4: Mutual similarities among states

States	Ignore	Closeby	Meeting	Follow
Ignore	10	8	4	4
Closeby	8	10	4	4
Meeting	4	4	10	8
Follow	4	4	8	10

Table 5.3 summarises four alternative mechanisms for MP selection and where they can be applied. In the table the first column lists the individual mechanisms: (i) State Replacement, (ii) Descriptor Rotation, (iii) Nearest Location and (iv) Mirroring. The second and third columns show the applicability of each mechanism with respect to the adopted MP representation (absolute or relative). Here it can be seen that although all four mechanisms are applicable in the context of relative movement patterns, only State Change and Nearest Location are applicable in the absolute case. The last three columns indicate the category of scenario where each of the mechanisms is applicable. Here it can be seen that State Change is only applicable with respect to the two rodent scenario category (because it is the only category, in the context of this thesis, that uses the concept of states). In the table the mechanism are listed in the ordered in which they

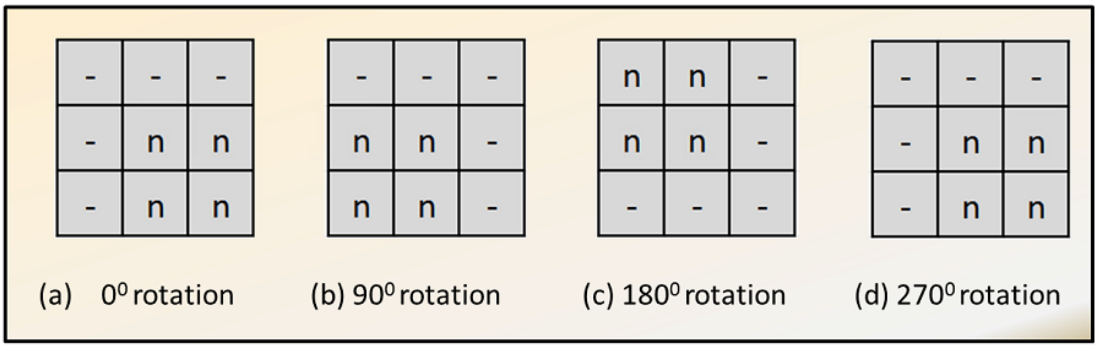
are applied, hence Nearest Location is listed last. Each of the mechanisms is considered in further detail below.

1. **State Replacement**. The State Replacement mechanism is applicable to both relative and absolute MPs. As discussed previously in Chapter 4, that concept of states exists only in the two-rodent scenario category, thus the state change mechanism is applicable only with respect to this category. The idea here is to replace the current state of an agent with the most similar state to the current state and attempt to find MPs using this alternative state where previously no suitable MPs could be found. To this end similarity weights were assigned to state pairings as shown in Table 5.4. The weightings range between 10 and 0, where 10 indicates two identical states and 0 two states that are as different as possible. So, with respect to Table 5.4, the “Closeby” state is more similar to the “Ignore” state than the “Meeting” and “Follow” states; and so on. According to observations made in the context of the research work considered in the thesis, in terms of the proposed state replacement mechanism, “Follow” and “Meeting” states are synonymous with each other, thus they were assigned equal weight equal to 4 against the “Ignore” state as shown in the Table 5.4. Like wise “Closeby” state has been assigned weight equal to 8 against of “Ignore” state because these states are relatively similar to each other. Thus, given a two rodent scenario where no MPs can be found, given a particular From description, the idea is to replace the state in the From description with the most similar state (as indicated by Table 5.4) and try again.

2. **Descriptor Rotation**. The Descriptor Rotation mechanism is applicable only in the case of relative MPs, but with respect to all categories of scenario considered in this thesis (Simple, Complex, and Two-Rodents). The basic idea behind the descriptor rotation mechanism is to rotate the From location through angles of 90°, 180° and 270°, and after each rotation attempt to find alternative MPs (recall that descriptors, as conceived of in this thesis, are rotation variant). In the current framework no consideration was given to rotating the descriptors associated with relative MPs. If “next move” decisions are made using only 3×3 ground type descriptors it makes sense that this should be applicable in all four cardinal directions. For example, given the 3×3 ground type configuration given in Figure 5.5(a) this should also apply in the situations given in Figures 5.5 (b), (c) and (d). How this should be achieved and used is a suggested matter for future research.

3. **Mirroring**. Using the mirroring mechanism the idea is to “flip” the descriptor in either the “x” or “y” direction. Thus when flipping in the x-direction the left and right columns will be switched, and in the y-direction the top and bottom rows will be switched. Note that this is only applicable with respect to relative MPs.

4. **Nearest Location**. Failing all else we can attempt to find MPs that are either physically near to the From location in the case of the absolute representation, or



-	-	-
-	n	n
-	n	n

(a) 0^0 rotation

-	-	-
n	n	-
n	n	-

(b) 90^0 rotation

n	n	-
n	n	-
-	-	-

(c) 180^0 rotation

-	-	-
-	n	n
-	n	n

(d) 270^0 rotation

FIGURE 5.5: Example of the rotation of a relative location descriptor: (a) 0° rotation, (b) 90° rotation, (c) 180° rotation and (d) 270° rotation.

similar in the case of relative representation. In the case of the absolute representation there will be eight next closest locations to be considered and so on. In the case of the relative representation we can “twist” the descriptor by replacing one ground type with another. This mechanism is thus applicable to all types of scenarios and can be used for both absolute and relative MPs.

5.3 Simulation Implementation

This chapter is concluded, for completeness, with a description of the implementation of the MP driven MABS. More specifically four aspects of the implementation are considered. Firstly, in Sub-sections 5.3.1 and 5.3.2, the attributes associated with environments and the agents representing simulation entities (rats and mice in the case of the work presented in this thesis) are considered. Next, in Sub-section 5.3.3, issues concerned with the concept of “smooth” simulations are considered in the context of sample time versus simulation time. The section is completed with a discussion of the visualisation module associated with the implementation of the MP driven MABS framework as depicted in Figure 5.1 (Sub-section 5.3.4).

5.3.1 Environment Definition

As noted in Chapter 3, the environments in which MABS agents were expected to operate were defined in term of tile worlds (2D grids). This section presents some further details concerning the nature of the environments used with respect to the realisation of the case studies considered for evaluation purposes later in this thesis. Using the implementation of the proposed MP driven MABS implementation, the grid environment used were defined in terms the following attributes:

1. **gridWidth**: The width of the environment, in terms of a number of grid squares, in the X direction (East-West).
2. **gridHeight**: The height of the environment, in terms of a number of grid squares, in the Y direction (North-South).

3. **gridComponentsList**: A list containing, zero, one or more items to be featured in the environment (nests, obstruction, blocked areas, tunnels and water points) and their location.

Given a $gridWidth$ and a $gridHeight$ expressed in terms of a number grid squares we can calculate the total number of grid squares in the environment, $totalCells = gridWidth \times gridHeight$.

5.3.2 Agent Definition

The agents (rodent agents with respect to this thesis) are the main players in a MP driven MABS. With respect to the proposed MP directed MABS implementation, agents featured the following attributes:

1. **fromLocation**: The agent's current location expressed in terms of a grid cell ID number.
2. **toLocation**: The cell ID where the agent intends to move to.
3. **previousLocation**: The cell location for the agent's previous location.
4. **state**: Only used in the context of two-rodent category scenarios, but a list of states describing the relationship between "this" agent and each other agent. (Empty if there are no other agents.)
5. **motionDirection**: The direction in which an agent is facing, one out of eight possible directions (North, North-East, East, South-East, South, South-West, West, North-West).
6. **route**: The path which an agent intends to follow.
7. **listOfSimulationPathPoints**: A list of those intermediate locations required for moving from: (i) a From location to a To location or (ii) a From location to a Way point location or (iii) from one Way point location to another Way point location or (iv) Way point to "To" location, defined in such a way that a "smooth" simulation results (the concept of smooth simulations is considered in further detail Sub-Section 5.3.3 below). When calculating these paths due account is taken of blocked areas or whether gate cells need to be used (in the case of the complex scenario category).

5.3.3 Simulation Time And Sample Time

The concept of smooth simulations was eluded to above. The overriding criteria for any form of simulation is that it is as realistic as possible. Thus simulations need to be "smooth" (in terms of their visualisation). To achieve this, there are two factors involved: (i) simulation time and (ii) simulation path points (or the intermediate points between

Table 5.5: Location relationship percentages with respect to current location for 1000 locations using a range of values for T_{samp} and a simple category scenario where the observed entities are rats

Scenario Category	Location recorded w.r.t current location	T_{samp}		
		20 frames	25 frames	30 frames
Simple	Immediate neighbour of current location	48%	86%	42%
	Same location as the current location	52%	12%	4%
	Not neighbours of current location	0%	2%	54%

every pair of way points associated with an MP). Using VDAS system, whereby location information is extracted from video data (from which MPs are derived), a sample interval is used expressed in terms of a number of frames. We refer to the rate at which MPs are extracted as the *sample time* (T_{samp}) and express this in terms of a number of frames.

Sample time is domain dependent, for the application domain considered with respect to the work presented in this thesis, a number of different T_{samp} values were used. In the case of the experiments conduced using simple category scenarios $T_{samp} = 25$ frames was used as this was found to produce good results. If a value of $T_{samp} > 25$ was used grid locations were frequently missed (jumped over), not necessarily a bad thing but it was felt that the value for T_{samp} should be set so that immediate neighbouring locations were recorded. If a value of $T_{samp} < 25$ was used the agent frequently did not have time to move. Of course if a video entity does not move this is also of interest, in this case same location is recorded after the specified time interval thus with a movement vector $v = 0$. In the case of the simple category scenario evaluations the video entities considered were rats; in the case of the complex and two-rodent category scenarios the video entities under consideration were mice. Mice move faster than rats, thus in the later case $T_{samp} = 5$ frames was used. Note that if the value of T_{samp} was reduced to below 5 the time required for processing became an issue.

In both of the above cases experiments were conducted using alternative values for T_{samp}. The results are presented in Tables 5.5 and 5.6 with respect to a range of T_{samp} values from 20 frames to 30 frames increasing in steps of 5 frames. For each experiment the sampling was conducted over 1000 iterations (at the given value for T_{samp}). For each iteration the location was recorded (thus over 1000 locations). The numbers in the tables show the percentage of locations, in each case, that were: (ii) the immediate neighbour of the current location, (ii) the same location as the current location and (iii) other locations (not neighbours of the current location). Inspection of the tables corroborates the observations made above.

However, for visualising simulations the sample time T_{samp} is too coarse, a more fine grained timing is required to ensure a smooth simulation visualisation, we refer to this

Table 5.6: Location relationship percentages with respect to current location for 1000 locations using a range of values for T_{samp} and complex and two-rodent category scenarios where the observed entities are mice

Categories of scenarios	Location recorded w.r.t current location	T_{samp}		
		5 frames	10 frames	20 frames
Complex	Immediate neighbour of current location	70%	32%	20%
	Same location as the current location	5%	5%	4%
	Not neighbours of current location	25%	63%	76%
Two Rodents	Immediate neighbour of current location	65%	28%	20%
	Same location as the current location	20%	15%	10%
	Not neighbours of current location	15%	57%	70%

as simulation time (T_{sim}) and it is expressed in terms of milliseconds. The relationship between T_{sim} and T_{samp} can be expressed as shown in Equation (5.2) where q is a constant

$$T_{sim} = \frac{T_{samp}}{q} \tag{5.2}$$

where T_{samp} is expressed in terms of milliseconds (not frames). From experimentation it was found that $q = 5$ produced best simulation visualisation results. If q was less than 5 the simulation became "juddery", while if q was greater the resource required to process the simulation became an issue.

As noted above, in the case of the experiments conducted with respect to this thesis (experiments discussed later in Chapter 6) the values used for T_{samp} were 25 and 5 (depending on whether the video entities considered were rats or mice). Thus the values for T_{sim} were $\frac{25}{5} = 5$ and $\frac{5}{5} = 1$ respectively. $T_{samp} = 25$ frames equates to 1000 milliseconds (1 second) while $T_{samp} = 5$ frames equates to 200 milliseconds; hence the values used for T_{samp} were 200 and 40 millisecond respectively. Consequently intermediate points (locations) had to be calculated to define the track which an agent will follow between two locations.

Given $q = 5$, this means that with respect to the *listOfSimulationPathPoints* agent attribute (see Sub-section 5.3.2 above)$T_{samp} \times q - 1$ intermediate points need to be calculated. In other words given a path associated with an MP ($MP.Path$) this means that $MP.Path \times (q - 1)$ intermediate locations (between From, Way point and To locations) need to be calculated.

The process for calculating intermediate locations is given in Algorithm 7. The algorithm takes as input a pair of locations, $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$, that might be a From location and a Way Point (WP) location, a WP location and another WP location, or a WP location and a To location depending on the nature of the MP under consideration. The output is a list of simulation location points (*Route*) connecting the two points. The number of such locations depends on the relationship between T_{samp} and T_{sim} calculated using Equation 5.3 (Equation (5.2) rearranged):

$$q = \frac{t_{samp}}{t_{sim}} \tag{5.3}$$

where q is the number of intermediate points we wish to find. The algorithm commences by calculating q (line 1). Then it calculates the x and y increments (δ_x and δ_y) which are applied to the start point so that the x and y coordinates for the intermediate points can be derived. If both Point 1 and Point 2 are in the same division, tested for in line 6, the list of intermediate points are calculated in the loop (from line 25 to 30). Where Point 1 and Point 2 are not in the same division, this is necessary to identify the appropriate gate cell(x_g, y_g) connecting the two points. This is achieved using the findGateCell function on line 7. Once gate cell has been identified, rodent agent will move first to gate cell and then proceeds towards "To" location. As the algorithm progresses the *Route* list is built up.

5.3.4 Simulation Interface and Visualisation

From Figure 5.1 the input/output to the proposed MABS framework comprises a "Simulator Interface" and a visualisation. Recall that the visualisation component of the proposed MABS is for observing the output of simulations. From an implementation perspective the simulator interface and the visualisation sit side by side in a single "user interface". This penultimate section of this chapter, for completeness, briefly describes this user interface.

The simulator interface is a menu driven mechanism for: (i) running simulations, (ii) setting up environments and (iii) specifying the number of agents to be included in a simulation and their default start locations. The interface has different views depending on the nature of the scenario (simple, complex and two-rodent). A number of "screen shots" illustrating the different views are given in Figure 5.6.

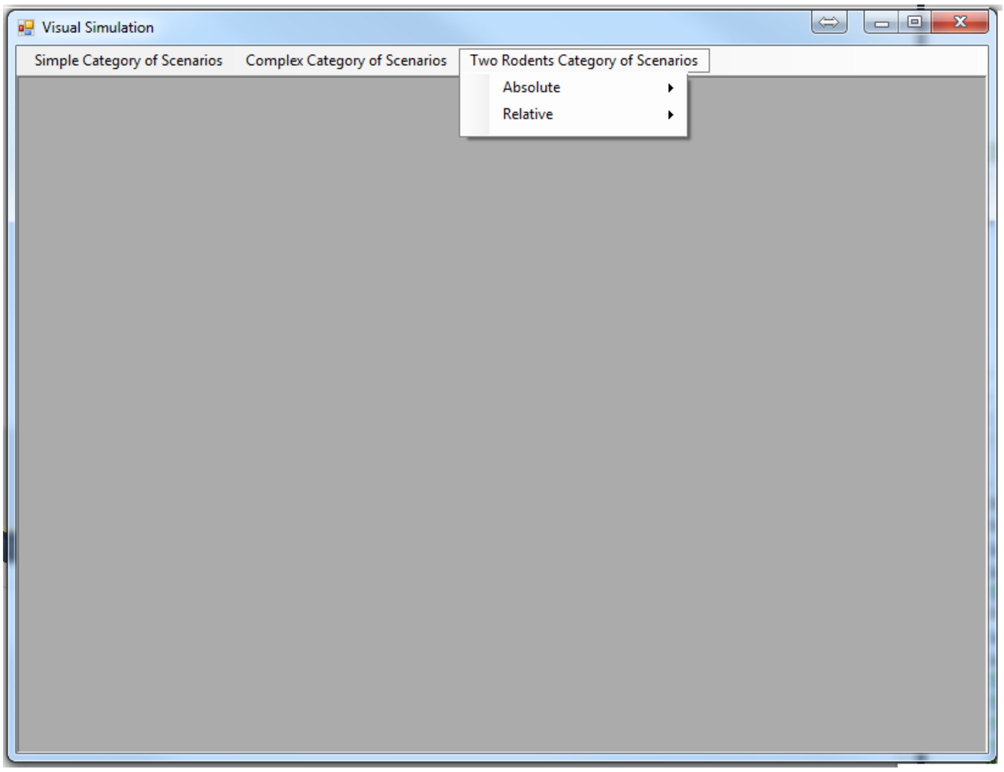
Figure 5.6(a) shows the start menu interface from which users can select a scenario category (Simple, Complex and Two-rodent). Each category of scenario has a dedicated sub menu. Facility is also provided whereby the user can select whether to use a relative or absolute representation, and to upload an appropriate MP database. Once the simulation starts the user can visualise the simulation as it progresses. Example visualisations are given in Figures 5.6(b), 5.6(c) and 5.6(d). Figure 5.6(b) shows an example of a simple category scenario, whilst Figure 5.6(c) shows a complex category scenario.

Algorithm 7 Algorithm for finding simulation path points between two locations

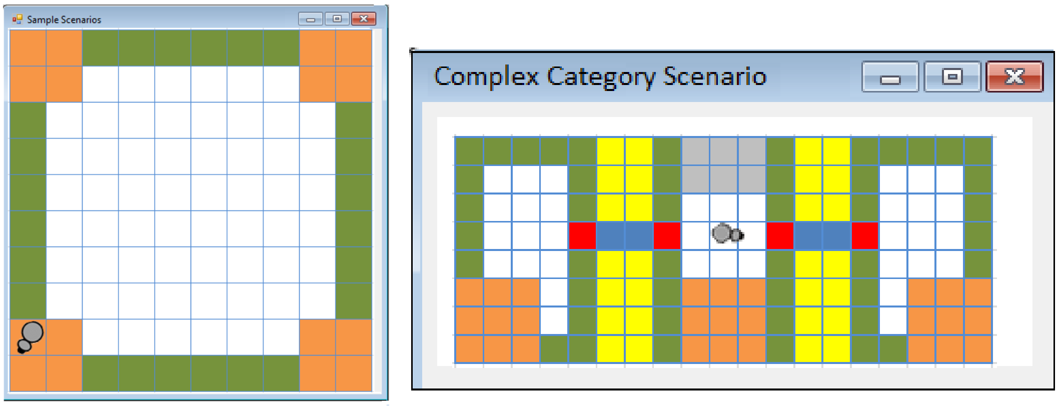
INPUT:
$\langle x_1, y_1 \rangle$ = Location for Point 1
$\langle x_2, y_2 \rangle$ = Location for Point 2
T_{samp} = sample time
T_{sim} = simulation time

OUTPUT: *Route* = A list on intermediate locations connecting points 1 and 2, $\{P_1, P_2, \dots\}$ where $P_i = \langle x_i, y_i \rangle$

```
q = $\frac{t_{sample}}{t_{simulation}}$
$Dist_x = \mid x_2 - x_1 \mid$
$Dist_y = \mid y_2 - y_1 \mid$
$\delta_x = \lceil \frac{Dist_x}{q} \rceil$
$\delta_y = \lceil \frac{Dist_y}{q} \rceil$
if ((Point 1 and Point 2) $\notin$ (Same Division)) then
    $x_g, y_g$ = findGateCell(Point1, Point 2)
    distSoFar = 0
    gDist = $\sqrt{(x_g - x_1)^2 + (y_g - y_1)^2}$
    for ($i = 0 \rightarrow q - 1$) do
        if (distSoFar < gDist) then
            $x = x_1$
            $y = y_1$
        else
            $x = x_g$
            $y = y_g$
        end if
        $x_i = x + (i \times \delta_x)$
        $y_i = y + (i \times \delta_y)$
        $P_i = \langle x_i, y_i \rangle$
        $Route = Route \cup P_i$
        distSoFar = $distSoFar + \sqrt{(x_i - x)^2 + (y_i - y)^2}$
    end for
else
    for ($i = 0 \rightarrow q - 1$)) do
        $x_i = x_1 + (i \times \delta_x)$
        $y_i = y_1 + (i \times \delta_y)$
        $P_i = \langle x_i, y_i \rangle$
        $Route = Route \cup P_i$
    end for
end if
```

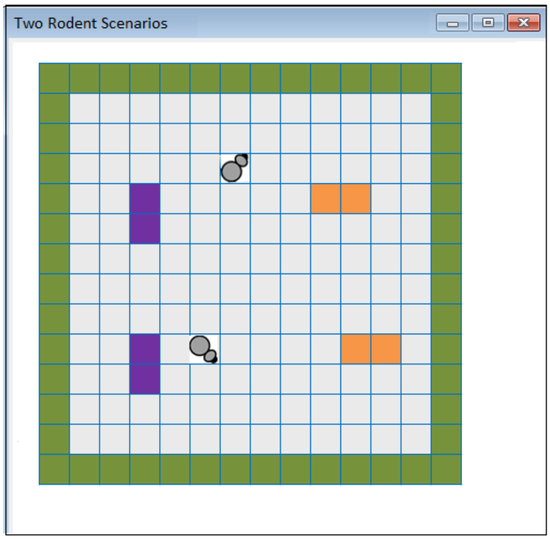


(a) Simulator interface



(b) Visualisation for the Simple scenarios

(c) Visualisation for the Complex scenarios



(d) Visualisation for Two rodents scenarios

FIGURE 5.6: User Interface for Different Categories of Scenarios

More specifically the later shows a complex scenario that includes three divisions. Figure 5.6(d) shows an example of a two-rodent scenario. The colour coding used in the figures is as follows: (i) wall proximity (green), (ii) nests (orange), (iii) water pot (grey), (iv) tunnels (blue) and (v) gate cells (red), (vi) nests (orange) and (vii) obstructions (purple).

5.4 Summary

This chapter has presented the proposed MP driven MABS framework. The chapter commenced by describing how MPs can be utilize within a MABS framework. The chapter then continued to describe the simulation process. Detail was presented concerning the selection of *legal* MPs using a weighted random probability mechanism. The situation where there may not be any legal MPs was also considered. In the last section of the chapter the proposed user interface to the MP driven MABS was discussed together with the nature of the attributes associated with environments and the individual agents, and the issues concerning the desire for smooth simulations. In the following chapter the evaluation of the proposed MABS driven by MPs idea is reported on.

Chapter 6

Evaluation of MABS using MPs

6.1 Introduction

The process for tracking objects and extracting location information from video data was presented in Chapter 3. The proposed mechanism for extracting (mining) movement patterns from location information to support Multiagent Based Simulation (MABS), was presented in Chapter 4. In Chapter 5 the mechanism whereby such movement patterns could be incorporated and utilised to “drive” a MABS was presented. In this chapter the movement pattern concept, as conceived of in this thesis, is evaluated and analysed.

It is difficult to evaluate the operation of simulations (MABS or otherwise) with respect to any “Gold standard”. However, the overriding criteria for the effectiveness of a simulation environment are that the simulations supported must be as realistic as possible. In [126] this is defined in terms of two requirements for simulation: (i) corroboration and (ii) internal consistency. The first is concerned with the requirement that a simulation system model operates in the same manner as the system being simulated. In other words that the simulation results are similar to real world results. The second is concerned with the requirement that the constituent parts of a simulation system function in line with underlying concepts and theories. In other words that the philosophical foundations on which a simulation is based conform to recognised theories and observations concerning the simulation domain. To address the first, the operation of the simulated system needs to be compared against its real world representation [21]. To address the second, the operation of the proposed simulation framework needs to be analysed in terms of the mechanisms used to realise the simulation.

For the evaluation presented in this chapter, corroboration checking was conducted by recording simulated experiments and conducting comparisons with video of the real life experiment. This is a process referred to as “closing the loop” for reasons that will become clear later in this chapter. The collection of real life video data, as noted earlier, was a resource intensive process, thus the same video data as used for training purposes was used. The adopted corroborative evaluation strategy is shown in Figure 6.1. In the figure the ovals indicate processes and the boxes data or information, the directed arcs indicate the flow of information. The top part of the figure illustrates the movement pattern mining process described earlier in this thesis where first a set of locations are extracted from the video data. These locations are then mined to generate movement patterns, which can then used to drive a MABS as described in the foregoing chapter. From this simulation we can extract simulated video data, from which locations can be extracted in the same way as before. The occurrence counts associated with these, "simulated" locations can then be compared with the occurrence counts associated with the real locations (hence "closing the loop"). If the occurrence counts associated with the video data were similar to those associated with the simulation video data it could be argued that the simulation was realistic (corroborated). Of course the simulation and video run times have to be the same for the comparison to be meaningful.

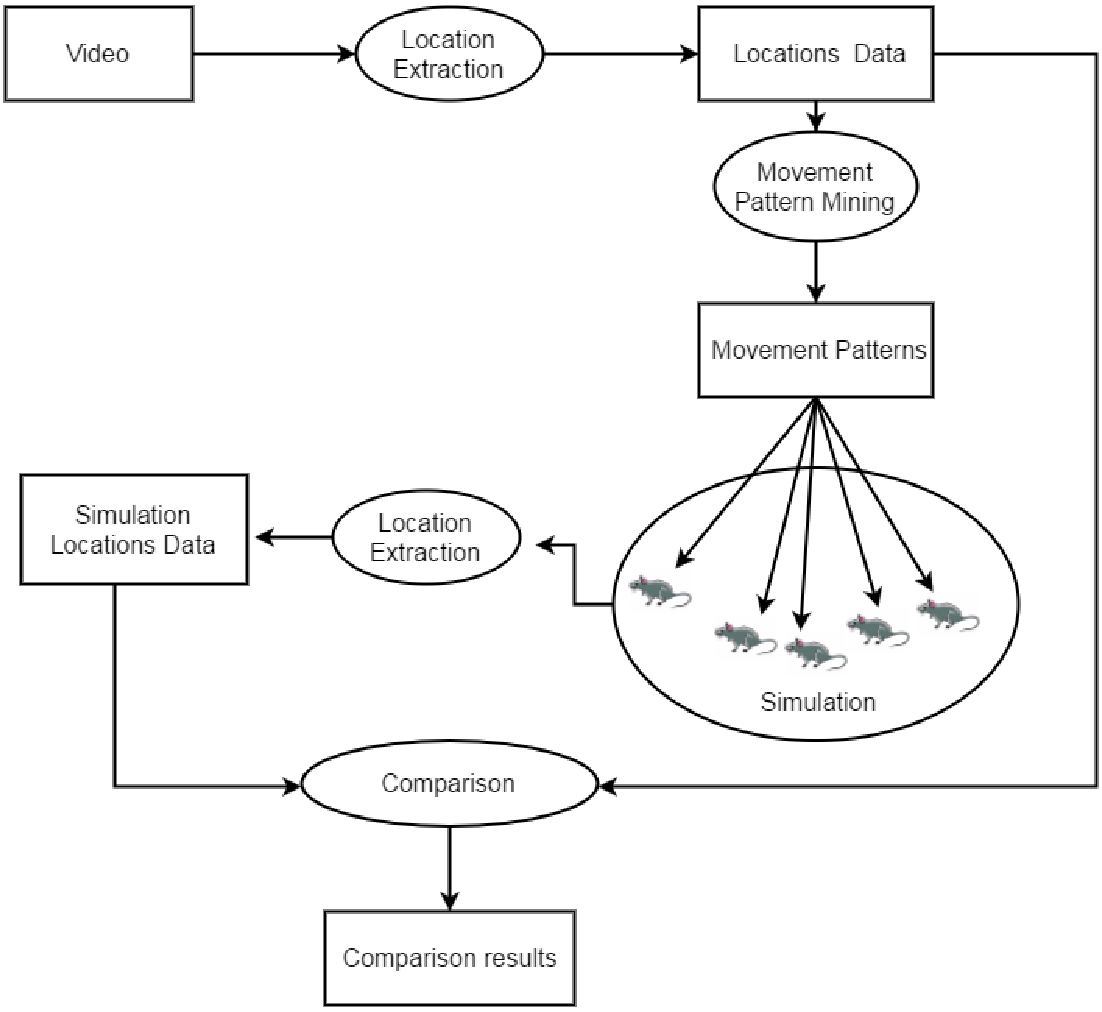


FIGURE 6.1: Adopted Corroborative Evaluation Strategy ("Closing the loop")

For comparison purpose the grid environment was divided into groups of grid cells referred to as blocks. Each block defined a location with respect to the corroborative evaluation considered in this chapter. The total number of visits for each block was then recorded instead of for each grid cell. For scenarios within the complex scenario category each division was, given particular evaluation requirements, also sometimes considered to be a “location”.

In the context of the evaluation presented in this chapter, the operation of the proposed movement pattern based MABS framework was evaluated in terms of the three categories of scenario: (i) simple scenarios featuring a single mouse in a box without obstructions, (ii) complex scenarios featuring a single mouse in a box with divisions and (iii) two mice scenarios that included obstructions (and more than one mouse). Recall also that two mechanisms for defining movement patterns have been proposed: (i) absolute and (ii) relative. Consequently, with respect to each scenario category, the evaluation was conducted by considering both absolute and relative movement patterns. The experimentation conducted with respect to each of the scenario categories is reported on in Sections 6.2, 6.3 and 6.4 respectively. The chapter is concluded, in Section 6.5, with a summary of the main outcomes of the presented evaluations.

6.2 Category 1: Simple Scenarios

The section reports on the evaluation results obtained with respect to the first category of scenario, Simple scenarios, involving a single mouse in a box without obstructions. The idea is to simulate the kind of laboratory experiments frequently conducted by rodent behaviourologists in laboratory conditions. The advantages of such simulations are of course that they entail less resource and do not require the usage of laboratory mice or rats (the usage of laboratory mice/rats is politically contentious in some areas).

The objectives of the evaluation were:

1. To compare the operation of Category 1 simulations founded on absolute patterns with the real life equivalent operation.
2. To compare the operation of Category 1 simulations founded on relative patterns with the real life equivalent operation.
3. To compare the frequency with which nest sites, wall locations and open space locations were visited in the context of both relative and absolute movement patterns and Category 1 simulations.
4. To compare the operation of the absolute and relative mechanisms in the context of Category 1 simulations.
5. To compare the effect of different path lengths as incorporated in the MP concept.

Each of the above objectives is considered, with respect to the experimental results obtained, in the following four sub-sections.

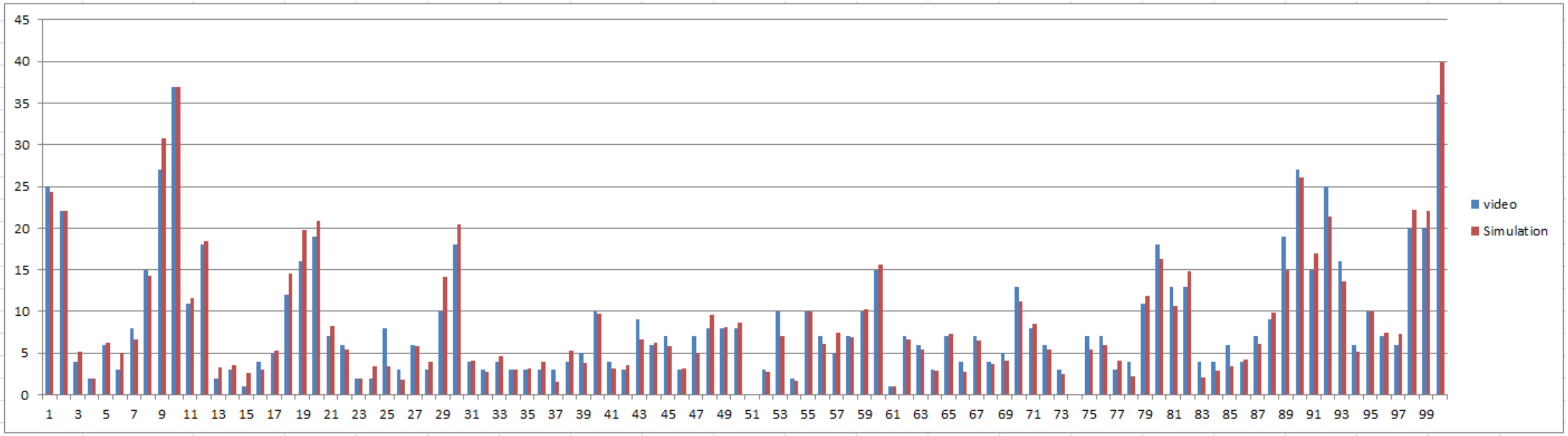


FIGURE 6.2: Comparison of Absolute Movement Pattern Simulation data with Real Life data in the context of the Simple scenario category and in terms of the number of standard grid cells visited

6.2.1 Comparison of the Usage of Absolute Movement Patterns with Real Life Experimentation (Simple scenarios)

The analysis of the operation of absolute movement patterns was conducted using the video data used for training purposes as described above. Recall that in the case of the simple scenario category the video featured rats. In all, 2 hours of video data was used. The simulation was set up so that the simulated experiment was identical to that presented in the video. Thus the simulation featured the same environment and the same duration as the two hour real life equivalent experiment. The simulation was run 10 times and the occurrence counts for each location recorded. A still from the video used was given in Chapter 3, Figure 3.5(a).

Occurrence count data was collected using two grids: (i) the standard grid used to define environments (as described in section 3.7.1 of chapter 3); and (ii) a coarse grid, comprised of blocks, designed to capture more general information. Recall that the standard grid size for the simple scenarios was 10, giving $10 \times 10 = 100$ grid cells. In terms of the video data used this equated to a 500×500 pixel area. The grid size used for the coarse block grid was designed so that a nest site would be encompassed by a single block. As it happened this meant that the size of the coarse grid cells, the blocks, were twice that of the "standard" grid cells. Thus a single coarse grid cell (block) measured $2 \times 2 = 4$ grid cells of the standard grid. The coarse grid thus measured $5 \times 5 = 25$ blocks.

Results using the absolute representation are presented in Figures 6.2 and 6.3. Figure 6.2 gives the results obtained using the standard environment grid while Figure 6.3 gives the results using the coarse block grid. In each case the blue bars indicate the frequency that each location was visited in the video data and the red bars the frequency abstracted from the simulation. From the figures it can be observed that the behaviour of the rat agent in the simulation is similar to that captured within the video data. In other words we can conclude that the operation of the simulation using absolute movement patterns is appropriate.

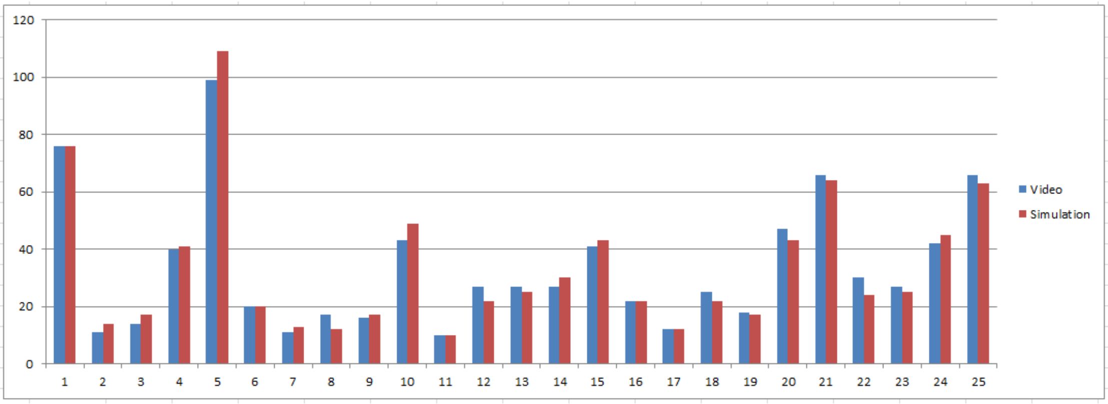


FIGURE 6.3: Comparison of Absolute Movement Pattern Simulation data with Real Life data in the context of the Simple scenario category and in terms of the number of coarse block grid cells visited

It should be noted that, because the simulation was run 10 times, average values are shown in the figures. Thus given further simulation runs the recorded frequency counts might not be the exactly same, but our experiments have indicated that consistent results are obtained.

Further evaluation was conducted by dividing the data set in two and using the first half (the training set) for generating MPs and the second half (the test set) for testing the result. Experiments were conducted using only the coarse grid (measuring $5 \times 5 = 25$ blocks). The results are given in Figure 6.4 where the x-axis features the block numbers ordered sequentially and the y-axis the occurrence counts. The blue bars indicate the frequency that each block was visited in the simulation and the red bars the frequency from the video test data. From the figure it can be seen that the operation of the MP driven simulation is similar to the video data; grid cells that are frequently visited in the simulation are also frequently visited in the video. Thus providing further evidence that the Movement Pattern concept can be successfully employed for simulation purposes.

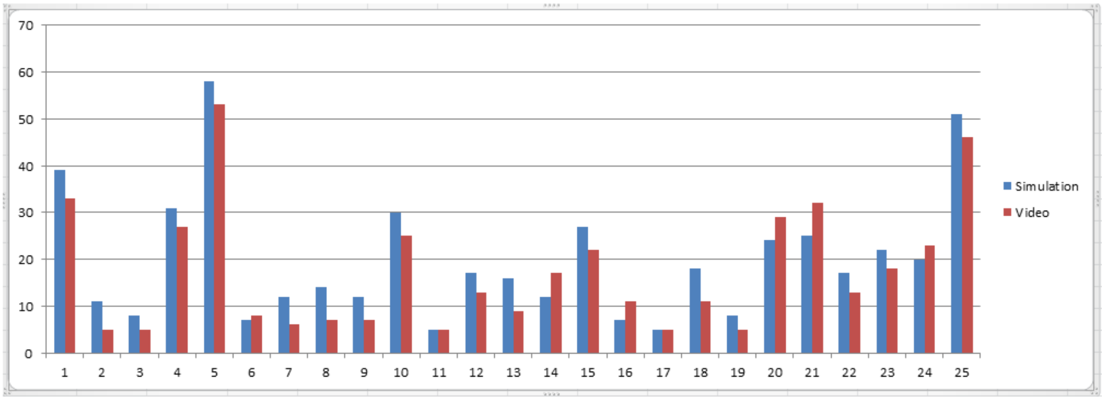


Figure 6.4: Comparison of video and simulation block counts where data was split 50:50

6.2.2 Comparison of the Usage of Relative Movement Patterns with Real Life Experimentation (Simple scenarios)

A similar set of experiments were conducted, to those described above, to analyse the relative movement pattern representation. However, in this case occurrence counts for visits to cells were not used, instead occurrence counts for individual location descriptors were used. Recall that, when using relative movement patterns, locations are described in terms of location descriptors (see Section 3.6 of Chapter 3). For the experiment the same video data as used with respect to the analysis of the absolute approach was used. For the environment featured in the video data a total of 45 location descriptors can be identified. For completeness these are listed in Appendix A, A.1. The results obtained with respect to the evaluation conducted using relative movement patterns in comparison with the real video data are presented in Figure 6.5 where the horizontal axis lists the different possible cell descriptors and the vertical axis the occurrence count associated with each descriptor. From the figure it can again be seen that the simulated location occurrence counts were similar to the real life occurrence counts, confirming that the relative movement pattern mechanism can be successfully used in the context of a MABS.

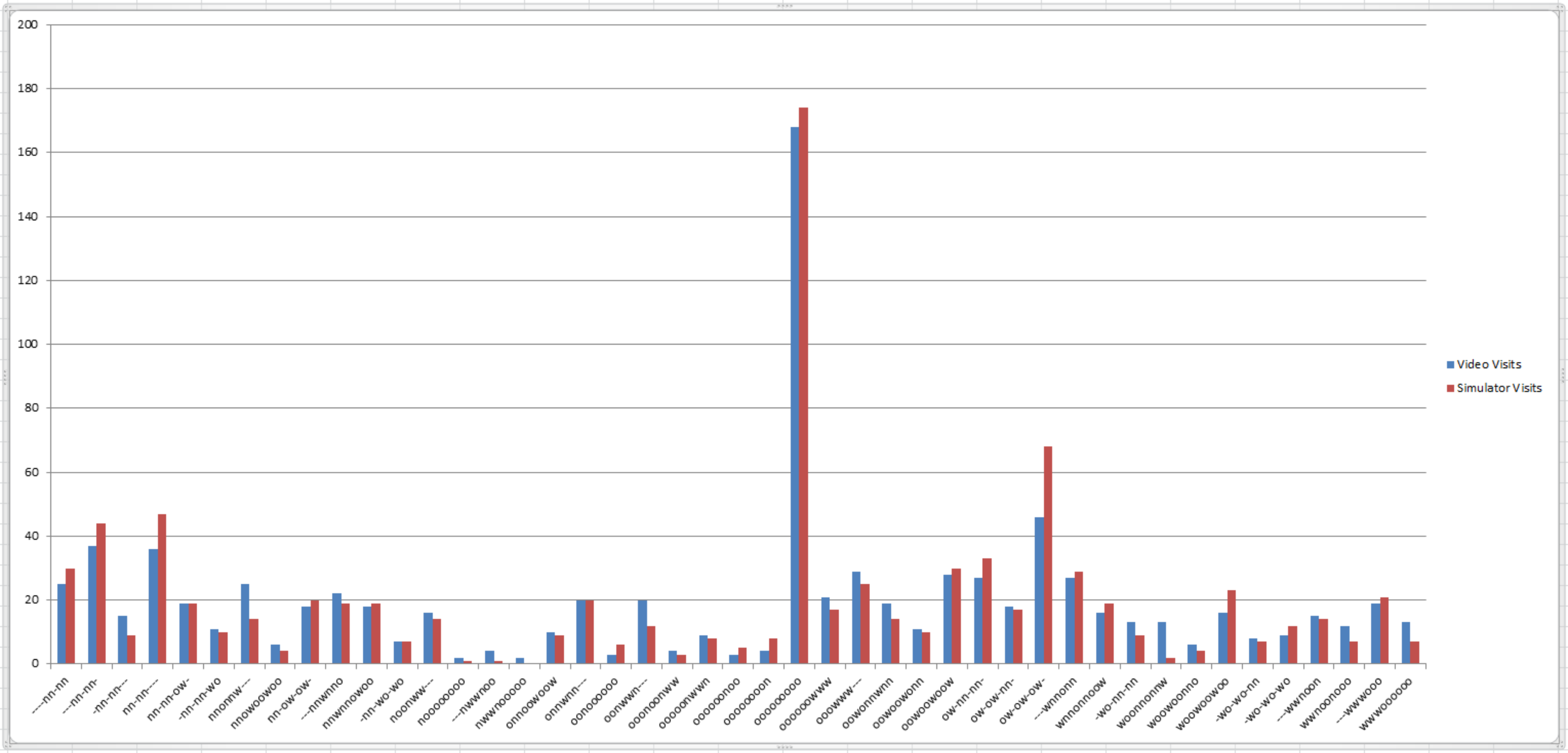


FIGURE 6.5: Comparison of Simulation Movement with Real Movement using the relative representation for Simple scenarios

6.2.3 Comparison of Frequency whereby nest sites, wall and open space locations are visited (Simple scenarios).

The evaluation directed at the frequency that nest sites, walls and open space locations were visited is presented in this sub-section. The aim here was to confirm two behaviours: (i) a behaviour known as *thigmotaxis* [150] and (ii) a tendency to spend significant time at nest sites. Thigmotaxis is an affinity for walls, a behaviour displayed by some animals including rodents (it is related to a desire for safety). For this purpose the standard 10×10 grid was used. With respect to this grid used to model environments associated with simple scenarios, it should be recalled from the figures given in Chapter 3 that: (i) cell numbers 1, 2, 11 and 12 represent the north-west nest box, (ii) cells 9, 10, 19 and 20 the north-east nest box, (iii) cell numbers 81, 82, 91 and 92 represents south-west nest box and (iv) cells 89, 90, 99 and 100 the south-east box. Similarly: (i) cell numbers from 3 to 8 represent the upper wall area locations, (ii) cells 30, 40, 50, 60, 70 and 80 the right side wall area locations, (iii) cell numbers 21, 31, 41, 51, 61 and 71 the left side wall area locations and (iv) cell numbers 93 to 98 the lower wall area locations. For the purpose of the experiments reported on in this sub-section an open space location was considered to be any location that was not a wall or nest location.

The frequency count results are presented in Figures 6.6, 6.7 and 6.8. Figure 6.6 shows the number of times each nest site was visited using absolute and relative movement pattern simulations, and in real life (the nest site identifiers used in the figure have the obvious interpretation). From the figure it can be noted that cells representing nest boxes, as might be expected, are frequently visited locations. The results also show that the simulation frequency counts are similar to the real life recorded counts, thus confirming that the simulation operates in an appropriate manner.

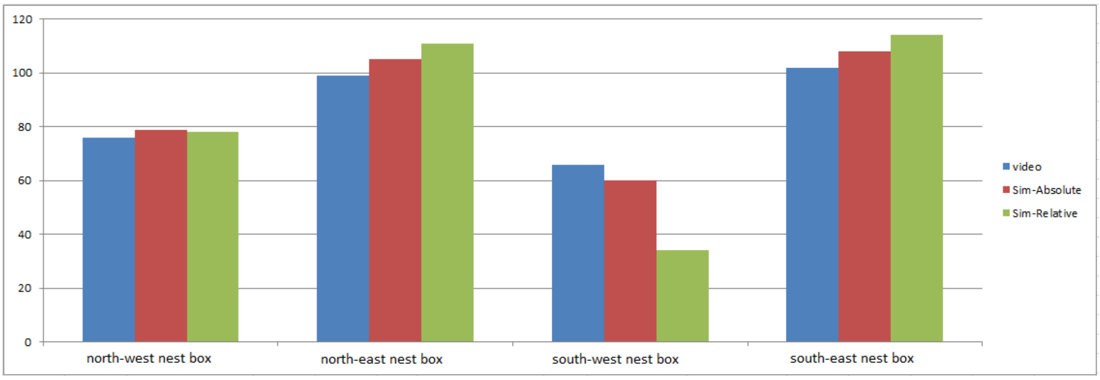


FIGURE 6.6: Comparison of nest box location counts obtained using video data versus those obtained using absolute and relative movement pattern simulations

Figure 6.7 shows the number of times each wall was visited using absolute and relative movement pattern simulation and in real life (the wall identifiers used in the figure again have the obvious interpretation). From the figure it can be observed that the frequency counts in all three cases are similar; there is a good correspondence between the simulation data and the video data, again indicating that the simulations are realistic.

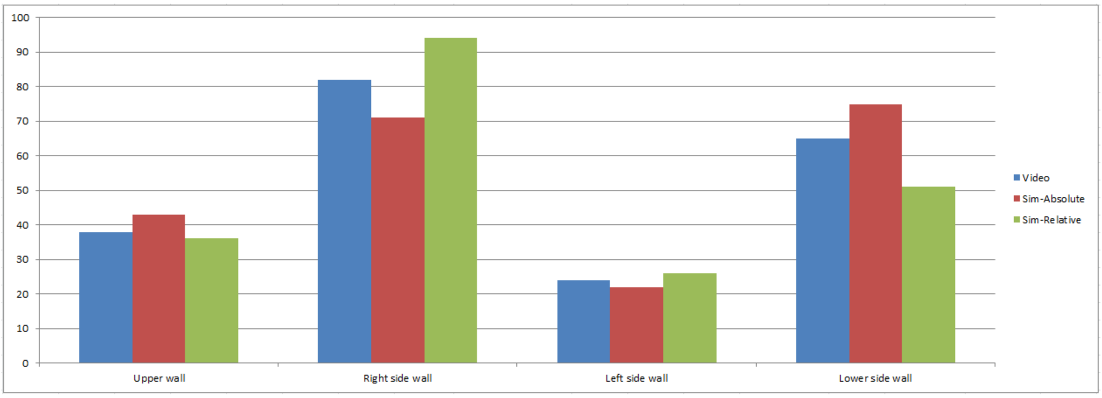


FIGURE 6.7: Comparison of wall location counts obtained using video data versus those obtained using absolute and relative movement pattern simulations

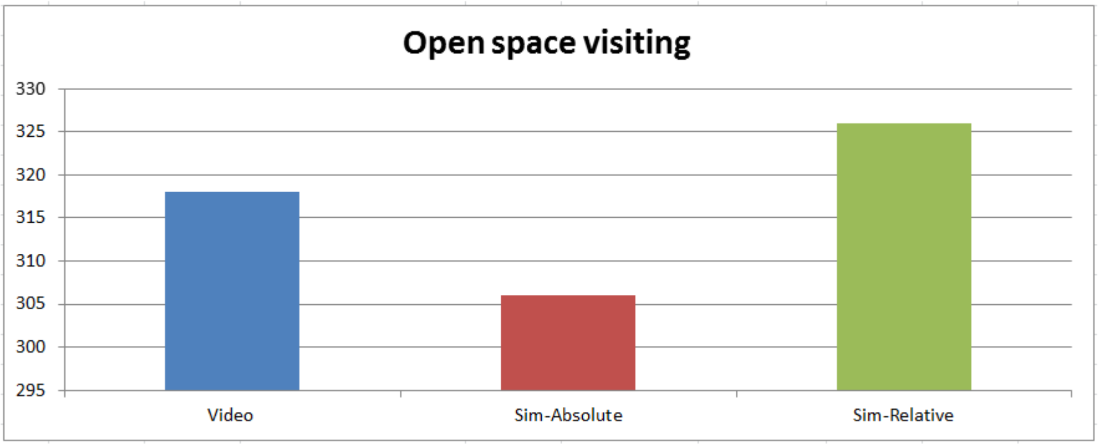


FIGURE 6.8: Comparison of open space locations counts obtained using video data versus those obtained using absolute and relative movement pattern simulations

For completeness Figure 6.8 shows the number of times open space locations were visited. Again a good correspondence between the simulated counts and the real counts can be observed.

6.2.4 Comparison of Absolute versus Relative Movement Patterns (Simple scenarios)

This section considers whether absolute or relative movement patterns produced the “best” simulations. The evaluation was conducted using results from the experiments described in Sub-section 6.2.1. Recall that this set of experiments considered the closeness of the simulation cell visit occurrence counts to the actual occurrence counts. Given a set of simulated occurrence counts $F_s = \{s_1, s_2, \dots\}$ and a set of corresponding real occurrence counts $F_r = \{r_1, r_2, \dots\}$ (such that $|F_s| = |F_r|$), we can use the average error (Avg_Error) of the differences, as a measure of “closeness” of the simulated data with the real data, calculated as follows:

$$Avg\_Error = \frac{\sum_{i=1}^{|F_s|} \mid s_i - r_i \mid}{\mid F_s \mid} \tag{6.1}$$

Using the results from Sub-sections 6.2.1 and 6.2.2, the average error values, using the standard grid and Equation (6.1), are: (i) 0.13 using absolute patterns and (ii) 0.20 using relative patterns. From these results it can be seen that the absolute patterns, using the coarse grid, produced the most accurate result; the results "closest to real life". Although, as noted earlier, the advantage offered by the use of relative movement patterns was that they were not tied to a particular environment, while absolute movement patterns were. This is explored further in Sub-section 6.3.6 below.

From the foregoing the argument can be made that the absolute representation was the better than the relative representation in the context of simulation realism; although it was also noted that the relative representation was more versatile. The question that remains is whether this result is statistically significant or not. The remainder of this Sub-section considers this statistical significance in further detail. The idea is that if the average error of the differences, obtained in both cases, is better than the average error of the differences obtained randomly, the results obtained using the movement pattern concept will be statistically significant.

TABLE 6.1: Statistical significance calculation

	A	B		C		D	
To Loc.	Video Counts	Sim. Counts	Diff. A and B	Random	Diff. A and C	Max	Diff. A and D
1	1	1	0	2.5	1.5	0	1
2	1	1	0	2.5	1.5	0	1
3	1	1	0	2.5	1.5	0	1
4	1	2	1	2.5	1.5	0	1
5	1	2	1	2.5	1.5	0	1
6	2	2	0	2.5	0.5	0	2
7	3	3	0	2.5	0.5	0	3
8	3	2	1	2.5	0.5	0	3
9	5	5	0	2.5	2.5	0	5
10	7	6	1	2.5	4.5	25	18
Total	25	25	4	25	16.0	25	36
Avg_Error			0.16		0.64		1.44

The adopted process is illustrated in Table 6.1. The table assumes, for convenience, a 10 cell scenario with 25 simulation iterations. In the table the first column lists the 10 cell IDs, the second column (Column A) the number of times each cell was visited in the "video" and in the third column (Column B) the number of times each cell was visited in the "simulation". The fourth column then lists the absolute differences between column A and B from which the average error of the differences, 0.16, is derived using Equation 6.1. The fifth column (Column C) then gives the number of times each cell would be visited assuming a Gaussian distribution, in other words the probability of a cell being visited is the same as the probability of any other cell being visited. Thus, in the example we have 2.5 visits per cell ($totalnumberofvisits/numberofcells = 25/10 = 2.5$). Column six gives the difference between columns A and C from which the average error of the differences can be derived, in this case 0.64. In the example the "simulated" average error of 0.16 lies between 0.64 and 1.44, hence we can conclude that the result is not because of some random allocation. Hence we can reject the null hypothesis that the application of MPs, within the context of the proposed simulation framework, has no effect.

However, we would also like to know how significant the outcome is. The idea is to calculate this using the maximum possible average error of the differences. This will occur when an agent remains in the same cell for the entire duration of a simulation. This is illustrated in column seven (Column D) of Table 6.1. The resulting average error of the differences is then 1.44 (column 8 of Table 6.1). Thus, in the example, the average error when cells are chosen at random, as a percentage of the maximum average error, is given by:

$$Avg\_Error\% = \frac{0.64}{1.44} \times 100 = 44\% \tag{6.2}$$

and the average error when cells are chosen using the proposed mechanism, as a percentage of the maximum average error, is given by:

$$Avg\_Error\% = \frac{0.16}{1.44} \times 100 = 11\% \tag{6.3}$$

Consequently we can conclude, in the example, that the result is "significantly statistically significant". Applying this analysis to the actual results obtained earlier and reported on, in Sub-Sections 6.2.1 and 6.2.2, we get the results given in Table 6.2. In Table 6.2, first column lists two mechanism used for the representation of movement patterns (absolute and relative), the second column denotes the actual recorded average error values obtained in Sub-sections 6.2.1 and 6.2.2. The third column then indicates the percentage of the recorded average error value calculated by Equation (6.3). The fourth column then represents the recorded average error values obtained by random allocation, while the fifth column contains the percentage of random average value calculated by Equation (6.2). The maximum possible average error values are listed in the last column of the table. Inspection of the table indicates that the results obtained are indeed statistically significant, better than a random allocation. Further, inspection of the percentage values indicates that the results are "significantly statistically significant".

Table 6.2: Statistical Significance testing results.

Rep.	Rec. Avg. Error	Rec. Avg. Error %	Ran. Avg. Error	Ran. Avg. Error %	Max. Avg. Error
Absolute	0.13	6.80%	0.66	34%	1.91
Relative	0.20	12.42%	0.59	36%	1.61

6.2.5 Comparison of Different Path Lengths (Simple scenarios)

The path length parameter ($|Path|$) used through out the thesis was set to 5. The reason for this was, that it was found to provide good simulations; especially in the context of multi-rodent scenarios where the state is only updated once the execution of an MP has been completed. Experiments were conducted in the same manner as discussed in Sub-Section 6.2.1, using a range of values for $|Path|$ from 4 to 6 increasing in steps of 1. The results are presented in Table 6.3. From the table it can be seen that the average error (Avg_Error) value increases when $|Path| > 5$. Recall that the path concept was used for two purposes: (i) to avoid "ping pong" situations in the simulation and (ii) to provide agents with "memory". Therefore, values below 4 were not considered because this would result in the rodent agents having very little or no "memory" (the later was the case when $|Path| = 1$). Paths of length 4 and 5 ($|Path| = 4$ and $|Path| = 5$), result in similar errors, therefore the longer of these two paths length, which is $|Path| = 5$ provides better "memory" feature to the agent was selected for this research work.

Table 6.3: Average Error results using different path lengths

Path length	Avg_Error		
	Absolute		Relative
	Standard Grid Size	Coarse Grid	Standard Grid Size
4	0.13	0.08	0.20
5	0.13	0.08	0.20
6	0.24	0.15	0.27

6.3 Category 2: Complex Scenarios

Recall that with respect to the complex scenario category the scenarios of interest comprised a number of areas (divisions) connected by tunnels (refer back to Chapter 3 for details). We use the term "division" to describe such areas. With respect to the video data obtained each scenario had three divisions: Left (L), Middle (M) and Right (R); each tunnel having gate cells at both ends. This section reports on the evaluation conducted with respect to the proposed movement pattern based MABS framework with respect to Category 2 scenarios. The objectives of the evaluation were as follows:

1. To compare the operation of Category 2 simulations founded on absolute patterns with the real life equivalent operation.
2. To compare the operation of Category 2 simulations founded on relative patterns with the real life equivalent operation.
3. To compare the frequency with which divisions were visited with respect to both absolute and relative movement patterns.
4. To compare the frequency with which mice agents moved from one divisions to another (including the same division) with respect to both absolute and relative movement patterns.
5. To compare the operation of the absolute and relative mechanisms in the context of Category 2 simulations.

6. To investigate usage of relative movement patterns with respect to Category 2 scenarios other than the scenarios used for training purposes.

The evaluation was undertaken in a similar manner to that described with respect to the simple scenarios, namely comparison with video data. In this case the video data featured mice (rats were used with respect to the evaluation presented in the previous section). About 2 hours of video data was used and the simulation was setup in such a way that the simulated experiment was identical to that presented in the video. A still from the video data was given in Chapter 3, Figure 3.6. As noted above the scenario comprised three divisions connected by two tunnels, and that much of the "playing area" is consequently "out of bounds". Each of the above objectives is discussed in terms of the evaluation results obtained in the following sub-sections.

6.3.1 Comparison of the Usage of Absolute Movement Patterns with Real Life Experimentation (Complex scenarios)

Data was collected using a 19 × 8 grid with grid cell sizes equivalent to the grid size used for the movement pattern mining as discussed in Chapter 3 (see Figure 3.4). Thus 152 grid cells in total, 40 grid cells associated with each of the three divisions, 4 with the tunnels and 28 with blocked areas. A coarse block grid was not used in this case because it was deemed unnecessary.

With respect to the comparison of the operation of the MABS using absolute movement patterns and real life data, because of the large number of occurrence counts obtained, for analysis purposes, these were grouped according to their relevant division (Left, Middle or Right) or tunnel (Tunnel 1 or Tunnel 2). The results are presented in Figures 6.9 to 6.13. Figures 6.9 to 6.11 show the results obtained with respect to the individual divisions, while Figures 6.12 and 6.13 show the results obtained with respect to the tunnels.

Figure 6.9 shows the results obtained with respect to the left division. From the figure it can be seen that grid cells close to the gate cells and close to nest box are frequently visited in the video as well as in the simulation. The gate cell in this case is number 65, cells 80, 81 and 62 are all close to the gate cell; cells 98 and 99 cover a nest box and cells 100 and 118 are close to this nest box. Figure 6.10 shows the results obtained for the middle division from which it can be seen that cells 65 to 69 are highly visited. These cells mark the direct route between the left division and the right division. The middle division contains a nest box (at cell numbers $104, 105, 106$, $123, 124, 125$ and $142, 143, 144$); the nest box and cells close to it are comparatively frequently visited, for example cells 86 and 87 as shown in the figure. The occurrence count data for the right division is shown in Figure 6.11. In this case cell 72 is a gate cell and, as before, is a highly visited cell, in both the video and simulation, compared to the other cells associated with the right division. Both Tunnels comprise two cells, 63 and 64 in the case of Tunnel 1, 70 and 71 in the case of Tunnel 2, as shown in the Figures 6.12 and 6.13.

Overall, from the figures, it can be seen the there is a close corroboration between the video and simulation data when using absolute movement patterns. Thus it can be concluded that simulation based on the proposed absolute movement patterns, in the context of complex scenarios, run in a realistic manner.

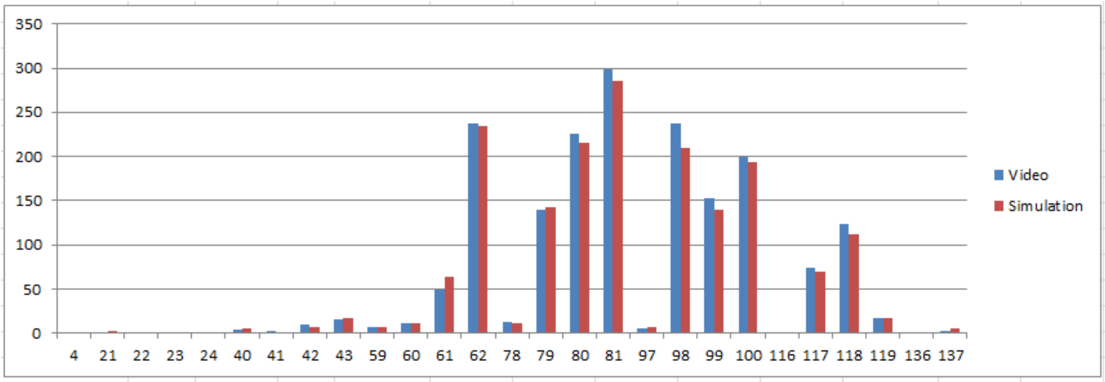


FIGURE 6.9: Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for left division of the example Complex scenario

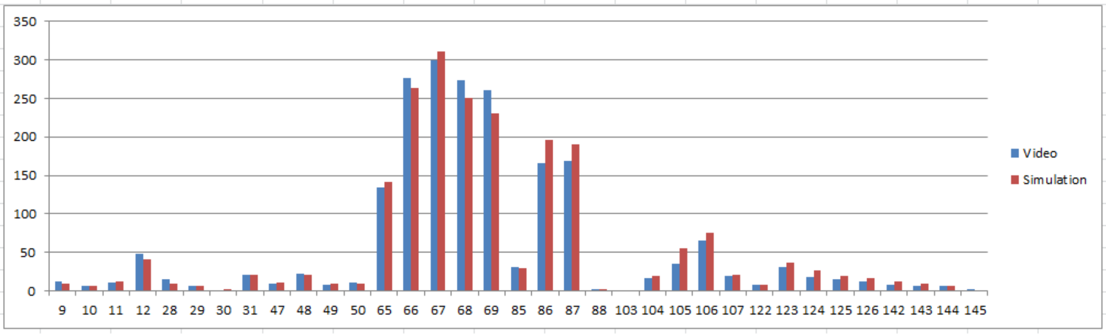


FIGURE 6.10: Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for middle division of the example Complex scenario

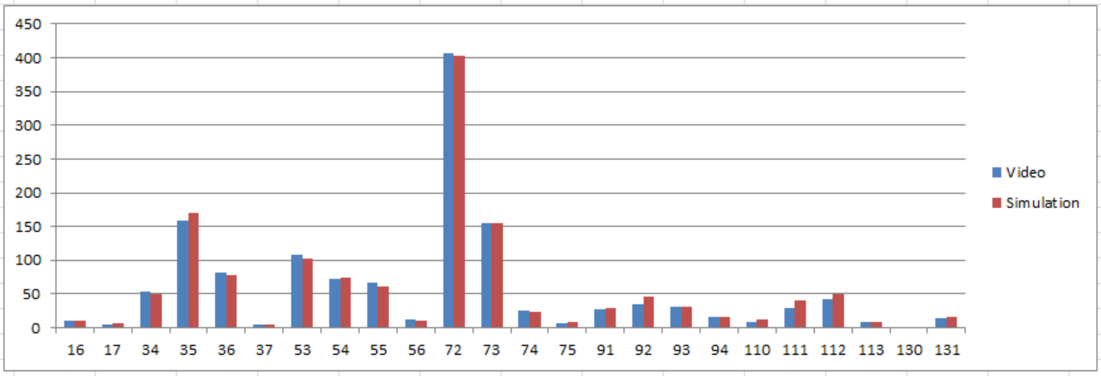


FIGURE 6.11: Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for right division of the example Complex scenario

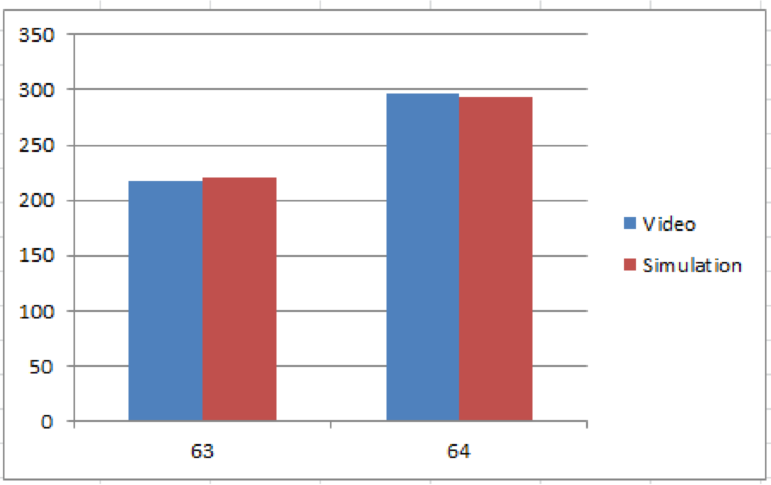


FIGURE 6.12: Comparison of cell counts in simulation data with real life data, using the absolute mechanism, for tunnel 1 of the example Complex scenario

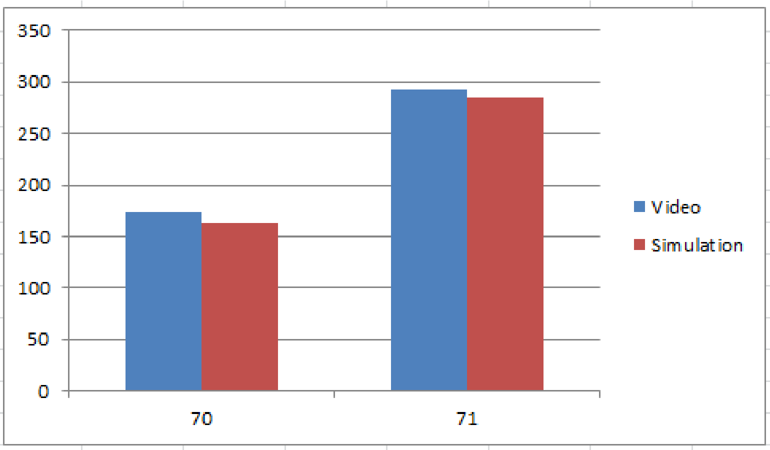


FIGURE 6.13: Comparison of cells counts in simulation data with real life data, using the absolute mechanism, for tunnel 2 of the example Complex scenario

6.3.2 Comparison of the Usage of Relative Movement Patterns with Real Life Experimentation (complex scenarios)

With respect to the comparison of the operation of the MABS using relative movement patterns and real life data the frequency counts for each potential location descriptor are given in Figures 6.14 to 6.18. Again, as in the case of the results presented with respect to absolute movement patterns, the results are presented with respect to the relevant division or tunnel (Left, Middle, Right, Tunnel 1 or Tunnel 2). Figure 6.14 shows the cell descriptors associated with the left division, Figure 6.15 for the middle division and Figure 6.16 for the right division. For reference, the complete set of descriptors are given in Appendix B. Figures 6.17 and 6.18 give the results for the two tunnels. Note that in the case of the tunnels there are only two location descriptors: $\langle bbwttgbbw \rangle$ and $\langle wbbgttwbb \rangle$ associated with each tunnel.

As discussed in Subsection 6.3.1 it is clear from the figures that those cells descriptors associated with gate cells, nest boxes and those nearby to gate cells and nest boxes, are highly visited with respect to each division. For example, with respect to the left division (Figure 6.14), the descriptors $\langle ogtowbowbL \rangle$ and $\langle oogoownowL \rangle$, representing locations close to gate cells, and the descriptors $\langle owbogtowbL \rangle$ representing the gate cell itself, are frequently visited. Similar observations can be made with respect to the Middle and Right division as shown in Figures 6.15 and 6.16 where those cell descriptors associated with gate cells and nest boxes, or close to gate cells and nest boxes, are highly visited. Once again, from the figures, it can be concluded that complex simulations run using the proposed relative movement patterns mechanism are realistic.

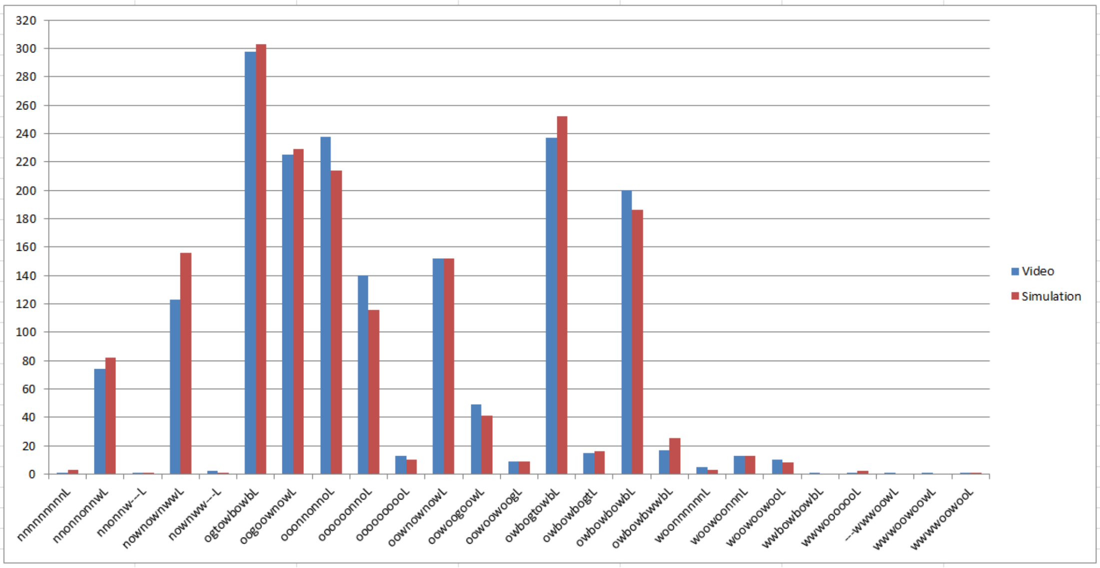


FIGURE 6.14: Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for the left division of the example Complex scenario

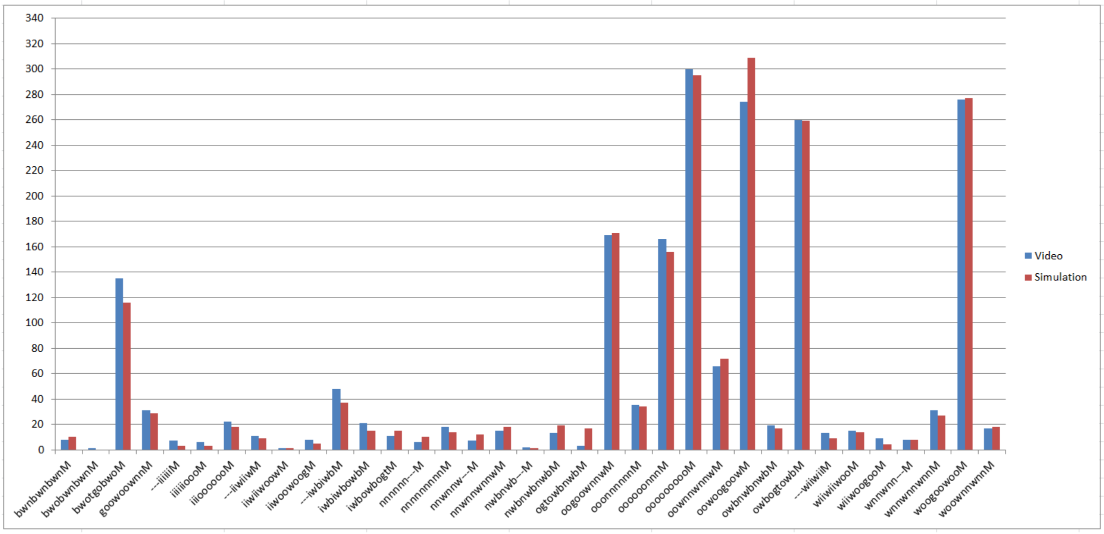


FIGURE 6.15: Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for the middle division of the example Complex scenario

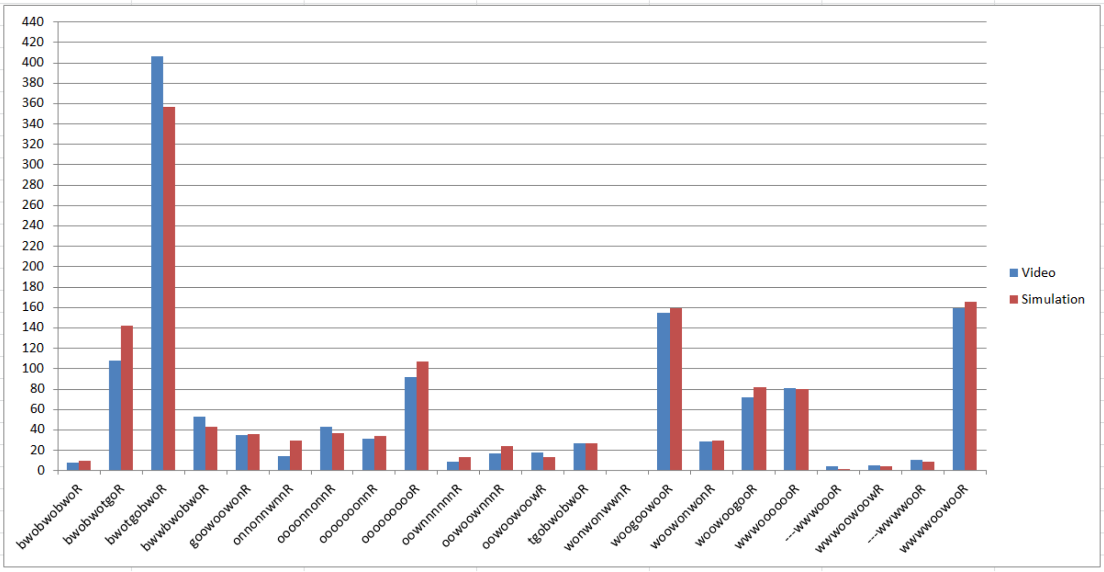


Figure 6.16: Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for the right division of the example Complex scenario

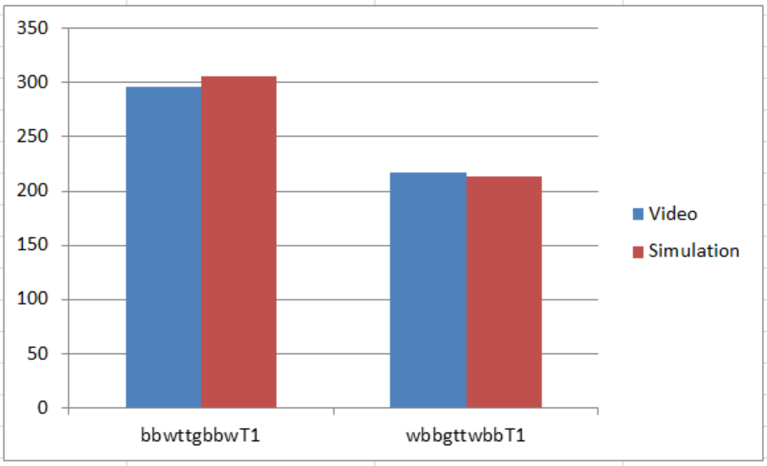


Figure 6.17: Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for tunnel 1 of the example Complex scenario

6.3.3 Comparison of Frequency with which Divisions were Visited (Complex Scenarios)

The results obtained with respect to the frequency with which individual divisions were visited are presented in Figure 6.19. Note that the figure also includes frequency counts for the two tunnel locations (Tunnel 1 and Tunnel 2). From the figure it can again be observed that there is a good correspondence between the simulation data, using both absolute and relative patterns, and the video data; indicating that the simulation is realistic.

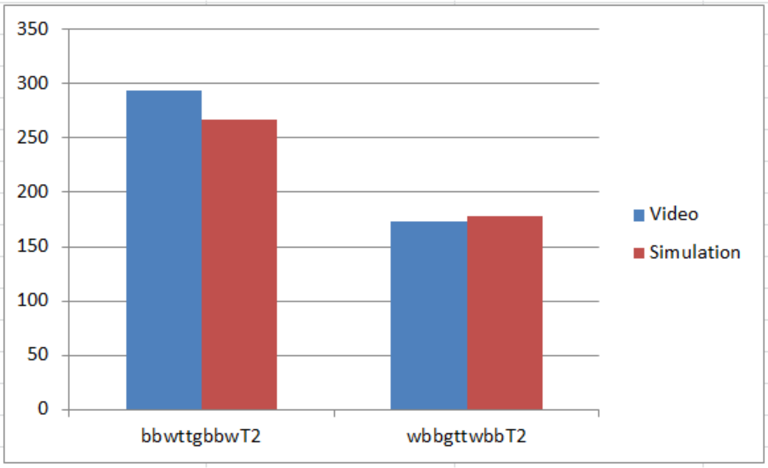


FIGURE 6.18: Comparison of cells descriptor counts in video data with simulation data, using the relative mechanism for tunnel 2 of the example Complex scenario

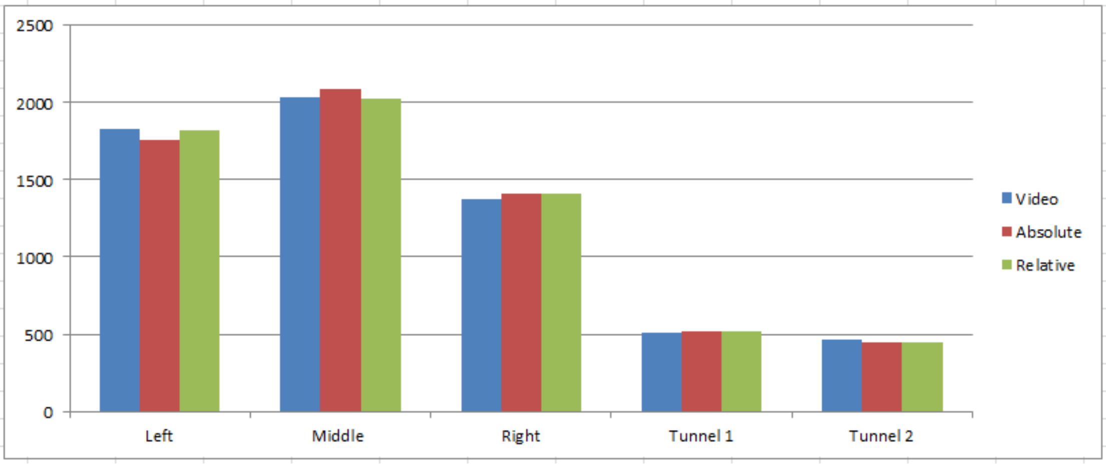


FIGURE 6.19: Comparison of divisions counts in video data, with the simulation data, using absolute and relative movement patterns, and a Complex scenario

6.3.4 Comparison of Frequency of Divisions Transits (Complex Scenarios)

This subsection presents the comparison of the results obtained by considering the number of times that a mouse agent moved from one division to another division (including the same division). This information was extracted from the results presented above (using absolute, relative and real patterns). To this end nine division transitions were identified: (i) $LtoL$, (ii) $LtoM$, (iii) $LtoR$, (iv) $MtoL$, (v) $MtoM$, (vi) $MtoR$, (vii) $RtoL$, (viii) $RtoM$ and (ix) $RtoR$.

With respect to the transition from division to division analysis the results are presented in Figure 6.20. In the figure the nine different transitions considered are listed on the X-axis, while the Y-axis represents the frequency counts. From the figure it can firstly be observed that there is a good correspondence between the simulation data and the real video data indicating that the simulation is realistic, thus corroborating the results previously presented.

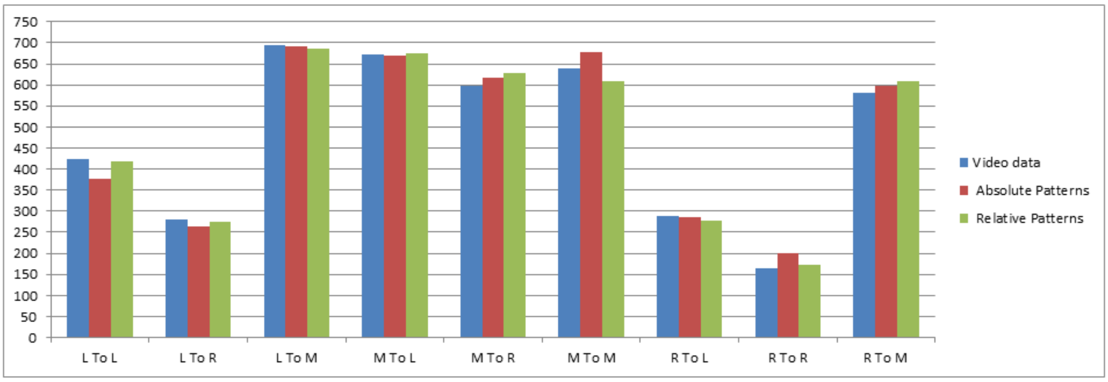


FIGURE 6.20: Comparison of division to division transit counts in real life video data with simulation data using absolute and relative movement patterns and the example complex scenario

6.3.5 Comparison of Absolute versus Relative Movement Patterns (Complex Scenarios)

In this section it was explored that, whether absolute or relative movement patterns produced the “best” simulation. The evaluation was conducted using two sets of experiments. The first set of experiments considered the closeness of the simulation occurrence counts to the actual occurrence counts. The second considered the simulation counts only by comparing the counts recorded in the first half of simulation with the counts obtained in the second half of the video.

In the first case the comparison of the operation of absolute movement patterns versus relative movement patterns was conducted firstly by comparing the differences in operation between the two mechanisms in terms of average error (Avg_Error, as calculated using Equation 6.1). Thus, with respect to the frequency counts presented in Sub-sections 6.3.1 and 6.3.2 above, Table 6.4 gives the Avg_Error values obtained. From these results it can be seen that the relative mechanism produced simulations that were closest to the real data. This contradicts the results presented in Sub-section 6.2.4 where the absolute mechanism was found to be more accurate. It is consequently conjectured here that the relative mechanism is better suited to complex scenarios.

In the second case, video data of Complex Category scenario into two parts, the first part was used to extract MPs and then to utilise those MPs in MABS. The results from the second set of experiments are given in Figure 6.21. Figure 6.21 gives the results obtained using the grid 19×8, where blue bars indicate the frequency that each division was visited in the simulation data and the red bars the frequency obtained from the remaining part of the video (not considered for MPs extracting). From the figure it can be observed that the MPs driven simulation operation is similar to the video data and

TABLE 6.4: Comparison of AMPs versus RMPs in Complex Scenarios

Name of Division	Absolute Mechanism (Avg_Error)	Relative Mechanism (Avg_Error)
Left Division	0.011	**0.001**
Middle Division	0.009	**0.002**
Right Division	**0.005**	**0.005**
Tunnel 1	**0.000**	0.001
Tunnel 2	**0.003**	**0.003**
Total	0.029	**0.011**

divisions highly visited in simulation data are also highly visited in video. Thus it could be concluded that Movement Patterns could be used for unseen video scenarios.

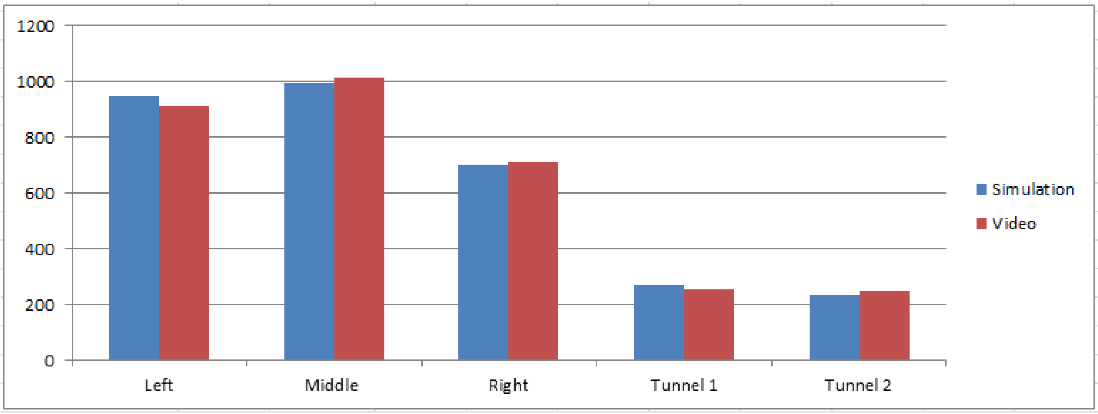


FIGURE 6.21: Comparison of division counts using simulation data split 50:50

6.3.6 Usage of Relative Movement Patterns With Respect to Alternative Category 2 Scenarios

Earlier in this thesis it was argued that an advantage of relative movement patterns is that they are much less restrictive than absolute movement patterns in that they are not limited to a very specific environment. This subsection provides some evidence that this is indeed the case by considering an alternative complex scenario to that considered for evaluation purposes above (for which video data was available). Figure 6.22 shows the environment used (the colour coding is the same as that used in Chapter 3, Figure 3.4). In this case the “playing area” is larger (40 × 11 using the same grid scale) and features six divisions, whereas the environment considered for training and evaluation purposes featured three[1]. In this case the grid cell locations can again be represented using relative addresses and consequently the same movement patterns used earlier can be utilised. However, in this case there is no ground truth” (real video data) to compare with, when running the simulation and observing its operation; therefore, a justification

---

[1] Note that the earlier area/division labelling, $\{L, M, R\}$, had to be reinterpreted with respect to this alternative scenario.

that it operated in a realistic manner has been obtained when shown to domain experts. Details concerning this scenario were published by the author in [152]. Simulations run using this environment also demonstrated that the previously generated movement patterns were entirely suited to generating realistic simulations using this environment and similar alternative environments (of course absolute movement patterns could not be used for this purpose).

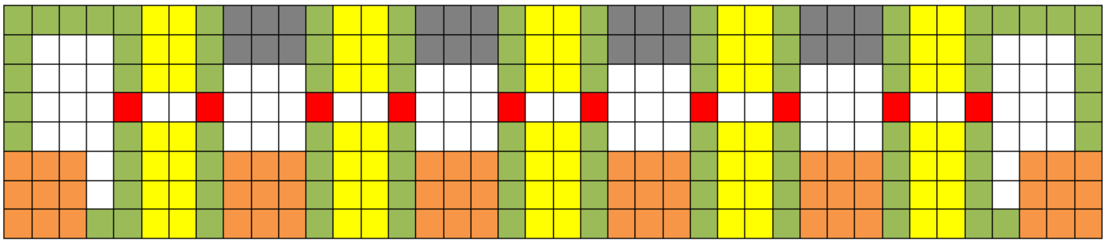


FIGURE 6.22: Case study: Usage of Relative Movement Patterns With Respect to an Alternative Category 2 Scenario.

6.4 Category Study 3: Two Rodents Scenarios

The third category of scenario considered was the two rodent scenario. As before the operation of the simulations was analysed by comparing the simulated behaviour with “real life” video data so that the number of occurrences of each grid location in the simulated data could be measured against the number produced using the real video data. In the case of the relative mechanism the comparison was again conducted by comparing the number of occasions that each descriptor was recorded with respect to the simulation and real life video data. The objectives of the evaluation were:

1. To compare the operation of Category 3 simulations founded on absolute patterns, with and without the usage of the state concept, with the real life equivalent operation.
2. To compare the operation of Category 3 simulations founded on relative patterns, with and without the usage of the state concept, with the real life equivalent operation.
3. To compare the operation of the absolute and relative mechanisms, with and without states, in the context of Category 3 simulations.
4. To investigate the usage of relative movement patterns with respect to Category 3 scenarios other than the scenarios used for training purposes.

For the evaluation two different real life video data sources were used, referred to simply as: (i) Video 1 and (ii) Video 2. The nature of this video data was presented in Chapter 3, Subsection 3.7.3. Note that the videos are not of the same length. Each of the above objectives is considered in the following sub-sections in terms of the results obtained.

6.4.1 Comparison of the Usage of Absolute Movement Patterns with Real Life Experimentation (Two Rodent Scenarios)

In this section evalualtion was conducted using two sets of experiments. The first set of experiments considered the closeness of the simulation occurrence counts to the actual occurrence counts. The second considered the simulation counts only by comparing the counts recorded in the first video with the counts obtained in the second video and vice versa.

In the first case, by considering, the absolute movement pattern mechanism the environment was divided into a grid comprising 14×14 cells. Thus a total of 196 absolute addresses, too many to display in bar chart form as in the case of the evaluation results presented earlier in this chapter. Recall that the principle for the grid cell size to be adopted was that a grid cell should act as a "minimum bounding box" surrounding the nature of the entity that the agents are to represent. In the case of the simple scenario category evaluation the entities were brown rats, in the case of the two rodent scenario category evaluation presented here laboratory mice were used. A brown rat is much bigger than a laboratory mouse; in the video data the body length of a rat is about 48 pixels (excluding the tail which can add a further 10 pixels) while a mouse measures 30 pixels. Thus the simulation environment grid size used for the evaluation presented here is much smaller when considering laboratory mice as oppose to brown rats, and consequently there were many more absolute addresses to consider than in the case of the earlier evaluations presented (note that in the case of the complex scenario evaluation the entire space was not used, some parts of the space were outside the "playing area", so the number of addresses was still manageable).

Thus, for the absolute analysis in the context of two rodent scenarios the concept of a coarse grid, as used previously in the context of the simple scenario category evaluation, was adopted. The environment was divided into 7×7 "blocks", each block representing $2 \times 2 = 4$ "standard" grid cells giving 49 blocks.

The results are presented in Figures 6.23 to 6.26. Figures 6.23 and 6.24 give the results using coarse block grid for videos 1 and 2 using absolute movement patterns without states, while Figures 6.25 and 6.26 present the results for the same videos using absolute movement patterns with states. In each case, the blue bar indicates the occurrence counts abstracted from video data while the red bar the occurrence counts for the simulation. From the figures it can be observed that the behaviour of mice agents in the simulation is similar to that featured within the video data. Thus it could be concluded, on the bases of the results shown in Figures 6.23 to 6.26, that the operation of the simulations using absolute movement patterns is appropriate.

The overall simulation Avg_Error obtained using absolute addressing and no states for Video 1 was 0.08 and for Video 2 was 0.11. When using states the accuracies obtained were 0.12 and 0.13 respectively. For reference the frequency counts obtained in the context of the evaluation using two rodents scenarios and absolute addressing have been included in Appendix B, Tables B.1, B.2, B.3 and B.4 of this thesis.

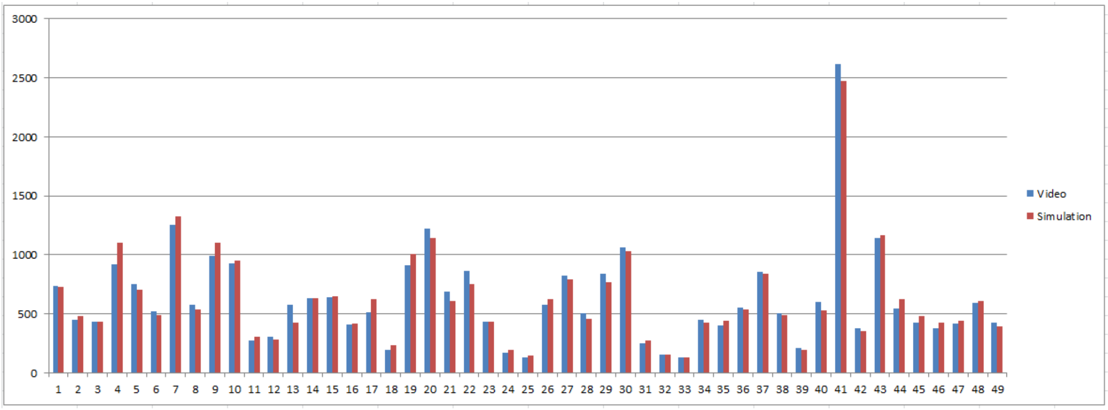


FIGURE 6.23: Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns without state and (ii) video 1 data; for the example scenario Two rodent category

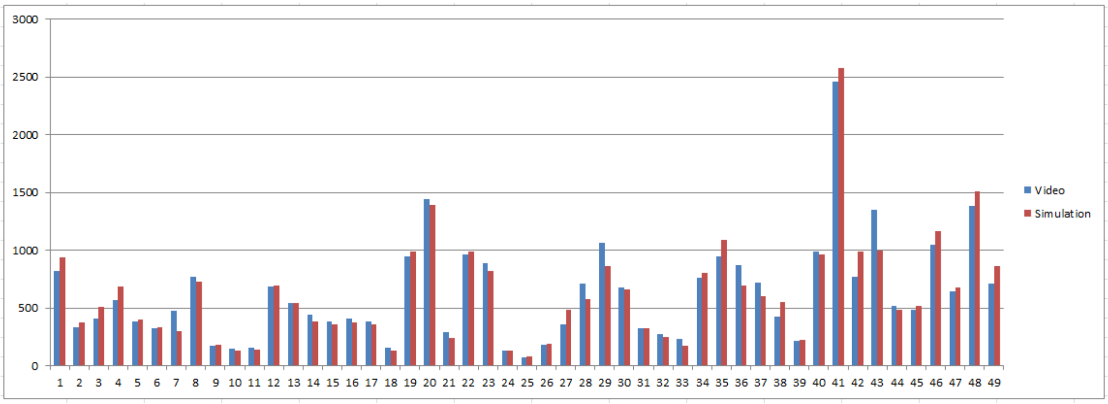


FIGURE 6.24: Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns without state and (ii) video 2 data; for the example scenario Two rodent category

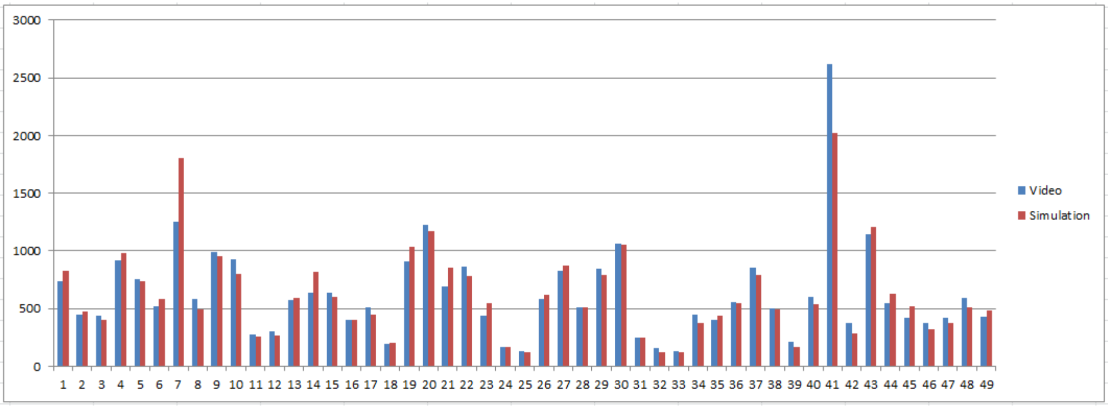


FIGURE 6.25: Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns with state and (ii) video 1 data; for the example scenario Two rodent category

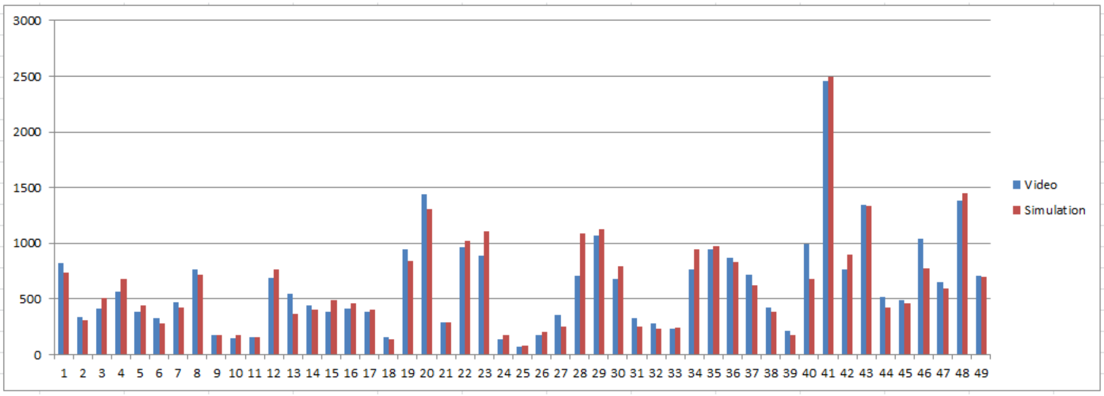


FIGURE 6.26: Comparison of simulation data with real life video data in terms of blocks visited using: (i) absolute movement patterns without state and (ii) video 2 data; for the example scenario Two rodent category

In the second case, video 1 data of Two Rodents Scenarios for the extraction of MPs and then to utilise in MABS. The evaluation of MPs was done in cross-validation, where video used for testing has not been seen in extracting process of MPs as given in Sub-Section 6.2.4 and 6.3.6. The out come of MPs driven MABS (based on video 1) was then compared with the MPs extracted from video 2 and vice versa. Figure 6.27 and 6.28 give the results obtained using the coarse grid thus measured $7 \times 7 = 49$ blocks, where blue bars indicate the frequency that each block was visited in the simulation data and the red bars the frequency obtained from video (not considered for MPs extracting, video 2 for Figure 6.27 and video 1 for Figure 6.28). From the figure it can be observed that the MPs driven simulation operation is similar to the video data. Thus it could be concluded that Movement Patterns could be used for unseen video scenarios.

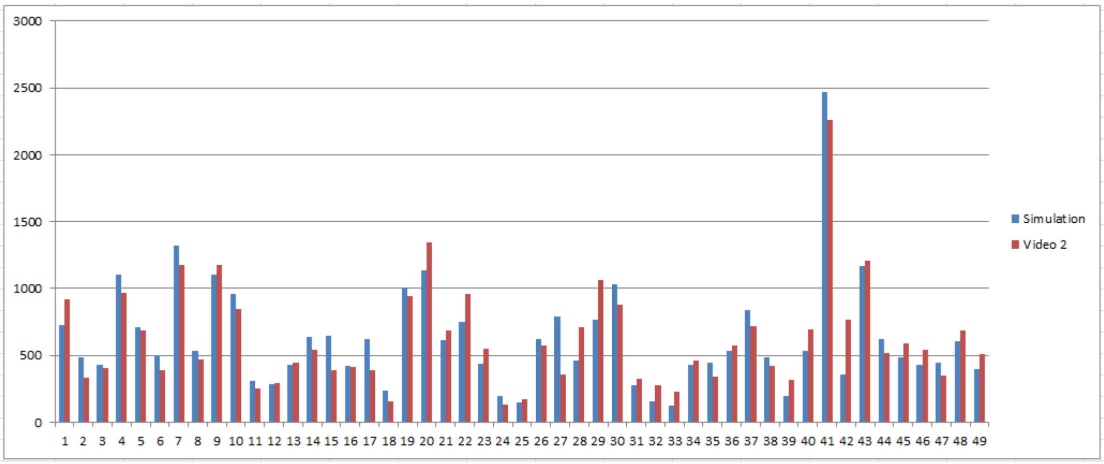


FIGURE 6.27: Comparison of location counts in MPs (extracted from video 1) driven simulation data with video 2 data not used for MP extraction

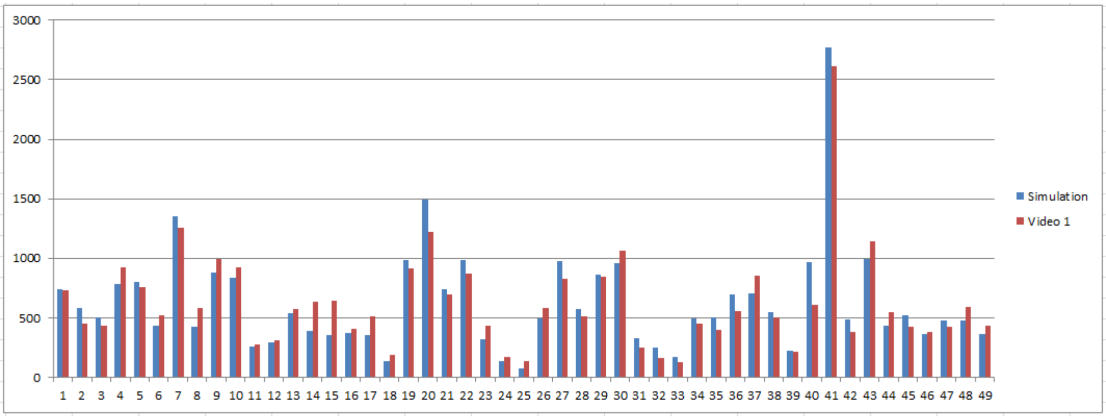


FIGURE 6.28: Comparison of location counts in MPs (extracted from video 2) driven simulation data with video 1 data not used for MP extraction

6.4.2 Comparison of the Usage of Relative Movement Patterns with Real Life Experimentation (Two Rodent Scenarios)

For the evaluation using relative movement patterns, with and without states, the environment featured 109 relative addresses (location descriptors). The available descriptors are similar to those used with respect to the simple scenario category and are listed in Appendix A, Table A.3. If we include states we have 125×4 state-address combinations. Again, too many to present in bar graph form. In this case it did not make sense to consider a coarse grid as in the case of the absolute addressing evaluation presented in the foregoing section. For reference the complete results are presented in tabular form in Appendix B included at the end of this thesis. From these tabulated results the overall computed simulation average error using relative addressing and no states, for Video 1 and 2, was 0.47 and 0.57 respectively, whilst when using states it was 0.58 and 0.62 respectively as given in Appendix C, Tables C.1, C.2, C.3 and C.4 of this thesis. From this we can conclude that the relative MP performance is comparatively produces better results.

6.4.3 Comparison of Absolute versus Relative Movement Patterns, With and Without States (Two Rodent Scenarios)

This subsection considers the relative performance of the four mechanisms considered in this section: (i) absolute addressing without states, (ii) absolute addressing with states, (iii) relative addressing without states and (iv) relative addressing with states. The comparison was conducted by considering the Avg_Error of the simulations (calculated using Equation (6.1) as in the case of the previous reported evaluations). The outcomes are presented in Table 6.5. From these tabulated results the overall computed simulation net average error, using absolute without and with state, relative addressing without and with state, for Video 1 and 2, was 0.09, 0.12, 0.34 and 0.35 respectively. From the table it can firstly be seen that, good results were obtained. Secondly, it is to be mentioned that surprisingly the absolute mechanism behaves better than the relative one, but this may be due to the small scale of the environment, while the relative mechanism is more general. Thirdly, and interestingly, it can be seen that, in case of absolute mechanism the states worsen the simulation results, while in case of relative mechanism the difference is negligible. Thus indicating that usage of states is unnecessary. Note that, the main role of states, is with respect to the MP selection process. The reason for including the work on states was, that it was anticipated that, this might make a difference, not that case as the evaluation eventually demonstrated. Overall it was thus concluded that, the absolute mechanism could perform better with the limitation of having been run in an identical environment, furthermore, that, large-scale experiments are needed to see that, what is the impact of relativeness and states to accuracy of the simulations.

Table 6.5: Summary of simulation average error results

Techniques	Absolute Avg_Error. Video 1	Absolute Avg_Error. Video 2	Net avg error. Video 1 and 2
Absolute	**0.08**	0.11	**0.09**
Absolute + state	**0.12**	0.13	**0.12**
Relative	**0.26**	0.42	**0.34**
Relative + state	**0.25**	0.45	**0.35**

6.4.4 Usage of Relative Movement Patterns With Respect to Alternative Category 3 Scenarios

Experiments were conducted using three mice simulations. In the case of the relative addressing mechanism, as already noted, this can be used in the context of alternative environments. The video data on which the training was conducted featured obstructions and thus experiments were also conducted using alternative environments that featured alternative dispositions of obstructions as shown in Figures 6.29 and 6.30. Observation of the operation of these simulations indicated that these seemed to work in a realistic manner (no ground truth was available so conclusive experiments could not be conducted).

6.5 Summary

This chapter has presented an evaluation of the absolute and relative movement pattern concepts as incorporated into the MABS framework presented in Chapter 4. It is difficult to evaluate the operation of MABS with respect to any gold standards but the mechanism adopted for the work presented in this thesis was to “complete the loop”. The operation of MABS runs were evaluated by “videoing” the simulation and repeating the process of extracting location data from the simulated video data. The occurrence counts whereby locations were visited were then compared with the original set of locations extracted from the training video data. The overall purpose of the evaluation was

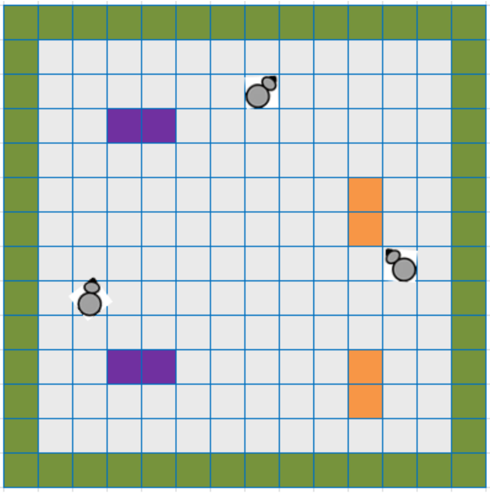


FIGURE 6.29: Usage of Relative Movement Patterns With Respect to Alternative Category 3 Scenarios (3 mice), for Video 1

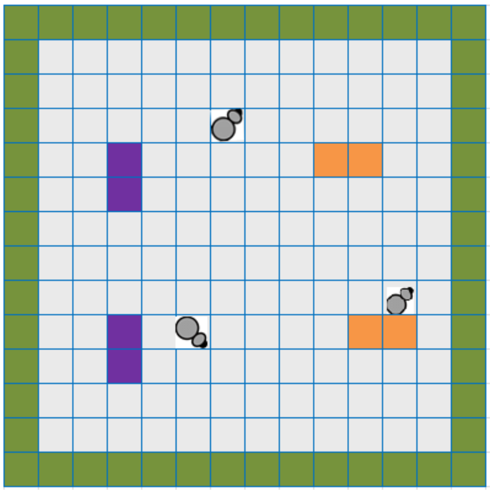


FIGURE 6.30: Usage of Relative Movement Patterns With Respect to Alternative Category 3 Scenarios (alternative obstruction arrangement), for Video 2

to demonstrate that the operation of the proposed MABS framework, founded on the concept of movement patterns, was as realistic as possible. The results were presented in bar graph form and in terms of an overall numeric average error. The average error results that were obtained are summarised in Table 6.6 (best results highlighted in bold font) and statistical significance holds, by arguments similar to that in Sub-section 6.2.4, for all results in this chapter. With respect to the results presented in the table it should be recalled that the average error values were obtained using two different types of rodent (rats and mice) and three different categories of environment. It should also be recalled that states were only of relevance with respect to the two rodent scenario category. However, it can be observed that, with respect to the individual representations, the absolute mechanism produced the best results. In the context of the different categories of scenario considered, the absolute mechanism without states was found to be the most effective with a average error value 0.09 as shown in Table 6.6; while when used along with states produced a slightly worse average error of 0.12. In the case of the relative mechanism (without states) this produced best result in the case of Complex scenario category (an average error of 0.01 as shown in table 6.6). Usage of the relative mechanism (without states) produces a better performance, for the scenarios discussed in this thesis, than the relative mechanism with states. Overall it can be concluded that the absolute mechanism without states is most appropriate where we have a fixed environment for which video training data is available; whilst the relative mechanism without states is most appropriate where we wish to consider alternative environments for which video training data is not specifically available.

The following chapter concludes the work on the usage of movement patterns, extracted from video data, to support MABS as presented in this thesis by presenting

Table 6.6: Summary of average error results with respect to the three different scenario categories (best results highlighted in bold font)

Scenarios	Absolute	Absolute + State	Relative	Relative + State	Average
Simple Scenario	**0.13**		0.20		**0.16**
Complex Scenario	0.02		**0.01**		**0.01**
Two Rodent Scenario (Video 1)	**0.08**	0.12	0.26	0.25	**0.17**
Two Rodent Scenario (Video 2)	**0.11**	0.13	0.42	0.45	**0.27**
Average	**0.09**	**0.12**	**0.17**	**0.35**	

some conclusions, main findings and some ideas for future research directions.

Chapter 7

Conclusion and Future Research Works

7.1 Introduction

This chapter concludes the work described in this thesis. The chapter presents a summary of the work described in Section 7.2, the main findings in Section 7.3 and then some suggestions for future work in Section 7.4.

7.2 Summary

The thesis commenced with a scene setting chapter, chapter 1, where the motivation for the work, the research issues to be addressed, the contributions of the work and the research methodology were outlined. The research methodology noted that three categories of simulation scenario were to be considered in the thesis: (i) Simple scenarios featuring a single rodent in a box with no obstructions, (ii) Complex scenarios featuring a number of divisions (and a singe rodent) and (iii) Two rodent scenarios that also featured obstructions.

From the literature a number of different computer simulation frameworks can be identified, the framework selected with respect to the research work presented in this thesis was the Multi Agent Based Simulation (MABS) framework. This was selected because each moving object in the video (rodents) can be encapsulated as an agent; the MABS approach to computer simulation therefore seemed like a natural choice. In Chapter 2 the main characteristics of software agents were listed. In the context of the proposed MABS the mouse agents conform to these characteristics as follows.

1. **Responsive behaviour**: Using the proposed Movement Patterns (MP) mechanism agents will "respond" to their environment where a considered MP entails moving to a blocked area, or a location outside of the environment.

2. **Pro-active behaviour**: Mouse agents, with the context of the proposed MABS, are pro-active in the sense that they decide their own behaviours in terms of MPs.

3. **Social behaviour**: In the research reported on in this thesis the concept of states is proposed to support MP selection. More specifically the concept of states was used to define the social relationship between mouse agents so as to support the selection of MPs.

4. **Flexible behaviour**: Mouse agents, as envisioned in this thesis, feature flexible behaviour in that the agents have the capability of considering alternative MPs.

Also in Chapter 2, where simulation were discussed, it was noted that the most challenging task of MABS, and simulation in general, was the acquisition of data with which to "encode" the operation of the simulation. Different methods for data acquisition were discussed and it was observed that translating expert domain knowledge into some appropriate format was time consuming and error prone, hence the idea of extracting knowledge from video of the real world scenarios to be simulated. Chapter 2 also discussed previous work on how to identify moving objects (rodents) in video data and to how extract data from video. It was noted that, with respect to the thesis, background subtraction was the method adopted for object identification. Once detected objects were expressed in terms of blobs; blob locations were tracked throughout the video for further processing.

Chapter 3 introduced the concept of Movement Patterns (MPs). The proposed mechanism for environment modelling and the process for extracting location information from video data (which could then be mined for MPs) was also described. The significance of the environment modelling was that that nature of this modelling dictated the nature of the video data extraction process and the subsequent MP mining. The fundamental idea was to divide the environment into equal sized grid cells so that the entire environment could then be linearized by assigning sequential location IDs to each grid cell (tile). Note that a linearised space can processed more efficiently than a non-linearized space. In addition a ground type descriptor was associated with each tile. Locations within the environment could be labelled using the ID numbers (absolute addressing) or in terms of the location ground type descriptor and its neighbouring descriptors (relative addressing). Both were considered in the thesis.

Chapter 4 presented the MP concept in more detail. Given that locations could be described using absolute or relative addressing, MPs could be either: (i) absolute or (ii) relative. From a functional perspective the significance was that absolute MPs could only be used with respect to simulations that featured the same environment as that from which the patterns were mined, whilst relative MPs were more versatile and could be used for a verity of simulations. The proposed structure of MPs, regardless of representation was:

$$MP = \langle F, S, v, Path \rangle$$

Where F was the "From" location, S was a collection of zero, one or more states describing the spatial relationship between agents featured in a scenario, v was the "movement vector" and $Path$ was the "path" which an agent needed to follow to get a "To" location. The length of the path could be anything between one (thus comprising only a "To" location) to some specified maximum. In the thesis this maximum was set to 5. The significance is that if $|Path| > 1$ the respective agent has a "memory"; it knows where it is going. For the purpose of mining MPs from the extracted location information a bespoke software system was developed, this was also presented in Chapter 4.

The operation of the rodent behaviour MABS was described in detail in Chapter 5. The proposed MABS framework included a mechanism whereby the mined MPs were used to drive the simulation. Chapter 5 also included details concerning the simulation interface and the visualization mechanism used to display simulations as they progressed.

It is difficult to evaluate the operation of MABS, and computer simulations in general, with respect to any "gold standard"; however, the main criteria for measuring the effectiveness of simulations is for them to be as realistic as possible. This was the "yard stick" used to evaluate the mechanisms proposed in the thesis as discussed in Chapter 6. More specifically the evaluated used two different realism criteria: (i) corroboration and (ii) internal consistency. The first one was concerned with establishing that a simulation operated in the same manner as the real world scenarios being simulated, in other words the simulation results should be at least similar to the real world results. AS evaluation of simulation as given in Chapter 6, shows that, grid cells which are highly visited in video are also highly visited in simulation. While internal consistency was concerned with the requirement that all parts of simulation system functioned in line with established underlying concepts and theories. To address the first one, the operation of simulations was compared against their real world representation. To address the second the operation of the simulated system was analysed in term of the mechanisms used to realised the simulations. For the simple and two rodents categories of scenario, using absolute MPs was found to produce better results than when using relative MPs, however, for the complex category of scenario the use of relative MPs produced better results.

7.3 Main Findings and Contribution

In this section the main findings and contributions of the research considered in this thesis are presented. The discussion is initially presented in terms of the research issues identified in Chapter 1 and then in terms of the overriding research question that this thesis seeks to address. Each of the research issues is considered in turn as follows.

1. *What is the most appropriate mechanism for extracting data from video?* Moving objects can be identified using different mechanism as discussed in Chapter 2. For example we can use: (i) the geometric shape of an object such as rectangle or ellipse; (ii) an articulated shape in the case of objects consisting of "body parts" that are held together with "joints" in the same way that a human body

comprises legs and arms and so on, and (iv) blob representations. The latter basically defines a group of connected pixels describing a moving object in the video data. The most appropriate mechanism with respect to the work presented in this thesis was considered to be this last mechanism because of its simplicity and versatility. Blobs were tracked using a Particle Filtering mechanism so that location information could be extracted from video data.

2. *Given that data can be successfully extracted from video how can this best be translated to form the input to a simulation framework (more specifically a MABS framework).* The solution presented in this thesis, and the central idea espoused by the thesis, is the concept of Movement Patterns (MPs). An MP, as noted earlier in this chapter, is a tuple of the form:

   $$MP = \langle F, S, v, Path \rangle$$

   Where: (i) F is the "From" location (where the movement represented by the MP starts); (ii) S is a collection of zero, one or more states describing the spatial relationship between agents featured in a scenario; (iii) v is a *movement vector*; and (iv) $Path$ is the *path*, encapsulated by the MP, which an agent needs to follow to get to the "To" location. MPs, as demonstrated in the thesis, can easily mined from location data which in turn has been extracted from video data. MPs, as demonstrated earlier in the thesis, can be readily used as input to drive a MABS framework. A MABS framework was proposed which had been specifically designed to operate with MPS. Two kinds of MP were considered: (i) Absolute MPs and (ii) Relative MPs. The distinction being how location addressing is conducted. A number of variations of these two kinds of MP were also proposed: with and without divisions, where divisions are interconnected areas; and with and without states where states are used to capture the relationships between pairs of rodents featured in a simulation. Note that divisions feature in the complex scenario category and states with respect to the two rodent scenario category.

3. *How best can a MABS model operate given the nature of the input data?* The fundamental idea was to use MPs in a weighted random probabilistic manner. Each MPs that had been mined (extracted) from video data had a probability value associated with it. The proposed idea was that the MP selection process be embedded in the MABS simulation. The process for selecting MPs starts by extracting a set of candidate MPs from an MP database, that feature the given agent's current location (the From location F, and possibly a division identifier, in the context of the MPs) and, in the case of scenarios featuring more than one agent, the contents of the current agent's set of states S describing its relationship with other agents that exists in the scenario. If there is no MP available these constrains are relaxed so that alternative, but similar, From locations are considered. When the relative representation is adopted the next step is to prune those MPs which

are not "legal" (this is not necessary where the absolute mechanisms is adopted because absolute MPs will always be legal as they only operate with environments identical to those for which they are generated; not the case using relative patters but as a consequence such patterns may not always be legal). For a potential MP to be legal the indicated To location, and any intervening way points, must be both *valid* and *accessible*. A MP is said to be accessible (features accessibility) if:

(a) The To location and associated way points can all be accessed from the agents current location.

(b) The To location is also a From location in a least one MP in the MP database.

The significance of the later is so as to ensure that an agent will not get "stuck" at a location where it cannot move out from. If there is no MP available after pruning the set of MPs for the given From locations, the constrains for the From location are again relaxed in the same manner as in the case where no suitable MPs were found in the first place. After finding a set of available MPs, one MP is selected using the proposed random weighted probability mechanism.

4. *What is the most appropriate mechanism for evaluating simulators that operate using data extracted from video?*. As noted earlier in the thesis there, is no "gold standard" with which to evaluate the operation of MABS. The proposed mechanism for evaluating MP enabled MABS was primarily by comparing the location visit counts obtained from both video data and simulation data; in effect videoing the simulation and extracting location information in the same manner as for the original video data. With respect to the comparison, in some cases, cell locations were compared. The results were typically recorded in bar chart from. In some cases the number of grid cells to be considered was too large and instead "blocks" of grid cells were considered. An accuracy measure was also derived.

Returning back to the original overriding research question:

> **"What are the most appropriate mechanisms that can be adopted so that some form of machine learning can be applied to video data describing animal (human) behaviour so as to extract sufficient information to populate and drive a MABS framework of some description?"**

The evaluation presented in the thesis indicated that the mining of MPs from video data, using these to populate a MABS and consequently using these in a random probabilistic manner, within the context of the MABS framework, produced good simulation results. Four different MP variants were considered: (i) absolute with state, (ii) absolute without state, (iii) relative with state, and (iv) relative without state for our proposed movement patterns (MPs). Best results were obtained using the absolute without states variant;

however, as noted above, absolute MPs feature some disadvantages with respect to general applicability. Thus there may be situations where more general applicability is desirable and hence the relative mechanism might be more appropriate.

The main contribution of the thesis were listed in Chapter 1. For convenience they are listed again here:

1. **Video Analysis Software.** Software, the Video Data Acquisition Software (VADS), for processing video data that incorporates a technique for identifying and tracking moving objects in such data. The proposed software can automatically track moving objects in video data and store captured location information in a database format.

2. **Absolute mechanism for mining movement patterns from video data.** A mechanism for representing agent (object) movements using the concept of "absolute" movement patterns referenced to some environment origin.

3. **Relative mechanism for mining movement patterns from video data.** A mechanism for representing agent (object) movements using the concept of "relative" movement patterns referenced in terms of the neighbourhood of a given agent.

4. **Movement pattern based simulation.** A mechanism whereby movement patterns, extracted from video data, can be used to drive computer simulations.

5. **A mechanism for agents to have a "Memory".** A mechanism whereby agents have "memory" in the sense that they have a planned route they wish to follow. Using this concept agents are also prevented from "pinging" back and forth between two locations in what may be considered to be an unrealistic (unnatural) manner.

6. **A mechanism for evaluating the quality of video data based computer simulations.** A process for comparing real scenarios with computer simulated scenarios by comparing the real video data with "video" extracted from the simulations (a process referred to as "closing the loop").

The consequent application contributions are then as follows:

1. **A Rodent movement behaviour MABS.** A mechanism whereby behaviourists can study rodent behaviour (movement) in a simulated environment.

2. **A generic MABS mechanism.** A mechanism which can be applied in the context of other behaviour studies such as fire exit simulation and behaviour at rail terminal/station forecourts or airport concourses.

The following methodologies are derived to this research work.

1. **An environment modelling** A mechanism used to represent environments in which the moving objects (agents in a MABS) of interest exist, using a uniform grid representation comprised of cells enumerated in such a way so as to facilitate ready translation across the space. In this context it is acknowledged that grid representations are well established, however, the novel contribution is how such representations have been adopted to represent the environments of interest, the nature of the proposed environment representation has a strong influence on the nature of the proposed MPS.

2. **State Representation.** A mechanism for representing the relationship between pairs of moving objects (agents) using the concept of states, and an associated *state graph*, to support the selection of MPs. Again it is acknowledged that the concept of states in its self is not novel, however, what is novel in the context of the research reported in the thesis is how states are represented so that they can be utilised in the context of the MP selection process.

For confirmation regarding the utility and effectiveness of the proposed MP driven MABS, and the general ideas presented in this thesis, Appendix D gives a letter from Prof. Jane Hurst, the consultant domain expert used with respect to the work presented in this thesis. It should also be noted here that practitioners can set up whatever scenarios they like through the inclusion of objects. The interface to the system presented in Chapter 5 allows non-specialists to do this. Similarly, when using the relative representation, the playing area can, within some operational limitations, be more or less of any size. Also it is easily possible to increase the number of agents/rodents with the limitation of the operational requirements of the system. Tests have been conducted demonstrating that the system can operate easily using five rodents. Note that the conduct of large scale studies has been included as an item for future work. Should a practitioner wish to include some alternative form of behaviour, or consider (for example) 3D environments this will require significant changes to the proposed system that will require software programming capabilities.

7.4 Future Work

The work presented in this thesis has addressed the issue of automatically extracting information (locations and consequently MPs) from video data which was then used to drive computer simulations. Knowledge acquisition is a significant “bottle neck” with respect to many forms of computer simulation. The work presented in this thesis has demonstrated that, in the context of MABS for rodents; using MPs extracted from video data provides for an effective mechanism for automating the computer simulation process. However, various improvements and enhancement to the work can be identified. A number of suggested areas for future work are presented in this section as follows:

1. **Additional video data.** The video data used with respect to the work described in this thesis comprised six hours of data. Although obtaining such video data is resource intensive it would be useful to obtain further data covering additional scenarios so that a much more sophisticated bank of MPs can be established. For example: (i) scenarios featuring more than two rodents, (ii) "maze" scenario as shown in Figure 7.1 (a) and (iii) "O-Box" scenario as shown in Figure 7.1 (b) both of these were considered in [2, 3], but in a different context.

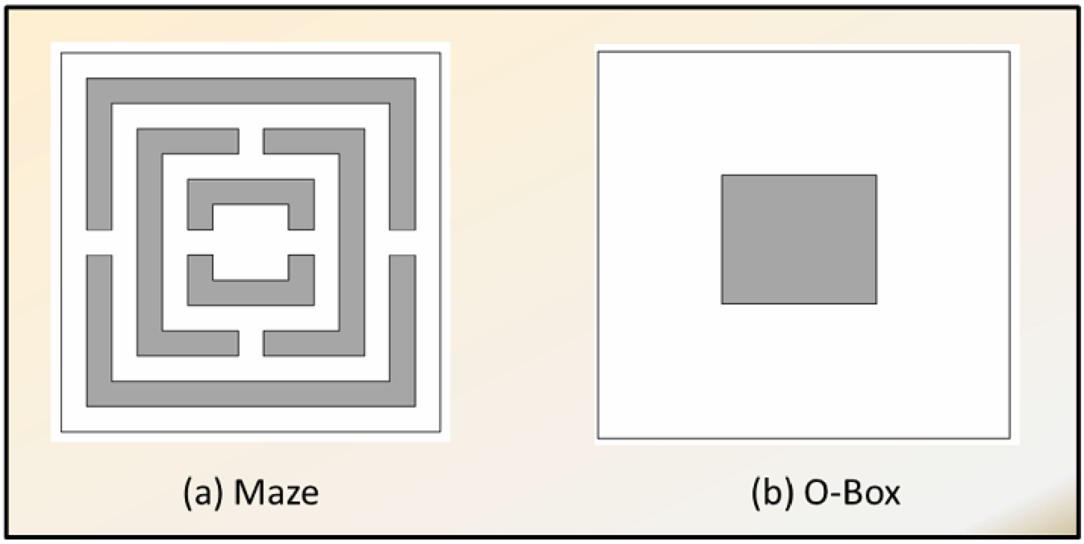


FIGURE 7.1: Additional Video Data for additional scenarios : (a) Maze, and (b) O-Box

2. **Consideration of alternative relative MP window sizes.** In the thesis the work on relative MPs was directed at a "window size" of 3 × 3. It would be interesting to conduct further evaluation with respect to alternative window sizes, and alternative configurations as shown in Figure 7.2 (b) and (c). The dots in the figure indicate locations that might be included in a descriptor.

3. **Consideration of alternative for square zones.** To capture the spatial relationship between rodents during the proposed video mining, a set of square zones were proposed with which to define the relationships between pairs of rodents (agents). Square zones were considered simply for ease of computation. It would be interesting to investigate the potential of capturing the relationship between agents using circular zones defined by a radius. Alternative square or rectangular zones, referenced to the direction in which an agent is facing, might be considered.

4. **Alternative mechanisms for relative location descriptions.** The descriptor mechanism proposed in the thesis has some similarity with mechanisms to describe texture in images. A fruitful avenue for further research would be to consider whether such techniques could be adopted with respect to relative location description. For example we might consider using Local Binary Patterns (LBPs)

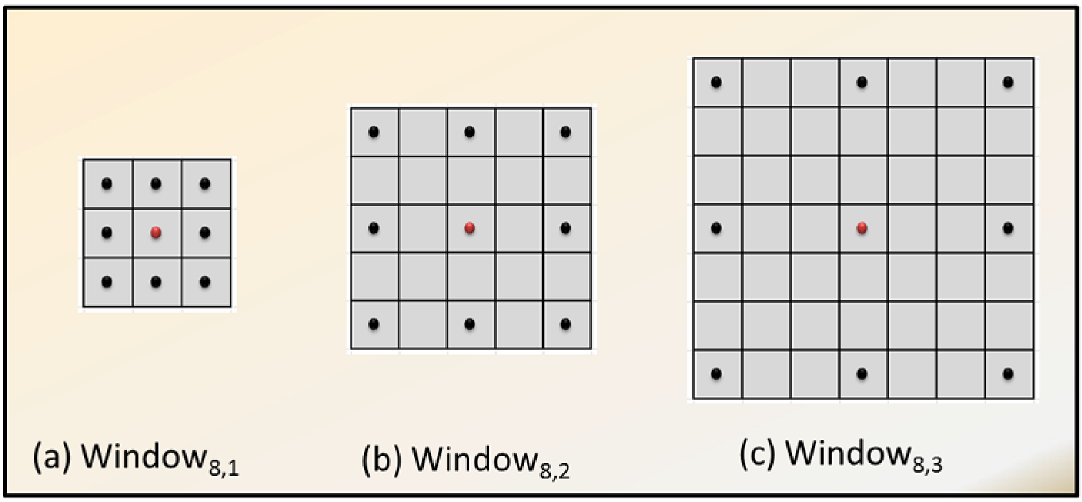


Figure 7.2: Some suggested alternative configuration for relative location descriptors.

as proposed in [60, 159] and used in [37] in the context of analysing Google earth satellite imagery.

5. **Missing MPs.** We can imagine a situation where there are no valid MPs for a given location. In the thesis this has been addressed by ensuring that there is sufficient data covering all locations. However, this may not be practical in the context of more sophisticated and/or larger simulations. Some mechanism is therefore required whereby the issue of missing MPs can be addressed. One idea is to consider a "nearest match" process, although the precise nature of such a process would require further investigation.

6. **Large scale studies.** Only small scale scenarios were considered in this thesis, mice in a box scenarios where the box measured $1.22 \times 1.22\text{m}^2$. It would be interesting to conduct larger scale studies although this would involve further resource and the usage of several video cameras. This in turn would mean that rodents need to be tracked across videos, a challenge that has been significantly investigated in the context of video surveillance [105]. This technology could clearly be adopted for larger scale studies of rodent behaviour. An extension to this would be heat sensitive cameras that could be used in a controlled outside environment (and at night when much rodent activity takes place).

7. **Alternative large scale application domains.** Earlier in the thesis it was noted that the proposed MP based MABS framework would be equally applicable to alternative simulation application domains. For example simulating how people move round station forecourts or airport terminals. This more general applicability of the mechanisms proposed in this thesis would also provide for a fruitful area for further investigation.

8. **Updating of states during MP execution.** With respect to the work presented in this thesis the concept of states is used in the context of MP selection. States are not updated whilst an MP is being executed. It would be interesting to consider mechanisms whereby states are updated during MP execution and how this might be used to abandon a current MP in favour of an alternative MP. This might conceivably enhance the realism of simulations.

In conclusion it is suggested that the proposed mechanism for extracting locations, and consequently mining MPs, from video data, to drive a MABS has provided a useful proof of concept approach to rodent simulation that is likely to have wide ranging benefits.

References

[1] Emmanuel Agiriga. *A Multi Agent Based Simulation Framework For Mammalian Behaviour*. PhD thesis, Department of Computer Science, Univeristy of Liverpool, 2016.

[2] Emmanuel Agiriga, Frans Coenen, Jane Hurst, Richard Beynon, and Darek Kowalski. Towards large-scale multi-agent based rodent simulation: The “mice in a box” scenario. In *Research and Development in Intelligent Systems XXVIII*, pages 369–382. Springer, 2011.

[3] Emmanuel Agiriga, Frans Coenen, Jane Hurst, and Darek Kowalski. A multiagent based framework for the simulation of mammalian behaviour. In *Research and Development in Intelligent Systems XXX*, pages 435–441. Springer, 2013.

[4] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

[5] Natalia Andrienko and Gennady Andrienko. Designing visual analytics methods for massive collections of movement data. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 42(2):117–138, 2007.

[6] Cláudia M Antunes and Arlindo L Oliveira. Temporal data mining: An overview. In *KDD workshop on temporal data mining*, volume 1, page 13, 2001.

[7] J Joshan Athanesious and P Suresh. Systematic survey on object tracking methods in video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) October*, pages 242–247, 2012.

[8] Jay Bakst, Joel Hoffner, and Kris Jacoby. Evaluation of a distribution center towline material handling system through simulation modeling. In *Proceedings of the 28th conference on Winter simulation*, pages 1099–1106. IEEE Computer Society, 1996.

[9] Jerry Banks et al. *Handbook of simulation*. Wiley Online Library, 2007.

[10] Vidit Bansal, Ramachandra Kota, and Kamalakar Karlapalem. System issues in multi-agent simulation of large crowds. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 8–19. Springer, 2007.

[11] Dr Stephen Battersby. *Pest control procedures in the housing sector.* Charted Institute of Environmentol Health, 2010.

[12] Claudia Battista, G Dello Stritto, Francesco Giordano, Raffaele Iannone, and Massimiliano M Schiraldi. Manufacturing systems modelling and simulation software design: a reference model. In *XXII DAAAM International World Symposium*, 2011.

[13] Michael Batty. Agent-based pedestrian modelling. *Advanced spatial analysis: The CASA book of GIS*, 81, 2003.

[14] Laura Betzig, Monique Borgerhoff Mulder, and Paul Turke. *Human reproductive behaviour: A Darwinian perspective.* CUP Archive, 1988.

[15] Tibor Bosse, Zulfiqar A Memon, and Jan Treur. Modelling animal behaviour based on interpretation of another animal's behaviour. In *ICCM'07: Proceedings of the 8th International Conference on Cognitive Modeling*, pages 193–198, 2007.

[16] Thierry Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11:31–66, 2014.

[17] Thierry Bouwmans, Fida El Baf, and Bertrand Vachon. Background modeling using mixture of gaussians for foreground detection-a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.

[18] Alan C Bovik. *Handbook of image and video processing.* Academic press, 2010.

[19] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 568–574. IEEE, 1997.

[20] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1):14–23, 1986.

[21] Joanna J Bryson, Yasushi Ando, and Hagen Lehmann. Agent-based modelling as scientific method: a case study analysing primate social behaviour. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 362(1485):1685–1699, 2007.

[22] Huiping Cao, Nikos Mamoulis, and David W Cheung. Mining frequent spatio-temporal sequential patterns. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[23] Juan P Caraça-Valente and Ignacio López-Chavarrías. Discovering similar patterns in time series. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–505. ACM, 2000.

[24] Peter J Carrington, John Scott, and Stanley Wasserman. *Models and methods in social network analysis*, volume 28. Cambridge university press, 2005.

[25] II Carson and S John. Introduction to modeling and simulation. In *Proceedings of the 36th conference on Winter simulation*, pages 9–16. Winter Simulation Conference, 2004.

[26] Kuei-Hu Chang, Yung-Chia Chang, and Hsiang-Yu Chung. A novel ahp-based benefit evaluation model of military simulation training systems. Hindawi Publishing Corporation, 2015.

[27] K.w. Choy, A.A. Hopgood, L. Nolle, and B.C. O'Neill. Implementation of a tileworld testbed on a distributed blackboard system. In *Proc. 18th European Simulation Multiconference (ESM2004)*, pages 129–135, 2004.

[28] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. volume 25, pages 564–577. IEEE, 2003.

[29] Stephane Cotin, S Dawson, Dwight Meglan, D Shaffer, M Ferrell, R Bardsley, F Morgan, Tad Nagano, Jacob Nikom, Paul Sherman, et al. Icts, an interventional cardiology training system. *Studies in health technology and informatics*, pages 59–65, 2000.

[30] Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Finding similar time series. In *European Symposium on Principles of Data Mining and Knowledge Discovery*, pages 88–100. Springer, 1997.

[31] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *KDD*, volume 98, pages 16–22, 1998.

[32] Sanchita Das, Mark S Rundell, Aashiq H Mirza, Maneesh R Pingle, Kristi Shigyo, Aura R Garrison, Jason Paragas, Scott K Smith, Victoria A Olson, Davise H Larone, et al. A multiplex pcr/ldr assay for the simultaneous identification of category a infectious pathogens: Agents of viral hemorrhagic fever and variola virus. volume 10. Public Library of Science, 2015.

[33] Paul Davidsson. Multi agent based simulation: beyond social simulation. In *Multi-Agent-Based Simulation*, pages 97–107. Springer, 2000.

[34] Christian Derksen, Cherif Branki, and Rainer Unland. Agent. gui: A multi-agent based simulation framework. In *Computer Science and Information Systems (FedCSIS), 2011 Federated Conference on*, pages 623–630. IEEE, 2011.

[35] Michael H Dickinson, Claire T Farley, Robert J Full, MAR Koehl, Rodger Kram, and Steven Lehman. How animals move: an integrative view. *Science*, 288(5463):100–106, 2000.

[36] Mark d'Inverno, David Kinny, Michael Luck, and Michael Wooldridge. A formal specification of dmars. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 155–176. Springer, 1997.

[37] Kwankamon Dittakan, Frans Coenen, and Rob Christley. Satellite image mining for census collection: a comparative study with respect to the ethiopian hinterland. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 260–274. Springer, 2013.

[38] Ajay Divakaran, Kudir Peker, Shih-Fu Chang, Rathnakumar Radhakrishnan, Lexing Xie, et al. Video mining: pattern discovery versus pattern recognition. In *Image Processing, 2004. ICIP'04. 2004 International Conference on*, volume 4, pages 2379–2382. IEEE, 2004.

[39] Shiloh L Dockstader and A Murat Tekalp. On the tracking of articulated and occluded video object motion. *Real-Time Imaging*, 7(5):415–432, 2001.

[40] Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information visualization*, 7(3-4):240–252, 2008.

[41] Bertrand Dumont and David RC Hill. Multi-agent simulation of group foraging in sheep: effects of spatial memory, conspecific attraction and plot size. volume 141, pages 201–215. Elsevier, 2001.

[42] Rajai El Dajani, Maryvonne Miquel, Marie-Claire Forlini, and Paul Rubel. Modeling of ventricular repolarisation time series by multi-layer perceptrons. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 152–155. Springer, 2001.

[43] Ahmed Elgammal, David Harwood, and Larry Davis. Non-parametric model for background subtraction. In *European conference on computer vision*, pages 751–767. Springer, 2000.

[44] Thomas P. Engel. Computer simulation techniques. *Journal of Clinical Monitoring and Computing*, 17(1):3–9, 2002.

[45] Joshua M Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up.* Brookings Institution Press, 1996.

[46] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. *Fast subsequence matching in time-series databases*, volume 23. ACM, 1994.

[47] Jerry G. Fox and Stephen K. Halladin. Effective application of simulation in the life cycle of a manufacturing cell project. In *Proceedings of the 23rd Conference on Winter Simulation*, WSC '91, pages 411–418, Washington, DC, USA, 1991. IEEE Computer Society.

[48] David M Gaba. The future vision of simulation in healthcare. *Simulation in Healthcare*, 2(2):126–135, 2007.

[49] David M Gaba and Abe DeAnda. A comprehensive anesthesia simulation environment: re-creating the operating room for research and training. *Anesthesiology*, 69(3):387–394, 1988.

[50] DR Gandham Sindhuja. Renuka devi sm 2015. a survey on detection and tracking of objects in video sequence. *International Journal of Engineering Research and General Science*, 3:418.

[51] Benoit Gaudou, Christophe Sibertin-Blanc, Olivier Therond, Frédéric Amblard, Yves Auda, Jean-Paul Arcangeli, Maud Balestrat, Marie-Hélène Charron-Moirez, Etienne Gondet, Yi Hong, et al. The maelia multi-agent platform for integrated analysis of interactions between agricultural land-use and low-water management strategies. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 85–100. Springer, 2013.

[52] Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, and Rajeev Motwani. Mining the stock market (extended abstract): which measure is best? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 487–496. ACM, 2000.

[53] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. In *International Workshop on Agent Theories, Architectures, and Languages*, pages 1–10. Springer, 1998.

[54] Luca Giancardo, Diego Sona, Huiping Huang, Sara Sannino, Francesca Managò, Diego Scheggia, Francesco Papaleo, and Vittorio Murino. Automatic visual tracking and social behaviour analysis with multiple mice. volume 8, page e74557. Public Library of Science, 2013.

[55] David W. Graehl. Insights into carrier control: A simulation of a power and free conveyor through an automotive paint shop. In *Proceedings of the 24th Conference on Winter Simulation*, WSC '92, pages 925–932, New York, NY, USA, 1992. ACM.

[56] Haisong Gu, Yoshiaki Shirai, and Minoru Asada. Mdl-based spatiotemporal segmentation from motion in a long image sequence. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 448–453. IEEE, 1994.

[57] Tarun Gupta and Subramaniam Arasakesari. Capacity planning in wood products industry using simulation. In *Proceedings of the 23rd conference on Winter simulation*, pages 435–440. IEEE Computer Society, 1991.

[58] Simon Hallé and Brahim Chaib-draa. A collaborative driving system based on multiagent modelling and simulations. *Transportation Research Part C: Emerging Technologies*, 13(4):320–345, 2005.

[59] Ismail Haritaoglu, David Harwood, and Larry S. Davis. W 4: Real-time surveillance of people and their activities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830, 2000.

[60] Dong-Chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. *IEEE transactions on Geoscience and Remote Sensing*, 28(4):509–512, 1990.

[61] MK Hegarty and MB Bloch. The use of simulators in intensive care training. *Current Anaesthesia & Critical Care*, 13(4):194–200, 2002.

[62] B Heisele and Christian Woehler. Motion-based recognition of pedestrians. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1325–1330. IEEE, 1998.

[63] Dirk Helbing. Agent-based modeling. In *Social self-organization*, pages 25–70. Springer, 2012.

[64] Islem Henane, Lamjed Ben Said, Sameh Hadouaj, and Nasr Ragged. Multi-agent based simulation of animal food selective behavior in a pastoral system. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 283–292. Springer, 2010.

[65] Colin M Henein and Tony White. Agent-based modelling of forces in crowds. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 173–184. Springer, 2004.

[66] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–43. IEEE, 2012.

[67] Petter Holme and Gourab Ghoshal. Dynamics of networking agents competing for high centrality and low degree. *Physical review letters*, 96(9):098701, 2006.

[68] Cars H Hommes. Heterogeneous agent models in economics and finance. *Handbook of computational economics*, 2:1109–1186, 2006.

[69] E.T. Hooperm. A systematic review of the harvest mice (genus reithrodontomys) of latin america. 1952.

[70] Kazuaki Iwata, Ma Onosato, Ka Teramoto, and S Osaki. A modelling and simulation architecture for virtual manufacturing systems. *CIRP Annals-Manufacturing Technology*, 44(1):399–402, 1995.

[71] Marco A Janssen and Elinor Ostrom. Empirically based, agent-based models. *Ecology and Society*, 11(2):37, 2006.

[72] Jacqueline Jarvis, Dennis Jarvis, and Ralph Rönnquist. *Holonic execution: a BDI approach*, volume 106. Springer Science & Business Media, 2008.

[73] Thomas Jefferson, Madhav Rangaswami, and Gregory Stoner. Simulation in the design of ground-based intrabay automation systems. In *Proceedings of the 28th conference on Winter simulation*, pages 1008–1013. IEEE Computer Society, 1996.

[74] Vadivelu Jeyabalan and Norman C. Otto. Simulation of material delivery systems with dolly trains. In *Proceedings of the 24th Conference on Winter Simulation*, WSC '92, pages 916–924, New York, NY, USA, 1992. ACM.

[75] Hueihan Jhuang, Estibaliz Garrote, Xinlin Yu, Vinita Khilnani, Tomaso Poggio, Andrew D Steele, and Thomas Serre. Automated home-cage behavioural phenotyping of mice. volume 1, page 68. Nature Publishing Group, 2010.

[76] Catholijn M Jonker and Jan Treur. Agent-based simulation of animal behaviour. *Applied Intelligence*, 15(2):83–115, 2001.

[77] Shanon X Ju, Michael J Black, and Yaser Yacoob. Cardboard people: A parameterized model of articulated image motion. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 38–44. IEEE, 1996.

[78] Leslie Pack Kaelbling. An architecture for intelligent reactive systems. In *Morgan Kaufmann, Proceedings of the 1986 Workshop: Reasoning about Actions and Plans, Editors: Georgeff, M, Lansky, A*, volume 30, pages 395–410, 1987.

[79] Yoshinari Kameda and Michihiko Minoh. A human motion estimation method using 3-successive video frames. In *International conference on virtual systems and multimedia*, pages 135–140, 1996.

[80] Franziska Klügl and Guido Rindsfüser. Large-scale agent-based pedestrian simulation. In *Multiagent System Technologies*, pages 145–156. Springer, 2007.

[81] Thomas M Krummel. Surgical simulation and virtual reality: the coming revolution. *Annals of surgery*, 228(5):635, 1998.

[82] Hans Kruuk. Scent marking by otters (lutra lutra): signaling the use of resources. *Behavioral Ecology*, 3(2):133–140, 1992.

[83] Feirouz Ksontini, René Mandiau, Zahia Guessoum, and Stéphane Espié. Affordance-based agent model for road traffic simulation. *Autonomous Agents and Multi-Agent Systems*, 29(5):821–849, 2015.

[84] Blake LeBaron et al. A builders guide to agent-based financial markets. *Quantitative Finance*, 1(2):254–261, 2001.

[85] Jun Wook Lee, Ok Hyun Paek, and Keun Ho Ryu. Temporal moving pattern mining for location-based service. *Journal of Systems and Software*, 73(3):481–490, 2004.

[86] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee. Group behavior from video: a data-driven approach to crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–118. Eurographics Association, 2007.

[87] Gerard S Letterie. How virtual reality may enhance training in obstetrics and gynecology. *American journal of obstetrics and gynecology*, 187(3):S37–S40, 2002.

[88] Geoffrey K Lighthall, Juliana Barr, Steven K Howard, Eran Gellar, Yasser Sowb, Edward Bertacini, and David Gaba. Use of a fully simulated intensive care unit environment for critical event management training for internal medicine residents. *Critical care medicine*, 31(10):2437–2443, 2003.

[89] Rake& Agrawal King-lp Lin and Harpreet S Sawhney Kyuseok Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceeding of the 21th International Conference on Very Large Data Bases*, pages 490–501. Citeseer, 1995.

[90] Eno Ludavicius and Ahad Ali. Modeling and simulation of automotive engine sub-assembly for production improvement. In *Proceedings of the International Conference on Industrial Engineering and Operations Management*, pages 2040–2049, 2014.

[91] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[92] Charles M Macal and Michael J North. Tutorial on agent-based modeling and simulation. In *Proceedings of the 37th conference on Winter simulation*, pages 2–15. Winter Simulation Conference, 2005.

[93] John MacCormick and Andrew Blake. Probabilistic exclusion and partitioned sampling for multiple object tracking. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[94] Heikki Mannila and Christopher Meek. Global partial orders from sequential data. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 161–168. ACM, 2000.

[95] Heikki Mannila and Hannu Toivonen. Discovering generalized episodes using minimal occurrences. In *KDD*, volume 96, pages 146–151, 1996.

[96] Anu Maria. Introduction to modeling and simulation. In *Proceedings of the 29th conference on Winter simulation*, pages 7–13. IEEE Computer Society, 1997.

[97] GJ Mason and KE Littin. The humaneness of rodent pest control. volume 12(1), pages 1–37. Journal of Animal Welfare. Animal Welfare, 2003.

[98] Bastiaan G Meerburg, Grant R Singleton, and Herwig Leirs. The year of the rat endstime to fight hunger! volume 65(4), pages 351–352. Journal of Pest Management Science. Wiley Online Library, 2009.

[99] Adriano Melo, Mairon Belchior, and Vasco Furtado. Analyzing police patrol routes by simulating the physical reorganization of agents. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 99–114. Springer, 2005.

[100] Harvey J Miller and Jiawei Han. *Geographic data mining and knowledge discovery.* CRC Press, 2009.

[101] Gerardo Miramontes-de Leon, Arturo Moreno-B, Claudia Sifuentes-Gallardo, Ernesto Garcia-Dominguez, Marisol Galv, et al. Analysis of avi files for mice behavior experiments in the morris water maze. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2011 IEEE*, pages 131–136. IEEE, 2011.

[102] Rokia Missaoui and Roman M Palenichka. Effective image and video mining: an overview of model-based approaches. In *Proceedings of the 6th international workshop on Multimedia data mining: mining integrated media and complex data*, pages 43–52. ACM, 2005.

[103] Thomas B Moeslund. *Introduction to video and image processing: Building real systems and applications.* Springer Science & Business Media, 2012.

[104] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. volume 81, pages 231–268. Elsevier, 2001.

[105] Brendan Tran Morris and Mohan Manubhai Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. volume 18, pages 1114–1127. IEEE, 2008.

[106] Ashkan Negahban and Jeffrey S Smith. Simulation for manufacturing system design and operation: Literature review and analysis. volume 33, pages 241–261. Elsevier, 2014.

[107] JV Neumann and AW Burks. Theory of self-reproduction automata. *Illinois: UP*, 1966.

[108] Lucas PJJ Noldus, Andrew J Spink, and Ruud AJ Tegelenbosch. Computerised video tracking, movement analysis and behaviour recognition in insects. *Computers and Electronics in Agriculture*, 35(2):201–227, 2002.

[109] Emma Jane Norling. Modelling human behaviour with bdi agents. 2009.

[110] Norbert Noury and Tareq Hadidi. Computer simulation of the activity of the elderly person living independently in a health smart home. *Computer methods and programs in biomedicine*, 108(3):1216–1228, 2012.

[111] Shay Ohayon, Ofer Avni, Adam L. Taylor, Pietro Perona, and S.E. Roian Egnor. Automated multi-day tracking of marked mice for the analysis of social behaviour. volume 219, pages 10 – 19. 2013.

[112] Shipra Ojha and Sachin Sakhare. Image processing techniques for object tracking in video surveillance-a survey. In *Pervasive Computing (ICPC), 2015 International Conference on*, pages 1–6. IEEE, 2015.

[113] Masaru Okaya and Tomoichi Takahashi. Bdi agent model based evacuation simulation. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1297–1298. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[114] Kenji Okuma, Ali Taleghani, Nando de Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. *Computer Vision-ECCV 2004*, pages 28–39, 2004.

[115] Tuncer I. Ören. Artificial intelligence in simulation. *Annals of Operations Research*, 53(1):287–319, 1994.

[116] Xiaoshan Pan, Charles S Han, Ken Dauber, and Kincho H Law. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. volume 22, pages 113–132. Springer, 2007.

[117] Himani S Parekh, Darshak G Thakore, and Udesang K Jaliya. A survey on object detection and tracking methods. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(2):2970–2979, 2014.

[118] Juan Pavón and Jorge Gómez-Sanz. Agent oriented software engineering with ingenias. In *Multi-Agent Systems and Applications III*, pages 394–403. Springer, 2003.

[119] Oliver P Pearson. Habits of harvest mice revealed by automatic photographic recorders. *Journal of Mammalogy*, 41(1):58–74, 1960.

[120] Massimo Piccardi. Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on*, volume 4, pages 3099–3104. IEEE, 2004.

[121] R Pittini, D Oepkes, K Macrury, R Reznick, J Beyene, and R Windrim. Teaching invasive perinatal procedures: assessment of a high fidelity simulator-based curriculum. *Ultrasound in Obstetrics and Gynecology*, 19(5):478–483, 2002.

[122] Joyce H Poole, Peter L Tyack, Angela S Stoeger-Horwath, and Stephanie Watwood. Animal behaviour: elephants are capable of vocal learning. *Nature*, 434(7032):455–456, 2005.

[123] Stephen C Pratt, David JT Sumpter, Eamonn B Mallon, and Nigel R Franks. An agent-based model of collective nest choice by the ant temnothorax albipennis. *Animal Behaviour*, 70(5):1023–1036, 2005.

[124] Till Quack, Vittorio Ferrari, and Luc Van Gool. Video mining with frequent itemset configurations. In *International Conference on Image and Video Retrieval*, pages 360–369. Springer, 2006.

[125] Marco Raberto, Silvano Cincotti, Sergio M Focardi, and Michele Marchesi. Agent-based simulation of a financial market. *Physica A: Statistical Mechanics and its Applications*, 299(1):319–327, 2001.

[126] Gordon C Rausser and SR Johnson. On the limitations of simulation in model evaluation and. *Simulation Gaming*, 6(2):1685–1699, 1975.

[127] Martin Reznek, Phillip Harter, and Thomas Krummel. Virtual reality and simulation: training the future emergency physician. *Academic Emergency Medicine*, 9(1):78–87, 2002.

[128] Fritz Rohrlich. Computer simulation in the physical sciences. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, pages 507–518. JSTOR, 1990.

[129] Dustin R Rubenstein and Keith A Hobson. From birds to butterflies: animal movement patterns and stable isotopes. *Trends in Ecology & Evolution*, 19(5):256–263, 2004.

[130] Imen Saffar, Arnaud Doniec, Jacques Boonaert, and Stéphane Lecoeuche. Multi-agent simulation design driven by real observations and clustering techniques. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 555–560. IEEE, 2011.

[131] Richard M Satava. Emerging medical applications of virtual reality: A surgeon's perspective. *Artificial Intelligence in Medicine*, 6(4):281–288, 1994.

[132] Richard M Satava. Surgical education and surgical simulation. *World Journal of Surgery*, 25(11):1484–1489, 2001.

[133] JW Schmidt. Introduction to systems analysis, modeling and simulation. In *Proceedings of the 17th conference on Winter simulation*, pages 3–14. ACM, 1985.

[134] Howard A Schwid. A flight simulator for general anesthesia training. *Computers and biomedical research*, 20(1):64–75, 1987.

[135] DV Chandra Shekar. Generic algorithm for tracking genaric objects in a motion picture. *Journal of Global Research in Computer Science*, 3(2):38–42, 2012.

[136] Weiming Shen, Qi Hao, Hyun Joong Yoon, and Douglas H Norrie. Applications of agent-based systems in intelligent manufacturing: An updated review. *Advanced engineering INFORMATICS*, 20(4):415–431, 2006.

[137] Ameya Shendarkar, Karthik Vasudevan, Seungho Lee, and Young-Jun Son. Crowd simulation for emergency response using bdi agent based on virtual reality. In *Proceedings of the 38th conference on Winter simulation*, pages 545–553. Winter Simulation Conference, 2006.

[138] Martin Shubik. Simulation of the industry and the firm. *The American Economic Review*, 50(5):908–919, 1960.

[139] Peer-Olaf Siebers and Uwe Aickelin. Introduction to multi-agent simulation. 2008.

[140] Revathi.R Sindhu.K. A survey on moving object tracking in video frame sequences. *International Journal of Advanced Computional Engineering and Networking*, 2(8), 2014.

[141] Grant R Singleton, Steven Belmain, Peter R Brown, Ken Aplin, and Nyo Me Htwe. Impacts of rodent outbreaks on food security in asia. *Wildlife Research*, 37(5):355–359, 2010.

[142] Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis.* BF Skinner Foundation, 1990.

[143] Stephen D Small, Richard C Wuerz, Robert Simon, Nathan Shapiro, Alasdair Conn, and Gary Setnik. Demonstration of high-fidelity simulation team training for emergency medicine. *Academic Emergency Medicine*, 6(4):312–323, 1999.

[144] Maartje Spoelstra and Elizabeth Sklar. Agent-based simulation of group learning. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 69–83. Springer, 2007.

[145] Ramzi Suleiman, Klaus G Troitzsch, and Nigel Gilbert. *Tools and techniques for social science simulation.* Springer Science & Business Media, 2012.

[146] Shipeng Sun and Steven M Manson. Simple agents, complex emergent city: Agent-based modeling of intraurban migration. In *Computational Approaches for Urban Environments*, pages 123–147. Springer, 2015.

[147] Soemon Takakuwa. Efficient module-based modeling for a large-scale as/rs-agv system. In *Proceedings of the 28th conference on Winter simulation*, pages 1141–1148. IEEE Computer Society, 1996.

[148] Lu Tan, Mingyuan Hu, and Hui Lin. Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency. *Information Sciences*, 295:53–66, 2015.

[149] Kardi Teknomo and Gloria Purugganan Gerilla. Pedestrian static trajectory analysis of a hypermarket. In *Proceedings of the Eastern Asia Society for Transportation Studies*, volume 2009, pages 220–220. Eastern Asia Society for Transportation Studies, 2009.

[150] Dallas Treit and M Fundytus. Thigmotaxis as a test for anxiolytic activity in rats. volume 31, pages 959–962. Elsevier, 1988.

[151] Maksim Tsvetovat and Maciej Łatek. Dynamics of agent organizations: application to modeling irregular warfare. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 60–70. Springer, 2008.

[152] Muhammad Tufail, Frans Coenen, Jane Hurst, and Tintin Mu. Multi agent based simulation using movement patterns mined from video data. In *Research and Development in Intelligent Systems XXXII*, pages 275–287. Springer, 2015.

[153] Cor J Veenman, Marcel JT Reinders, and Eric Backer. Resolving motion correspondence for densely moving points. volume 23, pages 54–72. IEEE, 2001.

[154] Neil J Vickers. Mechanisms of animal navigation in odor plumes. *The Biological Bulletin*, 198(2):203–212, 2000.

[155] V Vijayakumar and R Nedunchezhian. Mining best-n frequent patterns in a video sequence. *International Journal on Computer Science and Engineering*, 3(11):3525, 2011.

[156] V Vijayakumar and R Nedunchezhian. A study on video data mining. *International journal of multimedia information retrieval*, 1(3):153–172, 2012.

[157] Jasna Vuk, Michael E Anders, Cynthia C Mercado, Robert L Kennedy, Jessie Casella, and Susan C Steelman. Impact of simulation training on self-efficacy of outpatient health care providers to use electronic health records. volume 84, pages 423–429. Elsevier, 2015.

[158] LA Walker, JS Chaplow, NR Llewellyn, MG Pereira, ED Potter, AW Sainsbury, and RF Shore. Anticoagulant rodenticides in predatory birds 2011: a predatory bird monitoring scheme (pbms) report. NERC/Centre for Ecology & Hydrology, 2013.

[159] Li Wang and Dong-Chen He. Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905–910, 1990.

[160] Yaoping Wang, Cheekong Chui, Honglip Lim, Yiyu Cai, and Koonhou Mak. Real-time interactive simulator for percutaneous coronary revascularization procedures. *Computer Aided Surgery*, 3(5):211–227, 1998.

[161] Shinichi Watanabe, Masako Izawa, Akiko Kato, Yan Ropert-Coudert, and Yasuhiko Naito. A new technique for monitoring the detailed behaviour of terrestrial animals: a case study with the domestic cat. *Applied Animal Behaviour Science*, 94(1):117–131, 2005.

[162] Jones Wayne, Vavra Pavel, Fournier Alexis, et al. *OECD-FAO Agricultural Outlook 2009-2018*. OECD/FAO, 2009.

[163] Aharon Weissbrod, Alexander Shapiro, Genadiy Vasserman, Liat Edry, Molly Dayan, Assif Yitzhaky, Libi Hertzberg, Ofer Feinerman, and Tali Kimchi. Automated long-term tracking and social behavioural phenotyping of animal colonies within a semi-natural environment. volume 4. Nature Publishing Group, 2013.

[164] Pablo Werlang, Michel Q Fagundes, Diana F Adamatti, Karina S Machado, Andrea von Groll, Pedro EA da Silva, and Adriano V Werhli. Multi-agent-based simulation of mycobacterium tuberculosis growth. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 131–142. Springer, 2013.

[165] Tham Wickenberg and Paul Davidsson. On multi agent based simulation of software development processes. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 171–180. Springer, 2002.

[166] Shawn R Wolfe, Maarten Sierhuis, and Peter A Jarvis. To bdi, or not to bdi: design choices in an agent-based traffic flow management simulation. In *Proceedings of the 2008 Spring simulation multiconference*, pages 63–70. Society for Computer Simulation International, 2008.

[167] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[168] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.

[169] Masanobu Yamamoto and Kazutada Koshikawa. Human motion analysis based on a robot arm model. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 664–665. IEEE, 1991.

[170] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. volume 38, page 13. Acm, 2006.

[171] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. volume 26, pages 1531–1536. IEEE, 2004.

Appendix A

Location descriptors for three categories of scenarios

In the context of the Simple category scenario simulations considered for evaluation purposes with respect to the work presented in this thesis the 45 different potential descriptors are listed Table A.1. Note that, with respect to Table A.1, that pattern numbers have been included simply to facilitate discussion, not because they are required by the pattern mining process or the eventual operation of the desired MABS. The identified set of location ground type labels in this case is $L = \{n, o, w, -\}$ as discussed previously in Chapter 3.

Table A.1: The complete set of location descriptors for the environment considered with respect to the Simple Scenario category

Num.	**Descriptor**	**Num.**	**Descriptor**	**Num.**	**Descriptor**
1	- - - -nn-nn	2	- - -nn-nn-	3	-nn-nn- - -
4	nn-nn- - - -	5	nn-nn-ow-	6	-nn-nn-wo
7	nnonnw- - -	8	nnowoowoo	9	nn-ow-ow-
10	- - -nnwnno	11	nnwnnowoo	12	-nn-wo-wo
13	noonww- - -	14	nooooooo0	15	- - -nwwnoo
16	nwwnooooo	17	onnoowoow	18	onnwnn- - -
19	oonoooooo	20	oonwwn- - -	21	ooonoonww
22	oooonwwn	23	ooooooonoo	24	oooooooon
25	ooooooooo	26	ooooooowww	27	ooowww- - -
28	oowonnwnn	29	oowoowonn	30	oowoowoow
31	ow-nn-nn-	32	ow-ow-nn-	33	ow-ow-ow-
34	- - -wnnonn	35	wnnonnoow	36	-wo-nn-nn
37	woonnonnw	38	woowoonno	39	woowoowoo
40	-wo-wo-nn	41	-wo-wo-wo	42	- - -wwnoon
43	wwnoonooo	44	- - -wwwooo	45	wwwoooooo

In the context of the Complex category scenario simulations considered for evaluation purposes with respect to the work described in this thesis the 119 different potential relative descriptors as listed in Table A.2. The identified set of location ground types labels in this case is $L = \{b, g, i, n, o, w, t, -\}$ as discussed previously in Chapter 3.

Table A.2: The complete set of location descriptors for the environment considered with respect to the Complex Scenario category

Num	**Descriptor**	**Num**	**Descriptor**	**Num**	**Descriptor**
1	- - -wwwwooL	41	nnwnnw- - -M	81	tgobwobwoR
2	bbwttgbbwT1	42	nnwnnwnnwM	82	wbbgttwbbT1
3	bbwttgbbwT2	43	nownownwwL	83	wbbgttwbbT2
4	bwibwibwoM	44	nownww- - -L	84	- - -wiiwiiM
5	- - -bwibwiM	45	nwbnwb- - -M	85	wiiwiiwooM
6	bwibwotgoM	46	nwbnwbnwbM	86	wiiwoogooM
7	bwnbwnbwnM	47	ogtowbnwbM	87	wnnwnn- - -M
8	bwnbwn- - -M	48	ogtowbowbL	88	wnnwnnwnnM
9	bwobwnbwnM	49	onnonnwnnR	89	-wo-nn-nnL
10	bwobwobwoR	50	onnwnn- - -R	90	wonwonwwnR
11	bwobwobwwR	51	oogoownnwM	91	wonwwn- - -R
12	bwobwotgoR	52	oogoownowL	92	woogoowooM
13	bwobww- - -R	53	ooonnnnnnM	93	woogoowooR
14	bwotgobwoM	54	ooonnonnoL	94	woonnnnnnL
15	bwotgobwoR	55	oooonnonnR	95	woownnwnnM
16	bwwbwobwoR	56	oooooonnnM	96	woowonwonR
17	- - -bwwbwoR	57	oooooonnoL	97	woowoogooR
18	goowoownnM	58	ooooooonnR	98	woowoonnnL
19	goowoowonR	59	oooooooooL	99	woowoowooL
20	- - -iiiiiiM	60	oooooooooM	100	-wo-wo-nnL
21	iiiiiioooM	61	oooooooooR	101	-wo-wo-woL
22	iiiooooooM	62	oownnnnnnR	102	- - -wwbowbL
23	- - -iiwiiwM	63	oownnwnnwM	103	wwbowbowbL
24	iiwiiwoowM	64	oownownowL	104	ww-ow-ow-R
25	iiwoowoogM	65	oowoogoowL	105	- - -ww-ow-R
26	- - -iwbiwbM	66	oowoogoowM	106	- - - -ww-woL
27	iwbiwbowbM	67	oowoownnnR	107	- - -wwwoooL
28	iwbowbogtM	68	oowoowoogL	108	wwwooooooL
29	-nn-nn- - -L	69	oowoowoowR	109	wwwooooooR
30	nnnnnn- - -L	70	owbnwbnwbM	110	- - -wwwoooR
31	-nn-nn-nnL	71	owbogtowbL	111	- - -wwwoowL
32	nnnnnn- - -M	72	owbogtowbM	112	wwwoowoowL
33	nnnnnnnnnL	73	owbowbogtL	113	wwwoowoowR

Continued on next page

Table A.2 – *Continued from previous page*

Num	Descriptor	Num	Descriptor	Num	Descriptor
34	nnnnnnnnnM	74	owbowbowbL	114	- - -wwwoowR
35	nnnnnnnnnR	75	owbowbwwbL	115	-ww-wo-woL
36	nnnnnn- - -R	76	owbwwb- - -L	116	- - -wwwwooL
37	nn-nn-nn-R	77	ow-nn-nn-R	117	- - -wwwwooR
38	nn-nn- - –R	78	ow-ow-nn-R	118	wwwwoowooL
39	nnonnonnwL	79	ow-ow-ow-R	119	wwwwoowooR
40	nnonnw- - -L	80	tgobwobwnM		

In the context of the Two rodent category scenario simulations considered for evaluation purposes with respect to the work described in this thesis the 125 different potential relative descriptors as listed in Table A.3. The identified set of location ground types labels in this case is $L = \{w, b, s, n, o, -\}$ as discussed previously in Chapter 3.

Table A.3: The complete set of location descriptors for the environment considered with respect to the Two Rodents Scenario category

Num	Descriptor	Num	Descriptor	Num	Descriptor
1	bbbooooooA	43	oooonnonnA	85	ssoooooooA
2	bbooboobo A	44	oooooboobA	86	ssossossoA
3	bboooooooA	45	oooooboooA	87	ssossowooA
4	boobbooboA	46	ooooonoonA	88	ssowoowooA
5	booboobboA	47	oooooobbbA	89	ss - ow - ow - A
6	booboobooA	48	oooooobboA	90	sssooooooA
7	booboooooA	49	oooooobooA	91	ss - ss - ow - A
8	booooooooA	50	oooooonnnA	92	ss - ss - ss - A
9	nnnnnnnnnA	51	oooooonnoA	93	- ss - ss - ssA
10	nnnnnnoooA	52	oooooonooA	94	- - - ssssssA
11	nnnooooooA	53	ooooooobbA	95	ssssssoooA
12	nnonnonnoA	54	oooooooboA	96	- ss - ss - woA
13	nnonnooooA	55	ooooooonnA	97	- ss - wo - woA
14	nnoooooooA	56	oooooooobA	98	- - - sswssoA
15	noonoonooA	57	oooooooonA	99	sswssooooA
16	noonoooooA	58	oooooooooA	100	- - - swwsooA
17	nooooooooA	59	oooooooosA	101	swwsoooooA
18	obbooboobA	60	oooooosooA	102	woossossoA
19	obbooooooA	61	oooooowwwA	103	woowoossoA
20	oboobboobA	62	ooooosoosA	104	woowoowooA
21	obooboobbA	63	ooosoosooA	105	woowoowwwA
22	oboobooboA	64	ooowww - - - A	106	woowww - - - A

Continued on next page

Table A.3 – *Continued from previous page*

Num	**Descriptor**	**Num**	**Descriptor**	**Num**	**Descriptor**
23	oboobooooA	65	oosoooooooA	107	- wo - ss - ssA
24	obooooooooA	66	oosoosoooA	108	- wo - wo - ssA
25	onnonnonnA	67	oosoosoosA	109	- wo - wo - woA
26	onnonnoooA	68	oowoowoowA	110	- wo - wo - wwA
27	onnoooooooA	69	oowoowossA	111	- wo - ww - - - A
28	ooboobooobA	70	oowoowwwwA	112	- - - wssossA
29	ooboobooooA	71	oowossossA	113	wssossoooA
30	oobooooooA	72	oowwww - - - A	114	- - - ww - ow - A
31	oonoonoonA	73	ossoooooooA	115	ww - ow - ow - A
32	oonoonoooA	74	ossoowoowA	116	- - - wwsoosA
33	oonoooooooA	75	ossossoowA	117	wwsoosoooA
34	ooobbboooA	76	ossossossA	118	- - - - ww - woA
35	ooobbooooA	77	ow - ow - ow - A	119	- - - wwwoooA
36	ooobooobooA	78	ow - ow - ss - A	120	wwwooooooA
37	ooobooooooA	79	ow - ow - ww - A	121	- - - wwwoowA
38	ooonnnnnnA	80	ow - ss - ss - A	122	wwwoowoowA
39	ooonnonnoA	81	ow - ww - - - - A	123	- ww - wo - woA
40	ooonoonooA	82	sooooooooA	124	- - - wwwwooA
41	oooobbboooA	83	soosooooooA	125	wwwwoowooA
42	ooooboooboA	84	soosoosooA		

Appendix B

Tables for Two Rodents Scenarios Video 1 and 2, Using Absolute Mechanism

Table B.1: Average absolute difference of two agents from video and simulation data using absolute mechanism without states for Two Rodents Scenarios Video-1

Block No	**Video** 1	**Simulation**	**Absolute difference**
0	735	726	10
1	453	485	33
2	435	434	2
3	923	1105	182
4	755	708	47
5	520	491	30
6	1253	1323	70
7	580	536	44
8	993	1102	109
9	925	956	31
10	278	307	29
11	308	282	26
12	578	431	147
13	635	636	1
14	643	650	8
15	408	419	12
16	515	623	108
17	193	239	46
18	913	1005	92
19	1223	1140	83

Continued on next page

Table B.1 – *Continued from previous page*

Block No	**Video** 1	**Simulation**	**Absolute difference**
20	693	614	79
21	868	751	117
22	435	439	4
23	170	197	27
24	133	148	16
25	580	626	46
26	825	793	33
27	510	461	49
28	843	769	74
29	1065	1029	37
30	250	275	25
31	160	159	1
32	130	129	1
33	448	427	21
34	399	446	47
35	558	536	22
36	857	843	14
37	503	489	14
38	213	195	18
39	605	531	75
40	2616	2471	146
41	380	356	25
42	1143	1169	26
43	545	626	81
44	425	486	61
45	378	428	51
46	423	447	24
47	590	610	20
48	430	395	36
Total absolute difference			0.08

Table B.2: Average absolute difference of two agents from video and simulation data using absolute mechanism without states for Two Rodents Scenarios Video-2

Block No	**Video** 2	**Simulation**	**Absolute difference**
0	819	939	120
1	334	379	46

Continued on next page

Table B.2 – *Continued from previous page*

Block No	**Video 2**	**Simulation**	**Absolute difference**
2	410	507	97
3	570	686	116
4	388	401	13
5	325	332	7
6	475	299	176
7	768	727	41
8	175	183	8
9	145	134	11
10	155	139	17
11	690	691	1
12	543	539	4
13	445	386	59
14	388	359	29
15	413	373	40
16	388	357	31
17	155	133	23
18	948	986	39
19	1443	1395	48
20	288	242	46
21	963	986	23
22	885	819	66
23	135	135	1
24	70	78	8
25	178	194	17
26	355	481	126
27	713	577	136
28	1065	865	201
29	680	661	19
30	328	327	1
31	278	248	30
32	233	176	57
33	763	800	38
34	945	1087	142
35	873	698	175
36	720	606	114
37	425	548	123
38	215	227	12

Continued on next page

Table B.2 – *Continued from previous page*

Block No	**Video** 2	**Simulation**	**Absolute difference**
39	993	967	26
40	2458	2574	117
41	768	987	220
42	1348	998	350
43	520	486	34
44	488	519	31
45	1045	1167	122
46	648	679	31
47	1385	1512	127
48	710	864	154
Total absolute difference			0.11

Table B.3: Average absolute difference of two agents from video and simulation data using absolute mechanism with states for Two Rodents Scenarios Video-1

Block No	**Video** 1	**Simulation**	**Absolute difference**
0	735	831	96
1	453	477	24
2	435	400	35
3	923	986	63
4	755	742	13
5	520	581	61
6	1253	1808	555
7	580	496	85
8	993	957	36
9	925	801	125
10	278	262	16
11	308	265	43
12	578	598	20
13	635	823	188
14	643	600	43
15	408	404	4
16	515	446	70
17	193	202	9
18	913	1040	127
19	1223	1172	51
20	693	853	161

Continued on next page

Table B.3 – *Continued from previous page*

Block No	**Video** 1	**Simulation**	**Absolute difference**
21	868	783	85
22	435	550	115
23	170	169	1
24	133	123	10
25	580	618	38
26	825	875	50
27	510	513	3
28	843	795	48
29	1065	1055	11
30	250	248	3
31	160	127	34
32	130	121	9
33	448	374	74
34	399	437	38
35	558	552	6
36	857	791	66
37	503	498	5
38	213	172	41
39	605	539	67
40	2616	2024	592
41	380	282	98
42	1143	1206	63
43	545	626	81
44	425	518	93
45	378	323	55
46	423	380	43
47	590	516	75
48	430	483	53
Total absolute difference			0.12

Table B.4: Average absolute difference of two agents from video and simulation data using absolute mechanism with states for Two Rodents Scenarios Video-2

Block No	**Video** 2	**Simulation**	**Absolute difference**
0	819	735	84
1	334	308	26
2	410	513	103

Continued on next page

Table B.4 – *Continued from previous page*

Block No	**Video 2**	**Simulation**	**Absolute difference**
3	570	682	112
4	388	446	58
5	325	282	44
6	475	422	53
7	768	721	47
8	175	177	2
9	145	176	31
10	155	161	6
11	690	766	76
12	543	365	178
13	445	407	39
14	388	486	98
15	413	465	53
16	388	402	15
17	155	138	18
18	948	845	103
19	1443	1310	133
20	288	295	8
21	963	1020	57
22	885	1109	224
23	135	175	40
24	70	86	16
25	178	201	23
26	355	252	103
27	713	1089	377
28	1065	1129	64
29	680	796	116
30	328	250	78
31	278	236	42
32	233	244	11
33	763	942	179
34	945	979	34
35	873	835	38
36	720	621	99
37	425	383	42
38	215	175	41
39	993	680	313

Continued on next page

Table B.4 – *Continued from previous page*

Block No	**Video 2**	**Simulation**	**Absolute difference**
40	2458	2491	34
41	768	903	135
42	1348	1331	17
43	520	428	92
44	488	465	23
45	1045	780	266
46	648	598	50
47	1385	1448	63
48	710	704	7
Total absolute difference			0.13

Appendix C

Tables for Two Rodent Scenario Videos 1 and 2, using Relative Mechanism

TABLE C.1: Recorded results for Video 1 and Simulation using relative mechanism without states

Location descriptor	**Video** 1	**Simulation**	**Absolute difference**
ow-ow-ww-A	17	32	15
os-os-ow-A	24	75	51
ooobooboo A	29	55	26
oooooobbA	30	74	44
- - -wwsoooA	32	55	23
onoooooooA	33	82	49
nooooooooA	35	90	55
oosoowoowA	37	77	40
oonooooooA	38	66	28
oosoosoowA	39	73	34
ooooobboA	40	84	44
oowoowwwwA	43	73	30
ow-ww- - - - A	44	9	35
ooboooooA	49	70	21
ooboobooobA	50	72	22
wwwwoowooA	54	82	28
booooowwwA	54	52	2
wwsoooooA	55	74	19
ow-os-os-A	56	119	63
obbooooooA	59	84	25

Continued on next page

Table C.1 – *Continued from previous page*

Location descriptor	**Video** 1	**Simulation**	**Absolute difference**
booboooooA	60	72	12
oboooowwwA	60	59	1
ow-ow-os-A	61	40	21
sswoooooooA	61	80	19
oooooboobA	61	65	4
soosoowooA	64	87	23
ooooooboooA	66	167	101
oonooowwwA	66	67	1
boooooooooA	68	74	6
oowwww- - -A	68	36	32
woowoosooA	69	52	17
oooooooonA	71	79	8
-ww-wo-woA	73	25	48
os-ow-ow-A	174	148	26
- - - wwwwooA	274	247	27
ooboooowwwA	175	149	26
woowww- - -A	276	233	43
ooboobooooA	276	266	10
swwoooooooA	277	260	17
ooooooooobA	279	380	101
oboobooooA	279	273	6
-wo-wo-soA	182	142	40
wssoooooooA	185	190	5
oowoosoosA	185	190	5
onooooowwwA	187	179	8
-wo-so-soA	287	241	46
-so-so-woA	190	174	16
noooooowwwA	90	71	19
soowoowooA	90	47	43
- - - wssoooA	93	71	22
- - - - ww-woA	93	18	75
bbooooooooA	94	92	2
- - - ww-ow-A	95	12	83
woowoowwwA	96	36	60
oooooboooA	102	86	16
boobooboooA	102	84	18
woosoosooA	106	59	47

Continued on next page

Table C.1 – *Continued from previous page*

Location descriptor	**Video** 1	**Simulation**	**Absolute difference**
oooboooooA	119	128	9
ooooooonooA	122	185	63
-wo-wo-wwA	122	39	83
oowoowoosA	123	89	34
-so-wo-woA	127	64	63
ww-ow-ow-A	128	21	107
- - - sswoooA	130	75	55
- - - swwoooA	133	58	75
- - - wwwoowA	137	129	8
obooboob oA	138	171	33
oooooooboA	140	111	29
wwwoowoowA	141	147	6
ooooooonoA	158	199	41
-wo-ww- - -A	64	90	26
ow-ow-ow-A	266	250	16
ooooboob oA	175	132	43
ooobboooo A	182	142	40
ooooooowwwA	182	272	90
- - - wwwoooA	223	217	6
oooobbooo A	140	135	5
oonoonoooA	246	206	40
ooonoonooA	250	264	14
-wo-wo-woA	272	251	21
wwwooooooA	330	421	91
ooowww - - - A	331	278	53
oooooonoonA	354	251	103
noonooooo A	364	209	155
oowoowoowA	381	482	101
woowoowooA	411	367	44
onoonooooA	413	392	21
oooonoonoA	610	589	21
oooooooooA	1387	1854	467
Total absolute difference			0.26

Table C.2: Recorded results for Video 2 and Simulation using relative mechanism without states

Location descriptor	**Video 2**	**Simulation**	**Absolute difference**
- - - wwwoowA	13	31	18
oooboooooA	14	94	80
oboboooooA	18	83	65
ow-ow-os-A	21	23	2
boooooooooA	28	139	111
oooooonooA	28	59	31
wwsoooooooA	28	86	58
sswoooooooA	32	64	32
obooooooooA	33	43	10
ww-ow-ow-A	36	21	15
oooooboobA	37	72	35
ooboobooobA	39	52	13
oosoosoowA	39	61	22
woowoosooA	41	84	43
ow-ww - - - - A	42	14	28
- - - ww-ow-A	42	15	27
swwoooooooA	44	41	3
boobooboooA	46	42	4
oooooboboA	47	90	43
oowoosoosA	47	77	30
oosoowoowA	48	59	11
ooboobooooA	48	73	25
oooboobooA	48	58	10
- - - - ww-woA	50	21	29
ooooboboooA	54	87	33
os-os-ow-A	55	53	2
oobooooooA	56	80	24
wssoooooooA	58	72	14
ow-ow-ww-A	59	47	12
booboooooA	60	50	10
- - - sswoooA	63	73	10
oowoowoosA	65	95	30
oooooobooA	65	133	68
-so-so-woA	69	28	41
-wo-wo-soA	69	38	31
booooowwwA	72	83	11

Continued on next page

Table C.2 – *Continued from previous page*

Location descriptor	**Video 2**	**Simulation**	**Absolute difference**
- - - wwwwooA	73	54	19
soowoowooA	74	83	9
- - - wssoooA	75	80	5
oowoowwwwA	76	51	25
oooooooonA	77	123	46
oobooowwwA	79	76	3
obooboobоA	80	113	33
noooooooоA	81	61	20
oboooowwwA	81	72	9
- - - swwoooA	81	47	34
wwwwoowooA	87	63	24
- - - wwsoooA	91	61	30
soosoowooA	91	54	37
ooonoooooA	92	85	7
oooooooobA	93	150	57
woosoosooA	98	92	6
oownowoowA	101	86	15
wwwoowoowA	101	52	49
woowww - - - A	102	39	63
oooooooboA	103	178	75
oowoownowA	105	81	24
oooooboooA	106	80	26
oowwww - - - A	50	31	19
os-ow-ow-A	45	37	8
-so-wo-woA	60	31	29
-ww-wo-woA	19	21	2
obooboooоA	60	85	25
-wo-so-soA	27	37	10
oowoowoowA	133	180	47
oonooooooA	133	152	19
-wo-ww - - - A	26	10	16
-wo-wo-wwA	40	60	20
ow-os-os-A	44	55	11
ooooboobоA	145	220	75
wwwooooooA	150	190	40
nowoowoowA	80	74	6
ooboboooоA	159	229	70

Continued on next page

Table C.2 – *Continued from previous page*

Location descriptor	**Video** 2	**Simulation**	**Absolute difference**
woowoowwwA	80	155	75
ooooonoooA	169	183	14
- - - wwwoooA	175	282	107
ow-ow-ow-A	239	219	20
nnoooooooA	260	223	37
oooooonnoA	293	202	91
ooooooonnA	338	225	113
onnooooooA	363	167	196
woowoowooA	368	421	53
ooonnooooA	459	381	78
-wo-wo-woA	256	225	31
ooowww - - - A	327	365	38
oooonnoooA	345	365	20
oooooowwwA	580	518	62
oooooooooA	3381	3258	123
Total absolute difference			0.25

Table C.3: Recorded results for Video 1 and Simulation using relative mechanism with states

Location descriptor	**State**	**Video** 1	**Simulation**	**Absolute difference**
bboooooooA	CloseBy	33	52	19
bboooooooA	Ignore	52	21	31
bboooooooA	Meeting	9	5	4
booboobooA	CloseBy	17	14	3
booboobooA	Ignore	64	50	14
booboobooA	Meeting	21	13	8
booboooooA	CloseBy	15	14	1
booboooooA	Ignore	39	42	3
booboooooA	Meeting	6	4	2
booooooooA	CloseBy	18	28	10
booooooooA	Ignore	24	45	21
booooooooA	Meeting	26	8	18
booooowwwA	CloseBy	2	5	3
booooowwwA	Ignore	49	35	14
booooowwwA	Meeting	3	3	0
noonoooooA	CloseBy	80	36	44

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video** 1	**Simulation**	**Absolute difference**
noonooooA	Ignore	200	150	50
noonooooA	Meeting	84	17	67
nooooooooA	CloseBy	13	11	2
nooooooooA	Ignore	21	53	32
nooooooooA	Meeting	1	3	2
nooooowwwA	CloseBy	8	3	5
nooooowwwA	Follow	1	0	1
nooooowwwA	Ignore	70	15	55
nooooowwwA	Meeting	11	3	8
obbooooooA	CloseBy	15	20	5
obbooooooA	Ignore	43	19	24
obbooooooA	Meeting	1	2	1
obooboobоA	CloseBy	20	15	5
obooboobоA	Ignore	108	29	79
obooboobоA	Meeting	10	7	3
oboobooooA	CloseBy	18	9	9
oboobooooA	Ignore	55	30	25
oboobooooA	Meeting	6	2	4
obooooowwwA	CloseBy	4	14	10
obooooowwwA	Ignore	53	60	7
obooooowwwA	Meeting	3	1	2
onoonooooA	CloseBy	42	74	32
onoonooooA	Ignore	255	250	5
onoonooooA	Meeting	116	140	24
onooooooooA	CloseBy	11	23	12
onooooooooA	Ignore	21	46	25
onooooooooA	Meeting	1	7	6
onooooowwwA	CloseBy	22	4	18
onooooowwwA	Ignore	49	32	17
onooooowwwA	Meeting	16	0	16
ooboobooobA	CloseBy	14	6	8
ooboobooobA	Ignore	26	53	27
ooboobooobA	Meeting	10	14	4
ooboobooooA	CloseBy	8	8	0
ooboobooooA	Ignore	39	43	4
ooboobooooA	Meeting	29	6	23
oobooooooA	CloseBy	17	19	2

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video** 1	**Simulation**	**Absolute difference**
ooboooooA	Ignore	31	40	9
ooboooooA	Meeting	1	3	2
ooboooowwwA	CloseBy	8	3	5
ooboooowwwA	Ignore	44	33	11
ooboooowwwA	Meeting	23	1	22
oonoonoooA	CloseBy	32	55	23
oonoonoooA	Ignore	176	180	4
oonoonoooA	Meeting	38	17	21
oonooooooA	CloseBy	13	34	21
oonooooooA	Ignore	21	26	5
oonooooooA	Meeting	4	7	3
oonooowwwA	CloseBy	10	11	1
oonooowwwA	Ignore	53	57	4
oonooowwwA	Meeting	3	1	2
ooobbooooA	CloseBy	44	22	22
ooobbooooA	Ignore	116	47	69
ooobbooooA	Meeting	22	5	17
oooboobooA	CloseBy	2	11	9
oooboobooA	Ignore	21	45	24
oooboobooA	Meeting	6	3	3
ooobooooooA	CloseBy	26	33	7
ooobooooooA	Ignore	65	42	23
ooobooooooA	Meeting	28	2	26
ooonoonooA	CloseBy	61	40	21
ooonoonooA	Ignore	126	101	25
ooonoonooA	Meeting	63	22	41
oooobboooA	CloseBy	45	34	11
oooobboooA	Ignore	174	83	91
oooobboooA	Meeting	17	12	5
ooooboob oA	CloseBy	16	21	5
oooobooboA	Ignore	50	80	30
oooobooboA	Meeting	38	4	34
oooonoonoA	CloseBy	91	89	2
oooonoonoA	Ignore	307	320	13
oooonoonoA	Meeting	110	115	5
ooooobboobA	CloseBy	9	19	10
ooooobboobA	Ignore	48	51	3

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video** 1	**Simulation**	**Absolute difference**
oooooboobA	Meeting	4	7	3
ooooobooooA	CloseBy	15	16	1
ooooobooooA	Ignore	50	46	4
ooooobooooA	Meeting	37	5	32
ooooonoonA	CloseBy	70	37	33
ooooonoonA	Ignore	226	290	64
ooooonoonA	Meeting	58	20	38
oooooobboA	CloseBy	7	12	5
oooooobboA	Ignore	29	36	7
oooooobboA	Meeting	4	3	1
oooooobooA	CloseBy	18	40	22
oooooobooA	Ignore	41	87	46
oooooobooA	Meeting	7	4	3
oooooonooA	CloseBy	23	44	21
oooooonooA	Ignore	83	68	15
oooooonooA	Meeting	16	12	4
ooooooobbA	CloseBy	4	3	1
ooooooobbA	Ignore	22	39	17
ooooooobbA	Meeting	4	4	0
oooooooboA	CloseBy	43	20	23
oooooooboA	Ignore	80	55	25
oooooooboA	Meeting	17	16	1
ooooooonoA	CloseBy	28	35	7
ooooooonoA	Ignore	95	79	16
ooooooonoA	Meeting	35	13	22
oooooooobA	CloseBy	11	27	16
oooooooobA	Ignore	61	80	19
oooooooobA	Meeting	7	9	2
oooooooonA	CloseBy	16	41	25
oooooooonA	Ignore	49	60	11
oooooooonA	Meeting	6	16	10
oooooooooA	CloseBy	406	787	381
oooooooooA	Follow	2	5	3
oooooooooA	Ignore	905	1022	117
oooooooooA	Meeting	74	225	151
oooooowwwA	CloseBy	38	41	3
oooooowwwA	Ignore	134	144	10

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video** 1	**Simulation**	**Absolute difference**
ooooowwwA	Meeting	10	7	3
ooowww - - - A	CloseBy	36	30	6
ooowww - - - A	Ignore	269	222	47
ooowww - - - A	Meeting	26	8	18
oosoosoowA	CloseBy	19	18	1
oosoosoowA	Ignore	20	36	16
oosoowoowA	CloseBy	22	16	6
oosoowoowA	Ignore	14	47	33
oosoowoowA	Meeting	1	5	4
oowoosoosA	CloseBy	14	18	4
oowoosoosA	Ignore	65	22	43
oowoosoosA	Meeting	6	10	4
oowoowoosA	CloseBy	23	15	8
oowoowoosA	Ignore	93	50	43
oowoowoosA	Meeting	7	3	4
oowoowoowA	CloseBy	62	44	18
oowoowoowA	Ignore	72	80	8
oowoowoowA	Meeting	47	19	28
oowoowwwwA	CloseBy	15	5	10
oowoowwwwA	Ignore	22	31	9
oowoowwwwA	Meeting	6	4	2
oowwww - - - A	CloseBy	9	5	4
oowwww - - - A	Ignore	59	19	40
os - os - ow - A	CloseBy	4	4	0
os - os - ow - A	Ignore	20	35	15
os - ow - ow - A	CloseBy	26	7	19
os - ow - ow - A	Ignore	48	32	16
ow - os - os - A	CloseBy	13	20	7
ow - os - os - A	Ignore	41	47	6
ow - os - os - A	Meeting	2	13	11
ow - ow - os - A	CloseBy	9	5	4
ow - ow - os - A	Ignore	48	28	20
ow - ow - os - A	Meeting	4	4	0
ow - ow - ow - A	CloseBy	17	23	6
ow - ow - ow - A	Ignore	137	126	11
ow - ow - ow - A	Meeting	12	3	9
ow - ow - ww - A	CloseBy	5	4	1

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video 1**	**Simulation**	**Absolute difference**
ow - ow - ww - A	Ignore	11	17	6
ow - ow - ww - A	Meeting	1	3	2
ow - ww - - - - A	CloseBy	7	0	7
ow - ww - - - - A	Follow	1	0	1
ow - ww - - - - A	Ignore	35	13	22
ow - ww - - - - A	Meeting	1	0	1
soosoowooA	CloseBy	14	12	2
soosoowooA	Ignore	39	33	6
soosoowooA	Meeting	11	3	8
soowoowooA	CloseBy	25	13	12
soowoowooA	Ignore	33	41	8
soowoowooA	Meeting	32	4	28
- so - so - woA	CloseBy	22	7	15
- so - so - woA	Ignore	54	50	4
- so - so - woA	Meeting	14	0	14
- so - wo - woA	CloseBy	32	6	26
- so - wo - woA	Ignore	71	44	27
- so - wo - woA	Meeting	24	1	23
- - - sswoooA	CloseBy	20	26	6
- - - sswoooA	Ignore	92	25	67
- - - sswoooA	Meeting	18	25	7
sswoooooooA	CloseBy	19	16	3
sswoooooooA	Ignore	39	51	12
sswoooooooA	Meeting	3	12	9
- - - swwoooA	CloseBy	25	14	11
- - - swwoooA	Ignore	98	49	49
- - - swwoooA	Meeting	10	3	7
swwoooooooA	CloseBy	35	10	25
swwoooooooA	Follow	1	0	1
swwoooooooA	Ignore	36	46	10
swwoooooooA	Meeting	5	15	10
woosoosooA	CloseBy	34	13	21
woosoosooA	Follow	1	0	1
woosoosooA	Ignore	49	41	8
woosoosooA	Meeting	22	3	19
woowoosooA	CloseBy	17	13	4
woowoosooA	Follow	1	0	1

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video 1**	**Simulation**	**Absolute difference**
woowoosooA	Ignore	41	26	15
woowoosooA	Meeting	10	6	4
woowoowooA	CloseBy	61	56	5
woowoowooA	Ignore	319	242	77
woowoowooA	Meeting	31	23	8
woowoowwwA	CloseBy	10	4	6
woowoowwwA	Ignore	58	42	16
woowoowwwA	Meeting	28	3	25
woowww - - - A	CloseBy	7	1	6
woowww - - - A	Ignore	53	32	21
woowww - - - A	Meeting	16	0	16
- wo - so - soA	CloseBy	9	9	0
- wo - so - soA	Ignore	34	34	0
- wo - so - soA	Meeting	44	5	39
- wo - wo - soA	CloseBy	14	14	0
- wo - wo - soA	Ignore	49	40	9
- wo - wo - soA	Meeting	19	2	17
- wo - wo - woA	CloseBy	42	35	7
- wo - wo - woA	Ignore	193	149	44
- wo - wo - woA	Meeting	37	29	8
- wo - wo - wwA	CloseBy	1	2	1
- wo - wo - wwA	Ignore	102	20	82
- wo - wo - wwA	Meeting	19	0	19
- wo - ww - - - A	CloseBy	10	0	10
- wo - ww - - - A	Ignore	136	12	124
- wo - ww - - - A	Meeting	18	0	18
- - - wssoooA	CloseBy	5	28	23
- - - wssoooA	Ignore	88	44	44
wssoooooooA	CloseBy	15	19	4
wssoooooooA	Ignore	69	33	36
wssoooooooA	Meeting	1	4	3
- - - ww - ow - A	CloseBy	6	2	4
- - - ww - ow - A	Ignore	80	14	66
- - - ww - ow - A	Meeting	9	0	9
ww - ow - ow - A	CloseBy	23	1	22
ww - ow - ow - A	Ignore	82	17	65
ww - ow - ow - A	Meeting	23	0	23

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video 1**	**Simulation**	**Absolute difference**
- - - wwsoooA	CloseBy	6	9	3
- - - wwsoooA	Ignore	25	18	7
- - - wwsoooA	Meeting	1	1	0
wwsooooooA	CloseBy	14	15	1
wwsooooooA	Follow	1	0	1
wwsooooooA	Ignore	39	42	3
wwsooooooA	Meeting	1	2	1
- - - - ww - woA	CloseBy	22	6	16
- - - - ww - woA	Ignore	60	5	55
- - - - ww - woA	Meeting	11	5	6
- - - wwwoooA	CloseBy	30	34	4
- - - wwwoooA	Ignore	166	179	13
- - - wwwoooA	Meeting	27	18	9
wwwooooooA	CloseBy	77	81	4
wwwooooooA	Follow	1	0	1
wwwooooooA	Ignore	220	309	89
wwwooooooA	Meeting	32	23	9
- - - wwwoowA	CloseBy	22	3	19
- - - wwwoowA	Ignore	99	43	56
- - - wwwoowA	Meeting	16	0	16
wwwoowoowA	CloseBy	23	5	18
wwwoowoowA	Ignore	87	50	37
wwwoowoowA	Meeting	31	1	30
- ww - wo - woA	CloseBy	13	7	6
- ww - wo - woA	Ignore	36	34	2
- ww - wo - woA	Meeting	24	5	19
- - - wwwwooA	CloseBy	7	8	1
- - - wwwwooA	Follow	1	0	1
- - - wwwwooA	Ignore	48	29	19
- - - wwwwooA	Meeting	18	6	12
wwwwoowooA	CloseBy	24	14	10
wwwwoowooA	Ignore	25	59	34
wwwwoowooA	Meeting	5	10	5
noonoooooA	Follow	0	1	1
oonoooooooA	Follow	0	1	1
ooowww - - - A	Follow	0	1	1
oosoosoowA	Meeting	0	5	5

Continued on next page

Table C.3 – *Continued from previous page*

Location descriptor	**State**	**Video** 1	**Simulation**	**Absolute difference**
oowoosoosA	Follow	0	1	1
oowoowoowA	Follow	0	1	1
oowwww - - - A	Meeting	0	1	1
os - os - ow - A	Meeting	0	1	1
os - ow - ow - A	Meeting	0	1	1
soowoowooA	Follow	0	1	1
woowoowooA	Follow	0	1	1
- - - wssoooA	Meeting	0	4	4
- - - wwwoooA	Follow	0	1	1
- ww - wo - woA	Follow	0	1	1
Total absolute difference				0.42

Table C.4: Recorded results for Video 2 and Simulation using relative mechanism with states

Location descriptor	**State**	**Video** 2	**Simulation**	**Absolute difference**
boobooboo A	CloseBy	2	23	21
booboobooA	Ignore	43	38	5
booboobooA	Meeting	1	5	4
boobooooo A	CloseBy	10	17	7
booboooooA	Ignore	45	70	25
booboooooA	Meeting	5	0	5
boooooooo A	CloseBy	5	62	57
booooooooA	Ignore	23	66	43
booooowwwA	CloseBy	8	15	7
booooowwwA	Ignore	57	76	19
booooowwwA	Meeting	7	2	5
nnoooooooA	CloseBy	63	76	13
nnoooooooA	Follow	1	0	1
nnoooooooA	Ignore	164	153	11
nnoooooooA	Meeting	32	27	5
nooooooooA	CloseBy	27	37	10
nooooooooA	Ignore	42	74	32
nooooooooA	Meeting	12	17	5
nowoowoowA	CloseBy	5	13	8
nowoowoowA	Ignore	141	41	100
nowoowoowA	Meeting	8	18	10

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video** 2	**Simulation**	**Absolute difference**
oboboooooA	CloseBy	4	35	31
oboboooooA	Follow	1	0	1
oboboooooA	Ignore	12	62	50
oboboooooA	Meeting	1	8	7
oboobooboA	CloseBy	34	36	2
oboobooboA	Ignore	44	101	57
oboobooboA	Meeting	2	9	7
oboobooooA	CloseBy	32	17	15
oboobooooA	Ignore	78	87	9
oboobooooA	Meeting	11	4	7
oboooooooA	CloseBy	8	35	27
oboooooooA	Ignore	19	51	32
oboooooooA	Meeting	6	3	3
oboooowwwA	CloseBy	10	10	0
oboooowwwA	Ignore	58	63	5
oboooowwwA	Meeting	13	9	4
onnooooooA	CloseBy	51	66	15
onnooooooA	Ignore	240	166	74
onnooooooA	Meeting	72	23	49
oobobooooA	CloseBy	48	45	3
oobobooooA	Ignore	111	124	13
oobooboobA	CloseBy	4	12	8
oobooboobA	Follow	1	0	1
oobooboobA	Ignore	30	65	35
oobooboobA	Meeting	4	4	0
oobooboooA	CloseBy	5	22	17
oobooboooA	Ignore	38	70	32
oobooboooA	Meeting	5	3	2
oobooooooA	CloseBy	19	28	9
oobooooooA	Ignore	33	80	47
oobooooooA	Meeting	4	6	2
oobooowwwA	CloseBy	6	13	7
oobooowwwA	Follow	1	0	1
oobooowwwA	Ignore	67	76	9
oobooowwwA	Meeting	5	5	0
oonooooooA	CloseBy	11	58	47
oonooooooA	Ignore	48	104	56

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video** 2	**Simulation**	**Absolute difference**
oonooooooA	Meeting	74	20	54
ooobooboo A	CloseBy	14	26	12
oooboobooA	Ignore	29	47	18
oooboobooA	Meeting	5	5	0
ooobooooo A	CloseBy	7	56	49
oooboooooA	Ignore	7	52	45
ooonnooooA	CloseBy	77	127	50
ooonnooooA	Follow	1	0	1
ooonnooooA	Ignore	324	329	5
ooonnooooA	Meeting	57	59	2
ooonooooooA	CloseBy	18	22	4
ooonooooooA	Ignore	55	55	0
ooonooooooA	Meeting	19	11	8
oooobobooA	CloseBy	9	35	26
oooobobooA	Ignore	43	65	22
oooobobooA	Meeting	2	19	17
ooooboobo A	CloseBy	42	34	8
oooobooboA	Ignore	101	140	39
oooobooboA	Meeting	2	6	4
ooooonnoooA	CloseBy	102	163	61
ooooonnoooA	Ignore	421	319	102
ooooonnoooA	Meeting	109	57	52
ooooobobo A	CloseBy	1	35	34
oooooboboA	Ignore	46	70	24
ooooooboobA	CloseBy	3	27	24
ooooooboobA	Ignore	30	84	54
ooooooboobA	Meeting	4	7	3
ooooooboooA	CloseBy	22	25	3
ooooooboooA	Ignore	79	99	20
ooooooboooA	Meeting	5	5	0
ooooonoooA	CloseBy	58	73	15
ooooonoooA	Ignore	90	108	18
ooooonoooA	Meeting	21	20	1
ooooooobooA	CloseBy	20	84	64
ooooooobooA	Ignore	35	115	80
ooooooobooA	Meeting	10	21	11
ooooooonnoA	CloseBy	68	67	1

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video** 2	**Simulation**	**Absolute difference**
oooooonnoA	Follow	1	1	0
oooooonnoA	Ignore	161	166	5
oooooonnoA	Meeting	63	35	28
oooooonooA	CloseBy	5	21	16
oooooonooA	Ignore	21	59	38
oooooonooA	Meeting	2	7	5
oooooooboA	CloseBy	19	83	64
oooooooboA	Ignore	82	132	50
oooooooboA	Meeting	2	23	21
ooooooonnA	CloseBy	40	60	20
ooooooonnA	Ignore	222	140	82
ooooooonnA	Meeting	76	25	51
oooooooobA	CloseBy	16	68	52
oooooooobA	Ignore	150	188	38
oooooooobA	Meeting	9	35	26
oooooooonA	CloseBy	20	55	35
oooooooonA	Ignore	54	132	78
oooooooonA	Meeting	3	24	21
oooooooooA	CloseBy	1055	975	80
oooooooooA	Follow	1	10	9
oooooooooA	Ignore	679	870	191
oooooooooA	Meeting	365	395	30
oooooowwwA	CloseBy	82	118	36
oooooowwwA	Ignore	604	396	208
oooooowwwA	Meeting	73	35	38
ooowww—A	CloseBy	59	80	21
ooowww—A	Ignore	513	330	183
ooowww—A	Meeting	55	21	34
oosoosoowA	CloseBy	15	26	11
oosoosoowA	Ignore	14	84	70
oosoosoowA	Meeting	10	7	3
oosoowoowA	CloseBy	16	21	5
oosoowoowA	Ignore	26	82	56
oosoowoowA	Meeting	6	9	3
oownowoowA	CloseBy	5	18	13
oownowoowA	Ignore	87	64	23
oownowoowA	Meeting	9	14	5

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video 2**	**Simulation**	**Absolute difference**
oowoosoosA	CloseBy	13	16	3
oowoosoosA	Ignore	28	71	43
oowoosoosA	Meeting	6	7	1
oowoownowA	CloseBy	8	27	19
oowoownowA	Ignore	94	80	14
oowoownowA	Meeting	3	6	3
oowoowoosA	CloseBy	5	18	13
oowoowoosA	Ignore	59	77	18
oowoowoosA	Meeting	1	17	16
oowoowoowA	CloseBy	12	43	31
oowoowoowA	Ignore	95	185	90
oowoowoowA	Meeting	26	32	6
oowoowwwwA	CloseBy	4	6	2
oowoowwwwA	Ignore	62	82	20
oowoowwwwA	Meeting	10	6	4
oowwww—A	CloseBy	2	3	1
oowwww—A	Ignore	104	31	73
oowwww—A	Meeting	1	3	2
os-os-ow-A	CloseBy	10	6	4
os-os-ow-A	Ignore	39	54	15
os-os-ow-A	Meeting	6	0	6
os-ow-ow-A	CloseBy	2	9	7
os-ow-ow-A	Ignore	109	40	69
os-ow-ow-A	Meeting	1	8	7
ow-os-os-A	CloseBy	72	31	41
ow-os-os-A	Ignore	53	89	36
ow-os-os-A	Meeting	19	4	15
ow-ow-os-A	CloseBy	9	9	0
ow-ow-os-A	Ignore	12	37	25
ow-ow-ow-A	CloseBy	19	51	32
ow-ow-ow-A	Ignore	218	234	16
ow-ow-ow-A	Meeting	2	17	15
ow-ow-ww-A	Ignore	57	42	15
ow-ow-ww-A	Meeting	2	2	0
ow-ww—-A	Ignore	42	20	22
soosoowooA	CloseBy	4	25	21
soosoowooA	Ignore	56	69	13

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video 2**	**Simulation**	**Absolute difference**
soosoowooA	Meeting	31	1	30
soowoowooA	CloseBy	17	19	2
soowoowooA	Ignore	40	90	50
soowoowooA	Meeting	17	5	12
-so-so-woA	CloseBy	9	5	4
-so-so-woA	Ignore	59	52	7
-so-so-woA	Meeting	1	2	1
-so-wo-woA	CloseBy	21	5	16
-so-wo-woA	Ignore	95	76	19
-so-wo-woA	Meeting	3	1	2
—sswoooA	CloseBy	1	16	15
—sswoooA	Ignore	62	85	23
sswoooooooA	CloseBy	7	38	31
sswoooooooA	Ignore	25	67	42
—swwoooA	CloseBy	2	15	13
—swwoooA	Ignore	79	38	41
swwoooooooA	CloseBy	1	31	30
swwoooooooA	Ignore	43	51	8
woosoosooA	CloseBy	25	18	7
woosoosooA	Ignore	55	78	23
woosoosooA	Meeting	18	5	13
woowoosooA	CloseBy	10	15	5
woowoosooA	Ignore	29	58	29
woowoosooA	Meeting	2	9	7
woowoowooA	CloseBy	57	92	35
woowoowooA	Ignore	276	255	21
woowoowooA	Meeting	35	26	9
woowoowwwA	CloseBy	15	8	7
woowoowwwA	Ignore	85	76	9
woowoowwwA	Meeting	59	0	59
woowww—A	CloseBy	14	4	10
woowww—A	Ignore	56	31	25
woowww—A	Meeting	32	1	31
-wo-so-soA	CloseBy	30	6	24
-wo-so-soA	Ignore	87	29	58
-wo-so-soA	Meeting	10	1	9
-wo-wo-soA	CloseBy	1	15	14

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video 2**	**Simulation**	**Absolute difference**
-wo-wo-soA	Ignore	68	41	27
-wo-wo-woA	CloseBy	75	38	37
-wo-wo-woA	Follow	1	0	1
-wo-wo-woA	Ignore	423	224	199
-wo-wo-woA	Meeting	67	0	67
-wo-wo-wwA	CloseBy	4	4	0
-wo-wo-wwA	Ignore	117	50	67
-wo-wo-wwA	Meeting	21	0	21
-wo-ww—A	CloseBy	12	1	11
-wo-ww—A	Ignore	107	30	77
-wo-ww—A	Meeting	17	0	17
—wssoooA	Ignore	75	99	24
wssooooooA	CloseBy	3	39	36
wssooooooA	Ignore	55	75	20
—ww-ow-A	Ignore	42	15	27
ww-ow-ow-A	Ignore	36	33	3
—wwsoooA	Ignore	91	27	64
wwsooooooA	CloseBy	2	25	23
wwsooooooA	Ignore	26	73	47
—-ww-woA	Ignore	49	35	14
—-ww-woA	Meeting	1	0	1
—wwwoooA	CloseBy	120	109	11
—wwwoooA	Ignore	467	366	101
wwwooooooA	CloseBy	18	179	161
wwwooooooA	Ignore	531	436	95
wwwooooooA	Meeting	1	62	61
—wwwoowA	Ignore	9	41	32
—wwwoowA	Meeting	4	2	2
wwwoowoowA	CloseBy	3	11	8
wwwoowoowA	Ignore	88	87	1
wwwoowoowA	Meeting	10	1	9
-ww-wo-woA	CloseBy	7	12	5
-ww-wo-woA	Ignore	97	58	39
-ww-wo-woA	Meeting	15	1	14
—wwwwooA	CloseBy	2	13	11
—wwwwooA	Ignore	57	39	18
—wwwwooA	Meeting	14	3	11

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video 2**	**Simulation**	**Absolute difference**
wwwwoowooA	CloseBy	6	22	16
wwwwoowooA	Ignore	69	90	21
wwwwoowooA	Meeting	12	6	6
boobooboo A	Follow	0	1	1
booooooooA	Meeting	0	19	19
nooooooooA	Follow	0	1	1
ooboboooo A	Meeting	0	22	22
ooobooooo A	Follow	0	2	2
ooobooooo A	Meeting	0	9	9
ooonooooo A	Follow	0	2	2
oooonnoooA	Follow	0	1	1
oooooboboA	Meeting	0	8	8
ooooonoooA	Follow	0	1	1
oooooonooA	Follow	0	1	1
oooooooonnA	Follow	0	1	1
ow-ow-os-A	Meeting	0	6	6
ow-ow-ow-A	Follow	0	1	1
ow-ow-ww-A	CloseBy	0	9	9
ow-ww—-A	CloseBy	0	1	1
—sswoooA	Meeting	0	4	4
sswoooooo A	Follow	0	1	1
sswoooooo A	Meeting	0	8	8
—swwoooA	Meeting	0	3	3
swwooooooA	Follow	0	1	1
swwooooooA	Meeting	0	8	8
woowoowooA	Follow	0	3	3
-wo-wo-soA	Meeting	0	1	1
—wssoooA	CloseBy	0	14	14
—wssoooA	Meeting	0	3	3
wssooooooA	Meeting	0	13	13
—ww-ow-A	CloseBy	0	2	2
ww-ow-ow-A	CloseBy	0	8	8
—wwsoooA	CloseBy	0	14	14
—wwsoooA	Meeting	0	7	7
wwsooooooA	Meeting	0	14	14
—wwwoooA	Meeting	0	23	23
—wwwoowA	CloseBy	0	3	3
Total absolute				

Continued on next page

Table C.4 – *Continued from previous page*

Location descriptor	**State**	**Video 2**	**Simulation**	**Absolute difference**
difference				0.45

Appendix D

Supporting Letter From Domain Expert

UNIVERSITY OF LIVERPOOL | INSTITUTE OF INTEGRATIVE BIOLOGY

Professor Jane Hurst BSc PhD
William Prescott Chair of Animal Science

Mammalian Behaviour & Evolution Group
Institute of Integrative Biology
University of Liverpool
Leahurst Campus
Neston, CH64 7TE, UK

Athena SWAN Silver Award

T +44 151 794 6100
E jane.hurst@liv.ac.uk
W http://www.liv.ac.uk/mbe

To whom it may concern

RE: The Extraction and Usage of Patterns from Video Data to Support Multi-Agent Based Simulation by Muhammad Tufail

This is to provide some additional commentary concerning the relevance of the work undertaken by Muhammad Tufail in the above PhD thesis for studies of mammalian behaviour and especially rodent behaviour. I am the William Prescott Professor of Animal Science in the Institute of Integrative Biology at the University's Leahurst Campus, where I direct the Mammalian Behaviour & Evolution research group and facilities. I was Muhammad's third PhD supervisor, with a particular remit to provide domain knowledge in support of this programme of studies (what I understand is referred to as a "domain expert" in computer science circles).

Simulation tools, of the kind proposed and realised in this thesis, have wide ranging potential benefits in the context of predicting the behaviour of rodent pest species. Such simulation tools have the potential for considerable benefit with respect to the development of effective rodent control strategies, all of which rely on our ability to predict specific aspects of rodent behaviour. This, in turn will have significant benefits in terms of food security, human and livestock health, the economic impact of infrastructure damage, and the potential for reduced ecological and other damage to non-target species. To the best of my knowledge, the availability of such simulation tools is more or less non-existent. I can think of only one example, and that required the information needed for the simulation to operate to be entered by hand. The idea of automating this process, "knowledge acquisition", is therefore a good one.

The "proof of concept work" reported in the thesis demonstrates that computational agents can indeed be generated using information extracted from video, and that this information can indeed be used to define and predict the behaviour of rodents, in complex environments, providing the potential to inform and shape rodent pest control strategies. The work presented in the thesis thus provides an excellent foundation for further work directed at the simulation of rodent behaviour in the context of pest control, but could also be extended to the wider context of mammalian behaviour more generally.

Yours sincerely

Professor Jane Hurst