



A Context-Aware Framework for Intersection Collision Avoidance

by

Flora Dilys Salim

Bachelor of Computing (Hons)

DISSERTATION

Submitted in Fulfilment of the Requirements for the Degree of

Doctor of Philosophy

Caulfield School of Information Technology

Faculty of Information Technology

Monash University

Australia

August 2008

© Copyright
by
Flora Dilys Salim
2008

DECLARATION

In accordance with Monash University Doctorate Regulation 17 / Doctor of Philosophy and Master of Philosophy (MPhil) regulations the following declarations are made:

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Flora Dilys Salim

Date: __/__/__

ACKNOWLEDGMENTS

I would like to take this opportunity to thank and acknowledge the people who have supported me throughout my PhD candidature. This thesis would not have been possible without their great support.

My deepest gratitude is bestowed to my husband, Rudy Hartono, who continually stands by me, encourages me not to give up, and cheer me up in the most special way. I thank my parents, Arifin Salim and Indahwati, my brother, Kenny, and my sister, Felicia for the continuous love and sustenance.

Special thanks are due to my great supervisors, Dr. Shonali Krishnaswamy, Prof. Bala Srinivasan, Dr. Seng Wai Loke, and Dr. Andry Rakotonirainy. I acknowledge the support, insights, ideas, advices, and continual encouragement from my four supervisors, to whom I am deeply grateful. Their outstanding support has made me a much better researcher I am today.

I thank Shonali for her continual confidence in me. Being one of my previous Honours supervisors, she has assured me of my potential as a researcher and encouraged me to take this PhD degree. I am often astonished with her great enthusiasm and passion for research and her incredible eyes for details. Her

advices about the nitty gritty of research, such as targeting the right publications, presenting at conferences, applying awards and grants, writing a CV, and much more have helped me to shape my thoughts and career direction. I thank Srimi for his availability to join the supervision board in the third year of my candidature. I am always amazed with his speed in reading my publications and thesis. With his immense experience and expertise, he has a great capability to paint the big picture. This is particularly helpful for my research that covers multidisciplinary areas. He has helped me to see the end goal of my thesis and encouraged me to finish well. I thank Seng for being there whenever I need feedback. I am so grateful for his dedication and availability for my research, knowing that he needs to travel a great deal of distance to meet his students in Monash and he has many other professional commitments to attend to. He has been a great example for me in his commitment to research. I thank Andry that he is only an email away. His expertise in road safety is particularly helpful in my research. His thoughts and ideas always challenge a different perspective that I have overlooked in my research and thesis.

I would like to express my appreciation and thanks to Ps. Yudah and Anna Soetopo for their words of affirmation, support, and prayer. You are both my wonderful encouragers. I am very grateful to Evo for her continual friendship and her help with the reference list. I am so glad and thankful to have my brothers and sisters from my cell group, Multicultural Student Fellowship/Blast, and Gardiner International Church of Christ as they have been such a caring, thoughtful and supportive community. I would like to thank Angel, Iven, Tince, Frida, Lucy, Hermin, Jeff, Edwin, San, K Sam, Geary, Nith, Selina, Susan, Eelaine, Chang Wei, Nana, Nini, Sari, Winaya, Phyllis, Fanny, Cecil, Jerry, Jan, Albert, Lydia, Ferryman, Adit, Kaming, Shandy, Olivia, Vivi, Lana, Hin, Brenda, Eric, Diana, Vicky, Tania, Yonk, Jenny Tantonio, and so many others for coloring my life throughout my challenging PhD years. I wish I can mention all of their names

one by one in this thesis but it will be a never ending list. I would like to mention the special care from Ron Lancashire and Ps. Pieter Petrusma, who always ask me “How are you? How is your thesis going?” I would also acknowledge Selina Xiao for her great help with the reference list. My gratitude is also due to my grandparents, Ibu, Cie, Pak Is, K Ike, Mas Yono, K Diana, Anthony, K Tommy, and K Inge for their love and support.

I am glad to have nice friends with whom I spent my PhD years with: John Page, Evi Syukur, Suan Khai Chong, Donny Kurniawan, Wanita Sherchan, Ruwini Kodikara, Philip Chen, Mohamed Gaber, Pari Delir Haghighi, Prem Jayaraman, Brett Gillick, Kate Lazarenko, Grace Xie, and many others. We have been on the same boat. I wish them all the best for their research and future career. I thank Licheng Cai for doing a minor thesis project as part of my research.

I thank the staff of Caulfield School of Information Technology for their support. I especially thank Dr. Maria Indrawan for her assistance and friendship. I also thank her for offering me teaching, tutoring, and supervising opportunities in the faculty. I acknowledge the support from Michelle Ketchen, Allison Mitchell, Aleisha Matthews, Akamon Kunkangkopun, Dianna Sussman, Duke Fonias, See Ngieng, Rafig Tjahjadi, Rob Gray, Dr. Chris Ling, Dr. Phu Dung Le, Dr. Campbell Wilson, A/Prof Arkady Zaslavsky, Dr. Simon Cuce, and also Julie Simon from Monash College.

I gratefully acknowledge the financial support throughout my candidature from Australian Postgraduate Awards.

Last but not least, I thank my GOD, the Lord Jesus Christ, for giving me the knowledge, wisdom, strength, and opportunity to achieve this academic level, which is truly quite beyond my aspirations and dreams.

DEDICATION

For my parents, *papa, mama*, and for my husband, *Rudy*
and for my *Lord Jesus Christ*
To whom I owe all that I am

LIST OF PUBLICATIONS

This thesis includes eight original peer reviewed publications, six published in international conferences, one to be published as a book chapter, and one journal article under review after changes based on the reviewers' feedback were made.

Book Chapter

1. Salim, F. D., Loke, S. W., Rakotonirainy, A. and Krishnaswamy, S., "U & I Aware (Ubiquitous Intersection Awareness): A framework for intersection safety", accepted for publication in November 2006 as a book chapter in *Handbook on Mobile and Ubiquitous Computing: Innovations and Perspectives*, to be published by American Scientific Publishers.

Journal

2. Salim, F. D., Loke, S. W., Rakotonirainy, A., Krishnaswamy, S., "Context-Awareness, Agents, and Data Mining for Efficient, Autonomous and Safe Transportation Systems: a Survey", *IEEE Transactions on Intelligent Transportation Systems*, IEEE Computer Society Press, Under Review.

International Conferences

3. Salim, F. D., (2008), "A generic and real-time collision warning and avoidance system in a ubiquitous intersection environment", *Proceedings of*

the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008): Google PhD Forum, 17-21 March, Hong Kong, China, pp. 37-40.

4. Salim, F. D., Cai, L., Indrawan, M. and Loke, S. W., (2008), "Road intersections as pervasive computing environments: towards a multiagent real-time collision warning system", *In Proceedings of the 1st IEEE Workshop on Agent Technologies for Pervasive Communities*, in conjunction with the Sixth IEEE International Conference on Pervasive Computing and Communications (Percom '08), 17-21 March, Hong Kong, China, IEEE Computer Society Press, pp. 621-626.
5. Salim, F. D., Loke, S. W., Rakotonirainy, A., Srinivasan, B. and Krishnaswamy, S., (2007), "Collision pattern modeling and real-time collision detection at road intersections", *Proceedings of The 10th International IEEE Conference on Intelligent Transportation Systems*, 30 September - 3 October, Seattle, Washington, USA, IEEE Intelligent Transportation Systems Society, pp. 161-166.
6. Salim, F. D., Loke, S. W., Rakotonirainy, A. and Krishnaswamy, S., (2007), "Simulated intersection environment and learning of collision and traffic data in the U&I Aware framework", *In Proceedings of The 4th International Conference on Ubiquitous Intelligence and Computing (UIC-07)*, 11-13 July, Hong Kong, China, Springer Berlin/Heidelberg, pp. 153-162.
7. Salim, F. D., Loke, S. W., Rakotonirainy, A. and Krishnaswamy, S., (2007), "U&I Aware: a framework using data mining and collision detection to increase awareness for intersection users", *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, in conjunction with AINA-2007, 21-23 May, Niagara Falls, Canada, IEEE Computer Society Press, pp. 530-535.

8. Salim, F.D., Krishnaswamy, S., Loke, S. W. and Rakotonirainy, A., (2005), “Context-aware ubiquitous data mining based agent model for intersection safety”, *Proceedings of the 2005 IFIP Conference on Embedded and Ubiquitous Computing Workshops (EUCW 2005)*, in conjunction with EUC 2005, 6-9 December, Nagasaki, Japan, Lecture Notes in Computer Science, Springer-Verlag, pp. 61-70.

ABSTRACT

The number of intersection accidents around the world has reached a plateau and has not decreased in spite of the innovation and improvement in road and vehicle safety technologies. The key challenge in enhancing intersection safety is to identify vehicles that have a high potential to be involved in a collision as early as possible and take preventative action thereof. Thus, there is a clear need for an intersection collision warning and avoidance system that is able to warn drivers of an impending potential collision.

Today's vehicles and on-road infrastructures are equipped with a large number of sophisticated sensory devices. These sensory devices are capable of monitoring and providing data pertaining to vehicle status, real-time traffic conditions, traffic incidents, and road crashes. The wealth of data available through these sensors provides a new opportunity for intersection safety. By analysing this sensor data, there is a potential to determine contextual knowledge about situations that can lead to crashes in particular intersections. Such knowledge can have a significant positive impact on the key issue of improving intersection safety. However, along with the opportunity come several challenges. While technology has advanced to provide important data, we still do not have adequate mechanisms to capture,

integrate, and analyse this information. Furthermore, current research has not addressed the key issue of how to usefully leverage contextual knowledge obtained through such an analysis.

In this thesis, we propose and develop a novel intersection safety framework that we term the U&I Aware (Ubiquitous Awareness Intersection) Framework. This framework addresses the need to analyse sensor data to extract important contextual knowledge about crashes at the intersection. We propose and develop mechanisms to use this knowledge in early identification of vehicles that have a high likelihood of colliding.

Through the use of contextual knowledge, we show that we can significantly improve on collision detection algorithms that typically compute collision points and Time-To-Collision (TTC) for all possible vehicle pairs in an intersection. We also show that we maintain high accuracy in identifying vehicles that have a potential to collide. Thus, our experimental evaluation demonstrates the clear advantage of the U&I Aware Framework in improving the speed and accuracy of identifying vehicles that are likely to collide at an intersection over conventional collision detection algorithms that compute all possible vehicle pairs in an intersection.

CONTENTS

Declaration	iii
Acknowledgments	iv
Dedication	vii
List of Publications	viii
Abstract	xi
Contents	xiii
List of Figures	xvi
List of Tables	xviii

Chapter 1

Introduction	1
1.1. Intelligent Transportation Systems	2
1.2. Intersection Safety	11
1.3. Motivations of the Thesis	14
1.4. Objectives of the Thesis	19
1.4.1 Approach	20
1.4.2 Contributions	24
1.5. Thesis Organization	24

Chapter 2

Pervasive Computing for Intersection Safety	26
2.1. Intersection Collision Analysis	27
2.2. Pre-Analysis: Data Collection	34

2.3.	Analysis	37
2.3.1	Context-Awareness	39
2.3.2	Knowledge Based Systems	43
2.3.3	Data Mining	47
2.3.4	Discussion	58
2.4.	Post Analysis	62
2.4.1	Infrastructure-Only Systems	64
2.4.2	Vehicle-based Intersection Collision Warning and Avoidance Systems	68
2.4.3	Cooperative Intersection Collision Warning and Avoidance Systems	70
2.5.	The need for a Real-Time, Generic, Adaptive, and Cooperative Intersection Safety Framework	74
2.5.1	Consideration of a Variety of Real-Time Sensor Data Sources	74
2.5.2	Performance and Scalability of Collision Detection	75
2.5.3	Adaptability and Learning	78
2.5.4	Relationship between Collision Detection and Warning	80
2.5.5	Communication Model and Protocol	82
2.6.	Summary	83

Chapter 3

	The Ubiquitous Intersection Awareness (U&I Aware) Framework	87
3.1.	U&I Aware Framework	89
3.1.1	Components of the U&I Aware Framework	90
3.1.2	Novelty of the U&I Aware Framework	93
3.1.3	Implementation Map and Scope	95
3.2.	Consideration about Variety of Data Sources	99
3.3.	Performance and Scalability of Collision Detection	101
3.4.	Adaptability and Learning	106
3.5.	Relationship between Collision Detection and Warning	109
3.6.	Communication Model and Protocol	115
3.6.1	Status Report	116
3.6.2	Registration Message	118
3.6.3	Warning Report	120
3.6.4	Evaluation	121
3.7.	Summary	126

Chapter 4

	Collision Learning	129
4.1.	Intersection Simulation	133
4.1.1	An Overview of the Simulation Environment	137
4.1.2	Designing the Simulation Model	140
4.1.3	Implementation of the Intersection Simulation	143
4.2.	Mining Intersection Traffic and Collision Data	152
4.2.1	Collision Patterns Learning	158
4.2.2	Traffic Trends during Non-Collision-Free Periods	175

4.2.3	Normal Behaviours of Drivers during Non-Collision-Free Periods	178
4.3.	Summary	180
 Chapter 5		
	Collision Detection	182
5.1.	Improving Existing Collision Detection and Warning Algorithms by Preselection	185
5.2.	Preselection	189
5.3.	Collision Detection Evaluation	193
5.4.	Collision Detection Evaluation	198
5.4.1	Speed of Detection	198
5.4.2	Accuracy: Precision and Coverage	199
5.5.	Summary	202
 Chapter 6		
	Conclusion	204
6.1.	Research Contributions	205
6.2.	Research Directions	210
	References	213

LIST OF FIGURES

Figure 1.1.	Sensors that Enhance Car Safety	3
Figure 1.2.	AIDE Project	7
Figure 2.1.	Perpendicular Path Collision	29
Figure 2.2.	PATH's IDS Uses Illuminated Stop Sign	66
Figure 2.3.	Reactive Intersection Safety System	79
Figure 2.4.	Reactive & Deliberative Intersection Safety System	80
Figure 3.1.	Collision Detection, Learning, and Warning Components in the U&I Aware Framework	90
Figure 3.2.	Intersection Administration Zone	96
Figure 3.3.	Mapping the U&I Aware Framework to Agent Implementations	97
Figure 3.4.	Agent Implementation and Messaging Protocol	98
Figure 3.5.	Pair-Wise Collision Detection Algorithm	102
Figure 3.6.	Performance Comparison between Brute Force and Preselection Method	106
Figure 3.7.	TTA Cost Model Diagram	112
Figure 3.8.	Status Message	116
Figure 3.9.	Registration Message	119
Figure 3.10.	Warning Message	120
Figure 3.11.	Simulation of an Intersection Agent	123
Figure 3.12.	Simulation of a Vehicle Agent	124
Figure 3.13.	TTAwarning and TTAc command range	126
Figure 4.1.	Collision Learning in the U & I Aware Framework	130
Figure 4.2.	Intersection Crash Scenarios	131
Figure 4.3.	The Highest Occurring Scenarios that Encompass More than 60% of Intersections Collisions	131
Figure 4.4.	Intersection Simulation with Traffic Lights	138
Figure 4.5.	Intersection Simulation without Traffic Lights	138
Figure 4.6.	Content of LegBuffer Hash Table	145

Figure 4.7.	Periodic Traffic Data	148
Figure 4.8.	Collision Event Data with Attributes of Speed, Distance, Traffic Light Colour, and Collision Point	149
Figure 4.9.	Collision Event Data with Attributes of Manoeuvre, Direction, Angle, and Type.	149
Figure 4.10.	Collision Patterns Clustered by DBScan Algorithm with Vehicle Direction as Visualisation Category	161
Figure 4.11.	Side Collision Event Data with Attributes of Manoeuvre, Direction, and Angle of Each Vehicle in a Pair	164
Figure 4.12.	Side Collision Patterns based on Vehicle Direction as classified by C4.5	165
Figure 4.13.	The Probability of Side Collision Patterns Based on Vehicle Direction as Classified by Bayesian Network	166
Figure 4.14.	The Probability of All Collision Patterns Based on Collision Types as Classified by Bayesian Network	168
Figure 4.15.	The Probability of All Collision Patterns Based on Vehicle Direction as Classified by Bayesian Network	169
Figure 5.1.	Collision Detection in the U&I Aware Framework	184
Figure 5.2.	Collision Detection Algorithm	186
Figure 5.3.	Cross Intersection without Traffic Lights Implementation	190
Figure 5.4.	Implementation of Specific Collision Pattern	194
Figure 5.5.	Implementation of Generic Collision Pattern	194
Figure 5.6.	getConflictingLegsAndManuevers Method	195
Figure 5.7.	CarState Class Constructor	196
Figure 5.8.	Pair Wise Collision Detection Algorithm Implementation	196
Figure 5.9.	Collision Object	196
Figure 5.10.	Preselection and Pair-Wise Collision Detection Pseudocode	197
Figure 5.11.	Speed Evaluation of Collision Detection	199
Figure 5.12.	Evaluation Terms	200

LIST OF TABLES

Table 1.1.	Available Sensors on Vehicles	4
Table 1.2.	Various RADAR/LIDAR Features	5
Table 1.3.	Advances of Wireless Communication Technology	9
Table 2.1.	Analysis of Intersection Collision Patterns	33
Table 2.2.	List of Sensors Used to Capture Traffic Data	35
Table 2.3.	Application of Knowledge Base, Context Awareness, and Data Mining in Various ITS areas	62
Table 3.1.	TTA Components Value Range	122
Table 3.2.	TTAwarning and TTAcommand Range for Various Velocity	125
Table 4.1.	Intersection Configuration File	143
Table 4.2.	Vehicle Configuration File	146
Table 4.3.	Clusters of Collision Event Data as Clustered by DBScan	161
Table 4.4.	Partial Side Collision Patterns Based on the Direction Pairs	167
Table 4.5.	Partial Collision Patterns Based on the Direction Pairs	169
Table 4.6.	Partial Collision Patterns Based on the Manoeuvre Pairs	171
Table 4.7.	Specific Collision Patterns	173
Table 4.8.	Generic Collision Patterns in the Knowledge Base	173

CHAPTER 1

INTRODUCTION

“Road crashes are a huge cause of human trauma”¹

Safety hazards on the road are faced by every road user in the world. Road tragedy is one of the highest causes of death universally. Every minute, on average, no less than one person dies in a crash worldwide [Jones02]. The statistics of road crashes worldwide are as follows:

- According to the International Road Traffic Accident Database, globally, there are likely to be 10 million road crashes every year, which claim one and a half million fatalities [Frye01].
- In 2004 alone, there were 42,636 lives claimed on U.S. roads [ATSB06a].
- Each year, over 2,000 people die on Australian roads, over 60,000 are injured, and over 20,000 suffer serious injuries [BITRE00].
- Financially, road crashes cost Australia \$17 billion a year [UQ06].
- In 2004, there were 1,583 people killed in 1,444 collisions in Australia [ATSB05]. A 3.3% increase happened in 2005, as 1,636 deaths occurred in 1,481 road crashes [ATSB05].

¹ Australian Transport Safety Bureau, <http://www.infrastructure.gov.au/roads/safety/> (Accessed on 10 June 2008)

- In Victoria, there were 343 fatalities in 2004, which was the highest count over all other states [ATSB06a].
- In Western Australia, from the year 1990 to 1999, the total of fatal and non-fatal crashes was 363,080 collisions [Hents00].

The above figures clearly signal the importance of improving road safety in human lives. Interdisciplinary research groups and automotive industries have come together to tackle the issues of road safety. Nonetheless, computer science plays a major part in the developments of tools and techniques for improving safety and performance of Intelligent Transportation Systems, which are discussed further in the next subsection.

1.1. Intelligent Transportation Systems

In today's world, mobility is a vital need of society. Therefore, there is an escalating requirement for the provision of transportation systems that are efficient, safe, and automated. Intelligent Transportation Systems (ITS) aim to improve the efficiency and safety of transport systems [Charles03]. ITS is described as “the application of computing, information and communications technologies to the vehicles and networks that move people and goods” [Charles03].

Road safety stakeholders around the world are joining forces to enhance safety and performance of traffic by implementing state-of-the-art technologies on the road and in vehicles. One of the rapidly developing technologies used in transportation systems is sensor technology. Sensors are designed and created to monitor the conditions of the vehicles, the road, and the environment in specific vicinities, such as weather information and traffic conditions. This enables

drivers and traffic authorities to be better informed when the information and knowledge gained from sensors are made available to them. In all the currently released vehicles, there are up to one hundred sensors on board each car [Knoll06] (see Figure 1.1 [Jones02]).

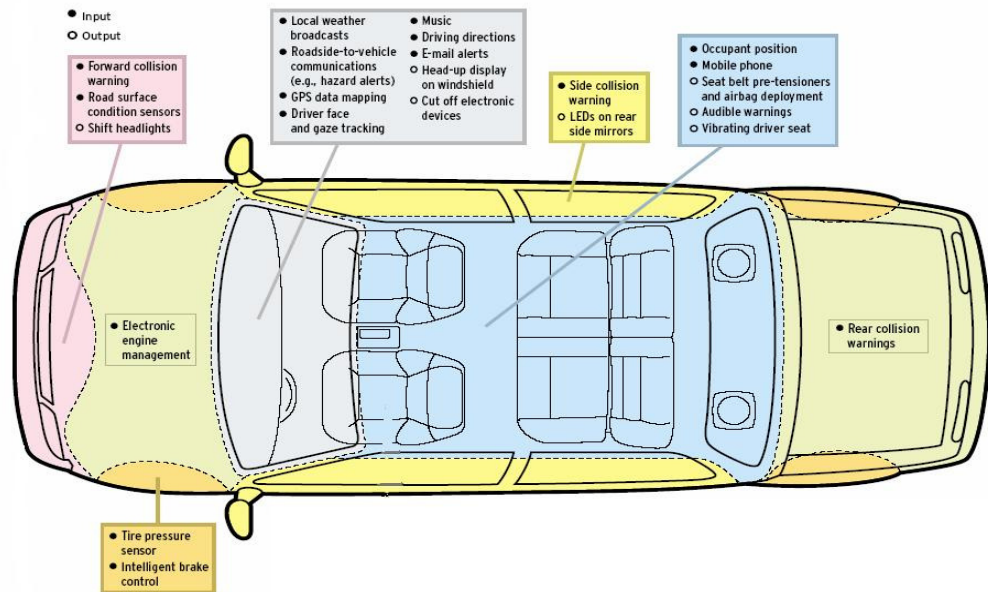


Figure 1.1. Sensors that Enhance Car Safety [Jones02]

The first generation collision-avoidance technology is already available in modern vehicles in the form of Adaptive Cruise Control (ACC). ACC systems are equipped with laser beams or radars to measure the distance of the vehicle from the vehicle ahead and compare both vehicles' relative speeds. ACC maintains the car's speed on a given value and distance between itself and the other cars that are ahead. However, ACC is mainly effective for driving on highways. Along with ACC technology, there are many sensors that enhance vehicular safety [Sharke03], [Strob04], [Jones02]. Sensors can also be used to monitor environmental conditions [Jones02], such as detection of wet, frozen, or snowy roads. Table 1.1 lists the various sensors that are currently available and the usage of each sensor type in this context.

Table 1.1. Available Sensors on Vehicles (adapted from [Strob04])

Sensor Types	Sensors	Usages
Imaging sensors	camera modules, 3D range cameras, infrared sensors, driver face and gaze trackers, road surface condition sensors	Lane detection, lane deviation, lane departures, obstacle detection, collision warning, driver vigilance monitoring, driver distraction monitoring
Range sensors	radar, laser scanner, ultrasonic sensors, forward/rear/side collision sensors	Lane deviation, lane departure, blind spot warning obstacle detection, collision warning, Adaptive Cruise Control, intelligent brake control
Digital maps	Global Positioning System (GPS), Geographic Information System (GIS)	Virtual sensors that provide information about the topology and the geometry of the infrastructure in the vehicle's environment
Communication devices	Wireless communication, local weather broadcast	Virtual sensors that detect hazards by receiving information or warning from external parties
Proprioceptive sensors	Inertial Navigation System, gyrometer, odometer, tachometer, speedometer	Motion detection, navigation aid, position tracking, orientation tracking, and velocity measurement
Mechanical sensors	Engine condition sensors, tire pressure sensors	Vehicle health monitoring

Existing range sensors, such as radar (long range sensor – using radio waves) and LIDAR (Light Detection and Ranging – using laser light) have been installed in vehicles to detect stationary objects, detect moving objects, measure distance, velocity, acceleration, and separation distance between two objects in traffic. The ranging and detection performance of radar and LIDAR vary according to products and manufacturers. For example, Bosch ‘Long Range RADAR’ sensor is able to detect 2 to 120 metres in terms of range with a 5% accuracy (maximum accuracy 0.5 metres), distinguish two objects in separation with minimum separation distance of 5 metres, measure up to 50 m/s relative speed, view objects within $\pm 8^\circ$ horizontal angle and $\pm 1.5^\circ$ vertical angle, and detect stationary objects [Strob04]. On the other hand, Continental Temic’s ‘Adaptive Cruise Control RADAR System ARS 300’ is capable of covering a range of 0.25 to 170 metres with 0.25 metres accuracy, detect separation distance of 2 metres, measure -24.7 to 73.6 m/s (-89 to 265 km/h) relative speed, measure -20 to 20 m/s² relative acceleration, and can handle stationary objects [Strob04]. More detailed

facts on range sensors [Strob04] are displayed in Table 1.2. Many of these range sensors are utilised as collision warning sensors in vehicles.

Table 1.2. Various RADAR/LIDAR Features (adapted from [Strob04])

Product / Company	Range	Separation Range	Relative Speed	Relative Acceleration	Horizontal / Vertical View Angle
Bosch Long Range RADAR Sensor	2 to 120 m	5 m	± 50 m/s	Not known	$\pm 8^\circ / \pm 1.5^\circ$
Continental Temic 'Adaptive Cruise Control RADAR System ARS 200'	1 to 150 m	7.5 m	-24.7 to 73.6 m/s (-89 to 265 km / h)	-20 to 20 m/s ²	$\pm 5.1^\circ / \pm 1.7^\circ$
Continental Temic 'Adaptive Cruise Control RADAR System ARS 300'	0.25 to 170 m	2 m	-24.7 to 73.6 m/s (-89 to 265 km / h)	-20 to 20 m/s ²	$\pm 9^\circ / \pm 2.1^\circ$
Continental Temic side looking short range RADAR 'SLR 100'	0.2 to 30 m	0.2 m	-35 to 35 m / s (-127 to 127 km / h)	None	$120^\circ / 15^\circ$
Continental Temic closing velocity detecting short range LIDAR 'CID 100'	10 m	Not known	1 to 56 m / s (5 to 200 km / h)	Not known	$36^\circ / 8^\circ$
Continental Temic short range LIDAR 'SIS 200'	0.5 to 50 m	Not known	-60 to 60 m / s	Not known	$\pm 15^\circ / 3$ to 6.5°
DENSO 'LIDAR Sensor'	0 to 130 m	Not known	51 m / s	6.35 m / s ²	$\pm 18.0^\circ / 4^\circ$
DENSO 'RADAR Sensor'	5 to 180 m	Not known	-55.5 +27.8 m / s	6.35 m / s ²	$\pm 10^\circ / 4^\circ$
DELPHI 'Long Range RADAR Sensor'	1 to 150 m	Not known	-63.9 to 31.9 m/s	Not known	Not known
DELPHI 'Short Range RADAR Sensor'	0 to 6 m	Not known	± 8.8 m / s	Not known	Not known
Hella 'Adaptive Cruise Control (ACC) B' (LIDAR sensor)	200 m	Not known	± 50 m / s	Not known	$16^\circ / 3^\circ$
Hella '24 GHz Short Range RADAR'	0.75 to 50 m	1.80 m	0 to 70 m / s	Not known	± 50 to $\pm 70^\circ / 13^\circ$
IBEO 'ALASCA'	0.3 to 80 m	0.5 to 1 m	Not available	Not available	$240^\circ / 3.2^\circ$
RoadEye 'Forward looking RADAR (FLR) sensor'	2 to 150 m	1.5 to 9 m	± 50 m/s	Not available	$\pm 18^\circ / 4^\circ$
TRW Automotive long range RADAR sensor 'AC 10'	200 m	0 m	± 50 m/s	Not known	$\pm 6^\circ / \pm 2.5^\circ$
Valeo 'Multiple Beam RADAR'	0.5 to 60 m	Not known	0 to 69.4 m / s	Not available	$150^\circ /$ Not known

Sensor systems that perceive where the driver is looking are also being developed [Fletch03], [Jones02], [Sara04], and [Seeing05]. This advancement is leading to enhance other research such as driver fatigue or inattention detection, driver distraction monitoring, pedestrian spotting, blind-spot checking, merging assistance to confirm whether adequate clearance exists between cars, driver warning for lane keeping, computer-augmented vision (that is, lane boundary or vehicle highlighting on a head-up display, traffic sign detection and recognition), and human factors research aids [Fletch03], [Seeing05]. An example of a facial imaging sensor system is FaceLAB, which is a head, face, eye, eyelid and gaze tracking system for human subjects and operates in a 3-dimensional volume using an entirely non-contact, video-based sensor that captures and processes facial images using a monochrome stereo camera hardwired to a workstation [Seeing05]. Proprietary algorithms use the image sequences to focus on facial landmarks such as the lips, nose, and eyes. This filtering generates head position and orientation measurements precise to within 1 mm and 2 degrees [Seeing05]. Such a facial imaging sensor system is a component that is generally included in Advanced Driving Assistance Systems (ADAS) [Gruyer05].

ADAS have been built to help drivers to manage driving tasks. An example of ADAS is AIDE (Adaptive Integrated Driver-vehicle Interface) [AIDE04]. Such driving assistance systems are developed to monitor the driver's condition by observing the face and gaze of the driver to detect drowsiness (left picture in Figure 1.2) and to provide information about road, vehicle, and other drivers and also to issue warnings when threats are present (right picture in Figure 1.2). Although ADAS have been developed in recent years, with features such as lane deviation detection, speed limit control, and face and gaze tracking to enhance drivers' vigilance, existing ADAS can only provide dedicated functions and display a partial view of driver behaviours [Gruyer05]. It is necessary for ADAS

to communicate with other vehicles and also sensors on the road for a holistic view of the driver, vehicle, and environment.



Figure 1.2. AIDE Project [AIDE04]

Long before in-vehicle sensors existed, many roadside sensors have been implemented and used for traffic monitoring. For example, to sense a vehicle's speed at a point, conventional inductive loop detectors, self-powered vehicle detectors, optical sensors, or radar sensors have been employed [Ferlis01]. Inductive loop detectors are used to detect the presence of vehicles in certain road segments. They are also used to measure traffic flow and estimate vehicle speed. In the past few years, inductive loop detectors have proven effective for detecting incidents, such as road blockage or traffic jam. A sensor named Traffic-Dot [Coleri05] is able to detect the presence, speed, length and size of vehicles with up to 97% accuracy, which is better than inductive loop detectors. Imaging sensors have also been recently installed to monitor traffic patterns and passing vehicle trajectories. Imaging sensors such as video cameras are used to monitor certain traffic violations, e.g., red light cameras for red light running detection and speed cameras for detecting speed limit violations.

However, despite the presence of range sensors, such as forward, rear, and side collision sensors in current vehicles and traffic monitoring sensors on the roadside, road collisions still occur. This is not merely because only few vehicles are currently equipped with those sensors and there are still limitations with sensor technologies (reliability and error rate of sensors), but also because in-

vehicle collision sensors alone cannot guarantee that a vehicle is free from impending collisions since collisions most likely involve more than one vehicle. Existing ITS devices such as obstacle detection or vehicle detection (radar or vision based) are not enough for intersections, since such sensors have limited visibility and detection. When a number of vehicles that are due to collide are approaching with a very high speed or from different intersection legs, it is possible that radar or vision based sensors are not able to detect the approaching vehicles until the collision becomes imminent. Since not all vehicles are equipped with such obstacle detection sensors, there is a need to communicate such information about incoming collision threats to other possible affected vehicles.

Similarly, traffic performance has not been greatly improved with the presence of digital maps and communication devices. With the increase in number of vehicles on roads, there is greater need to optimize the traffic network. Traffic information should be analysed and learnt so that road users can be better informed about public transport, parking, traffic conditions, best travel routes, and much more. Therefore, there is a clear need to leverage sensory information for more intelligent decision making in ITS.

Apart from the advances in sensor technology, the wireless technology has also been advancing (see Table 1.3). Along with the advances of wireless communication technology, short and long range communication technology between vehicle and infrastructure and between vehicles is being developed. The standard of IEEE 802.11p (Wireless Access for the Vehicular Environment, WAVE) is currently being formulated [Kerry08]. It is an extension of the 802.11 wireless network standards to support ITS applications. It enables high speed data exchange between vehicles and between vehicles and road infrastructures.

Table 1.3. Advances of Wireless Communication Technology

Time	Event
1896	Guglielmo Marconi invented wireless telegraph devices [Duben03]
1897	The birth of radio – Marconi’s invention of wireless telegraph was patented [Shea00]
1901 – 1902	Marconi’s telegraph device is able to send and receive a telegraph across the Atlantic Ocean[Jensen94], [Shea00]
1914	First voice over radio transmission [Shea00]
1927	First commercial radiotelephone service between UK and US [Duben03]
1946	First interconnection of mobile users to public switched telephone network (PSTN) [Shea00].
1946	First car-based mobile telephone set up using ‘push-to-talk’ technology [Duben03]
1950s	A number of ‘push-to-talk’ mobile services established in major cities. The first paging access control equipment (PACE) paging systems launched. [Duben03]
1960s	Improved Mobile Telephone Service (IMTS) launched; supports full-duplex, with more channels and more power [Shea00], [Duben03]
1962	The first communication satellite, Telstar, launched into orbit [Duben03]
1968	Defense Advanced Research Projects Agency (DARPA) in US developed the Advanced Research Projects Agency Network (ARPANET), the father of the modern Internet [Duben03]
1976	Bell Mobile Phone has 543 pay customers utilising 12 channels in the New York City region [Shea00]
1977	The Advanced Mobile Phone System (AMPS), invented by Bell Labs, installed in the US with geographic regions partitioned into ‘cells’ [Duben03]
1980s	The era of analogue signals (1G) [Light02]
1983	January 1, TCP/IP selected as the official protocol for the ARPANET, causing rapid growth in Internet technology [Duben03]
1989	The European digital cellular standard, GSM, was defined by Groupe Spécial Mobile [Shea00]
1990s	The era of digital signals (2G) [Light02]
1992	There were 1 million users of Internet [Duben03]
1994	Ericsson telecommunications company began to develop a technology to connect portable devices without cables, it was later named Bluetooth [Morr02]
2000	802.11(b) wireless based networks are in high demand [Duban03]. 802.11 wireless local area network (WLAN) standards are utilised to build Wi-Fi Hot-Spot networks and metropolitan area network (MAN) [Jha04].
2000	The era of third generation cellular system (3G) [Shea00]. Bluetooth standards launched [Shea00].
2001	WiMAX, the Worldwide Interoperability for Microwave Access, introduced by Wimax Forum [Wimax07], to support delivery of wireless broadband access over long distances as an alternative to wired broadband like cable and DSL, from point-to-point links to full mobile cellular type access, with expected capacity up to 40 Mbps per channel. WiMAX is also used to connect Wi-Fi hotspots.
Now	Development of the next generation wireless communication systems (the fourth generation (4G) or beyond 3G (B3G) systems) to support up to 100 Mbps in outdoor environments and up to 1 Gbps in indoor environments [Bharga06], an all-IP end-to-end solution and will combine mobility with multimedia-rich content, high bit rate, and IP transport [Jha04]. Development of IEEE 802.11p (Wireless Access for the Vehicular Environment, WAVE) [Kerry08].

Communications devices can be used to capture local weather broadcasts and forewarn the driver about upcoming dangers, such as an oil spill or a major accident, transmitted from the road infrastructure by digital short-range communications. Such special purpose devices are being developed to facilitate vehicle-to-vehicle communication. However, existing small and mobile devices such as mobile phones or PDA that have wireless or Bluetooth technology can also be used for vehicle-to-vehicle and vehicle-to-infrastructure communication. Therefore, sensor data can be transmitted easily from one point to another for further analysis and processing.

Additionally, it is also necessary to increase safety in road transportation systems and traffic networks by automation. Autonomy is a desired attribute for transportation coordination. Many human operated machines in transportation systems, including vehicles and rule based traffic controls, are now being developed into semi-autonomous machines (where human intervention is still required) and fully autonomous machines (which are able to be independent without the need for human intervention). In order to integrate automation into roads and traffic networks for, multi-disciplinary approaches should be taken into account. One approach that can be applied into ITS is to integrate intelligent pervasive computing techniques for road safety advancement. This is supported by the fact that computing and sensory devices are becoming more ubiquitous in the road environment.

As stated by the U.S. Department of Transportation, there are eight areas where ITS can advance safety [Sharke03]. Those major areas are categorised into four types of collision avoidances (rear-end, lane change and merge, road departure, and intersection), two types of enhancements (vision and vehicle stability), and two types of monitoring (driver condition and driver distraction). One of the main focuses of ITS is to improve intersection safety, which is a complex issue that

requires support from all areas of ITS [IVI02]. Therefore, the following section discusses specifically the issues and challenges in intersection safety.

1.2. Intersection Safety

The need for enhancing intersection safety is supported by the fact that the figure of the annual toll of human loss caused by intersection crashes has not significantly changed, regardless of improved intersection design and more sophisticated ITS technology over the years [USDOT04]. The following facts and figures explain the necessity of an effective and efficient intersection collision warning and avoidance systems on the road.

- Intersections are among the most dangerous locations on U.S. roads [Frye01]. The figure of fatal motor vehicle crashes at traffic signals is increasing more rapidly than any other type of fatal crash in USA. 9,612 fatalities (22 percent of total fatalities), and roughly 1.5 million injuries and 3 million crashes took place at or within an intersection [USDOT04].
- Yearly, 27 percent of the crashes in the United States occur at intersections [Frye01]. However, in 2002, in the USA, approximately 3.2 million intersection-related crashes occurred, corresponding to 50 percent of all reported crashes [USDOT04].
- Financially, intersection crashes cost \$96 billion annually in the USA [USDOT04].
- In Japan, intersection collision figures are even more devastating, with more than 58 percent of all traffic crashes occurring at intersections. Intersection-related fatalities in Japan are approximately 30 percent of all Japanese traffic accidents, and those fatal crashes mainly happen at intersections without traffic signals [Frye01].
- In Western Australia, almost half (49%) of all crashes that occurred in the years 1990 to 1999 took place at intersections [Hents00].

- In Queensland, there were 40863 collisions that occurred at intersections in the years 2002 to 2005. This figure constitutes 45 percent of all collisions during that period [Queens07]. During the same period, 0.61% of all intersection crashes were fatal and 19.28% of all intersection crashes caused serious injury [Queens07].

Intersection collisions are multifaceted problems. It affects all types of vehicle platforms, i.e. light vehicles, commercial vehicles, transit vehicles, and specialty vehicles [IVI02]. The complexity of intersections is mainly due to the varied characteristics of intersections [Stubbs03], such as:

- Different intersection geometry: shapes, number of legs, median width, number of lanes. The number and frequency of accidents in any particular intersection is affected by the geometry of the intersection. Each intersection normally has a different treatment for its safety based on its geometry;
- Different intersection characteristics: signalised/unsignalised, rural/urban setting;
- Different usage of intersections: traffic volume, types of vehicles, various average traffic speed, and road turn types;
- Different users of intersections should also be considered when dealing with intersection safety issues: pedestrians of all ages including those with cognitive and physical abilities/disabilities, cyclists, older drivers, younger drivers, transit/light rail/trolley vehicles, trucks including loading/unloading manoeuvres, emergency vehicles, adjacent driveways serving commercial properties, and commuters [USDOT04].

Negotiating intersections is one of the most difficult tasks a driver needs to cope with [USDOT04]. To successfully perform a vehicle manoeuvre through an intersection, the driver must integrate diverse types and amounts of information,

make a decision and perform the desired action. One shortcoming is that the human brain resembles serial processors and the load of the cognitive task at intersections can be quite onerous. There are a number of matters a driver must consider when nearing an intersection, such as observing and regulating speed, maintaining lane position, watching for other vehicles, observing traffic signals or signs, watching for pedestrians, bicyclists, people in wheelchairs and blind or visually-impaired people, decelerating for a stop, searching for path guidance, and selecting the proper lane [USDOT04].

Research suggests that driver error may account for roughly 90 percent of all crashes in the U.S. [Funder04, Sara04, Sharke03, USDOT04], whereas in Australia, road crashes that are ascribed to driver error is at the rate of 95 percent [Krish05]. Although technologies in automotive safety and highway design are advancing, the one factor that has not changed is the driver. Therefore, the key factor to prevent collisions in intersections is to understand collisions in each intersection and to help drivers in being aware of the potential threats they face.

From the above characteristics that pertain to intersection collisions, a driving assistance or collision warning system for intersections is both highly desirable and necessary. Such a system must in particular be able to detect potential collisions and warn drivers of those threats. There have been a number of initiatives in developing intersection collision warning and avoidance systems. As far as the current state of the art is concerned, no existing intersection collision warning and avoidance system can tackle intersection collision problems entirely. This is mainly because many of these systems cannot guarantee an effective and efficient real-time collision warning delivery, since:

- The data source only comes from either roadside infrastructure or vehicle and is therefore not comprehensive;

- Available resources and communication means for cooperative methods have not been considered;
- Most methods of the warning delivery only rely on roadside infrastructure (e.g. through LED displays);
- These systems have been designed just for specific intersections and thus create the difficulty of transferring the technology and systems to different intersection types.

Therefore, an intersection collision warning and avoidance system should be developed to meet the above issues by incorporating:

- the ability to detect and warn of collisions in real time so that impending collisions can be avoided by potentially affected drivers;
- the adaptability of the system to various kinds of intersections.

Research in intersection safety should investigate and propose novel methods for detecting and issuing about warning intersection collisions in real time that can be used in any intersection type. Research in this thesis is motivated by these two important issues in intersection safety.

1.3. Motivations of the Thesis

An intersection collision warning and/or avoidance systems should achieve the goal of real-time collision detection in order to avoid imminent crashes. A fast and accurate detection would allow time for the system to warn about a potential collision, for drivers to respond to warnings, and for avoidance systems or drivers to steer clear from the potential collision. Therefore, a collision avoidance system should consider different components that make up the available time to avoid the collision: detection time, communication time, time taken by vehicles or

infrastructures to issue warnings, and driver's response time including time to brake or change manoeuvre. Given that time to avoid a collision is typically in the order of seconds, various methods or techniques that can reduce computational time and warning time are of paramount significance.

Existing intersection collision warning and avoidance systems [USDOT99], [Ferlis01], [Funder04], [Stubbs03], [Veera02], [Verid00] mainly consist of two components: *detection* and *warning*. None of these systems have considered the enormous value of learning from sensor data. The advances in computational data analysis techniques provide valuable research that can leverage the vast amount of sensor data available in vehicles and on the road. The information and knowledge learnt from this sensor data can be useful for both the adaptability of the system for various intersections and also improve the efficiency and effectiveness of the system to detect threats, issue warnings, and avoid collisions in real time. As stated in the previous section, these features are greatly desired in an intersection collision warning and avoidance system.

Since the development and installation of sensors in vehicles and on the road, there is a need to understand sensor data for better situation recognition at an intersection. In order to comprehend driver behaviours and traffic conditions for uses in safety applications, simply relying on raw conventional sensor data, such as from ground loop sensors installed on the road, is insufficient [Chan04]. Information that significantly enhances understanding and knowledge about the intersection can be gained from analysing sensor data. This very important dimension has largely been unaddressed in the current systems and in the literature. An intersection collision warning and avoidance system should take into account the availability of sensor data and incorporate techniques to analyse this data for better understanding of the intersection. As data becomes easily available and accessible, new knowledge and interesting patterns can be learnt

and extracted, such as collision patterns, driver behaviours, vehicle conditions, best travel routes, traffic trends, and so on.

The key patterns that vary across intersections and are useful for determining the causes of collisions in an intersection are collision patterns. In order to improve the safety and design of an intersection, one of the first procedures is to execute a field observation and statistical analysis of collision patterns, since understanding patterns of collisions in an intersection can assist in planning for countermeasures. However, manual observation and manual analysis of collision patterns are expensive. Besides, given the vast volume of observed data, a manual approach is potentially infeasible. For example, as analysed by Veridian Engineering [Verid00], the collision patterns that occur in cross intersections are across path turn, perpendicular paths without violation of the traffic control, perpendicular paths with violation of traffic control, and premature intersection entry. Those collision patterns have not included other common collision patterns, such as rear-end collisions. It is necessary to have a comprehensive set of collision patterns, because in the future, impending collisions that match the collision patterns can then be detected. Such knowledge can also be utilised to improve the intersection design or safety measurements. Since it is necessary to have a comprehensive collection of collision patterns in an intersection safety system, human manual observation (without the help of the state-of-the-art computing technology) alone is not ideal due to the higher cost as well as the potential for error. Therefore, there is a need to investigate the application of machine learning or data mining techniques to extract collision patterns in an intersection. Collision patterns can be learnt from traffic data accumulated by sensors and historical collision data.

Furthermore, results of those studies [Verid00] cannot be applied for all types of intersections due to the uniqueness of each intersection. Due to the fact that each

intersection has a different set of collision patterns from another (i.e. a set of collision patterns is only applicable to a particular type of intersection), existing intersection collision detection and warning systems are built only to suit a particular intersection or a certain intersection type, such as the IDS (Intersection Decision Support) installed in an intersection in California [Funder04]. These systems cannot deal with emergent and changing patterns in the intersection.

Given the uniqueness of each intersection, rather than manually fine-tuning a system for each intersection, ideally, an intelligent system for intersection safety should be able to adapt to different types of intersections automatically [Salim05]. Changes and emergent trends are important characteristics of intersections particularly since variability is very high. Situations in road intersections, such as traffic trends, weather changes, and collision patterns, are very dynamic and vary from one intersection to another and can vary even within an intersection as conditions change. The ability to cope with subtle incremental changes in patterns has not been considered in the current intersection collision warning and avoidance systems. As a result, these systems have applied a fixed or static knowledge base rather than a dynamic knowledge base that is able to evolve in the presence of changes and emergent trends by incrementally adding new and relevant patterns and rules learnt from analysis of sensor data.

When the data from sensors on the road and in the vehicle are learnt and the results of such learning are added into the knowledge base, the intersection collision warning and avoidance system is made aware of possible collision patterns and able to detect future collisions based on relevant patterns. The knowledge base that contains relevant collision patterns learnt at the intersection can be used as the basis for the detection component in an intersection collision warning and avoidance system. In existing systems, all vehicles and users at the intersection are considered in the collision detection calculation. If the number of

vehicles and road users increases, the time required for collision detection calculation also increases exponentially. If a knowledge base was used as the basis for collision detection calculation, the vehicles and road users that do not match the collision patterns in the knowledge base can be ruled out for performing calculations. This in turn can significantly improve the efficiency and effectiveness of an intersection safety system.

Lastly, in order to have a real-time intersection collision warning, a real-time messaging protocol that enables communication between vehicles and road infrastructure should be established as the telecommunication infrastructure, such as the wireless broadband and mobile phone networks, are already available. This allows exchange of useful information needed for collision detection and warning messages required for collision avoidance. Due to the limitation in warning time available, the cost involved in issuing a warning message should be calculated in order to ensure that the warning message is received in time by the intended recipient. The messaging protocol designed in such a system should be simple and asynchronous, as we need to avoid unnecessary delay in transmitting warning messages.

In a nutshell, an intersection safety framework that is able to cope with the issues of learning collision patterns and issuing timely warning through early detection of potential collisions is required. It should monitor continuously and learn the occurrences of collision patterns that are not learnt merely through manual field observation conducted from time to time. The need to know collision patterns comprehensively is not only for the purpose of having an intersection safety framework that is able to adapt to various intersections (i.e. a generic intersection collision warning and avoidance system with incremental learning at local vicinities), but also for the effectiveness and efficiency of the system in detecting collisions and issuing warnings to potentially affected drivers in real-time. A

communication model and protocol that are designed specifically with intersection safety in mind are required. Therefore, in this thesis we propose and develop a real-time and generic context-aware framework for collision detection and warning in road intersections, which is elaborated further in the next subsection.

1.4. Objectives of the Thesis

We aim to develop a collision avoidance framework, which has the ability to deal with the following three main research questions [Salim08b]:

- How to develop an intersection safety system that can adapt to all kinds of intersections?
- How to detect collisions at road intersections in real time?
- How to warn drivers of potential collisions or hazards in real time?

Therefore, this research aims to develop a framework that is capable of real-time collision detection and warning to avoid impending threats. Further, it must be adaptive to different intersection types through the knowledge acquisition of intersection accidents. As such, the main objective of this thesis is to *propose a real-time and generic context-aware framework for collision avoidance in road intersections*.

The notion of context-awareness implies the framework could understand the situation of its surroundings and change its behaviour accordingly [Dey99]. We need to design a framework that takes into account all possible data sources in order to comprehend the situations in an intersection so that the framework can assist road users to be aware of the threats at the intersection surroundings. This can be facilitated by having a framework that is able to learn characteristics of

collisions, detect potential collisions, and warn accordingly. Thus, the framework must possess *learning*, *detection*, and *warning* components. The approach and contribution of this research are discussed in the following subsections.

1.4.1 Approach

This thesis approaches the need for a collision avoidance framework from the pervasive computing perspective. Due to the rapid development of sensors, ubiquitous and mobile devices, and wireless networking, we envision a road traffic network and vehicles equipped with devices that are interconnected with each other and sharing real-time messages. With this provision in mind, it is necessary to view the intersection safety problems from a pervasive computing perspective. The following discussion presents a number of pervasive computing techniques that can be utilised for advancement of intersection safety. There are still many other techniques that are not mentioned in this thesis but yet can be found useful in improving road safety or ITS in general.

Learning of collision patterns is performed using data mining techniques. As these patterns are extracted from historical collision and near-collision events in an intersection, the collision patterns are comprehensive up to the time of learning. Therefore, this approach helps to deal with the possibility of incompleteness of collision patterns and human error in manual field observation. The set of collision patterns that are localised to each intersection can be stored in the knowledge base of the intersection safety system as the basis of threat detection. A dynamic knowledge base technique for robotic collision avoidance [Mani93] can be adapted to road collision avoidance, instead of a static knowledge base. A dynamic knowledge base is extensible and adaptable. It involves learning to accumulate and refine rules to adapt to situational changes. Conversely, no new rules are added to a static knowledge base. Since an

intersection safety system should also have the ability to adjust and adapt to any intersection's vicinity, a dynamic knowledge base that keeps all the collision patterns that are only relevant to a particular intersection is needed. All possible collision patterns in the particular intersection where the system is installed needs to be learnt and stored in the dynamic knowledge base.

As collision detection must take place in real time, the methodology for collision detection should be simple and optimized. However, a simple collision detection algorithm involves kinematics equations to calculate point of collision and time to collision between two vehicles [Miller02]. Therefore, in order to optimize collision detection, the number of vehicle pairs to be calculated in real time needs to be minimised to reduce the computational time. This is because calculating each possible pair of vehicles located at an intersection for a potential collision is not prudent due to real time considerations. If we need to take into account all possible vehicle pairs in an intersection to be calculated for collision detection, detection time will take longer than it should. In fact, not all possible vehicle pairs will lead to collisions. The number of possible pairs of vehicles that need to be calculated for potential collisions should be reduced. A means of filtering the vehicle pairs that have the potential of colliding with each other through the patterns in the knowledge base needs to be proposed and developed in order to reduce the number of collision detection computations. We propose that patterns can be used as preselection criteria for finding and matching a pair of vehicles. In our framework, only vehicle pairs that match particular collision patterns will be calculated for collision detection. We evaluate the performance of the collision detection by measuring the speed and accuracy of the detection. As the accuracy of collision detection algorithms must be reliable, we propose a method adapted from information retrieval techniques to evaluate performance, which are *precision* and *recall* [Singhal01]. In this thesis, we term recall as *coverage* as we do not actually recall an existing collision, but use it to predict a future collision.

The formula to measure the precision and coverage of collision detection are proposed in this thesis and used to evaluate the performance of the system.

In order to avoid an imminent collision, the message structure and protocol, and avoidance mechanisms should be effective and efficient. The time available before a future collision occurs must be known and compared against the time available for avoiding the collision. In order to avoid a collision, the time to warn drivers of an impending collision must be lesser than the predicted time of collision. If there is not enough time to warn the drivers involved, the warning message should not be sent to the driver, instead, a direct command message needs to be sent to the vehicle system. Depending on the time available to avoid a collision, different schemes of warning messages in order to deal with different situations are required. A model that describes and calculates the required cost to issue a warning must be established, so that we can calculate the feasibility of a warning message to reach the intended recipient. As there can be different types of warning messages, the cost model should also consider calculation for different components involved in each warning type. We also need to have a short, straightforward, and simple message structure and protocol in order to lessen the message transmission time.

The aim of this research is to facilitate early warnings so that collisions can be avoided. Therefore, we actually focus on the early stage of pre-impact. Post-impact behaviours, actions, or methods that are necessary to alleviate the burden of the impact are not the scope of this research. In this research, it is assumed that the required sensing technologies and wireless communication are readily available on road and in vehicles. We, in this study, assume that the sensor data is accurate. This work assumes that the data used in the framework have been filtered by other mechanisms proposed in other research or studies. The particulars of the sensors assumed in the system are discussed in detail in Chapter

3. Network issues such as bandwidth, latency, etc. are not in the scope of this research. Also, human factor issues such as driver's distraction by warnings, user acceptance of the technology, and privacy issues are not part of this thesis.

This research is implemented and evaluated on a computer based simulation where the road and vehicle sensors used in the implementation are simulated, because the resources and licenses to do such extensive experimentation in the real world are not feasible without sufficient evidence in a simulated environment [Sicking00]. In order to simulate the collision and traffic data, we use computer based simulation to generate vehicles and traffic movements that eventually lead to collisions. This collision data and also traffic data generated from the simulation (representing data that can possibly be captured in the real-world by ITS sensors) are recorded into log files for further analysis. Although this research uses the notion of pervasive computing, which implies that computing resources are everywhere, the cost/benefit analysis to investigate the feasibility of real-world deployment of such pervasive framework is not considered in this thesis. There are also no empirical data in a small scale that would allow us to assess and extrapolate at larger scale on the cost/benefit of such a deployment. This thesis focuses solely on the safety aspects.

In particular, this research aspires to investigate an integration of knowledge based systems, data mining, and kinematics for a novel context-aware framework that is able to:

- monitor an intersection to learn for patterns of collisions and factors leading to a collision using data mining;
- detect potential hazards in intersections in an efficient manner from information communicated by road infrastructures, approaching and passing vehicles, and external entities;
- warn particular threatened vehicles that are at the intersection.

1.4.2 Contributions

The contributions of the thesis are:

- a generic intersection safety framework that is adaptive to different intersection types as the knowledge base is initialised with collision patterns learnt from traffic and accident data from that particular intersection;
- real-time collision detection through reduction of the number of vehicle pairs to be calculated;
- real-time communication protocol for intersection collision avoidance, including the communication cost model;
- performance evaluation methods to calculate the precision and coverage of the collision detection.

The central focus of this thesis is the real-time collision detection and warning, which are supported by sub-components: design and development of a computer-based intersection traffic simulation, learning of traffic sensor data, development of a knowledge base of collision patterns, development of a pre-selection algorithm for efficient collision detection, and design of collision warning message structures and protocols. The organisation of the thesis is presented in the next section.

1.5. Thesis Organization

This thesis is organized into six chapters. Chapter 2 reviews the related work of this research in the following areas: existing intersection collision warning systems, knowledge based systems in road and transportation, and data mining research in Intelligent Transportation Systems.

Chapter 3 describes our proposed U&I Aware (Ubiquitous Intersection Awareness) Framework to achieve the objectives of this research. The U&I Aware Framework consists of three components, which are collision learning, collision detection, and collision warning. The collision avoidance process through these components is elaborated on further in this chapter.

Subsequently, Chapter 4 discusses knowledge acquisition of intersection data using data mining techniques. For the purpose of data generation, the implementation of the test bed of the framework, which is a computer based simulation of intersections and sensors, is discussed here. The parameters of the simulation and the data generated from the simulation are explained. In this chapter, we demonstrate the process of pattern acquisition using data mining techniques on the data generated from the simulation. Data mining in this research is used to acquire collision patterns and traffic patterns.

Chapter 5 presents the existing collision detection algorithms that are currently available along with our proposed method to improve the speed of those collision detection algorithms. We present the pair wise route contention algorithm. We discuss the proposed preselection method that used the knowledge base. Preselection is applied to identify potentially colliding vehicles based on the rules in the knowledge base. We also discuss how this approach can help reduce computation time of collision detection. Finally, we present the evaluation methods and results in terms of speed and accuracy of the collision detection process.

To conclude, Chapter 6 summarizes the thesis and the future directions of this research.

CHAPTER 2

PERVASIVE COMPUTING FOR INTERSECTION SAFETY

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.”²

Pervasive (or ubiquitous) computing suggests that computing devices and applications are seamlessly connected “anytime, anywhere” [Weiser91]. This has become a reality since computing devices can now be found everywhere, in mobile phones, Personal Digital Assistants (PDA), and everyday appliances embedded with tiny chips and sensors. Pervasive computing research, which has been developing rapidly in recent years, has introduced the notion of bringing computation out to the physical world where activities happen, yielding sub-areas such as context-awareness and the use of artificial intelligence techniques (including multiagent technology). Branches of artificial intelligence such as intelligent agents, machine learning, and data mining have been found useful in ITS, because they can take into account the social aspect of computer systems, including human-computer interaction, distributed problem solving, and

² Weiser, M., “The Computer for the 21st Century,” *Scientific American*, Sept., 1991, pp. 94-104; reprinted in *IEEE Pervasive Computing*, Jan.-Mar. 2002, pp. 19-25.

simulation of social systems [Schlei02]. This has motivated the application of such intelligent systems to emerge in transportation systems. This progression has been enabled through the development of state-of-the-art on-the-road and in-vehicle sensors, wireless networking, and power efficient computing.

In the light of the advances in pervasive computing techniques, this chapter discusses how these techniques can potentially address the intersection safety issues. This chapter is organised as follows. Firstly, we review the conventional methods of analysing intersection collisions and set the background for the subsequent sections by presenting the three stages of road safety examination in Section 2.1. Section 2.2 discusses the pre-analysis stage of road safety examination by presenting various ways of how data are collected to be further processed. Section 2.3 discusses the analysis stage and pervasive computing techniques that can be used to analyse the collected data. Section 2.4 reviews existing intersection collision warning and avoidance systems that are designed, developed, and implemented after analysis is done (post-analysis stage). Section 2.5 presents the desirable properties of an intersection collision warning and avoidance systems. Section 2.6 concludes the chapter.

2.1. Intersection Collision Analysis

The complexity of intersection safety issues, as previously stated, is mainly contributed by the variety and variability of intersection characteristics. Therefore, each intersection requires a different safety treatment from another. Road characteristics and safety analyses are performed at each site to find the factors contributing to collisions and solutions to reduce or eliminate them. In this section, we focus on discussing the outcomes of research and field study that have investigated the cause of intersection collision and the collision patterns

found in those intersections. Research groups and road safety stakeholders worldwide have made attempts to analyse collision patterns in intersections in order to find the root of collisions and prevent them. However, each group has a different set of findings of intersection collision patterns, simply because they work on different intersections (or types of intersections). The following discussions review their findings with regards to the cause of collisions in intersections.

The U.S. Department of Transportation performed an exhaustive analysis of the intersection crash problem [Mitre99], [Verid00], [USDOT00]. Four different crash scenarios are classified in a four-legged cross intersection type: left turn across path, perpendicular path entry with inadequate gap, perpendicular path with violation of traffic control, and premature intersection entry with violation of traffic control signal [Mitre99]. These crash scenarios are only applicable to crash patterns within the specific geometric alignment of a four-legged cross intersection. Left turn across path in the U.S. is similar or equivalent to right turn across path in Australia. In case of a four-legs cross intersection, an example of right turn across path is a turn from lower/south leg of the intersection to the right/east leg of intersection across the incoming traffic from the upper/north leg of the intersection. Collision can possibly happen between the vehicle from the lower/south leg and a vehicle from the upper/north leg when making such a turn. Perpendicular path collision involves vehicles that travel from two perpendicular legs. For example, a vehicle that travels from the left/west leg of a four-legs cross intersection collides with a vehicle that travels from the lower/south leg (see Figure 2.1).

The distribution of the crash scenario based on the 1994 U.S. intersection crash database are as follows: 23.8 percent occurred when executing left turn across path, 30.2 percent happened during perpendicular path entry with inadequate gap,

43.9 percent occurred when taking perpendicular path with violation of traffic control, and 2.1 percent happened when there was premature intersection entry with violation of traffic control signal [Mitre99]. For each of the scenarios, particular attributes associated with the traffic control device, driver response, intended manoeuvre, and underlying factors were recognised. There are a number of factors contributing to a collision: driver did not see obstacles or incoming cars, driver attempted to beat incoming vehicles, driver's vision obstructed or impaired, driver inattention, deliberate violation of stop sign, and deliberate violation of traffic signal [Mitre99]. The collision scenario that has the highest percentage, perpendicular path with violation of traffic control, can be caused by either driver inattention or deliberate violation of stop sign/traffic signal.

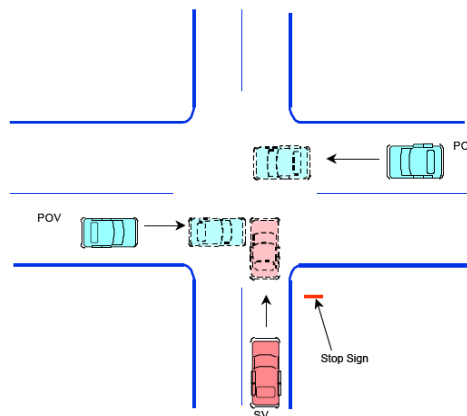


Figure 2.1. Perpendicular Path Collision [Verid00]

The fatal intersection crashes in U.S.A, during 2002, are analysed in two categories. If it is categorised by traffic control devices, 37 percent occurs at intersections with a stop sign, 32 percent at intersections with traffic signals, 28 percent at intersections without traffic signals, and 3 percent at intersections with other traffic control devices [USDOT04]. If it is categorised by manner of collision, the same data is classified into 62 percent side impact collision, 28 percent single vehicle collision (without another vehicle in motion), 5 percent head on collision, and 5 percent rear-end collision [USDOT04]. In this study,

issues that are found to be associated with intersection collisions are: traffic control misuse (for example: STOP sign that cannot be seen or misinterpreted), red light running, pedestrian safety, mature age drivers, accessibility for disabled, and human factors corresponding to all drivers. At the intersections with traffic signals (signalised intersections), usual driver errors include: indecisive dilemma whether to proceed or stop at a yellow-signal indication; miscalculating time to reach an intersection; miscalculating time to make a smooth stop; failure to notice signal and proper lane assignment; and misinterpreting sign information [USDOT04]. At the intersections with no traffic signals (unsignalised intersections), usual driver errors include: unsafe gap taking; inaccurate estimation of approaching vehicles' speed; miscalculating time to accelerate after making a turn; and failure to give up right-of-way [USDOT04].

The analysis on German accident data by the INVENT research project [Lages04] reveals that the types of collisions found are as follows (ordered from the highest to the lowest number of fatalities): accidents on curves, accidents on straight road, following/rear-end collisions, collisions with pedestrian or animal, lane change accidents, orthogonal/perpendicular path collisions, stand alone accidents (without a collision partner), and cross path turns. In accordance with the INVENT accident analysis, one focus of intersection safety system obviously needs to be on the crossing and turning assistance as well as on the right of way assistance [Lages04].

In Australia, research on Queensland's unsignalised intersections [Arndt03] provides evidence that the most frequent vehicle accidents at this type of intersections are: angle (right-turn or through-movement from minor leg colliding with a vehicle drifting through on the major road), right-turn from major road (colliding with an approaching major road vehicle), and rear-end. Single vehicle, head-on, sideswipe and left-turn from minor road (colliding with a vehicle

drifting through on the major road) are frequent, but not as frequent as the first three categories. The main factor that can contribute to crashes at unsignalised intersections is the failure to give way to the vehicle on the minor road. This typically causes a collision with a vehicle on the major road. The common errors of drivers failing to give way can be caused by not seeing the other vehicle, miscalculating the speed and position of the other vehicle, not recognising the intersection, or not realising the need to give way. A number of failure-to-give-way accidents are affected by obstructions to vision particularly by other vehicles. Other factors that can increase accident rates are an increase in relative speed between vehicles, an increase in the number of traffic flows to be observed (an increase in driver's workload), an increase in visibility restrictions, and a decrease in the levels of perception of an intersection.

Signalised intersections in Victoria, Australia, have been analysed using 1987 – 1991 crash data [Ogden94]. There are four main types of crashes at signalised intersections in Victoria, which are: right through crashes, rear-end crashes, adjacent approach crashes, and pedestrian crashes [Ogden94]. Intersections crashes involving pedestrians were the most fatal. Spatial clustering on the signalised intersections crash data explained that 86 percent of those crashes occurred at 50 percent of all the intersection sites. However, the severity per type of crashes in different intersections varies greatly. This signifies how every intersection is quite unique and has its own issues. Due to the simpler intersection geometry, T-intersections are safer than cross-intersections [Arndt03]. Most intersection crashes occurred in clear weather, daytime, on dry roads, and in the afternoon peak period. Thirty four percent of daily crashes at intersections occurred between 3 to 8 pm. The highest number of crashes at intersections or within 100 m of an intersection in an hour occurs between 5 and 6 pm, which is the daily rush hour. From drivers of all age groups, genders, and license types,

the drivers involved in intersection crashes are mostly young, inexperienced, and particularly male drivers.

After reviewing accident types or patterns in different intersections around the world, it is obvious that we cannot generalise collision patterns of one intersection with another. Table 2.1 portrays different sets of collision patterns extracted from various intersections.

As a set of collision patterns is unique to a particular intersection, hazardous site analyses is always performed for each intersection by road safety experts [Boury00]. The safety examination for a specific road site, which is also referred as Road Safety Audit (RSA) [Kwas07], can be categorised into pre-analysis, analysis, and post-analysis [Boury00]:

- pre-analysis: includes data collection;
- analysis: includes identification of problems, accidents and the characteristics of accidents;
- post-analysis is then executed by implementing the necessary actions to prevent accidents.

The integration of computer systems into RSA is desired to automate those tasks.

Similarly, we review the existing work in collision avoidance systems and related ITS technology within those three categories. The next three sections present existing research projects and literature in each relevant stage. The conventional data collection through collating expert knowledge and sensor data collection is discussed further in Section 2.2. The pervasive computing techniques that have been used in the road safety and related ITS areas for data analysis are described in Section 2.3. Development of collision warning and avoidance systems, which is an integral part of post-analysis, is discussed in Section 2.4.

Table 2.1. Analysis of Intersection Collision Patterns

Intersection Type, Traffic Signal, Location	Collision Patterns	Other Factors that Increase Crash Rates
Four-legged cross intersection, signalised, U.S. [Mitre99], [Verid00]	<ul style="list-style-type: none"> • left turn across path • perpendicular path entry with inadequate gap • perpendicular path with violation of traffic control • premature intersection entry with violation of traffic control signal 	<ul style="list-style-type: none"> • obstructed vision • driver's inattention • failures to give way • deliberate violation of stop sign or traffic signal
All intersection types, signalised and unsignalised, U.S. [USDOT04]	<p>Categorised by manner of collision:</p> <ul style="list-style-type: none"> • 62% side impact collision • 28% single vehicle collision • 5% head on collision • 5% rear-end collision <p>Categorised by traffic signal:</p> <ul style="list-style-type: none"> • 37% at intersections with stop sign • 32% at intersections with traffic signals • 28% at intersections without traffic signals • 3% at intersections with other traffic control devices 	<ul style="list-style-type: none"> • Traffic control misuse • red light running • pedestrian safety • mature age drivers • accessibility for disabled • human factors corresponding to all drivers
All intersection types, signalised and unsignalised, Germany [Lages04]	<ul style="list-style-type: none"> • Accident on curves • accident on straight road • following/rear-end • collision with pedestrian or animal • lane change accident • orthogonal/perpendicular path collision • stand alone accident (without a collision partner) • cross path turns 	
Unsignalised intersections, Queensland, Australia [Ogden94]	<ul style="list-style-type: none"> • angle (right-turn or through-movement from minor leg colliding with a vehicle on the major road) • right-turn from major road • rear-end • single vehicle • head-on • sideswipe • left-turn from minor road 	<ul style="list-style-type: none"> • intersection's geometry • weather • dry road • peak period • daily rush hour • age-gender group

The next section describes the existing techniques used and the desired improvements in pre-analysis stage.

2.2. Pre-Analysis: Data Collection

Before we can proceed to analyse intersection data, we need to first collect information. The main information source is from examining domain-related documents [Boury00], such as:

- textbooks on human factors to establish the integration of the three stages of driver information process (perception, cognition, and action) with safety consideration [Kwas07];
- publications on safety facts and figures to recognise potential collision patterns, e.g. categorised by crash type, time of crash, type of participants [Kwas07];
- guidelines and manuals for RSA [Kwas07];
- observations of road safety specialists [Kwas07].

Another information source is from road safety experts, thus, interview with road safety experts can be conducted in order to collect information and validate findings, new theories, and methodology [Boury00].

However, due to the flood of data available through sensor technology installed on the road and in vehicles, we should also collect and utilise data we gather from sensors collectively. The availability of real-world sensor data helps reducing the need for examining data contained mostly in paper-based documents, which requires meticulous efforts to access and exploit them. Therefore, the sensors that are available on the road and the kinds of data that can be retrieved from such sensors should be examined.

As discussed in Chapter 1, there are a range of different sensors now available on the road and in-vehicle. We need to decide whether to collect data that we need for intersection site analyses from roadside sensors or in-vehicle sensors. The selection of sensor data sources depends on the existence and availability of in-

vehicle sensors, roadside sensors, and communication infrastructure. Table 2.2 lists possible sensors that can be used to detect the required information for collision detection.

Table 2.2. List of Sensors Used to Capture Traffic Data

Data	In-vehicle sensors	Roadside sensors
Speed	Speedometer, Global Positioning System (GPS)	Camera, Inductive loop detector, Traffic-Dot
Vehicle Size	Built-in information	Traffic-Dot
Travel Direction	Camera, Compass, GPS	Camera
Current Position	GPS, GIS	Camera, Inductive loop detector, Traffic-Dot
Angle	Camera, Steering Wheel, GPS	Camera, Inductive loop detector, Traffic-Dot
Vehicle Registration Number	Built-in information	Camera, ANPR
Vehicle Manoeuvre	Eye and Gaze Sensors, GPS	N/A

The accuracy of the information detected by the sensor depends on the sensor itself, not on the sensor location (roadside or in vehicles). Each sensor type has different sensor products that in turn have a different level of accuracy. However, roadside sensors require a longer time to process the data that is pertinent to a particular vehicle rather than the sensors in the vehicle itself. The data listed in Table 2.2 is pertinent to a specific vehicle. For example, vehicle size and registration number can be recorded into the vehicle computer system as built-in information; therefore retrieving these data from the vehicle can be done speedily. On the other hand, retrieving data of vehicle size and registration number can also be done by roadside sensors with a longer processing time using Automatic Number Plate Recognition (ANPR) [Wiki07a], which is a recent technology that has been installed on roads, especially in the United Kingdom, to detect the number plate of a vehicle using Optical Character Recognition (OCR) on captured images. As of the year 2006, ANPR is able to scan number plates around 1 vehicle per second on vehicles moving up to 100 mph (160 km/h) [Wiki07a]. Similarly, the speed of a vehicle can be retrieved using Global

Positioning System (GPS) in a vehicle or a speed camera (a traffic rule enforcement control) on the road. Retrieving vehicle speed from GPS is instant. Conversely, most of the installed speed cameras on the roads are the conventional speed cameras that capture still images, which need to be interpreted by human operators. Only the most recent speed cameras have ANPR capabilities. Furthermore, there are still many issues to deal with vision sensors. Vision sensors' capability to detect object (e.g. vehicle, pedestrian, etc) during low light period (night, dawn, dusk) and in the presence of obstacles and shadows is still being developed and improved.

In terms of availability, many roadside sensors listed in Table 2.2 are already installed on the roads. Equally, many of the in-vehicle sensors listed in Table 2.2 are readily available in all vehicles, e.g. speedometer, compass, and GPS, which are the main component of driver's navigation systems and a common feature of today's mobile phones. GPS service is already installed in today's vehicles, including sedans, taxis, and trucks. GPS has been used widely in navigation, map creation, land surveying [Wiki07b], and also tracking vehicle manoeuvres (for example, lane change manoeuvre detection [Xuan06]). GPS helps to provide awareness to driver of the current location, speed, direction, and angle of the vehicle. These data are useful when collected and analysed for the purpose of improving safety. Nevertheless, collection of sensor data would depend on availability of sensors. Therefore, one must first choose the sensors that are required (and whether additional sensors are required to be installed for the purpose of data collection) in order to obtain the necessary data before preceding any analysis.

Once sensor data are collected, the right techniques are required to analyse the data and extract interesting patterns and useful information about the intersection to assist in collision detection, warning, and avoidance. This can be achieved

through the existing pervasive computing technology and intelligent systems, which are discussed in the next section. Section 2.3 reviews methods in pervasive computing techniques, such as knowledge based systems, data mining, and context awareness, in ITS that can be used to analyse the acquired intersection data.

2.3. Analysis

The next stage after pre-analysis (data collection) in the Road Safety Audit (RSA) is the analysis stage. In this stage, traffic, intersection, and crash data are analysed to identify the root causes of crashes and the possible avoidance or mitigation techniques. This section aims to review different approaches in intelligent pervasive computing that are used to enhance data analyses in road safety and also in wider ITS application areas, such as in traffic optimisation and automation.

Successful research projects using hybrid and cross disciplinary techniques of artificial intelligence, traffic, and transportation technologies have been seen since 1980s – e.g., expert systems, such as those built for traffic light controllers [Bazzan05]. However, since traffic and transportation systems are becoming more complex, both individual choices and global conditions of traffic and transportation systems must be better understood for greater efficiency and safety. Therefore, transportation systems are now being viewed and analysed at both the individual (micro) and the societal (macro) levels [Bazzan05]. To facilitate the analysis of the micro and macro view of intersection's vicinity, sensor data inputs must be analysed with tools and techniques from the pervasive computing paradigm.

We focus on the paradigms of *knowledge base systems*, *data mining* and *context-awareness* as we recognise that they have been widely used in ITS. Since the progression of sensor networks in ITS, these techniques are becoming increasingly relevant as they take into account sensor data and each has the capability to process this data efficiently for various purposes. *Knowledge based systems* consume all the given information sources, analyse them, and store them efficiently to be used to solve a specific problem. The application of knowledge based systems has existed for Road Safety Analyses (RSA), decision support in transportation systems, highway safety monitoring, and driver monitoring. *Data mining* is used for semi-automatic discovery of patterns, associations, changes, anomalies and significant structures from data [Gross98]. Data mining has a considerable value since it has the potential to process large amounts of sensor data to yield interesting, understandable, and applicable information for traffic efficiency and road safety. *Context* is “any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves” [Dey99]. The context awareness paradigm is useful to define the awareness of any computing application to its context, which can be the driving context, temporal, location, environment, vehicle, or driver. When an application is aware of its contexts, it is able to self-adjust accordingly.

Since each of those paradigms offer different capability and contribution to ITS, as can be seen from the discussion in the following subsections, it is observed that the right amalgamation and integration of those techniques are able to promote safety and efficiency in traffic and transportation systems. We note that the combination of those paradigms enables *cooperative and adaptive distributed situation awareness* of the physical world so that appropriate measures or actions can be taken autonomously:

- *distributed* in the sense that a system can be made concurrently aware of multiple points or places in the physical world;
- *cooperative* in the sense that information from nodes at different locations are integrated to form an overall picture;
- *adaptive* in the sense that a system can adapt to the situational changes and varying conditions in the physical world.

We review each of these paradigms and its data processing capabilities in the following subsections. Subsection 2.3.1 describes the notion of context-awareness and how context has been used in ITS. Subsection 2.3.2 reviews existing research in knowledge based systems in ITS. Subsection 2.3.3 presents the definition and usage of data mining as a widely accepted data analysis technique. Subsection 2.3.4 concludes section 2.3.

2.3.1 Context-Awareness

The notion of context-awareness has been adopted in ITS, since a context-aware application has the capability to adapt to situation changes informed by the sensors. The availability of context information may influence the behaviours of the application or device [Chen00], [Moran01]. There are three important features of context: where you are, who you are with, and what resources are close by [Schil94]. The most important types of context are identity, location, time, and activity. Therefore, context-aware applications observe the “who’s, where’s, when’s, and what’s” of entities and use this information to find out why the situation is happening [Dey99]. Therefore, context is reclassified into four categories, which are computing context (such as communication bandwidth, connectivity, nearby resources e.g. printer), user context (such as user’s activity, profile, location, nearby entity), physical context (such as lighting, traffic, weather), and time context (such as day, hour, season) [Chen00]. An application

can then use available context information to adapt to environment changes. Hence, context awareness is useful in ubiquitous and mobile computing to provide fault-tolerance and an adaptive computing infrastructure [Chen00].

Given the advances of sensors networks and ubiquitous computing devices that are found in vehicles and road infrastructures, it is apparent that computing applications have the potential to adapt to the changes in the environment. In any ITS applications, contextual information about vehicle dynamics, environments, and driver behaviours should be integrated to improve accuracy and gain better assessment of the current situation [Gruyer05]. According to [Vidal02], observing environmental conditions and driver behaviours in steering, braking, accelerating makes early detection and warning of dangerous driving situations possible. Also, recognizing driving behavioural patterns is necessary for many ITS applications, such as for vehicle navigation and driver's monitoring, because it provides better situation awareness [Oliver00], [Mitro05]. There are a number of areas where context-awareness is applicable in Intelligent Transportation Systems: driver's behaviour recognition, cooperative and autonomous vehicle navigation, traffic modelling and monitoring, and environment map and monitoring.

It is essential to incorporate knowledge about context to properly make decisions in complex dynamic environments such as in driving [Oliver00] or in analysing intersection safety [Salim05]. For example, in the intersection and highway scenarios described in [Julien02], resource-aware and location-aware concepts are employed, since the presence of other computing entities, the availability of resources associated with them, the connectivity, and their particular location or movement are the traits that can influence the behaviour of the application. In the following discussions, we review various applications of context-awareness in ITS and the data analysis of each application.

A context-aware route profiling application was presented in [Harr04] to evaluate the performance of road networks in the Republic of Ireland. The sensor data sources were traffic flow data, meteorological data, and road event data, from which the following contexts were extracted: time, weather, and road usage context [Harr04]. The proposed system merely detected and reported traffic condition. It is desirable for such system to have reasoning capability to improve traffic throughput efficiency, such as by recommending better alternative routes when incidents or road blocks are detected to be present. Those desired features can be met by applying data mining techniques, such as in [Gross05], which is discussed in 2.3.3.

The CORTEXT project [Verís02] proposed a model for collections of sentient objects, which were mobile intelligent software components that accepted input from different sensors that sense the environment before the system decided how to react. The scenarios for context-aware cooperating autonomous cars in the CORTEXT project were: cooperative behaviour without human control and autonomous vehicle navigation from a starting place to the predetermined destination [Verís02]. The next generation cars [Sivaha04] designed by CORTEXT were able to publish events (such as emergency braking) to other relevant vehicles (such as cars following within a certain distance). Sentient objects within other cars could ask to obtain the braking event and when notified could do their own correct braking action, which was then followed by publishing their own braking events. Context-awareness was achieved by consuming events from different sensors and event channels, combining those events to higher-level contexts (using Gaussian modelling, one of machine learning techniques, and Bayesian networks), and reasoning about those contexts using expert logic [Sivaha04]. In summary, the CORTEXT project created a system to sense and react to environmental changes. However, it did not incorporate analysis and

learning of data from sensors or mechanisms for reasoning about dangerous driving behaviours and predicting imminent threats. This can be achieved by applying data mining techniques.

Application of data mining techniques in a context-aware application can yield enormous context information that helps the application to be more context-aware. This was demonstrated in [Oliver00]. In order to recognise driver behaviour and manoeuvres in various driving scenarios, it is important to know the driver and vehicle context, such as the driver's gaze, position, speed and direction of the traffic [Oliver00]. Therefore, real-time context information, such as the car physical state (speed, brake, acceleration throttle, steering wheel angle, and gear), road state (road geometry, exit information), traffic state (relative speeds and positions of neighbouring cars), and driver's state (driver's face and gaze position and driver's viewpoint) were used in [Oliver00]. This project, using the above context information, was able to accurately recognize a driver's driving manoeuvres (stopping, turning, passing, changing lane left, changing lane right, turning left, turning right, starting) one second before the actual vehicle signals take place. The results concluded from the experiments were as follows [Oliver00]:

- although some driving manoeuvres could be recognized by using car information only, passing and changing lane manoeuvres require external context information for more accurate results;
- the usage of context was necessary for recognising certain manoeuvres such as passing and changing lanes;
- driver's gaze, as it was strongly correlated with driver's mental states, was a significant attribute in detecting lane changes, passing, and turnings;
- each manoeuvre could be predicted on average one second before any car signals or obvious changes in the car occur.

The details of the data mining techniques used and the learning results are further discussed in Section 2.3.3.

For further improvement of context-aware applications, information about the contexts of an application can be stored in a knowledge base, as the existence of knowledge-base helps decision making. The next section discusses the benefits of knowledge based systems in ITS.

2.3.2 Knowledge Based Systems

The first generation of knowledge based systems (also known as expert systems) was designed to fully automate human's reasoning and decision making process. However, the systems were only able to deal with easy problems and limited in the knowledge representation, reasoning and justification capabilities. Therefore, the second generation of expert systems was developed not for fully automation, but to assist users in decision making by providing advice through better knowledge representation and justification capabilities [Boury00].

In ITS, knowledge based systems have been applied for road safety and traffic management, such as in SICAS (System with Intelligent and Cooperative functions to help in the Analysis of Sites) [Boury00] and RSIT (Road-Site Investigation Tool) [Kwas07]. SICAS was developed to accommodate incremental collection and representation of knowledge from analysts and experts through a computer based interface, electronic database and knowledge base (contains case based scenarios or patterns); whereas RSIT was developed to guide high-crash rate site investigation by analysing and associating roadway characteristics, driver behaviours, and traffic controls with crash patterns, which would result in a set of proposals for road and traffic control improvements in the high-crash site [Kwas07].

Data analyses in knowledge based systems involve processing incoming data and store them electronically for basis of decision making. In SICAS [Boury00], result of analyses were stored in forms of databases (useful for obtaining detailed historical facts), electronic documents (useful for formalisation of guides, manuals, or descriptions), domain ontology (useful for defining and illustrating a concept or a word in order to remove ambiguity), task models (useful for organising task hierarchies), knowledge bases (store expert knowledge, rules and norms, and patterns), and case bases (store a set of solved case studies) [Boury00]. In RSIT (Road-Site Investigation Tool) [Kwas07], in order to help automating decision making process, the acquired knowledge was represented in a decision tree structure.

The implementation of RSIT focused on two-way stop-controlled intersections. The main research components in this project consisted of three steps: knowledge acquisition, knowledge representation and knowledge implementation. The knowledge acquisition included information from manuscripts on driver information process (perception, cognition and action), publications on safety facts and crash patterns, guidelines and manuals for road safety, final reports for road safety audits, and observation of specialists [Kwas07]. The knowledge was converted into rules through natural language programming as in expert systems. The knowledge representation imitated the decision-making process during site investigation. It started with identifying all applicable crash patterns, which were determined by crash type, time of collision, weather condition, pedestrian/bicycle crash, and the visibility of the intersection and the traffic signals). As for the knowledge implementation, a graphical user interface was developed to guide the users through the investigation by: capturing the user-defined knowledge base, displaying the questions, stating the selected answers, prompting possible subsequent question after a selected answer, receiving written comments from the

investigator, and summarizing the investigation in a report. However, we found that the knowledge acquisition in RSIT was done manually with a historical data. It did not utilise computing techniques that could help finding new knowledge from sensor data. Nevertheless, the instalment of RSIT has helped reducing investigation time needed from the checklist-based manual investigation [Kwas07].

Similarly, a knowledge-based system was required in Greece [Chass05] to assist in making priority lists of road maintenance, given the high rates of accidents and fatality. Safety has become a primary factor to identify whether a particular vicinity becomes a priority for maintenance. Hence, in order to assist in arranging the right priorities of rural highways that require maintenance, the knowledge-based decision support system has been designed and developed. It consists of a database, analysis tools, and a knowledge base. There are three sources of the knowledge base: road and accident data, results from past research, and expert opinions [Chass05]. The combination of different sources can increase the quality of the knowledge base and overcome the limitation of a single source. The database includes information about accident features, road geometry, pavement condition, traffic, operating and environmental situations, other characteristics of the environment, and maintenance history. The next steps are priority setting and feasible treatment assessment. The arrangement of priority is based on rates of collision, fatality, and repeating accident types. Then, a set of rules is used as a basis to determine a course of actions. Finally, heuristics, priority list, and resource budget were used to decide the safety improvement projects and resource allocation. Nevertheless, this system only focuses on the road repair work or maintenance in order to increase road safety. On-site prevention techniques that can be applied by utilising the knowledge base are not considered.

An adaptive knowledge-based driver monitoring and warning system, DAISY [Onken94], developed to work on German motorways, could generate warning messages based on the driving situations and adapted to the style of driving of each driver. It was based on situation awareness models, which consisted of driving situational model (e.g. lane change, overtake another vehicle), danger model, driver target speed model and model of the actual driver. However, only three types of danger model (collision patterns) that were modelled to capture potential accidents: collision with a vehicle ahead due to inadequate deceleration behaviour, violation of lane boundaries due to inadequate steering behaviour, and violation road boundary lines in curves because of inadequate velocity [Onken94]. The model of the actual driver was adaptive as it was learned through neural nets, on the driver's speed profile on different types of road geometry (i.e. straight, left curve, right curve). The experiments that were executed by test drivers yield a positive result in terms of increasing driver's safety although the warnings can be unnecessary and cause significant distractions to the driver [Onken94]. It would be better if the neural net learning that was applied on the driver model could also be applied on the danger model. This is because different road sites have different road characteristics (e.g. no curves or fixed lane boundaries); hence, those three types of danger models will not be applicable.

Existing knowledge-based systems mostly have a static knowledge base that does not update itself to changes in the environment. However, variations in intersection characteristics and environments require new generation knowledge-based systems that are flexible and extensible through exploiting real-time information sources. The advances in sensor technology and wireless communication have opened the opportunities for nearly seamless capability for knowledge sharing and collaboration. The requirement for adaptive intersection safety systems can be achieved by developing a knowledge-based system that implements a dynamic knowledge base. When an intersection safety system is to

be built to suit a particular intersection, it should have a knowledge base that stores collision patterns of that intersection. As the knowledge base in a system needs to be populated dynamically, we explore existing intelligent computing techniques, such as data mining, that have been used in ITS to gain new or interesting knowledge and patterns. Data mining is stated as a future work or suggestion of how interesting knowledge can be obtained after data collection in knowledge based systems [Boury00]. Therefore, the next discussion reviews data mining concept and its implementation in ITS.

2.3.3 Data Mining

Apart from the distributed aspects of the traffic and transportation systems, there is a considerable amount of data from in-vehicles and roadside sensors. Hence, it is essential to understand sensor data and act accordingly. Given the huge amount of data, a question arises whether computer systems can learn and improve automatically from past experience. There are many success stories of machine learning and data mining in producing knowledge bases. Algorithms have been formulated for certain types of learning such as classification, clustering, and association rules [Mitch97]. Effective algorithms would be those that are able to facilitate better understanding of data, better ways for tasks to be executed, or performance improvement through experience [Mitch97].

Machine learning has been implemented widely in Intelligent Transportation System, such as to train a computer-controlled vehicle to manoeuvre correctly when driving on a variety of road types [Mitch97]. The ALVINN system, as cited in [Mitch97], applied machine learning to drive unaided at 70 miles per hour for the distance of 90 miles on public highways among other vehicles. In [Moriar98], supervised machine learning and reinforcement learning have been used for cooperative lane selection in highways. Performance improvement was aimed to

be achieved by applying learning and generating rules in each car by coordinating lane changes. An example of a common lane change operation is when a car ahead has a slower speed, the car should move to an open lane on the left or the right side. The rules that were employed in this system were used for optimising lane usage. Slower vehicles had to move to the slower speed lane and give way for faster vehicles to pass. SANE (Symbiotic, Adaptive Neuro-Evolution), a form of reinforcement learning, consists of neural networks that represent the rules that map sensory input to decision output. The input and output layers are fixed in SANE, but the connections and the middle layers can evolve. During training, the supervised learning was performed with an existing knowledge base and SANE algorithm. The intelligent lane selection was able to improve traffic performance as shown in the simulation. However, the only input data that were simulated were velocity, position, acceleration, and car size. As the actual behavioural data of the driver, environment, and vehicle were not taken into account, the actual effectiveness of the algorithm was unknown.

With the huge amount of data available at present in databases, spreadsheets, data from sensors, and many other organizational data, data mining has become popular over the last decade. Data mining is the development of methods and techniques for making sense of data by pattern discovery and extraction [Fayyad96]. Data analysis techniques have the potential to facilitate better understanding of the vehicle, the driver, and the road environment for different purposes. Hence, information about the vehicles, infrastructures, and environment (road, traffic) extracted from sensors and further data analysis of sensor data, can potentially be utilized for better situation recognition and management. Furthermore, this is also supported by the advancing wireless technology (see Table 1.3), which facilitates communication among vehicles, road infrastructures, and traffic authorities. As data has become easily available and accessible, new knowledge and interesting patterns can be extracted and

learnt, for example, collision patterns, driver behaviours, vehicle conditions, best travel route, and so on. Such information can enable safer and more efficient transportation. There have been a number of research projects on data mining in the area of ITS, such as for driver's behaviour recognition, traffic optimization, and incident detection. We review these applications in following discussions.

Oliver and Pentland [Oliver00], as mentioned in Section 2.3.1 applied learning on sensor information to predict driver behaviours or manoeuvres. Hidden Markov Models (HMM), a set of discrete states and probabilities of transitions between them [Rabin89], was considered inadequate to characterize multiple interacting processes [Oliver00], as the basic HMM only has a single state variable. It is necessary to model real-time systems that have the temporal and spatial states. However, to represent it in HMM is intractable. Therefore, a new algorithm named Coupled Hidden Markov Models (CHMMs) has been proposed for modelling multiple interacting processes. Coupled Hidden Markov Models (CHMM) was utilized to: (i) learn driver behaviours that are captured by in-vehicle sensors such as video camera, face and gaze movement trackers, and the car's internal state (speed, acceleration, steering wheel angle, gear, and brake), and (ii) predict the next intended manoeuvre accurately [Oliver00]. The limitation of this system was in the data source. The data used in this system was originated from instrumented commands in a driving simulator, while the data used in [Mitro05], the next project discussed, was obtained from sensors in a real vehicle in normal driving situations.

A similar project that used contextual information from sensors to recognize driving patterns was proposed in [Mitro05]. This project considered actual responses from vehicle and the environment as data were collected from vehicle sensors. The model of the current situation for a driver, vehicle, and environment was made up of a patterns history and the currently detected driving event. In

[Mitro05], HMM was implemented to recognize driving patterns. In HMM, the changes in states of a Markov process could only be viewed by observables, but were hidden from the outside users. The pattern recognition in HMM consists of training and evaluation. Seven HMM models have been developed in [Mitro05], each to recognize one of seven most common driving events according to their experience, which are: driving along left and right curves, turning left and right on intersections (with and without roundabouts), and driving straight across an intersection with a roundabout. The training of each model used training set for each event type, which was about 30% of the complete set. The evaluation used the complete set. Although the processing power required for real-time recognition in current CPUs was low, the training of the HMMs real-time environment was said to be demanding due to its iterative character [Mitro05].

In [Chan04], data mining was used to analyse driver behaviours in an intersection. To facilitate understanding of driver behaviours for uses in safety applications, simply relying on raw conventional sensor data, such as from ground loop sensors installed on the road, is insufficient, as data analysis techniques is necessary to extract significant traffic parameters, such as time gap estimation in crossing paths [Chan04]. For example, in implementing intersection safety solutions, monitoring the speed, location, and movement of each vehicle is essential. For data collection in this study, which was performed in California, USA, a set of video cameras and radar were set up at an intersection to determine distance to intersection and speed of each vehicle at up to seven targets. The data was used to estimate time to reach the intersection, which was then mined together with distance to reach the intersection data to produce interesting knowledge [Chan04]. Two scenarios were analysed in mining the data: firstly, left turn across path subject vehicles versus other vehicles from opposite direction; secondly, red light running and dilemma zone [Chan04]. In the case of the second scenario, to detect potential violators of traffic signal early in real-

time, it was essential to determine the dilemma zone in a particular intersection. A dilemma zone is where and when the drivers are indecisive as to slow down or to speed up on the yellow light. It was found through mining the traffic data in the field observation of this study that a dilemma zone is located within about 10 – 30 meters from the intersection. At this site, the average speed of normal traffic is approximately 10 m/sec; hence, the dilemma zone corresponds to 1 – 3 seconds before a vehicle arrives at the stop line. To effectively detect a potential violator, the ranges of the dilemma zone should be monitored precisely [Chan04]. In this project report, the types of data mining algorithms, techniques, and evaluation methods used are not specified.

Apart from driver behaviours learning [Oliver00], [Mitro05], [Chan04], as previously discussed, data mining can also be applied to learn traffic and collision patterns [Gross05], [Chong04] and factors and conditions attributed to collisions [Abdel05], [Singh03].

The Pantheon Gateway Project [Gross05] records real time highway data from more than 830 traffic sensors installed in Chicago highways every six minutes, which accumulates to 173,000 sensor readings every day being added to the database. The purpose of this research is to detect real time changes in traffic conditions (speed, volume, occupancy). Using a tree-based classifier, the condition change is further analysed to detect the cause of it, which can either be weather related, accident, special events, or road construction [Gross05]. Therefore, traffic condition changes, present accidents, and special events that affect the traffic can be detected in real-time based on the learnt traffic patterns. This study signifies that from the enormous amount of live sensor data collected from highways, data mining can learn and deliver useful patterns of traffic which can be associated with various incidents on highways. These patterns are used to predict traffic incidents. Therefore, it can be deduced that such an approach can

also be applicable for learning collision patterns at intersections for collision avoidance.

In another study, automobile accident data was analysed using a hybrid approach of machine learning, which involves neural network, decision tree, support vector machines, and a hybrid decision tree for the purpose of building models to predict the severity of accident injuries [Chong04]. Based on the manner in which the collision occurs, the data was classified into seven categories: not collision, rear-end, head-on, rear-to-rear, angle, sideswipe same direction, and sideswipe opposite direction. The output classes that were learnt by the machine learning algorithms were categorized into no-injury, possible injury, non-incapacitating injury, incapacitating injury, and fatal injury. The test results revealed that it was best to use neural networks to classify non-incapacitating injury, incapacitating injury, and fatal injury, because of its accuracy. However, for the non-injury and possible injury classes, it was better to use decision trees [Chong04].

A road safety project in Florida conducted experiments and implementations to identify collision-prone conditions in freeways in order to predict freeway crashes in Advanced Traffic Management and Information Systems (ATMIS) environment [Abdel06]. Traffic flow information from traffic loop detectors, historical crash data and rain data has been collected for this study. Using online loop and rain data, identification of high risk situations in freeways can be done in real time. Learning algorithms that were utilized to build the crash-prediction model were Principal Component Analysis (PCA) and Logistic Regression (LR). The early result of this study displayed the association between turbulence and rainfall index with hazard levels for freeways. The higher the turbulence and rainfall index, the higher was the potential for crashes to happen [Abdel06]. However, implementation details such as the warning time and response time

required, and the right timing of warning messages so that crash can be avoided were not specified in [Abdel06].

Based on the databases accumulated by the U. S. National Highway Traffic Safety Administration (NHTSA), data mining tools have been applied to find patterns in drivers and vehicles that contribute to highway crashes [Singh03]. Driver characteristics that were analysed included driver attributes such as age and gender and driver-related crash attributes such as involvement of alcohol, distraction, speeding, wrong manoeuvres, and corrective action. Vehicle characteristics that were examined include vehicle body type, vehicle stability, vehicle path, and vehicle contributing factors such as steering, brakes, suspension, power train system, and wheels [Singh03]. These characteristics were examined with a data mining technique, Principal Component Analysis (PCA). PCA is able to deal with a large number of correlated variables. The study used PCA to compare sets of crash variables to produce maximum discrimination among groups (of drivers derived from age and gender, and of vehicles derived from body type) with regards to the original crash variables. There were a number of interesting correlations derived from the study, such as that the involvement of teenage drivers in highway crashes were highly related to speeding, while young drivers were highly related to drinking [Singh03]. Those kinds of findings are useful for further study. For example, young drivers should be the focus in monitoring for speed limit violation.

Data mining can also be used to improve traffic performance and throughput. In [Zhang05], data mining is used to detect incidents that can cause delay traffic. The data mining implementation in [Nakata04] is intended for travel time prediction. These projects are discussed next.

Congestion problems often occur in highways because of the presence of incidents [Zhang05]. The TSC Algorithm is a Bayesian Network (BN) model for freeway incident detection [Zhang05]. The BN includes two traffic events, which are incident and congestion, and seven traffic variables, which are traffic volume upstream and downstream, speed upstream and downstream, occupancies upstream and downstream, and occupancy difference between upstream and downstream. Expert knowledge of incident and incident-free traffic patterns are saved in the Conditional Probability Tables (CPT) of the BN. For testing, the CPT was firstly initialized with general traffic patterns, and then adjusted by data generated from a traffic simulation, since comprehensive real incident data were difficult to find. The algorithm was then adapted to different freeways by modifying the BN. There are two ways to modify the BN. Firstly, by updating entries of CPT by an expert, for example, by modifying the thresholds of traffic parameters, such as the lane volume. Secondly, by adapting incident data to the CPTs. However, only high quality data can be adapted, as noise in data can eliminate the generality of the knowledge base. The algorithm was evaluated by measuring the detection rate and false alarm rate. The algorithm was said to be effective as it had a high detection rate and a low false alarm rate in the evaluation, even when the algorithm was adapted to various freeway situations.

Travel time prediction has been implemented based on real-time data from probe cars (i.e., moving vehicles that are used to collect actual traffic information) in [Nakata04]. This approach aims to improve the usual way of travel time prediction by using a predefined travel timetable. The travel time data collected by each trip is regarded as a time series data. For the purpose of time series modelling, Auto Regression (AR) model and state space models were used. AR models used spatial and temporal data from locality to execute prediction at a certain location. State space models were used to characterize various time series models and deal with non-stationary time series data comprising AR models,

seasonal components, and trend components [Nakata04]. As a result, the AR model that was used with travel timetable has a much higher accuracy than a state space model, and the AR model was said to be more effective than the usage of timetable alone. However, the stability and reliability of data from probe cars were questionable as results from two different models vary greatly [Nakata04].

Data mining is proven to be effective for extracting patterns and trends in traffic. Data mining has been effectively used for extracting useful knowledge from persistent or stored data. The advances in sensor technology have resulted in very large amount of sensor data being generated making it infeasible for storage and consequent processing. Sensor data needs to typically be analysed, understood, and applied in real-time.

According to [Hsu02], Ubiquitous Data Stream Mining is one of the current trends in data mining. As stated by Gaber [Gaber04a], Ubiquitous Data Mining (UDM) is the analysis of data streams to discover useful knowledge such as patterns and association rules on mobile, embedded, and ubiquitous devices. There are only a limited number of research projects that have applied ubiquitous data mining for traffic and transportation systems, which include monitoring of drunk-driving behaviours [Horo06], vehicle health monitoring, and driving pattern recognition [Kargup04].

The Vehicle Data Stream Mining System (VEDAS) was proposed for analyzing onboard streams of vehicle data [Kargup04]. Data from sensors in moving vehicles was analysed in real-time for monitoring vehicle's health and recognising driving patterns. VEDAS used Principal Component Analysis (PCA), Fourier transformation, and online linear transformations to perform onboard pre-processing of sensor data by decreasing the dimensionality of data. The onboard system was connected to a central server that performs the following functions:

visualizations for global and local models, central controllers for onboard data mining operations, an event management service to notify users of unusual events, and map retrievals by connecting to a Geographical Information System. VEDAS only implemented online unsupervised learning, or clustering, for detecting new patterns. It has not applied any supervised and predictive learning techniques. In terms of performance, supervised learning/classification was a better approach as it performed better in detecting new or unusual patterns in real-time [Kargup04]. The basic models to be used for detecting unusual events can be developed offline. The models learnt offline can then be used as a basis for classifying new events.

Algorithms to identify drunk driving behaviours in real-time were proposed in [Horo06]. Two stages of ubiquitous data mining were applied, which is data synopsis, or clustering, and classification of driver behaviours. The clustering process used Lightweight Clustering (LWC) algorithm, introduced by Gaber [Gaber04b]. The major challenge was in linking the results of clustering models with the existing expert knowledge in the road safety field. Therefore, a fuzzy logic approach was implemented for labelling of clusters and determining probabilistic degree of membership of each driver to a particular behaviour group. Sensor data for the evaluation was generated using simulation based on an expert study, which categorised drunk driving behaviour into sober, borderline, drunk, and very drunk. The result of an online clustering and offline labelling was three clusters of drunk driving behaviours', from least drunk to most drunk, although originally the generated data is sourced from four different categories. This was because of very little distinction between the borderline and the drunk category. The offline labelled model was then used in real-time to classify data into one of the three clusters. The classification rules consisted of the following variables: number of correct responses, number of collisions, time over speed limit, reaction time, speed deviation, and lane deviation [Horo06]. The last step

of the process was to determine the degree of membership to each drink driving behavioural cluster. This LWC and fuzzy logic approach has been effective in identifying patterns in one-dimensional numerical data.

SAWUR (Situation-Awareness With Ubiquitous data mining for Road safety) is an ADAS that is based on Ubiquitous Data Mining and Context Awareness [Krish05]. SAWUR integrates contextual information of three main components of driving situations: the driver, the vehicle, and the environment. Driver behaviours and profiles, vehicle dynamics, and environmental situations are continually analysed on real-time for threat detection and effective delivery of warning. SAWUR contains an onboard system and a central server. The central server has a historical database, on which data is mined to build event classification models that is kept in the server and also injected into the onboard system. The onboard system uses pre-built models for event detection based on classification algorithms. When a potential threat is detected, the event is sent to a black box recorder that registers threats and also to an action or communication module that responds to the event either by issuing warnings to drivers or sending a message to other vehicles. The data that are kept in the black box recorder are sent periodically to the central server in order to update the database. An online data synopsis, created using a Lightweight Clustering (LWC) algorithm introduced in [Gaber04b] is used for building the online clustering models of driving behaviours. This approach eliminates the needs for frequent transmission of huge amounts of data [Krish05].

It is clearly shown that machine learning and data mining on ITS have made substantial contributions to improving safety on the road by finding new patterns that add to previous knowledge, such as characteristics of accidents or dangerous driving behaviours.

2.3.4 Discussion

We have reviewed the existing research in Intelligent Transportation Systems (ITS) that utilise pervasive computing techniques, such as knowledge-based systems, data mining, and context awareness, for improving safety, efficiency, and autonomy. Such pervasive computing techniques have been applied in ITS in order to accommodate cooperative and adaptive distributed situation awareness in a road environment. Road safety applications that have used knowledge-based systems, data mining, or context awareness has improved the analysis stage of the Road Safety Analyses (RSA) as the ability to learn and analyse data based on a given situation is incorporated. Therefore, it is necessary to incorporate these techniques to an intersection collision avoidance system.

None of the context-aware applications in the area of ITS has addressed the issue of intersection safety. The incorporation of context-awareness techniques facilitates adaptability of an application to the given situation. In other words, an application becomes aware of its situations/surroundings. Most existing context-aware projects use predefined context models. In a context-aware application, an event with a certain condition must be responded to in a certain fixed way. This event-condition-action is most likely predefined in the system. However, it is quite possible that a system changes after a period of time, or it must be adapted to a new environment. Hence, integrating learning into context-aware applications will be useful in order to discover new and interesting situations or additional contextual information that a safety system can employ. Data mining can be used to discover new contexts, which can be stored in a knowledge base for better decision making.

Knowledge-based systems in ITS have also been reviewed. Most of the existing knowledge-based systems only deal with road and intersection site maintenance

issues. None of the existing knowledge-based systems have dealt with the issue of intersection collisions and collision avoidance. Since each intersection has different and varying characteristics, it is necessary to have an intersection collision avoidance system that possesses a knowledge-base that contains knowledge and rules that are specific to a particular intersection, but can remain generic at the application level, with the intention that the same collision avoidance system can be adapted to other intersections with different contents in the knowledge base. Unfortunately, the knowledge acquisition in the existing systems still relies on traditional methods of manual observation and raw data collection and analysis. Furthermore, existing knowledge-based systems employ a static knowledge base, hence, no new rules or information can be added to it. Due to the varying conditions of an intersection, it is essential that those changes should be recorded in a dynamic knowledge base of an intersection safety system. Data mining can assist in adaptation of a dynamic knowledge base and automating the task of acquiring knowledge (which is to be stored in a knowledge base of any ITS system).

Data mining (including machine learning) enables new, interesting, and useful patterns to be extracted from data. The combination of variety and alternating attributes of intersection types can lead to various collision patterns. In order to increase situational awareness of an intersection collision avoidance system, it is necessary to learn and extract collision patterns in each intersection. None of the existing ITS applications that utilise data mining have addressed the need for collision pattern learning at intersections. Furthermore, learning of dangerous driver behaviour in an intersection is essential. The project that utilises data mining for driver behaviour learning applies it only for dilemma zone and red light running violation behaviours [Chan04]. Therefore, in order to have a holistic, automated, and adaptive intersection collision avoidance system, it is necessary to incorporate data mining techniques.

Given the ubiquity of small computing and mobile devices today, it is essential to explore the possibility of processing data in such devices. Nevertheless, mining real-time data on a resource-constrained mobile device is not possible due to the high cost of processing power of traditional data mining techniques. Also, transferring real-time data back and forth to a central server for further processing is not appropriate because of the high communication costs involved (sensor data that are accumulated daily may reach the size of hundreds of Megabytes, e.g. Pantheon Gateway Project [Gross05]). The information delivered to the systems will be from a myriad of sensors that continuously and rapidly stream data to the systems. Given this context, it is evident that Ubiquitous Data Mining (UDM) technique is a suitable option and one that can facilitate incremental learning. UDM does not merely correspond to applying data mining algorithms on resource-constrained devices, but focuses on dealing with the requirements of ubiquitous devices, such as providing time-critical data analysis in a mobile context [Krish05]. UDM is very appropriate for analysing data streams anywhere, anytime, in a resource-constrained device. The ITS projects that apply UDM for discovering new knowledge from streams of data have also been reviewed. As for processing efficiency in mobile and resource-constrained devices, UDM has shown its usefulness when compared to traditional data mining and machine learning techniques. However, as this is a new research area in knowledge discovery, there are only a few number of UDM algorithms when compared with machine learning and traditional data mining techniques. There are a number of challenges in applying UDM on ITS applications, which are as follows:

- The absence of contextual models of road environments that can help better situation recognition.
- The current shortage of real-world data (though sensors can be massively deployed to obtain this), which is preferred than simulation generated data, as real-world data is more comprehensive and accurate.

- Although ITS applications must scale to a great number of vehicles and infrastructures, the availability and bandwidth power of wireless networks are limited and the cost of data transmission is high. Therefore, the amount of data to be transferred should be reduced by transferring only processed data in forms of simple patterns or models, rather than transferring all the data [Horo06], [Kargup04].

Table 2.3 summarizes what has been done and our discussions in the earlier sections, in the area of improving efficiency, safety and autonomy in ITS, where knowledge-based systems, context-awareness, and data mining have played their parts.

From the review, we contend that work in the disparate fields of knowledge-based systems, context-aware computing, and data mining can complement each other in dealing with ubiquitous computing environments with mobile entities. The integration of all the paradigms (and their related technologies) is a powerful combination to achieve the purpose of efficiency, autonomy, and safety in road transportation systems. We deduce that context awareness is useful when traffic or driver conditions must be known or monitored. Data mining is a powerful complement to both knowledge-based systems and context awareness notion, as it improves the knowledge of the overall system by analysing historical data or data stream to achieve its purpose. It is also evident from the review that knowledge base systems, context awareness, and data mining are useful particularly for collision pattern analysis, collision detection, and collision warning for intersection safety applications, which are the focus of this thesis. However, many existing intersection collision warning and avoidance systems, which are discussed in Section 2.4, do not utilise these techniques. As a result, many of these systems are limited in many ways. These are discussed in the next section.

Table 2.3. Application of Knowledge Base, Context Awareness, and Data Mining in Various ITS areas

Project	Knowledge Base	Context Awareness	Data Mining	Application Areas
[Harr04]		✓		Traffic route profiling
[Chan04]			✓	Traffic monitoring
[Zhang05]			✓	Traffic optimization and control
Pantheon Gateway Project [Gross05]			✓	Incident detection
[Nakata04]			✓	Travel time prediction
CORTEXT [Veris02]		✓		Automated driving
[Mitch97]			✓	Automated driving
[Moriar98]			✓	Automated driving
[Weev03]			✓	Driving instructor
[Gruyer05]		✓		Vigilance monitoring
VEDAS [Kargup04]			✓	Vehicle, driver, and environment monitoring
[Gruyer05]		✓		Vehicle, driver, and environment monitoring
[Vidal02]		✓		Environment monitoring
[Oliver00]		✓	✓	Driving manoeuvre learning and prediction
[Mitro05]			✓	Driving manoeuvre learning and prediction
[Chan04]			✓	Driving behaviour learning in intersection
DAISY [Onken94]	✓			Collision Detection and Warning
[Horo06]			✓	Drunk driving monitoring
SICAS [Boury00]	✓			Road Site Analysis
RSIT [Kwas07]	✓			Road Site Analysis
[Chass05]	✓			Road maintenance
ATMIS [Abdel06]			✓	Post-collision data analysis
[Singh03]			✓	Post-collision data analysis
[Chong04]			✓	Post-collision data learning to predict severity of injuries

2.4. Post Analysis

Due to the fatality rates of intersection crashes, it is necessary to develop an intersection safety system that can assist drivers to navigate through intersections well. Once analysis of crashes are performed, a suitable intersection collision

warning and avoidance systems that can provide mitigation strategies for potential crashes needs to be developed. The following are the features and various intersection data that are desirable for intersection collision warning and avoidance systems [Stubbs03]:

- must incorporate and coordinate temporal traffic information from a variety of sensors;
- must process this information, identify collisions or near-misses, and deliver countermeasures in real-time;
- should give an explanation for different trajectories of the vehicles;
- must elucidate various vehicle speeds and acceleration/deceleration in the area of the intersection;
- should be able to adapt to different traffic volumes and average traffic speeds (e.g. at urban / suburban intersections, there are larger numbers of vehicles and they move relatively slower when compared to when they are at rural intersections);
- ought to differentiate different types of vehicles (e.g., buses are longer than cars, hence, they make wider and slower turns, consequently, bigger risk of collision);
- should comprise pedestrians and cyclists crossing at the intersection;
- must have effective means for issuing countermeasures;
- must consider traffic signals and vehicles' signals;
- must consider the shape of the intersection;
- must consider external factors such as weather conditions;
- must consider driver distraction issue in relation to countermeasures issued.

The above list requires information from sensors on the road and in-vehicle to be processed and analysed to determine comprehensive knowledge about the intersection and its locality. There have been a number of initiatives in developing intersection collision warning systems and/or avoidance systems.

Currently, no existing intersection collision warning and avoidance systems can tackle intersection collision problems entirely.

Intersection collision warning and avoidance systems can be categorized as either *vehicle-based*, *infrastructure-only* or as *infrastructure-vehicle cooperative* [Ferlis01].

- *Infrastructure-only systems* utilise roadside sensors, warning devices, vehicle-to-infrastructure communication, other roadside informational or warning devices, and traffic signals to provide driving assistance to road users [Ferlis01]. Infrastructure-only systems rely only on roadside warning devices to inform drivers.
- *Vehicle-based systems* rely only on in-vehicle sensors, processors, and interface to detect threats and produce warnings. There is no communication means existed in these systems.
- *Cooperative systems* communicate information straight to vehicles and drivers. The main advantage of cooperative systems rests in their potential to improve the interface to the driver, and thus to almost guarantee that a warning is received. Another potential of such a system is that it can apply control over the vehicle, at least in situations where the system can be recognised as trustworthy and the driver cannot be expected to take appropriate actions given the imminent danger and short response time. Cooperative systems include vehicle to vehicle communication and also infrastructure to vehicle communication.

2.4.1 Infrastructure-Only Systems

The U.S. Department of Transportation has successfully deployed an Intersection Collision Warning System (ICWS) for unsignalised intersections [USDOT99]. This system has demonstrated a significant impact on driver's behaviour that it

has reduced the degree of collision risk at the intersection installed with ICWS. The effectiveness of ICWS installed on unsignalised intersections is measured by: (1) the increase of sign response speed, (2) acceptable intersection arrival speed, (3) first speed reduction, (4) second speed reduction, (5) overall speed reduction, and (6) the increase of Projected Times to Collision (PTC) which allows accident-avoidance manoeuvres [USDOT99]. In spite of the effectiveness of the system, sensors and visual warnings are given only from roadside infrastructure, which does not guarantee that warnings are attended by approaching drivers.

The Intelligent Vehicle Initiative of the U.S. Department of Transportation proposed initial concepts for intersection collision avoidance systems [Ferlis01], which include: traffic signal violation warning, stop sign violation warning, traffic signal left turn assistance, and stop sign movement assistance. They aimed to install sensors on the roadside to detect speed, acceleration rate, deceleration rate, stopping, and movement of each vehicle approaching the intersection from all directions. Warnings are issued to drivers as violations or potential conflicts are detected. Warnings are given by: (1) activating warning lights to notify a need for caution and possibly to point out the source of the conflict; (2) activating intelligent rumble strips to notify the other motorist to slow down and advance carefully at the intersection; (3) using a Variable Message Sign (VMS) or graphic display sign to notify drivers of the potential conflict with the signal violator. These systems are still categorised into infrastructure-only systems, because there is no direct infrastructure-to-vehicle communication. As warning messages are given from the roadside, warning can be distractive and less effective.

Three different Intersection Decision Support (IDS) systems, which have been installed in three different U. S. states in June 2003, are being tested for acceptance [Funder04]. These systems are designed to significantly reduce the

number of intersection crashes in the intelligent intersection sites. However, this is yet another infrastructure version of intersection collision warning system, as warnings are issued by illuminating a LED stop sign (Figure 2.2 [Funder04]) and activating an intelligent rumble strip [Chan03], [Werner03], [Funder04]. In the near future, vehicles will also be equipped to receive intersection collision warnings from a driver-interface with the development of in-vehicle sensor and communication technologies [Funder04].



Figure 2.2. PATH's IDS Uses Illuminated Stop Sign [Funder04]

A vision-based sensing system for monitoring an intersection and predicting vehicle collisions is currently under development [Stubbs03], [Veera02]. It uses a single camera arbitrarily positioned at an intersection to observe the traffic flows. The system classifies moving objects (such as vehicles and pedestrians), tracks each of their movements [Veera02] and collects traffic data such as vehicle speeds, positions, routes, accelerations/decelerations, vehicle sizes, and signal status [Veera02]. The proposed system is also able to compute promptly the potential collisions and near-misses by applying algorithms that analyse the speeds and routes of the moving objects being studied. However, the proposed

system still has unsolved issues, such as obstructions of the image (such as trees) and shadows in the vehicle image. In addition, vehicles that are not moving cannot be tracked by this vision based system [Stubbs03].

In summary, existing intersection collision warning systems are still infrastructure-only systems, and are limited in certain aspects [Salim05], such as:

- In delivery of a warning message, it is distractive and less effective as warnings are only displayed on the roadside. There is no guarantee that a message is received by the intended recipient. Warning displays from the roadside may also somehow distract other drivers who do not need to receive the message (see Figure 2.1 and Figure 2.2).
- There is no means for communication that exists between road infrastructure and vehicles, and therefore, there is no exchange of useful information between them.
- Information about the intersection might not be comprehensive as the only data source is roadside sensors.
- The systems are mostly reactive. Reactive behaviour is required for such a real-time solution; however, deliberative reasoning can supplement and enhance these systems.
- Each system is built for a particular intersection and cannot be generalised for other types of intersections, and therefore, each application requires a field study on that intersection. As previously discussed, this is due to the different characteristics of each intersection that requires a different treatment for its safety.

The next section discusses research in the vehicle-based intersection collision warning and/or avoidance systems.

2.4.2 Vehicle-based Intersection Collision Warning and Avoidance Systems

Safety countermeasures for a single car have been developed by the National Highway Traffic Safety Administration (part of U.S. Department of Transportation) [Verid00] to cope with four different cross intersection crash scenarios. Nonetheless, the system does not include any means of communication between the infrastructure and the vehicle. However, communication between infrastructure and vehicles is being considered for implementation as it will improve the effectiveness of a vehicle-based intersection collision warning system.

Most ongoing research for in-vehicle collision warning systems (e.g. forward collision, rear-end collision, and side collision warning system) enable the system to work in all road types, either in rural or urban areas, on highways or small streets, and also at intersections. These vehicle-based collision warning systems are fairly effective for a single vehicle. However, in an intersection, a potential danger normally impacts more than one vehicle and it is important that other possible affected vehicles are also warned about the impending collisions.

A multiagent system for intersection collision warning system has been proposed by Miller and Huang [Miller02]. According to Wooldridge, an agent is autonomous intelligent program acting on behalf of the user [Woold02]. A societal group of intelligent agents, which are interacting with each other for common goals, can therefore be regarded as a multiagent system. Multiagent technology is very fitting for coordination of entities on the road. The abstraction of independent and autonomous entities, which are able to communicate with other entities and make independent decisions, maps suitably to the situation of an on-road scenario, where each entity, such as a vehicle, or a traffic light, can be

represented by an intelligent agent [Salim05]. This project uses a peer-to-peer concept where information and messages are communicated between a pair of vehicles. Threat detection relies on location, velocity, and acceleration information shared by other vehicles that use the system. Their proposed collision detection algorithms consist of algorithm to detect a potential collision point, detect collision time, and issue timely warnings. However, their collision detection algorithm is based on the conventional speed formula, and it requires calculation for each possible pair of vehicles at the intersection. In addition, since it is a peer-to-peer vehicle based collision warning and avoidance system, each vehicle needs to know the status of every other vehicle. Thus, each time the vehicle moves, all other vehicles should be informed. This incurs high communication cost. The algorithms are further discussed in Chapter 5. It is also suggested that based on the available time to reach the predicted collision point, either a collision warning message is sent to the driver or a command message is directly issued to the vehicle. A multiagent system approach is implemented for the intersection collision warning system. Each vehicle has a multi-agent-based software architecture and hardware architecture installed to detect potential dangers. The software architecture consists of three layers [Miller02]:

- sensory agents (i.e. Global Positioning System agent, brake sensor agent);
- decision / control agents (i.e. collision warning system agent);
- presentation agents (e.g. speaker agents) that deliver warnings to driver.

However, this system has some limitations [Salim07a]. Firstly, the agent architecture is reactive, without learning new knowledge, such as driving behaviours and crash patterns of the intersection, which can enhance the system to react better. Secondly, the algorithm for collision prediction used in this project is not efficient since every possible pair at the intersection is required to be computed for collision prediction. Finally, useful information about the infrastructure and environment are not incorporated here as there is no communication between infrastructure and vehicles, and between vehicles and

external parties. This makes this system categorised into vehicle-based collision system.

Both infrastructure to vehicle communication (and vice versa) and vehicle to vehicle communication are desirable in a collision warning and avoidance system. Therefore, there have been new initiatives to develop cooperative intersection collision warning systems, as discussed in the next section.

2.4.3 Cooperative Intersection Collision Warning and Avoidance Systems

Research projects on cooperative intersection warning and/or collision avoidance systems have been initiated to improve intersection safety. One of the recent initiatives for developing cooperative safety system for intersection is the Cooperative Intersection Collision Avoidance Systems (CICAS) [USDOT07] by the U. S. Department of Transportation, which seeks to develop vehicle-based systems, infrastructure-only systems, and finally, infrastructure-vehicle cooperative systems. Vehicle-based systems include sensors, processors and interfaces for driver inside each vehicle. Infrastructure-only systems depend on roadside sensors and processors to identify vehicles and threats and then generate signals through messaging signs to warn motorists of potential collisions. Infrastructure-only operations typically necessitate data processing techniques, an essential evolutionary move towards deployment of subsequent cooperative systems. Infrastructure-vehicle cooperative systems use infrastructure-only systems, and also utilise a communications system, i.e. Dedicated Short Range Communications (DSRC) to exchange warnings and data directly with drivers in vehicles capable of accepting and displaying the warnings within the vehicle. It has been stated that data processing and analysis techniques are required to assess situations in such contexts [USDOT07].

The other initiative in progress is based in Europe, which is the INTERSAFE project, part of the Integrated Project PReVENT [Fuers05], which aims to develop an intersection safety system to improve safety and reduce collisions. The INTERSAFE project develops an onboard system that utilises a combination of sensors to identify crossing traffic and all other objects on the intersection, in addition to sensors for locating the host vehicle position in the adjacent intersection. A communication transmitter between the host vehicle and the infrastructure is used to exchange additional information such as weather, traffic, and road conditions [Fuers05]. The INTERSAFE project employs two different methods in parallel [Fuers05]. The first method is to develop the Basic Intersection Safety System that is implemented on a Volkswagen test vehicle with two laser scanners for object detection, one video camera for road marking detection and vehicle-to-infrastructure communication. Communication units are to be installed at selected intersections to enable communication between the vehicle and traffic lights. A static world model is constructed from object detection, road marking detection, landmark navigation, GPS, and map [Fuers05]. The second method is to develop an Advanced Intersection Safety System that is implemented on a BMW driving simulator. This driving simulator examines dangerous states beyond the limitation of sensors in detecting the environment. In this second method, a dynamic risk assessment is executed based on object tracking and classification, communication with traffic management, and driver intention. Hence, potential threats to other road users and conflicts with traffic controls can be detected. As a result, the system by INTERSAFE is able to provide stop sign assistance, traffic light assistance, turning assistance, and right of way assistance [Fuers05]. INTERSAFE also identifies the need for analysing the situation and collision risks at an intersection, but specifics of how to learn and what techniques are appropriate have not been investigated or addressed.

There is also ongoing research to develop Cooperative Adaptive Cruise Control (CACC), which will use inter-vehicle communication [Bruin04] in order to maintain a safe distance from one another. CACC is the improvement from the Adaptive Cruise Control (ACC), as has been mentioned in Section 1.1. However, ACC, which is used to maintain a steady forward vehicle speed when driving in sparsely populated roads, is only intended for a single vehicle. CACC that utilises inter-vehicle communication allows vehicle to cooperate together, so that collision can be avoided. Moreover, a traffic flow study involving CACC has suggested that CACC will improve traffic flow performance [Arem06].

In summary, research initiatives in developing cooperative systems for intersection safety such as INTERSAFE [Fuers05] and Cooperative Intersection Collision Avoidance Systems (CICAS) [USDOT07] have recently commenced. To our knowledge, these projects are still in their early stages, and do not mention techniques to discover crash patterns and pre-crash behaviour associations, which are essential to detecting and reacting to potential threats. A generic framework that can automatically adapt to different intersections is required for efficient deployment; however, these projects have not addressed this issue [Salim05].

Existing intersection collision warning and avoidance systems, including the infrastructure-only and vehicle-based systems are still limited in many ways, which are restated as follows:

- The systems are mostly built to suit a particular intersection and therefore lack the capabilities to adapt to different types of intersections and to detect various collisions. Furthermore, while the systems can react to potential threats or collisions, however, there is no learning from past history, experiences, or traffic data for better situational awareness.

- The data sources that are used in existing intersection collision warning and avoidance systems to feed and trigger the collision detection comes only from one type of source, which is either roadside sensors or in-vehicle sensors, not both. Both roadside and vehicle sources should be considered to help broaden the view and information about the intersection, vehicles and drivers that pass through the intersection. It is necessary to choose the most appropriate sensors that can provide real-time data source for real-time collision detection and warning.
- The performance and scalability of the existing system are questionable [Miller02] since the collision detection computation requires every possible pair at the intersection to be calculated for the possibility of collision detection.
- There are no existing communication means used between road infrastructures and vehicles (as the cooperative systems are still under development). Hence, no existing real-time communication protocols established between vehicles and road infrastructures. Such communication means are useful for status update and warning messages.
- Effective and contextual means of delivering warning based on the available time before collision are required, but only [Miller02] presents such an argument.

Thus the issues of adaptability, learning, leveraging multiple data sources, and real-time communication are the major challenges that need to be considered in designing intersection collision warning and avoidance system. These are elements that constitute the desirable properties of an intersection collision warning and avoidance systems. We discuss the necessity of those elements in the next section.

2.5. The need for a Real-Time, Generic, Adaptive, and Cooperative Intersection Safety Framework

When it comes to modelling an intersection safety framework that can be used as a basis for developing an intersection collision warning and avoidance system, there are a number of issues and challenges to consider and draw our assumptions from: consideration of variety of data sources, performance and scalability, the issue of adaptability and learning, formalising/specifying communication cost models, and relationships between collision detection and warning. Pervasive computing techniques, as discussed in Section 2.3, have the potential to meet the desired properties. These are discussed in the following subsections with our recommendations on each issue.

2.5.1 Consideration of a Variety of Real-Time Sensor Data Sources

The information required in real-time to detect a collision based on the conventional speed formula are speed, vehicle size, travel direction, current position, and angle. Additionally, the vehicle registration number is required to identify each dataset uniquely. Vehicle manoeuvre data is also necessary for faster collision detection calculation (this is further explained in Chapter 5). Those data can be retrieved either from sensors on the road or in vehicles (See Table 2.2). All those data required to calculate possibility of collision are about individual vehicles. Therefore, it is important to decide whether those data are collected using roadside sensors or vehicle (on-board) sensors, and whether vehicle-to-infrastructure or vehicle-to-vehicle communication is required to communicate those sensor data to the location where collision detection algorithm is computed and warning messages are to be generated.

As stated in 2.4, the existing intersection collision warning and avoidance systems merely employ data from roadside sensors. The existing vehicle-based intersection collision warning and avoidance systems do not employ cooperative coordination between vehicle and infrastructure components. This causes a number of issues, such as partial information for collision warning and avoidance purposes and non real-time data sources. Hence, there is a need for real-time data sources to be supplied from both vehicle and roadside sensors.

The assumption on data source and availability leads us to ponder on where the collision detection should be performed. We discuss this further in the next subsection that analyse the location of computation and efficiency of the collision detection algorithm in terms of performance and scalability.

2.5.2 Performance and Scalability of Collision Detection

The performance of the collision detection algorithm should be optimised in order to achieve real-time collision warning and avoidance. Moreover, due to the growing number of vehicles that use an intersection, the collision detection algorithm should also be scalable, in order to accommodate all the vehicles in the vicinity. There are few facets to this issue: firstly, where the collision detection should be computed; secondly, how are we going to compute the collision detection.

The choice of the location of computation should not only be based on information availability, but also the efficiency of the method. In general, there are two ways of calculating collision detection based on the location of computations:

- centralised, where calculations are done at the intersection's vicinity (the algorithm running on some situated stationary computer server);

- distributed, where computations are done in each vehicle.

When the centralised approach is adopted and no information is required from vehicles (by simply relying on roadside sensors), the only communication is sending warning messages to relevant vehicles from the central intersection server. However, if centralised approach is adopted and certain information from vehicles is required, communication must be established in a robust way. Information from the car, such as current position and angle, must always be transmitted to the central server every few milliseconds for accurate collision detection.

When the computation is done locally on each vehicle (distributed approach), the intersection collision warning and avoidance system should be installed on every vehicle at the intersection. The system should rely on availability of in-vehicle sensor data retrieved from the each vehicle, and each local system should notify each other of its existence and status overtime as each system should identify and communicate with every other vehicle in the vicinity (peer-to-peer communication), such as the case in the Miller and Huang's peer-to-peer collision detection system. If the number of vehicle at the intersection grows, each vehicle should notify all other vehicles at the intersection of its status change, and each vehicle should also process a number of different status messages from different vehicles in every split-second. Sending and deciphering multiple messages on a small computer system in a vehicle can cause a performance bottleneck. On the other hand, when data from roadside sensors are necessary, data from those roadside sensors must be sent to all vehicles at the intersection so that the most recent data are captured for collision detection calculation in each vehicle. The transmission of such messages from roadside sensors to passing vehicles is not trivial.

Therefore, it is necessary to consider where computations and learning are going to be performed. Note that it is essential to have a bird's eye view of the whole intersection at a time so that the status of all vehicles at the intersection can be known and incoming collisions can be foreseen.

Apart from the location of computation, performance issues also arise from how the collision detection is computed. Conventionally, the method of collision detection computation is pair-wise [Miller02], as discussed in Section 2.3.4. Every possible pair of vehicles at the intersection should be computed for collision possibility. Therefore, we need to consider which pairs of vehicles the algorithm should be applied to; otherwise, it would be applied to each vehicle pair at the intersection. In our view, there are two approaches for choosing which vehicles in the vicinity the collision detection algorithm should be applied to:

- brute force: perform collision detection for each car, between the car and with every other car at the intersection;
- preselection: perform collision detection only for the cars that have the possibility of collisions based on the known intersection collision patterns.

In order to reduce computational time of collision detection, it is important to reduce the number of vehicle pairs to be calculated for collision possibility. Consider a four leg intersection with 60 vehicles in the vicinity: 10 in the left leg, 25 in the right leg, 10 in the upper leg, and 15 in the bottom leg. Each leg has 6 lanes; 3 for vehicles approaching the intersection; another 3 for outgoing vehicles. Collision detection is performed for the vehicle that is currently in the left leg and doing a right turn. With the brute force approach, calculation of possible conflicting pairs must be done 59 times for every other car within each calculation period. With the preselection criteria, the only colliding possibilities for the current vehicle are with vehicles in the bottom leg that are approaching the intersection with a straight manoeuvre, and only 10 vehicles satisfy such

criteria; thus, the calculation will be performed only 10 times within each calculation period. The existing intersection collision warning and avoidance systems employ brute force approach, which is not scalable when the number of vehicles at the intersection increases. Therefore, it is necessary to reduce computational time through well-established heuristics. Those heuristics can be stored in the knowledge base of the system for an intersection collision warning and avoidance system that is adaptive and able to learn dynamically. This is further discussed in the next section 2.5.3.

2.5.3 Adaptability and Learning

Currently, collision warning systems that have been installed are mostly reactive as they only focus on responding to events of collision detection, as illustrated in Figure 2.3. Data from each vehicle is used to calculate collision point with data from every other vehicle at the intersection. When the point of collision found, then time to collision is calculated. And if a collision is predicted, warning is issued, and then it does not take into consideration whether or not the predicted collision has actually happened. The information is discarded straightaway. This is mainly because the conventional collision detection algorithms [Miller02] are merely reactive.

Nevertheless, it is also important for intersection collision warning and avoidance systems to be deliberative as well as reactive. This is because there is a need for systems that are generic (i.e. applicable to various types of intersections) and adaptive (i.e. capable of making adjustments to specific traits and patterns of collisions in a particular intersection). When an intersection collision warning and avoidance system has learning capabilities, it becomes generic and adaptive because the overall system is generic but the results of learning are knowledge specific to the particular intersection. Learning also helps to improve the

performance of collision detection. This is because the collision patterns learnt from the historical and real-time collision data are stored in the knowledge base, which is to be used for the basis of the preselection approach. In addition, ad-hoc changes at the intersection environment can also be learnt to further knowledge and awareness of the system about the intersection.

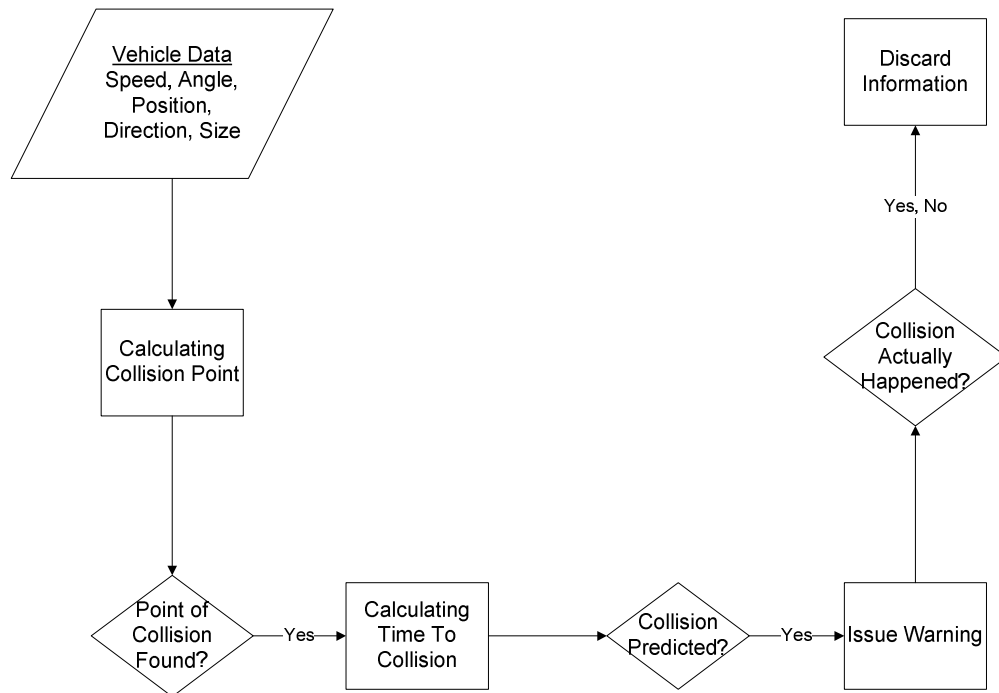


Figure 2.3. Reactive Intersection Safety System

Therefore, a learning component should be included as part of an intersection collision warning and avoidance system (Figure 2.4). By learning from historical collision and real-time data, automatic adaptations, better detection, and improved reactive behaviour can be achieved. None of the existing intersection collision warning and avoidance systems have incorporated learning for generic and adaptive collision detection, warning, and avoidance at various and varying intersections.

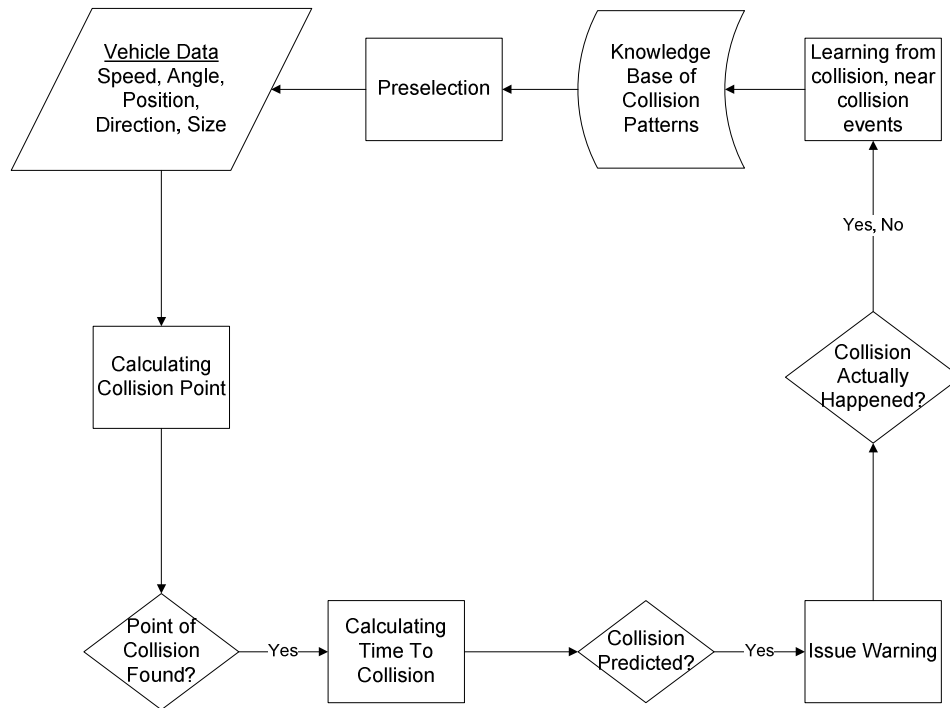


Figure 2.4. Reactive & Deliberative Intersection Safety System

Apart from reliable collision learning and detection, an effective collision warning component is required. When a collision is detected and warning needs to be issued, we need to consider a timely warning. We need to avoid issuing warnings too frequently such that it becomes an annoyance to the drivers. Conversely, we need to issue warning in time before a potential collision actually takes place. Therefore, the relationship between collision detection is an important factor to be considered as well in developing an intersection collision warning and avoidance system.

2.5.4 Relationship between Collision Detection and Warning

Collision detection misses should be avoided; however the number of alarms should also be regulated, as too many false alarms can cause a nuisance to the driver [Horst93]. If a collision is detected, it is important to realise whether the time and distance needed to respond to the warning and avoid the collision is sufficient [Miller02].

In general, there are two main temporal dimensions that need to be considered in collision warning, which is composed of Time-To-Collision (TTC) and Time-To-Avoidance (TTA). TTC is the remaining time predicted before a vehicle reaches the predicted collision point. TTA is the time available to avoid a collision which includes time to issue warning, human reaction time, and vehicle response time.

Time-To-Avoidance (TTA) in Miller and Huang's peer-to-peer collision warning system [Miller02] is computed based on vehicle kinetics, network latency, and human response time. If Time-To-Collision (TTC) is much greater than TTA, a warning is not issued. However, if TTC is close to TTA and driver is not braking, then warning is issued. Otherwise, if TTC is less than TTA, a mitigation unit is executed to lessen the collision effect [Miller02]. The best timing to warn drivers vary based on driver's skills and experience, therefore they also proposed a parameter γ for tweaking the timing of effective warning [Miller02].

$$TTC - TTA < \gamma \quad (2.1)$$

When γ is large, the algorithm will be more conservative. When it is too conservative, it can be a distraction rather than assistance to a driver. Thus γ must be adjusted well based on the best probable driver experience. However, according to Horst and Hogema [Horst93], the best time to warn driver using collision avoidance systems is when TTC is equal to 4 seconds. Nevertheless, when there is a fog that reduces visibility range to between 40 and 120 m, the most appropriate time for activating collision avoidance systems is between 4.5 to 5 seconds before collisions.

As TTA is greatly determined by velocity and distance to the collision point, sometimes, rather than issuing a warning to the drivers of the affected vehicles, it can be more useful to issue a command to the relevant vehicles to so that

appropriate action can be taken automatically rather than wait for the user to react to a warning message (thereby adding a further delay). For example, if TTC is greater than TTA, a warning message will be generated for driver to take an action. However, if the TTC is less than TTA, then the system can issue a command to change the steering angle or even change lane to avoid collision automatically without driver's intervention. The higher is the velocity; the lower is the available time to avoid collision, and a greater chance that a direct command message to the machine can be more effective.

Since the communication between all the components involved in the intersection collision warning and avoidance system should be efficient and effective, the next section reviews various elements involved in communication between the central computing infrastructure and the vehicle.

2.5.5 Communication Model and Protocol

When considering communication between the components in the intersection collision warning system, which include vehicles and the central intersection component, we must consider all the possible costs involved and the model of the communication. Therefore, we need to know where to host the computations for collision detection and generate warnings, as discussed in the subsection 2.5.1 (centrally or distributed), so that the cost of generating warnings as well as the computation cost of collision detection can be reduced.

We also need to consider the message protocol to use, as it is important to consider the effectiveness and efficiency of the message. Road Web Mark-up Language (RWML) [Kajiya04] is an initiative to setup a web service and its protocol for the purpose of exchanging road traffic information. It contains static information on environment surrounding the traffic, such as weather, road closure

event, etc. RWML is based on XML. It has been an issue that XML web services can cause a heavy bottleneck that generates a performance problem. It is suitable for the purpose of general road traffic information, but it is not able to serve the purpose of exchanging sensor information intensively and issuing collision warnings in real time. Therefore, it is necessary to create a lightweight message protocol for intersection collision warnings. At this stage, there is no real-time messaging protocol that has been proposed for intersection collision warning. The next section summarizes the previous discussions and discusses the challenges for the future of intelligent software systems in ITS.

2.6. Summary

In this chapter, we have reviewed road safety analyses that have been conducted in various intersections in different countries worldwide. The results of those analyses vary from one site to another, due to different and varying characteristics of each intersection. Hence, initiatives and efforts in advancing intersection safety also vary from one intersection to another. However, essentially, road safety analyses are performed in uniform stages in each intersection. Road safety analyses consist of three stages, which are pre-analysis (data collection), analysis (data investigation that yield knowledge, patterns, etc to help decision making), and post-analysis (implementing the solution). Current road safety analyses mainly involve manual observation and traditional ways (e.g. paper database, survey and interviews, statistic generation, etc). Computing techniques can help to automate parts of the analyses' tasks that are normally conducted through manual observation. For example, pre-analysis can involve data collection from roadside and vehicle sensors in addition to the domain expert knowledge extracted from manual observation. Analysis can take into account pervasive computing technologies, such as context awareness, knowledge based

systems, intelligent agents, and data mining, to generate, present, and communicate interesting knowledge and patterns about intersections. These components can then be implemented in post-analysis as a holistic solution to intersection safety problems. Thus, these leveraging of automated techniques to supplement and enhance analyses tasks have not been extensively performed. Such techniques demonstrate clear applicability and benefits for enhancing currently prevalent manual approaches.

The current or existing intersection collision warning systems are mainly infrastructure and vehicle-based only, and are limited in many ways because those systems are mostly built for certain type of intersections, do not have the ability to learn from past collisions, have performance and scalability issues, and do not explore the availability and potential of sensor data. Communication is not developed as part of the systems (hence, there is no real-time communication protocol), and some do not define a clear relationship between detection and warning.

Consequently, there is a clear need for a cooperative intersection collision warning and avoidance system that:

- incorporates various and real-time data from vehicle and roadside sensors;
- is generic and adaptive to various intersections;
- is able to perform real-time detection and warning no matter how busy an intersection is;
- has an established real-time communication protocol;
- is able to send a warning message effectively and efficiently in order to avoid a future immediate collision.

Our review has established that current research projects in cooperative intersection collision warning and avoidance system are still in progress and do not possess the above requisite properties.

Therefore, an intersection safety system that can adapt to all kinds of intersection, detect collisions at road intersections, and warn drivers of potential collisions or hazards in real time, is required. In designing and developing an intersection collision and warning system, there are few issues we need to take into account, which includes variety of real-time data sources, performance and scalability, the issue of adaptability and learning, communication cost and model, and relationships between collision detection and warning. We need to consider whether the most appropriate real-time data sources are either from the intersection's infrastructure sensors or in-vehicle sensors. The collision detection and warning should be cost-efficient and robust that it is able to accommodate any increasing number of vehicles in the vicinity and still able to communicate messages and warning in real-time. Learning is an important component of the system as it helps in reducing pairs of vehicles at the intersection based on the learnt collision patterns, also allows the system to be more flexible. Communication cost, model, and its relationship to detection should be part of the design of the framework to deliver a cooperative intersection collision warning and avoidance system.

Such requirements of a generic, adaptable, real-time, efficient, and cooperative intersection collision warning and avoidance system can be satisfied by pervasive computing techniques. Integration of knowledge based systems, context awareness, and data mining bring powerful facets to the development of an intersection collision warning and avoidance system. The concept of knowledge based system shows how knowledge can be acquired and represented for decision making in intersection. When a knowledge base is used for intersection safety,

the domain expert knowledge and results of analyses from historical and real-time data of the intersection can be stored and used for collision handling. Context awareness helps an intersection collision warning and avoidance system to realise the surrounding states that can affect the application behaviour and then act accordingly. When an intersection collision warning and avoidance system is context-aware, it is able to correspond to the intersection characteristics and the current situation in handling collisions. Data mining is used to extract useful and interesting patterns from the data collected either manually or from sensor or both. Hence, when an intersection safety system is equipped with data mining capabilities, it is able to learn collision and traffic patterns that pertain to intersection, and thus, enhance the knowledge base of the system.

In Chapter 3, a framework for a cooperative intersection collision warning and avoidance system that is underpinned by pervasive computing techniques is proposed. The components that make up the framework and how each component contributes to the solution are discussed in detail in that chapter.

CHAPTER 3

THE UBIQUITOUS INTERSECTION AWARENESS (U&I AWARE) FRAMEWORK

Due to the high rates of accidents and fatalities worldwide, developing a reliable intersection safety system that is able to detect and warn of potential collisions is a priority in many countries. Essentially, a fast, accurate, and efficient approach is required to detect and avoid potential threats at a road intersection. As discussed in the previous chapter, there is a requirement for an intersection collision warning and avoidance system that is able to:

- incorporate various real-time data sources from vehicle sensors and roadside sensors;
- adapt to different intersections;
- perform real-time detection and warning;
- send a warning message effectively and efficiently in order to avoid a future collision.

In order for a vehicle to effectively avoid an imminent collision, the time needed to avoid a collision (i.e. Time-To-Avoidance or TTA) should be less than the time left before a potential collision is predicted to occur (i.e. Time-To-Collision or TTC). Therefore, it is essential to increase the speed of detection (thus increasing the TTC value) and reduce the communication costs (thus decreasing

TTA value). After reviewing the desirable properties of an intersection collision warning and avoidance system addressed in Section 2.5, in order to achieve a real-time intersection safety system, it is evident that we need to address the following issues.

To speed up collision detection (increase TTC), we need to reduce the number of vehicle pairs for which collision detection needs to be computed. Thus, there needs to be a mechanism for a preselection or filtering process to reduce the total computation time. In this thesis, we propose that using data mining from a variety of historical/real-time sources has the potential to provide collision patterns that are intersection specific. These patterns, which can then be made readily available and accessible through a knowledge base, can form the basis for preselection.

Furthermore, to reduce the time needed to avoid collision (reducing TTA), we need accurate cost models of the Time-To-Avoidance (TTA) to assess timely warning or command messages and a communication protocol that can operate in an efficient and real-time manner.

Thus, there is a need to incorporate pervasive computing techniques to analyse collision and traffic data as well as perform efficient collision detection. A real-time communication protocol that is able to send warning message only to potentially affected drivers is also required. Furthermore, due to the variety and varying characteristics of intersections, a framework that is generic and adaptive to different types of intersections is necessary. Learning of intersection characteristics and dangerous driver behaviours should be an integral part of such a framework. Techniques, such as data mining, can be used to support generality and adaptability of the framework by learning collision patterns that are pertinent to a specific intersection.

The term framework in this thesis is used to refer to a chain of multiple software processes that collaborate together to achieve a goal. This framework is flexible, generic, and can be used to generate and represent various intersection collision warning and avoidance systems. Hence, the desirable properties of intersection collision warning and avoidance systems should be part of this framework.

We aspire to design and develop a framework that is generic, adaptable and performs situation recognition at road intersections through learning and detecting potential threats and generating warnings to relevant road users at an intersection. This chapter discusses our generic, adaptive, and real-time safety framework for road intersections. The work presented in this chapter has been previously published in [Salim06], [Salim07a], [Salim07c], [Salim08a]. The framework is introduced in 3.1. Sections 3.2 through 3.6 elucidate how the facets of the framework satisfy the requirements of a safety framework laid out in the previous chapter (in particular, Section 2.5.). Section 3.7 concludes the chapter.

3.1. U&I Aware Framework

In this thesis, we propose the Ubiquitous Intersection Awareness (U&I Aware) framework to achieve situation recognition and real-time collision detection and warning at road intersections. The components of the U&I Aware Framework are portrayed in subsection 3.1.1. In 3.1.2, novel traits of the U&I Aware Framework are discussed. Lastly, in 3.1.3, the mapping of the framework to agents is presented.

3.1.1 Components of the U&I Aware Framework

The U&I Aware Framework is the basis of a cooperative intersection collision warning and avoidance systems. Figure 3.1 illustrates the framework's components, which consist of learning, detection, and warning of collisions at an intersection. It also demonstrates the elements in each of the components in the U&I Aware Framework and the iterative operational process among the components. This figure is based on ANSI/ISO flowchart standards [ISO85]. The aim of the diagram is to conceptualise the components of the proposed framework in terms of their functionality at an abstract level.

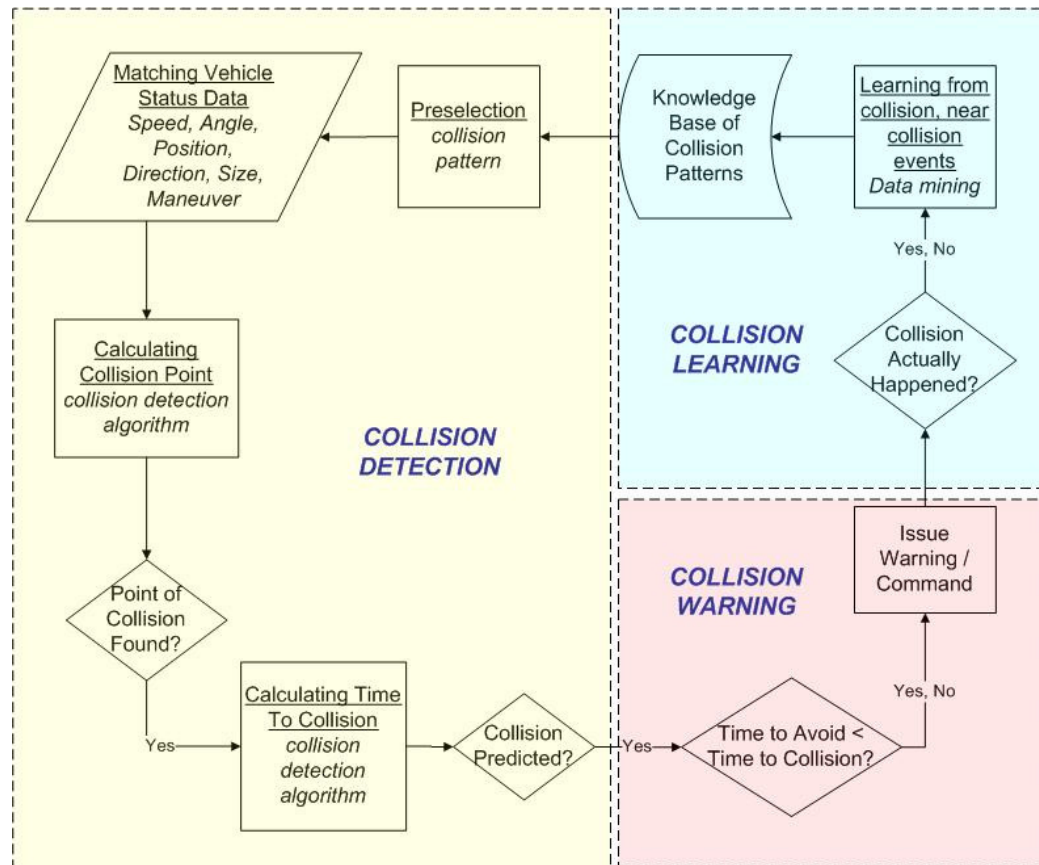


Figure 3.1. Collision Detection, Learning, and Warning Components in the U&I Aware Framework

Each of the components is described as follows:

- *Collision Learning*. This component belongs to the pre-analysis and analysis phase of the Road Safety Analyses (RSA). The collision learning component consists of the following elements:
 - i. *Data collection*. Historical collisions as well as online real-time vehicular and traffic data from the intersection's vicinity are collected to be analysed. Since collision data are rare, "near collision" or "near miss" events [Hayw72] are also captured to support data collection. The U&I Aware Framework only consumes sensor data, it does not perform any sensor data fusion or processing of raw sensor data from a particular sensor or a sensor network. Learning can start as soon as data is collected. A minimum quantity of data required is not specified since patterns (such as collision patterns) can be extracted once there is data. However, as a general rule of thumb, the more data is acquired, the higher is the support and confidence of the patterns and rules extracted from it.
 - ii. *Data mining*. Due to the need for a generic intersection collision warning and avoidance system, learning of specific collision patterns that are relevant for each particular intersection needs to be performed using data mining techniques. Once data are collected, data mining is applied on the collected data.
 - iii. *Knowledge Base Integration*. The results of learning that are relevant only for that particular intersection are integrated into the knowledge base of the framework for that intersection. Hence, the knowledge base is specific to an intersection and the situations that occur at the intersection. The knowledge base is used as the basis for preselection, which is an algorithm to match the vehicles that pass through the intersection with the collision patterns in the knowledge base. This is the key to reducing TTC.
- *Collision Detection*. This component belongs to the post-analysis stage of the RSA. In this thesis, the term "*collision detection*" and "*collision prediction*" are used interchangeably as both refer to recognising potential collisions (i.e.

future threats). These terms do not refer to identifying past or existing collision events. The collision detection component contains the following elements:

- i. *Preselection*. Based on the status data of a vehicle and key collision patterns in the knowledge base of the intersection, the preselection algorithm identifies vehicle pair combinations that have possibilities to collide.
 - ii. *Calculate future collision point*. The potentially colliding vehicles provide data to the collision detection algorithm. Each vehicle pair selected by preselection is assessed to see if a future collision point exists.
 - iii. *Calculate TTC*. If a future collision point is detected, then the TTC of each vehicle in the pair to the future collision point is calculated and compared. When the TTC of both vehicles are almost equivalent, then a future collision is imminent.
- *Collision Warning*. This component also belongs to the post-analysis stage of the RSA. The elements of the collision warning component are as listed below:
 - i. *Calculate TTA*. TTA of both vehicles are calculated using the TTA cost model. We present our proposed TTA cost model in this chapter that addresses the need for a real-time communication protocol.
 - ii. *Issue warning or command*. Depending on the TTA of each vehicle, either warning messages are issued to drivers of the relevant vehicles or command messages are generated and sent directly to the vehicle systems to avoid or minimise impact of an impending collision.

Within each component of the U&I Aware Framework, the processes are performed sequentially, because each of these processes is executed interdependently of each other. However, the component itself is executed continuously. Firstly, the collision data are monitored at all times and the results

of collision learning are incrementally added to the knowledge base. Secondly, the collision detection component continually monitors the status data of passing vehicles. Thirdly, the collision warning component is performed whenever it receives a new future collision event prediction. Thus, the U&I Aware Framework is a parallel and continual process of learning, detection, and warning of collisions, which are highly correlated to each other. Next, we discuss the traits of the U&I Aware Framework that distinguish it from existing collision warning and avoidance systems.

3.1.2 Novelty of the U&I Aware Framework

Currently, existing collision warning and avoidance systems only have detection and warning components (such as in a reactive intersection safety system – Figure 2.3). Consequently, they can merely react and respond to certain events as pre-programmed. However, since the U&I Aware Framework incorporates the *learning component* (such as in a reactive and deliberative intersection safety system – Figure 2.4), which does not exist in other collision avoidance systems, it carries two major positive contributions. Firstly, *adaptability*, as the framework is able to adapt to different and varying intersection characteristics that are learnt over a period of time. Secondly, improvement in *performance and scalability* of collision detection at intersections, as given the collision patterns learnt in the intersection, not all pairs of vehicles need to be computed for collision possibility.

The novelty of collision learning enables new intersection collision warning and avoidance systems (that can suit to various intersections) to be developed on the basis of the U&I Aware Framework as the governing principle. This is because the adaptation of new knowledge and information gained from mining of sensor and historical data at the intersection are performed as an integral part of the

framework. By learning from historical data of collision and near-collision events, improved detection and reactive behaviour can be achieved since the knowledge base of the intersection continues to evolve. Thus, the system can operate in any intersection where it is installed and learns of collisions that are specific to that intersection.

The U&I Aware Framework, as a basis for a cooperative and generic intersection collision warning and avoidance system that works on various intersections, is inspired by the notion of context-awareness, since a context-aware application is capable of being conscious of the changes in its environment and adjusting its behaviour accordingly. A context-aware application consists of a set of context attributes that become the basis for recognising a situation, adjusting the behaviour of the application, and issuing a specific response.

As the U&I Aware framework is generic and adaptable to different locations, it can be considered as a context-aware application (or to be more specific a location-aware application). The framework can be aware of changes in the location context and able to use the context information (e.g. collision patterns, traffic patterns, road user behaviours) as stored in the knowledge base to adapt to location changes by learning from sensor data. This knowledge base has the ability to grow over a period of time if incremental learning from current events is incorporated into the system. In fact, collision patterns are the main context attributes that are used in the U&I Aware Framework that makes it a context-aware application. There are multiple context attributes that can determine the behaviour of the application, which are applicable to this application domain. Examples of context attributes that can be used in an intersection safety system might be: speed profile of a driver, acceleration behaviour of a driver, speed limit of the intersection, traffic patterns during different times of the day or different days of the week etc. In this thesis, collision patterns are the context attributes

that determine the circumstances in which collision detection is performed. These patterns are explained further in Chapter 4. Collision detection is only performed when matching vehicle status data with the context attributes (i.e. collision patterns) are found.

The key to the context-awareness of the U&I Aware Framework lies in the integration of data mining techniques and a knowledge base to facilitate the framework to learn from its environment (*and accumulate context attributes of a specific intersection location*), be aware of the occurrence of learnt events or incoming threats in the environment (*monitor the intersection for events that can be identified with the context attributes*), and respond to the incoming threats contextually (*based on a given context attribute, the system yields a certain action, e.g. issuing a specific warning to the relevant drivers*).

The next subsection presents the mapping of the U&I Aware Framework to an implementation driven by software agents at the intersection's vicinity.

3.1.3 Implementation Map and Scope

For implementation, the U&I Aware Framework is mapped to agents in intersections and vehicles. The notion of an agent is used to signify a piece of software that can act autonomously on behalf of the user. Ideally, each agent needs to be capable of learning from sensory and historical data, detecting threats, and issuing warning to one another. Learning needs to be enabled in each agent depending on the context. For example, an intersection agent can learn patterns of collisions and traffic at the intersection's vicinity. A vehicle agent can learn driver behavioural patterns in driving context as well as dangerous driving behaviours (such as drink driving and drowsiness). Threat detection can also be enabled in every vehicle and intersection agent based on the patterns learnt on the

agent. For example, a vehicle agent is able to detect drink driving behaviour and threats that are faced by the driver when such behaviours are learnt. However, collision detection based on collision patterns can only be done by the intersection agent, since collision patterns are learnt by each respective intersection agent and not by vehicle agents. When a threat is detected, the agent can then issue warning messages to other agents that may be impacted.

The communication between the intersection agent and vehicle agents in the U&I Aware Framework is regulated inside the *administration zone* (Figure 3.2). An *administration zone* is the spatial domain that determines the region of authority of an intersection agent to coordinate vehicle agents in the approaching and passing vehicles. A wireless infrastructure is required for the administration zone and messaging of U&I Aware Framework to operate. A wireless network router can be installed in each intersection centre. Each vehicle needs to be equipped with a wireless device, or at least with Bluetooth. The size of an administration zone is dependent on the effective wireless signal strength. The maximum radius of an administration zone is 100 meters from the intersection centre. This is because both wireless network and Bluetooth can cover the range of 100 meters well.

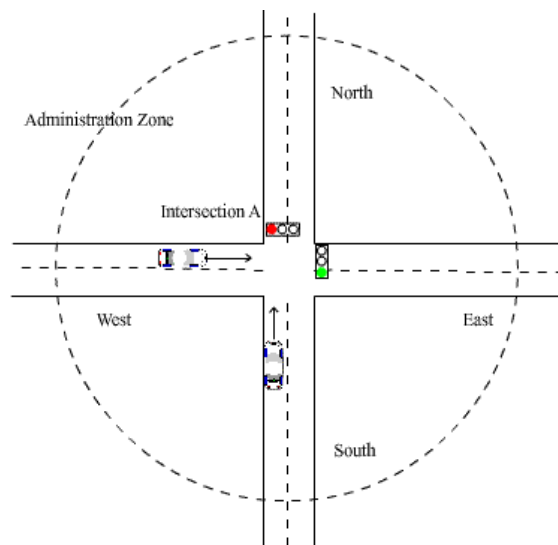


Figure 3.2. Intersection Administration Zone

Figure 3.3 portrays the relationships among vehicle agents and the intersection agent. Status messages, which are used to communicate real-time sensory information of each vehicle, are broadcasted by each vehicle agent to all other agents continuously. A registration message is initiated by the intersection agent and sent to the vehicle agent that enters the intersection's administration zone. A registration message is used by the intersection agent to retrieve specific information about the vehicles that are relevant for collision detection. Warning messages can be sent by any agent at the intersection to other relevant agents. Figure 3.3 displays a full scale implementation of the U&I Aware Framework since learning, detection, and warning components are implemented in both vehicle agents and intersection agent. However, based on the given scenario, a full scale implementation may not be necessary. In this thesis, learning, detection, and warning components are only implemented on intersection agent.

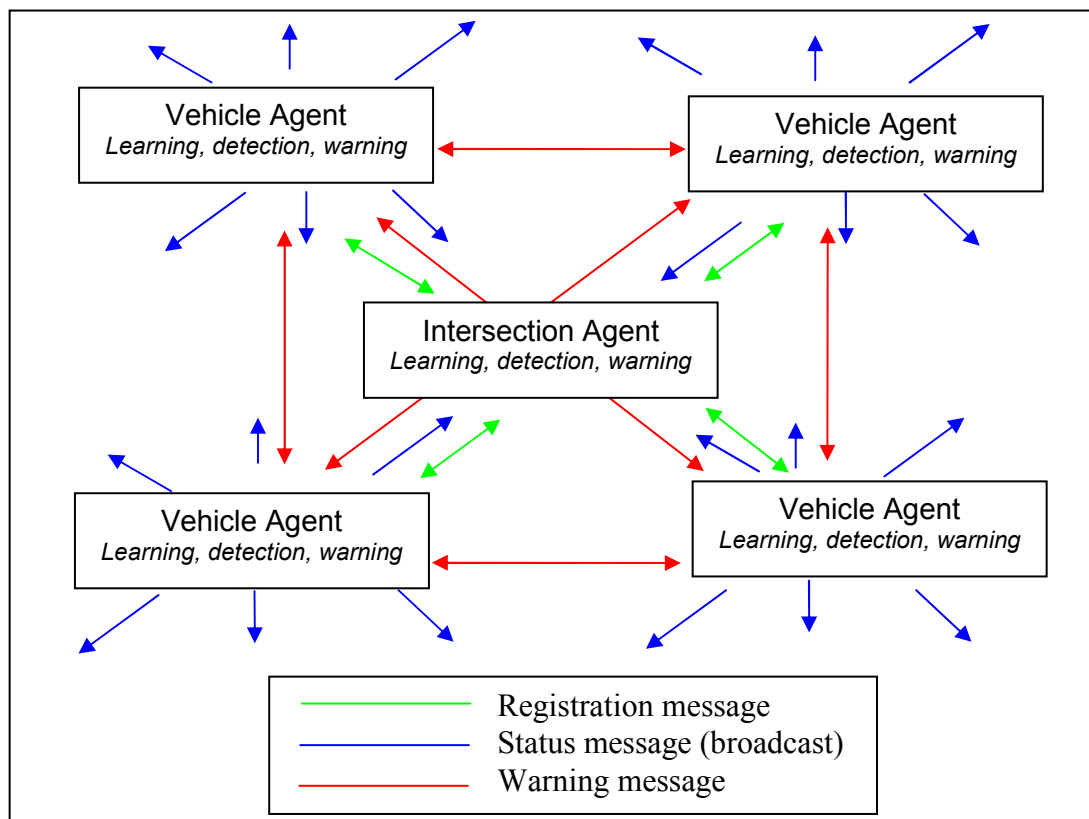


Figure 3.3. Mapping the U&I Aware Framework to Agent Implementations

As mentioned in Section 1.4, the aim of this thesis is to establish a generic framework that assists in collision detection and avoidance at road intersections. We propose learning of collision patterns to add to pair-wise collision detection for higher efficiency in terms of TTC. The implementation and messaging scenario adapted in this thesis is depicted in Figure 3.4. An intersection agent acts as a central traffic authority to learn collision patterns, detect threats and warn possibly affected vehicles of incoming hazards. The vehicle agent of each car at the intersection should always report to the intersection agent of its entry into and exit from a designated area in the vicinity of the intersection and also send its status periodically. Vehicle information and driver's behaviour information, such as driving manoeuvres are retrieved from in-vehicle sensors. An intersection agent manages the tasks of communication, learning, detection, and warning. The protocol of the communication is described further in section 3.6.

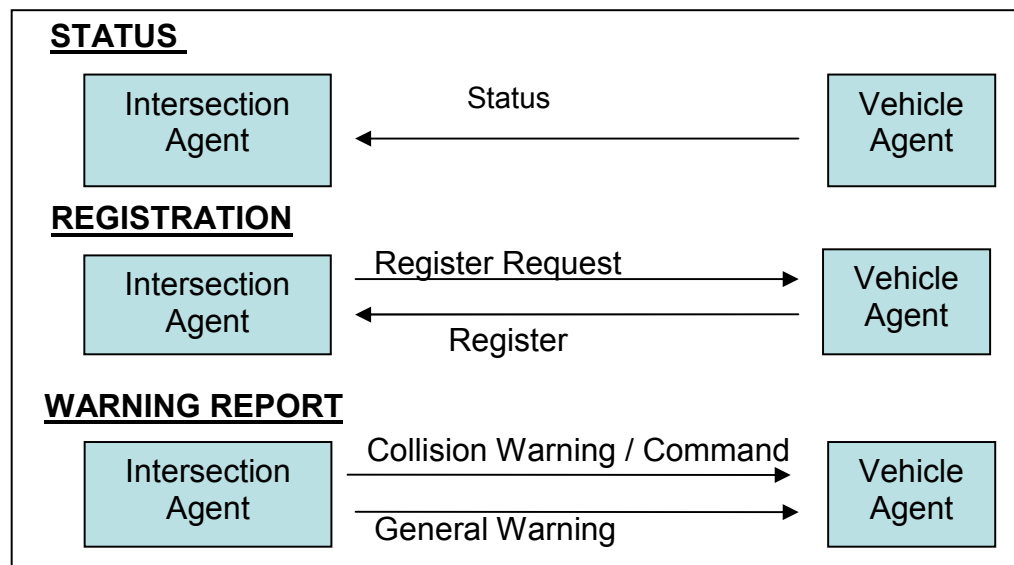


Figure 3.4. Agent Implementation and Messaging Protocol

Since collision patterns of an intersection are appropriately learnt by intersection agent and not vehicle agents (as discussed further in section 3.3), we do not implement learning, detection, and warning capabilities in vehicle agents in this

thesis. Collision learning and collision detection components are evaluated in this research and discussed further in Chapter 4 and 5. The cost model and functional evaluation of collision warning component are discussed in section 3.5 and section 3.6 respectively, but the field evaluation of collision warning is beyond the scope of this research. Furthermore, new model or applications for road safety or ITS in general require evaluations to be conducted in a simulation first before real-world evaluations [Sicking00]. Therefore, computer based simulations are developed to evaluate this research.

As mentioned previously in section 2.5, there are a number of desirable properties for an intersection collision warning and avoidance system based on: integration of a variety of real-time data sources for collision detection, performance and scalability, the issue of adaptability and learning, communication cost and model, and relationships between collision detection and warning. How the U&I Aware Framework meets each of the desirable properties is discussed in the following subsections 3.2 to 3.6.

3.2. Consideration about Variety of Data Sources

As discussed in section 2.5.1, in order to detect collisions efficiently, real-time data sources are required. The system needs to be aware of the current status of each vehicle travelling at the intersection. It is mentioned in section 2.2 that in terms of retrieving data that is pertinent to a particular vehicle, the speed of processing data from sensors in the vehicle itself is faster than roadside sensors. We can get reliable real-time data from vehicle sensors, but information from roadside sensors cannot be obtained in real-time. Hence, it can be more efficient if every vehicle can provide and transmit the data directly over a wireless communication link to other entities that require them, than purely relying on

infrastructure sensors. This bypasses the need for complex data fusion of roadside sensors. For example, the current speed of a particular vehicle can be measured by the speedometer in the vehicle and registration number information can be built-in or saved in the vehicle's profile. This approach is simpler and faster than using roadside sensors, such as a camera, to measure the speed of the vehicle and detect the registration number of the relevant vehicle. It has been tested that performance of a system using both on-board sensors and inter-vehicle communications is better than that of the system using merely on-board sensors or merely inter-vehicle communications [Satake07]. Therefore, we suggest utilising in-vehicle sensors as a useful data source rather than roadside sensors to retrieve information about a vehicle.

Nevertheless, there is information about the intersection that can only be provided by roadside sensors. For example, to know the traffic light rules and operation at the intersection, the traffic control sensors must be incorporated into the system. Fundamentally, both roadside and vehicle sensors are needed for a global comprehensive view of an intersection. However, the selection of sensors to be used for certain data should be decided based on the accuracy of the data and the speed with which it is available and accessible.

As suggested in section 2.5.1, the usage of vehicle sensors for data collection about each vehicle and roadside traffic control for traffic rules information is recommended. Vehicle manoeuvre, which is also required for collision detection, can be detected from GPS data (e.g. lane change manoeuvre detection [Xuan06]), face and gaze sensor data, or video cameras data with CHMM implementation [Oliver00]. Predicting intended driving manoeuvres can be done one second before the actual manoeuvre takes place. Hence, the best scenario of sensors to be used and applied for collision detection based on accuracy and performance are:

- vehicle sensors: GPS (to detect vehicle speed, angle, direction and position), GPS or face and gaze sensor (for manoeuvre detection), built-in profile information of a vehicle (vehicle size, registration number);
- roadside traffic light sensors (if the intersection has traffic light controllers).

Next, in Section 3.3, the facets of the U&I Aware Framework that ensures performance and scalability of collision detection are presented.

3.3. Performance and Scalability of Collision Detection

Performance and scalability of the collision detection component is affected by two factors (see Section 2.5.2): (i) the location where data analysis, data processing, and collision detection are performed, which are either centralised (where calculations are done centrally at the intersection's vicinity) or distributed (where calculations are done locally in each vehicle), and (ii), the method of filtering the vehicles that have a likelihood of being involved in a collision, which is either brute-force (no filtering as collision detection is computed for every possible pair at the intersection) or using some mechanism for filtering or preselecting certain vehicles (collision detection is performed only on vehicles that match the known collision patterns).

The centralised computation has a greater advantage of having the bird's eye view of the whole intersection at a time, since the central component knows the status of all vehicles at the intersection. Hence, in order to have a global bird's eye view of the intersection and reduce the overhead of vehicle-to-vehicle communication (which is the case if the distributed approach is adopted), it is recommended to have a central component where computations and learning of collision patterns are going to be performed. With a centralised computation

method, the matter of the sensor location is not that significant in comparison to a distributed computation method. In this thesis, we investigate a centralised computation strategy. Thus, to facilitate the centralised computation, the U&I Aware Framework uses an intersection agent that is located at the intersection's vicinity. This section only covers discussion on collision detection. The other tasks of the intersection agent are presented in subsequent sections.

In order to improve performance of collision detection, we need to revisit the conventional way of computing collision detection, which must be performed each time a car moves from its current position [Miller02], as mentioned in Section 2.3.4. The peer-to-peer collision detection system by Miller and Huang [Miller02] implies a brute force approach as computation is done locally in each vehicle against every other vehicle at the intersection. Figure 3.5 presents the pseudocode of the conventional pair-wise collision detection algorithm used in [Miller02].

```

for each vehicle at the intersection
  V1 = current vehicle
  for each other vehicle at the intersection
    V2 = other vehicle
    calculate point of collision (X)
    if point of collision found then
      calculate Time-To-Collision of V1 to X (TTC1)
      calculate Time-To-Collision of V2 to X (TTC2)
      if (TTC1 +- (V1 size/speed)) = (TTC2 +- (V2 size/speed))
        collision is predicted between V1 and V2
      else
        no collision is predicted between V1 and V2

```

Figure 3.5. Pair-Wise Collision Detection Algorithm [Miller02]

Each time a vehicle moves, it must know about possible collisions with any other vehicle in the vicinity. However, vehicles that move in a non-discrete time must send its information in a series of discrete intervals. The collision detection algorithm in Miller and Huang's approach is computed whenever new information is received, which is at 10 Hz, i.e. 10 signals sent per second [Miller02]. If there are 10 vehicles in the vicinity, each of the 10 cars will send 9 signals, one to every other vehicle that is in the vicinity per second. In total, there will be 90 signals broadcast every second by all other cars which will need to be processed by each car. When the number of vehicles in the vicinity increases (such as during peak hours); the computation cost for brute-force also increases exponentially. Therefore, when the brute-force method is applied, the issue of performance and scalability becomes prominent.

Given that the brute-force method implies collision detection computation to be applied on every possible pair of vehicles at the intersection, it is not efficient to do so in real-time, especially when there is an increasing number of vehicles at the intersection. Consequently, to enable real-time collision detection, we need to reduce the number of vehicle pairs for which collision detection computation is performed. Therefore, we propose that before the collision detection computation is performed, we need to "preselect" the vehicles that are most are most likely to collide based on the collision patterns learnt at the intersection. Preselection requires a holistic or global view of the intersection. Therefore, it needs to be performed centrally by the intersection agent. This is yet another underpinning factor for the design decision of the U&I Aware Framework, having a centralised rather than distributed mode of operation. Thus, the intersection agent needs to store the useful and related information of that particular intersection to enhance the performance and scalability of the collision detection. An example of such information is the most common collision pattern in the intersection that involves certain manoeuvres and driving directions. When such a pattern is found as being

relevant or applicable to passing vehicles, collision detection computation must be performed.

Since the real-time considerations clearly imply that preselection is more advantageous and preferable to the brute-force approach. The U&I Aware Framework proposes and develops novel strategies for performing preselection. We also evaluate the performance impact and benefit of preselection in comparison with the brute force approach. To improve the efficiency of a collision detection algorithm, the preselection method is applied in the U&I Aware Framework, so that collision detection is only performed on pairs of cars that have the possibility of collisions based on known intersection collision patterns.

We propose that data mining can be used to learn patterns of collisions in an intersection. These patterns can form the basis for performing preselection, since these patterns demonstrate the combinations of vehicle pair' characteristics (i.e. direction, manoeuvre, angle, intersection leg position) that have the high likelihood to collide. By choosing only the vehicles that exhibit behaviours, or are in specific relative locations, or are involved in driving manoeuvres that match specific collision patterns in the knowledge base for collision detection calculation, the performance of detection can be improved, while still maintaining the accuracy. This is because the number of vehicle pairs at the intersection that need to be computed for collision detection is reduced. As stated earlier, preselection is performed by the intersection agent, as this agent given its centralised operation knows the status of all vehicles in the vicinity. Hence, mining collision patterns is also performed in the intersection agent.

For example, given a cross intersection where the knowledge base contains a collision pattern “perpendicular straight”, which implies to collisions that happens between vehicles that have a straight manoeuvre movement when

entering the intersection and their conflicting paths intersecting at an angle of around 90 degrees. If a car enters the intersection from the south leg of the cross intersection detection with a straight manoeuvre movement, collision detection will be performed on this car against every other car that is currently located on perpendicular paths (i.e., west and east legs of the intersection), or moving straight towards the intersection. Therefore, performance can be improved and the cost of collision detection computation can be more efficient by not needing to check every vehicle pair at the intersection for collision detection computation. Although there is also an additional cost involved for preselecting the pair of vehicles at the intersection in addition to the cost of collision detection computation of the preselected vehicle pairs, it is relatively less expensive in comparison with the reduction of the cost of computing the pair-wise collision detection algorithm. The cost of computing the pair-wise collision detection with preselection can be considerably more efficient than without preselection, particularly when there are more vehicles at the intersection. However, there are cases when there are only a very small number of vehicles that pass through the intersection and mere application of pair-wise collision detection without preselection could serve the same efficiency.

The graph in Figure 3.6 displays the comparison of algorithm counts (i.e. the number of times the collision detection algorithm is executed to calculate possibilities of collisions in an intersection at a given time and the number of vehicles in an intersection) between brute force (merely conventional collision detection algorithm) and having preselection within each computation interval (which is 5 milliseconds in our simulation). Data is sampled from our simulation. The figure reveals that preselection performs better than brute force. The preselection method reduces the algorithm count greatly. At times, although the number of vehicles increases, when there is no vehicle that fulfils the preselection

criteria, the algorithm count is zero. However, with the brute force approach, an algorithm is executed at all times.

However, when the preselection approach is to be applied, the knowledge base should have a set of collision patterns that pertain to the intersection. Otherwise, the system may miss detecting a potential collision that is not stored as a known pattern in the knowledge base. Hence, collision patterns that become the heuristics for preselection kept in the knowledge base need to be learnt over a period of time to accommodate new changes at the intersection. The issue of adaptability and learning are further discussed in the following subsection.

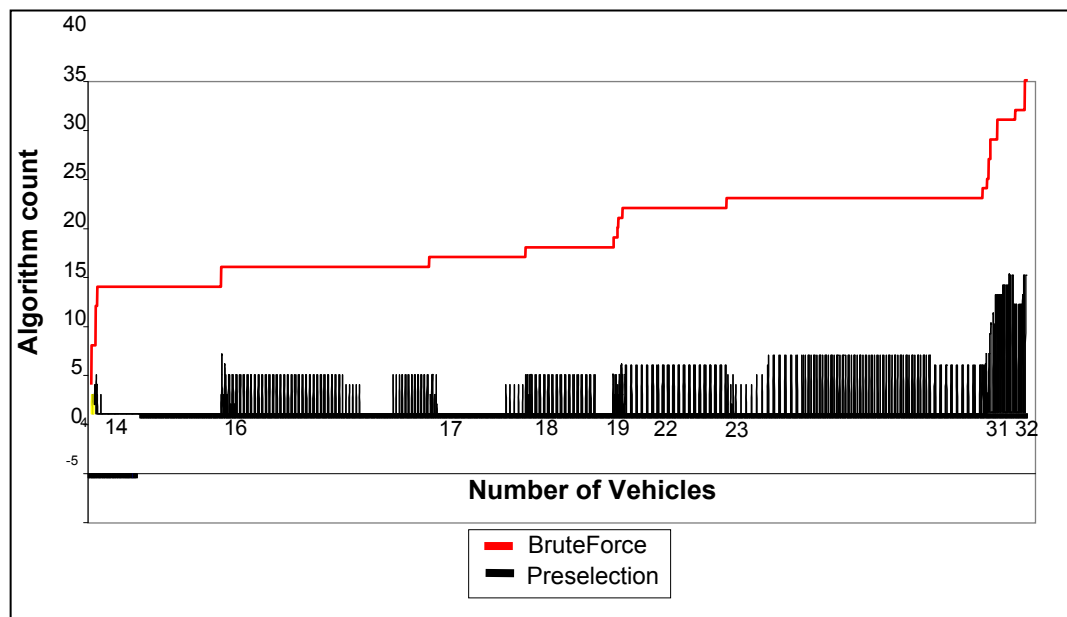


Figure 3.6. Performance Comparison between Brute Force and Preselection Method

3.4. Adaptability and Learning

As has been previously discussed in Chapter 1, it is desirable to have a generic intersection safety system that is adaptable to different types of intersections. As stated in Section 2.5.3, an intersection collision warning and avoidance system

should be *generic* (i.e. applicable to various types of intersections) and *adaptable* (i.e. capable of making adjustments to specific traits and patterns of collisions at a particular intersection). We propose that appropriate data mining techniques can be applied to sensor data (from vehicles and road infrastructure). We also suggest that the learning results obtained can be maintained in an enduring and dynamic knowledge base. The knowledge base can first be populated with collision patterns obtained from expert knowledge, while learning is performed. However, once collision data of the intersection is accumulated, the collision patterns can be learnt and maintained along with or on top of the existing knowledge. When learning is integrated into the intersection agent, its knowledge will improve, and the system can evolve and adapt to changes at the intersection over a period of time.

The key element of a generic intersection collision warning and avoidance system is the knowledge base, where specialised information that is only applicable and useful for that particular intersection is stored. The other elements of the intersection safety system can still be generic, thereby, allowing the system to be adaptable to different types of intersections. Each intersection has a different knowledge base that is specifically initialised with the characteristics of that particular intersection and possible collision patterns that may occur. With the inclusion of a specialised knowledge base for each intersection, a generic and adaptive intersection safety framework is made possible.

Therefore, it is necessary to learn from a history of events at the intersection (such as collision and near collision events) and real-time traffic data in order for the system to adapt to a specific intersection or new changes at the intersection. Learning of historical data in the U&I Aware Framework is performed to enhance the knowledge base of the intersection agent for better collision detection. Learning is performed through computational data analysis, rather than

manual field observation as the typical *modus operandi* for such systems. We propose the use of data mining for learning patterns and trends of a particular intersection, as data mining is very appropriate to extract knowledge and patterns and it has been widely used for learning traffic safety and trends in highways (as discussed in Chapter 2, Section 2.3.3). With advances of data mining (such as the recent ubiquitous data stream mining) as discussed in Chapter 2, learning also can be done onboard the vehicle utilising driver's profile and vehicle sensor data, thus making the vehicle agent that sits in the vehicle to be aware of the vehicle and the driver's behavioural contexts. As a result, the intersection collision warning system can be more informed when a driver exhibits dangerous driving behaviours. For example, when a vehicle that enters the intersection's vicinity exhibits drunk or tired driving behaviours (that is detected from drink driving patterns that are previously learnt) [Horo06], other vehicles at the intersection that are possibly affected would be warned about this vehicle. In this case, drink driving behaviours can be learnt and detected [Horo06] using a vehicle agent in each vehicle system so that the vehicle agent can inform the intersection agent to warn other vehicle agents in the vicinity of such threatening behaviours.

Integrating knowledge about an intersection from data mining results into the knowledge base helps road users to understand and be aware of threats at the particular intersection. Having a comprehensive set of collision patterns of an intersection assists in faster collision detection, as all passing vehicles can be initially matched with patterns in the intersection using the preselection approach. Hence, the knowledge base of the U&I Aware Framework may have different collision patterns when it is moved from one intersection to another, because every intersection is unique and will typically have different collision patterns (because of varying intersection characteristics). The crash pattern knowledge base, which is the basis for preselection, is filled with crash patterns and each

crash pattern consists of a name, a manoeuvre, a direction, an intersection leg location, and a function to find conflicting direction and manoeuvres.

As the results of collision learning become the basis for preselection that is designed to improve the efficiency of the collision detection, it is also necessary to improve the method and protocol of the collision warning in order to achieve a timely warning for relevant drivers of potentially affected vehicles about predicted collisions. The next section discusses the relationship between collision detection and warning.

3.5. Relationship between Collision Detection and Warning

As discussed in 2.5.4, the two main temporal dimensions that need to be considered in collision warning, which are the Time-To-Avoidance (TTA) and Time-To-Collision (TTC), should be taken into account in modelling the relationship between collision detection and warning. In order to avoid a collision, TTA must be lesser than TTC. However, warnings should not be issued in a manner whereby TTC is substantially greater than TTA. In such case, warning may not be necessary, as a potential collision might have been spotted by the driver or avoided in a due time. A warning that is too early can become an annoyance to the driver.

The value of TTC is determined by the collision detection computation, which is the time of a vehicle to reach the predicted future collision point. The value of TTA is computed based on the cost model of TTA, which needs to consider various factors such as the time to generate warning messages, network latency time, human response time, and vehicle response time. This section presents our

proposed cost model of computing TTA, which improves the two existing cost models proposed by Miller and Huang [Miller02] and INTERSAFE [INTER05] by combining the two models and adding finer abstraction details.

The formula to calculate Time-To-Avoidance (TTA) proposed by Miller and Huang [Miller02] is:

$$TTA = t_r + \frac{\beta v}{\mu g} \quad (3.1)$$

where t_r is the response time of the driver, β is the speed reduction factor (its range is from 0 to 1 that indicates the level of brake), v is the current speed, μ is the anticipated tire-road friction coefficient, and g is the acceleration of gravity.

However, there are different factors that can be considered in calculating TTA as proposed by INTERSAFE [INTER05]. The minimum warning distance required to inform a driver in order to stop in front of the intersection or behind the stop line [INTER05], is:

$$D_{\text{information}} = \frac{V_0^2}{2a} + (t_{\text{driver}} + t_{\text{machine}} + t_{\text{information}}) \times V_0 \quad (3.2)$$

where V_0 is the velocity of the vehicle, a is the vehicle braking deceleration, t_{driver} is the driver's response time to brake, t_{machine} is the combination of braking system and warning system response time, and $t_{\text{information}}$ is the constant information time, which is a time determined by the assistance system to allow the driver to react and prepare the driver to stop. Similarly, the formula 3.2 can be used to calculate TTA by adding t_{driver} , t_{machine} , and $t_{\text{information}}$ with the current vehicle speed divided by deceleration rate.

The t_r factor in Miller and Huang's algorithm [Miller02] is the same as t_{driver} in INTERSAFE's formula [INTER05]. The main differences between the INTERSAFE formula and Miller and Huang's proposal are that Miller and

Huang do not consider $t_{information}$, which is necessary in a warning system, and secondly, $t_{machine}$ in INTERSAFE is more comprehensive by including warning system response time. The time components in INTERSAFE's TTA formula have been evaluated on real-world tests and can be used as a point of reference in our system. Nevertheless, none of the two models for TTA consider communication or messaging cost with external systems such as a traffic control entity or a central component (such as an intersection agent). Although Miller and Huang's system is a vehicle-based warning system and in INTERSAFE, a central intersection computer system is assumed, both must consider communication cost and network latency in the cost model of TTA.

In the U&I Aware Framework, it is important that we only send messages to affected vehicles when a potential collision is detected. For collision warning, point-to-point messaging should be used between vehicle and intersection agents instead of broadcasting. As there is a need for real-time warning, the messages sent should be short, and thus, would only require short processing time. However, as discussed in Section 2.5.4, when TTA is not enough to issue a warning to notify the driver, it is more appropriate to send a command message to the vehicle agent directly to brake (in this thesis, we do not consider turning, increasing speed, or other avoidance methods and see these as future directions for this research). Therefore, we propose two types of avoidance messages with two types of TTA accordingly:

- warning message, intended for the driver, measured by $TTA_{warning}$;
- command message, intended for the vehicle braking system, measured by $TTA_{command}$.

These are described in Figure 3.7. So, to decide when a warning or command message should be generated, the rule of thumb to follow is:

- if $TTC > TTA_{warning}$, send collision warning message; else

- if $TTC \leq TTA_{warning}$, send command message.

Figure 3.7 portrays the cost model of collision avoidance, which is as follows:

- If a collision warning message is generated:
 - the message should be sent to the vehicle computer (the cost variable is $t_{message}$);
 - the vehicle computer alerts the driver (the cost variable is $t_{receive}$) by means of audio warning;
 - the driver reacts to the warning message by applying the brake (the cost variable is $t_{response}$);
 - and the brake system is slowing down the vehicle until it stops (the cost variable is t_{brake} and v/a).
- But if a command message is issued, the message is sent to the vehicle computer (the cost variable is $t_{message}$) and the vehicle computer directly applies brake to the vehicle to slow down the vehicle until it stops (the cost variable is t_{brake} and v/a), thereby passing delays due to the driver's reaction time.

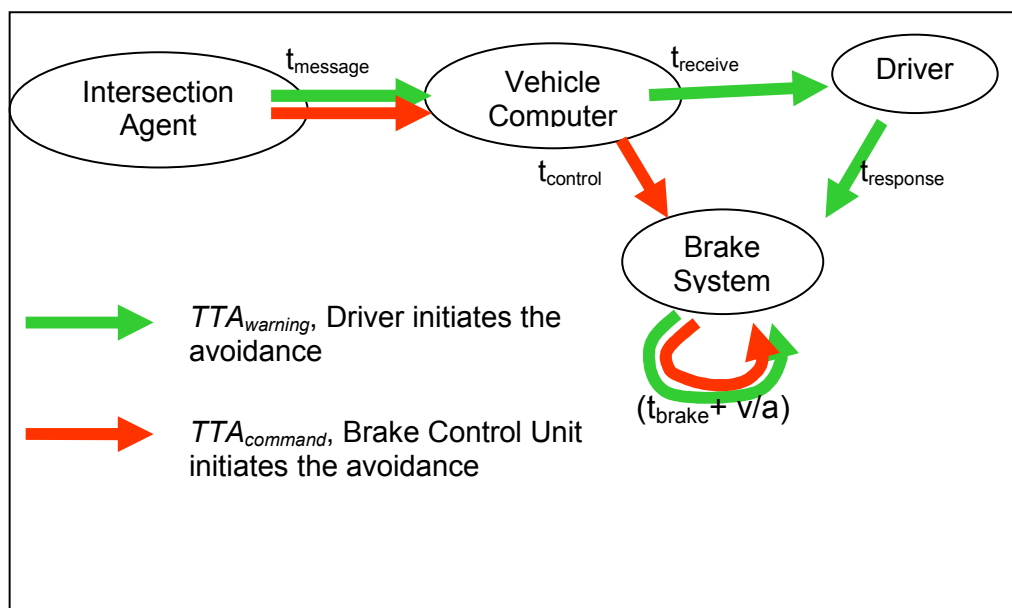


Figure 3.7. TTA Cost Model Diagram

When the intersection agent detects a collision, firstly it calculates the $TTA_{warning}$ using the following formula (3.3). If the $TTA_{warning}$ is less than the TTC, the intersection agent sends a warning message to the vehicle agent. When the vehicle agent receives the message, it generates an audio warning to warn the driver to stop the vehicle.

The cost model for $TTA_{warning}$ (driver initiates the avoidance) is:

$$TTA_{warning} = t_{message} + t_{receive} + t_{response} + t_{brake} + \frac{v}{a} \quad (3.3)$$

where $t_{message}$ is the required time to generate, transmit and read a warning message by the software, $t_{receive}$ is the time for a driver to receive the message, $t_{response}$ is the response time for a driver to take an action, t_{brake} is the response time of braking system, and v/a is the time to full stop (v is velocity and a is acceleration).

Nevertheless, if the $TTA_{warning}$ is larger than or equal to TTC, there is not enough time to inform the driver to avoid the collision. Therefore, a command message is going to be sent directly to the vehicle agent, and then the Brake Control Unit (i.e. the system that controls the brake automation) in the vehicle initiates the brake action to stop the vehicle directly without driver's interruption. The cost model for $TTA_{command}$ (Brake Control Unit initiates the avoidance) is:

$$TTA_{command} = t_{message} + t_{control} + t_{brake} + \frac{v}{a} \quad (3.4)$$

where $t_{control}$ is the response time of the Brake Control Unit.

The cost of issuing, transmitting, and reading a warning (in time units), $t_{message}$, after notification of new information or event is computed by:

$$t_{message} = t_{generate} + t_{transmit} + t_{read} \quad (3.5)$$

where $t_{generate}$ is time for the intersection agent to generate the message, $t_{transmit}$ is time for message transmission from the intersection agent to the vehicle agent,

and t_{read} is time for the vehicle's computer to read the message. The message transmission time, $t_{transmit}$ can be calculated by:

$$t_{transmit} = \frac{message_size}{bandwidth} + latency \quad (3.6)$$

where $message_size$ is the size of the message in bits, $bandwidth$ is the capacity of the communication channel in bits per second, and latency is the delay time in the communication channel that can be contributed by various factors such as bottleneck, queuing, message propagation, etc.

Besides $t_{message}$, the values of other components of $TTA_{warning}$ ($t_{receive}$, $t_{response}$, and t_{brake}) are beyond our control and are not affected by the design and protocol of the communication. We cannot manipulate $t_{receive}$ and $t_{response}$ since the reasoning and reaction time of the driver are parts of human factors. The only way we can improve $t_{receive}$ is by ensuring effective warning delivery (however, human-computer interaction is outside the scope of this thesis). The value of t_{brake} also varies from one vehicle to another. Since the only component of $TTA_{warning}$ that we can improve and manipulate is $t_{message}$, we aim to reduce $t_{message}$ as much as possible. Therefore, it is necessary to:

- generate the warning message rapidly;
- construct a short message to achieve a short transmission time;
- read and decipher the message promptly.

Clearly, high network bandwidth with low latency is also required. The real-world deployment of the U&I Aware Framework implies such networking infrastructure to be in place. The next section presents the model and protocol of communication between the intersection agent and the vehicle agent.

3.6. Communication Model and Protocol

This section discusses the communication model and protocol that are transmitted within the intersection agent's administration zone (Figure 3.2). As the intersection agent needs to work together with all vehicles in vicinity, there are at least three different types of messaging required between vehicle agents and the intersection agent (Figure 3.4), which are as follows:

- First, status messages need to be sent periodically from vehicles to intersection agent to keep the intersection agent up to date of vehicle's data for collision detection computation. As the status message is sent periodically, we need to also consider if the message is initiated by the vehicle agent (push method) or the intersection agent (pull method). In real-time terms, it is better to employ push method, as the status request message from the intersection agent is eliminated, thereby reducing communication cost.
- The second message type is the registration message. The presence of each vehicle needs to be known to the intersection agent.
- The third message type is warning message from the intersection agent sent to vehicle agents. At this point, there are two options of message delivery: point-to-point or broadcast. As false warning or alarms need to be avoided as much as possible, broadcast is not an option for collision warning. Point-to-point message delivery can ensure that only affected vehicles will be warned. However, if there is a general warning (e.g. weather warning or speed limit warning) that needs to be issued then a broadcast can be used.

We also need to consider the message protocol to use, as it is important to consider the effectiveness and efficiency of the message. Clearly, this solution is based on assumptions that communication network may fail. It is possible that communication is not reliable on wireless medium, however other works have

addressed solutions, e.g. [Huang04]. We propose a lightweight (i.e. concise and compact) message protocol for intersection collision warnings, since there is no real-time messaging protocol that has been designed specifically for intersection collision warning. We propose three types of messages transmitted within the administration zone (Figure 3.5), which are: status report, registration, and warning report. These are discussed in the following subsections 3.6.1 – 3.6.3. The evaluation is presented in 3.6.4.

3.6.1 Status Report

The purpose of status message is to report the existence of a vehicle travelling in the intersection's administration zone and to inform about the vehicle's dynamic information so that the intersection agent can track the vehicle's position correctly and use the information to predict collision. When a vehicle enters the intersection's administration zone, the vehicle agent (VA) detects the wireless signal from the intersection agent (IA) that signifies an intersection's administration zone is in place. The VA then commences to send status message to the IA repeatedly (Figure 3.8).

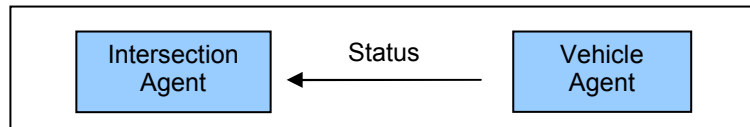


Figure 3.8. Status Message

This message includes the vehicle's dynamic information, such as vehicle ID, speed, position, angle, and manoeuvre. This information is retrieved from the vehicle's sensors, such as described in 3.2. The message structure is:

status		<vehicle_ID>		<x>		<y>		<speed>		<acceleration>		<direction>		<angle>		< manoeuvre >
--------	--	--------------	--	-----	--	-----	--	---------	--	----------------	--	-------------	--	---------	--	---------------

The word “*status*” is to indicate the message type. The *vehicle_ID* is the registration number of the vehicle, e.g. “VICABC001”. The x, y values are the coordinate values of the vehicle’s position, e.g. 213, 320. The *speed* is the velocity of the vehicle, e.g. 16.666. The measurement used for speed is meter/second. The *acceleration* is the acceleration of the vehicle, e.g. 1.471, which is measured in meter/second². The *direction* is the travel direction of the vehicle. The value of direction could include 0.00 (if the vehicle travels towards north), 90.00 (if the vehicle travels towards east), 180.00 (if the vehicle travels towards south), 270.00 (if the vehicle travels towards west). The *angle* is the steering angle of the vehicle, e.g. 0.00 for going straight. If the vehicle turns 5 degree to the left, the value is -5.00. If it turns 5 degree to the right, the value is 5.00. The *manoeuvre* is the intended driving manoeuvre that is predicted by in-vehicle devices. Typically it has been shown that such manoeuvres can be predicted one second before it occurs [Oliver00]. The value of manoeuvre could include Passing, TurnLeft, TurnRight, ChangeLaneLeft, ChangeLaneRight, Starting, and Stopping. Each parameter in the message is separated by a vertical bar “|”.

Existing research suggest various figures for the interval time of reporting the vehicle’s status. Miller and Huang [Miller02] suggested the interval time as one second. Kosch and Strassberger from BMW Group Research and Technology suggested that the interval time is around 100ms [Farkas06]. In our proposed communication protocol, we suggest a variable interval time. The interval time should depend on how far a vehicle has moved. We propose a variable interval time because the speed of vehicles at the intersection varies. When a vehicle travels at a higher speed, the position of the vehicle then changes more rapidly. Hence, it is necessary for a VA to report its status to be more frequent. Therefore, the variable interval time should be determined by the vehicle’s travelling distance. After a vehicle moves a certain distance, e.g. 0.5 metre, it is required to

report its status. If a vehicle travels at a high speed such as 60km/hr or 80km/hr, the interval time should be shorter, e.g. 30ms or 23ms. If there is a traffic jam, the interval time can be longer since vehicles are travelling slower. Another reason to employ a variable interval time is to prevent network congestion. Consider a scenario where there is a traffic jam at an intersection and there are a large number of vehicles at the intersection. If a constant short interval time is employed in such condition, network congestion may occur. For these reasons, we introduce the variability of interval time, which is determined by the travelling distance of each vehicle.

Given that another purpose of status message is to indicate whether a vehicle is still existent in the administration zone, we also propose a maximum interval time threshold, which is one second. Therefore, although a vehicle is not moving on the road (e.g. the vehicle is parked), it still needs to report its status at least every one second.

After a vehicle has passed through the intersection and is outside of the administration zone, the VA no longer receives the wireless signal from the IA. The VA stops sending status message to this particular IA.

3.6.2 Registration Message

There is static information about a vehicle that is necessary to be included in collision detection computation, such as vehicle size. However, since such information does not change over time, it is not necessary to be included in the status message and sent periodically. Therefore, we propose that a registration message is used to communicate vehicle's static information (Figure 3.9). The registration message is sent only once and the content of the registration message needs to be maintained by the IA as long as the respective vehicle exists in the

administration zone. After the IA receives a status message from the VA, the IA should check for the existence of the vehicle's static information, such as the vehicle's registration number and the size of the vehicle. If the IA does not have this information in its database, the IA sends a *Register Request* message to the vehicle. When a VA receives it, it replies the IA with a *Register* message.

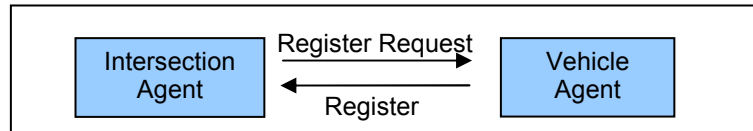


Figure 3.9. Registration Message

The content of the Register Request Message is very simple. Its structure is:

regreq <vehicle ID>

The word “regreq” is to indicate the message type.

The Register Message includes the vehicle's static information, such as vehicle ID and size. The message structure is:

regist <vehicle ID> <length> <width>
--

The word “regist” indicates the message type. The *length* is the length of the vehicle in meter. The *width* is the width of the vehicle in meter.

After the IA receives the register message, it should store the static information of the vehicle. When the IA receives another status message from the VA, as long as the IA has the static information of this particular vehicle, it should not send a register request to this vehicle agent. If the VA has exited the intersection's administration zone, the IA should no longer receive the status message from the VA. After a period not receiving any status message (e.g. 3 seconds), the static

information of the particular vehicle should be removed from the IA. The vehicle's final status message, which includes the vehicle's last position, speed and travel direction, can be used to determine whether the vehicle has actually exited the administration zone.

3.6.3 Warning Report

After status data is received, IA performs collision detection computation. When an imminent collision is predicted, a warning message is issued (Figure 3.10).

There are two types of warning messages. Firstly, *General Warning message*, which is broadcast to all vehicles including information for speed limit and dangerous behaviour warning, such as drink driving. The message structure can be one of the following type:

- *spdlmt* | *<value>*, e.g. “spdlmt|60.000” means that speed limit is 60 kilometers per hour
- *drkdrv* | *<vehicle ID>* | *<x value>* | *<y value>*, e.g. “drkdrv|VICPAD123|221|578” means that a drunk driver is driving vehicle “PAD-123” at the position (221, 578).

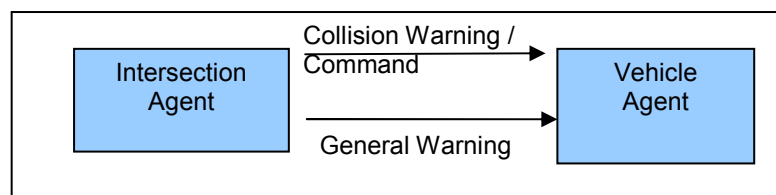


Figure 3.10. Warning Message

Secondly, *Collision Avoidance message*, which can either be a *Collision Warning* or *Command* message. If an intersection system detects that a collision will happen, its IA send *Collision Warning* to notify the driver of the pair of involved

vehicles. This message includes data: vehicle ID, Time-To-Collision (TTC), collision position, and collision type.

The message structure is:

$collwn \mid \langle vehicle\ ID \rangle \mid \langle TTC \rangle \mid \langle x \rangle \mid \langle y \rangle \mid \langle type \rangle$
--

The word “collwn” is used to indicate the message type. The *TTC* is the time to collision for the particular vehicle. The x, y are the position of the collision point in our simulated environment. The *type* is the collision type, e.g. *Side* or *RearEnd*. The vehicle agent receives it, processes it, and warns the driver.

However, if the TTA is less than the TTC, the IA sends a *Command* message to the VA so that the vehicle takes an action automatically without the driver’s intervention. This message includes data: vehicle ID and action. The message structure is:

$commnd \mid \langle vehicle\ ID \rangle \mid \langle acceleration \rangle$

The word “commnd” indicates the message type. If *acceleration* is negative, the vehicle needs to slow down. Otherwise, the vehicle needs to speed up.

The next section presents the evaluation performed on the protocol.

3.6.4 Evaluation

This section presents the evaluation of our proposed communication protocols and its cost borne to the overall TTA cost model. There are four contributing factors to $TTA_{warning}$, which are $t_{message}$, $t_{receive}$, $t_{response}$, and t_{brake} , and three factors contributing to $TTA_{command}$, which are $t_{message}$, $t_{control}$, and t_{brake} . As previously discussed in 3.5, the value of $t_{receive}$, $t_{response}$, t_{brake} , and $t_{control}$ are beyond our

control. Only $t_{message}$ can be evaluated by prototyping the IA – VA communication model.

According to the INTERSAFE project [INTER05], based on real-world experimentations, the value range for $t_{response}$, t_{brake} and $1/a$ (gravity acceleration) are as displayed in the Table 3.1 [INTER05]. The $t_{receive}$ value is 1.1 seconds, since that is the average reaction time for elderly [Green00]. The value of $t_{receive}$ of drivers from other age groups should be smaller than 1.1 seconds. Therefore we consider 1.1 seconds as the maximum $t_{receive}$.

Table 3.1. TTA Components Value Range

	Min. Value	Max. Value
$t_{receive}$	1.1 seconds (average reaction time for elderly)	
$t_{response}$	0.8 second	2 second
t_{brake}	0.3 second	0.5 second
A	0.31 g = 3.038 m/s ²	0.7 g = 6.86 m/s ²
$1/a$	1/(6.86 m/s ²) = 0.1458 s ² /m	1/(3.038 m/s ²) = 0.329 s ² /m

In order to evaluate the communication cost, the protocols are implemented on a simulated intersection agent and vehicle agent. The implementation prototype is described as follows:

- The IA is simulated on a powerful computer or server. Since the IA is stationary and needs to perform learning, predict collision, communicate with numerous vehicle agents, and calculate the TTA, it needs to run on a powerful and stable machine. The IA is implemented on Java Virtual Machine (see Figure 3.11). It is developed by using NetBeans 5.5.1 and Java 2 Standard Edition 1.6.0.02.
- The VA is simulated on a small device. Since this agent only needs to communicate with one intersection agent, it does not need much computing power. Furthermore, because the device needs to sit in a vehicle, it is easier if the VA is installed on a small device rather than a huge full size computer.

The VA is implemented on Java Kilobyte Virtual Machine (see Figure 3.12). It is developed by using NetBeans 5.5.1 with the Connected Limited Device Configuration (CLDC) 1.1 and the Mobile Information Device Profile (MIDP) 2.0, which together provides a standard Java runtime environment for mobile device such as cell phones and Personal Digital Assistants (PDA) [Sun07]. It is important to note that a VA does not have to run on cell phone or PDA. Although it is implemented with a cell phone interface, it demonstrates that VA can run on a small device.

- Since our proposed communication protocol is an application layer protocol in the ISO-OSI Reference Model, it needs to work with the protocols in the lower layer. In our implemented prototype, we employ TCP and UDP for the transmission layer protocol, IP for the network layer protocol, and IEEE 802.3 for the data-link layer and physical layer protocol. The status message should be sent through the UDP/IP protocol. Although UDP is not reliable protocol, it is faster than TCP. Since status message is sent frequently, transmission speed is more important than the reliability of the connection protocol. Other message types should be sent through the TCP/IP protocol because TCP provides a reliable connection.

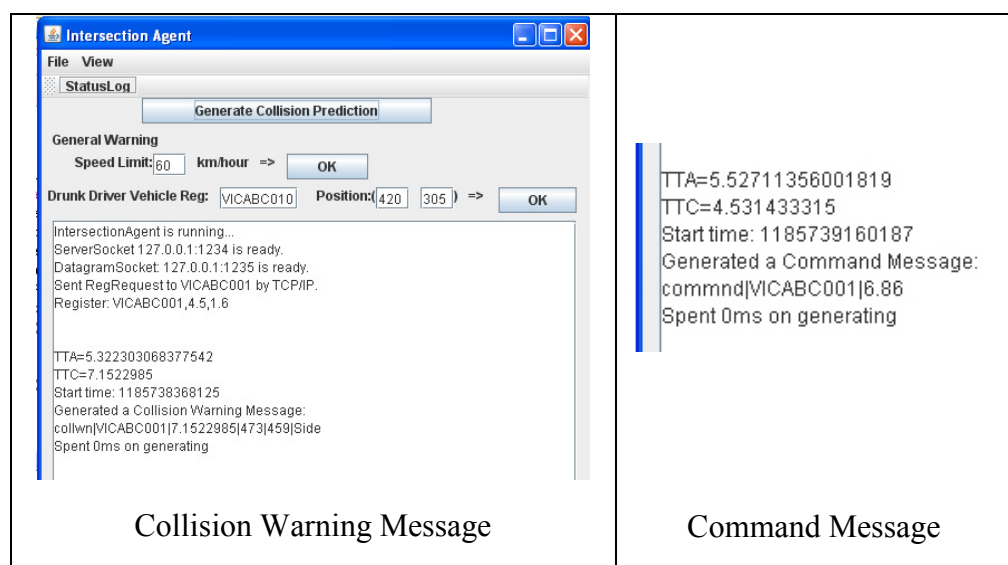


Figure 3.11. Simulation of an Intersection Agent

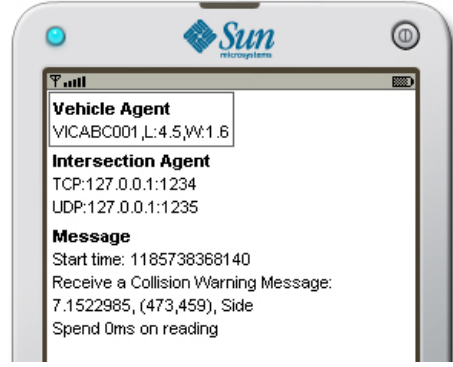


Figure 3.12. Simulation of a Vehicle Agent

The proposed communication model and protocol is comprehensively evaluated on their functionality. However, there is a limitation on the evaluation since it is implemented on a single machine. Bandwidth and latency are not yet taken into account into this evaluation, since we only perform the evaluation on a computer simulation. Furthermore, the standardisation of the wireless network protocol, which is IEEE 802.11p (Wireless Access for the Vehicular Environment, WAVE), is underway [Kerry08]. The wireless band or frequency of 5.9 GHz is licensed to be used by Intelligent Transportation Systems (ITS) [Kerry08] but there is no further details given about the available bandwidth and possible latency.

Based on the IA-VA prototype, we measure the value of $t_{message}$ in our implementation based on the formula (3.5) and (3.6). $t_{message}$ is the total of $t_{generate}$, $t_{transmit}$, and t_{read} . As seen in Figure 3.11 and Figure 3.12, the time for IA to generate the collision warning or command message and the time for VA to read the message are both 0ms, hence $t_{generate}$ and t_{read} are negligible. $t_{transmit}$ is the division of message size by the available bandwidth. The size of a collision warning message can be calculated based on its structure, which is described in section 3.6.3. This message consists of approximately 40 characters. Since we use

UTF-8 for encoding, each character can be encoded using one byte. Therefore, the size of the collision warning message is approximately 40 bytes. If the bandwidth is 10Mbit per second, the $t_{transmit}$ of a collision warning message will be 0.032 milliseconds. Considering the extra wrapper size from the TCP / IPv6 and other lower layer protocol, the value of $t_{transmit}$ is around 0.11 milliseconds. This demonstrates the efficiency of our proposed messaging protocol.

Hence, given the current speed is 60km/h (16.67 m/s), the minimum value of $TTA_{warning}$ is 4.630 seconds and the maximum value of $TTA_{warning}$ is 9.083 seconds. We assume the $t_{control}$ to be far less than t_{brake} and v/a , hence $t_{control}$ is negligible. Given the current speed of 60 km/h, the minimum value of $TTA_{command}$ is 2.73 seconds and the maximum value of $TTA_{command}$ is 5.983 seconds. Most intersections would have speed limit below 50 km/h, hence, the $TTA_{warning}$ and $TTA_{command}$ are smaller (see Table 3.2 and Figure 3.13). The smaller the $TTA_{warning}$ and $TTA_{command}$, the higher is the chance for the collision to be avoided.

Table 3.2. $TTA_{warning}$ and $TTA_{command}$ Range for Various Velocity

Velocity (km/h)	min $TTA_{warning}$ (secs)	max $TTA_{warning}$ (secs)	min $TTA_{command}$ (secs)	max $TTA_{command}$ (secs)
10	2.605	4.514	0.705	1.414
15	2.808	4.971	0.908	1.871
20	3.010	5.428	1.110	2.328
25	3.213	5.885	1.313	2.785
30	3.415	6.342	1.515	3.242
35	3.618	6.799	1.718	3.699
40	3.820	7.256	1.920	4.156
45	4.023	7.713	2.123	4.613
50	4.225	8.169	2.325	5.069
55	4.428	8.626	2.528	5.526
60	4.630	9.083	2.730	5.983
65	4.833	9.540	2.933	6.440
70	5.035	9.997	3.135	6.897

Figure 3.13 portrays the relationships between speed and TTA for $TTA_{warning}$ and $TTA_{command}$. When factors contributing to $TTA_{command}$ are at minimum, $TTA_{command}$ is always less than $TTA_{warning}$, hence it is more efficient to avoid an imminent collision when a command message is issued. However, it is interesting to note that when factors contributing to $TTA_{command}$ are at maximum, the speed of the vehicle is above 30 km/h, and factors contributing to $TTA_{command}$ are at minimum, it is better to issue a warning message. This phenomenon might be displayed because a vehicle requires a larger response time for it to stop when at a faster travelling speed.

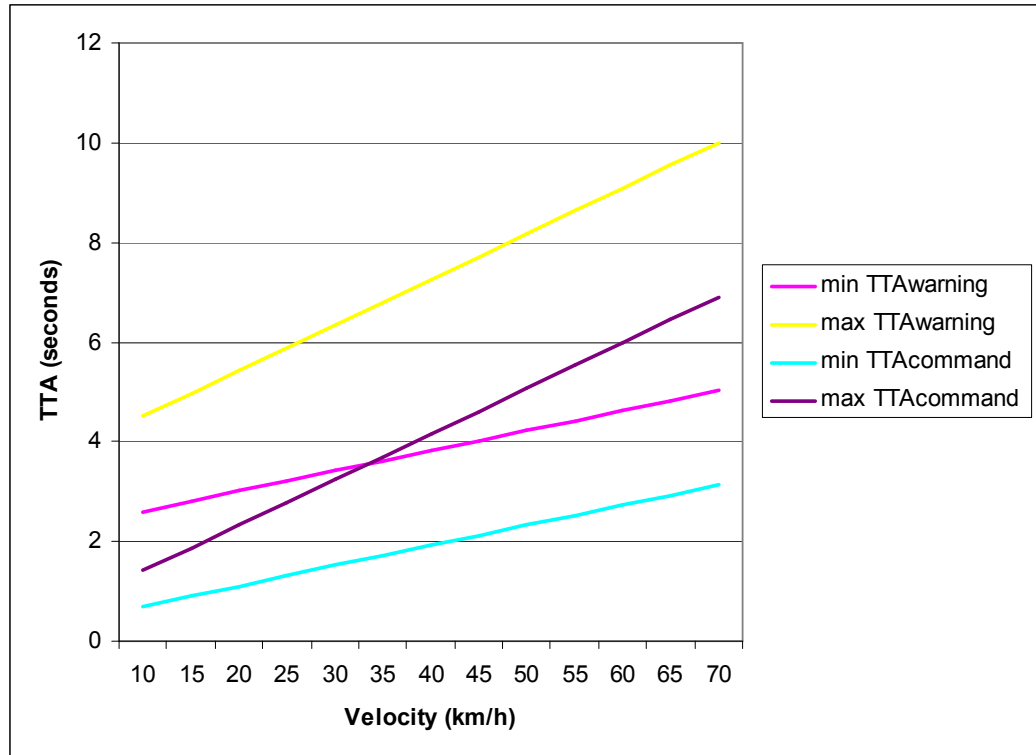


Figure 3.13. $TTA_{warning}$ and $TTA_{command}$ Range

3.7. Summary

In this chapter, we have proposed the Ubiquitous Intersection Awareness (U&I Aware) framework, which is a generic and real-time context-aware framework

for collision detection and warning at road intersections. The distinguishing feature of the U&I Aware Framework in comparison with existing intersection collision warning and avoidance systems is the presence of the learning component, whereas other systems typically include detection and warning components. This is because the U&I Aware framework is designed to meet the following desirable properties of an intersection collision warning and avoidance system, which are: usage of a variety of real-time data sources, performance and scalability, learning and adaptability, relationship between collision detection and warning, and real-time communication model and protocol. In order to have real-time collision avoidance, Time-To-Avoid (TTA) should be less than Time-To-Collision (TTC). Hence, we need to increase TTC by making the collision detection process faster and decrease TTA by reducing the communication and warning process.

To facilitate this, the U&I Aware Framework uses an intersection agent to centralise computations to avoid communication overhead and complexity in vehicle based systems. The U&I Aware framework utilises mainly vehicle sensor data as real-time data sources for collision learning and detection. The historical collision and near-collision data, and real-time traffic data are mined to extract collision patterns and other interesting traffic trends from the particular intersection. These patterns can help to determine which vehicles are likely to be involved in a collision.

The learning of collision patterns in the U&I Aware framework allows the framework to be generic yet adaptable to different types of intersections. A knowledge base populated with the results of learning from historical traffic data and collision events of that particular intersection is proposed. The knowledge base acts as a means for tailoring the system to specific intersections. Thus, the framework is adaptable and can be used at different intersections. The results of

collision patterns learning, which are kept in the knowledge base, are employed as a basis for preliminary selection or identification of potential colliding vehicles. This filtering enables reducing computation time and increasing TTC.

We have also proposed and developed a communication protocol for real-time intersection collision avoidance systems. The cost model for TTA has also been formulated and proposed. When a collision is detected, a warning is generated to warn the drivers of respective vehicles. However, if there is not enough time to avoid the collision through collision warning, a command message is generated, bypassing the driver, straight to the vehicle agent to send message to the Brake Control unit to stop the vehicle directly. As a result, impending collisions can be avoided in real-world situations.

Components of the U&I Aware Framework are developed and evaluated in the Chapter 4 and 5. The collision learning component is further discussed in Chapter 4. The simulation of an intersection that is developed to generate data (simulating sensors) is also described. The sample data collected, types of analysis performed, learning results, and how they are stored in the knowledge base are discussed next. The development and evaluation of the collision detection component are presented in Chapter 5. The functionality of collision warning component has been presented in this chapter. The performance and real-world evaluation of the collision warning is a future work of this research.

COLLISION LEARNING

Since the innovation of in-vehicles and roadside sensors, there is a significant amount of sensor data that is available. This data is a valuable source from which important information can be distilled via data analysis techniques. Data mining is proven to be effective for extracting traffic patterns and trends, such as in the Pantheon Gateway Project [Gross05] that monitors traffic conditions and patterns at highways (see review in Chapter 2.3.3). In this research, we focus on learning patterns and trends in road intersections, such as collision patterns, as the basis for the knowledge base, which is used to perform preselection of vehicles for faster collision detection. Therefore, collision learning is an integral element in the U & I Aware Framework (Figure 4.1).

Throughout this chapter, when a collision is discussed, the terms *Subject Vehicle* (SV) and *Principal Other Vehicle* (POV) are used, since the collisions analysed in this thesis involve at least two vehicles. The term “collision pattern” used in this chapter and throughout the thesis is not the same as “collision type”. A collision type refers to a generic classification of collisions, such as rear-end collisions, head-on collisions, and side collisions, without specific regards to any intersection type. For example:

- Rear-end collision [USDOT04], [Lages04] is a type of collision in which two or more vehicles that are travelling on the same trajectory (i.e. same direction, same manoeuvre, and same angle) in a car-following model (i.e. travelling on the same lane) collide.
- Side collision, or namely perpendicular path in [Verid00] (see Figure 4.2), or straight crossing in [Fuers05] (see Figure 4.3), involves two vehicles that collide in a perpendicular angle while travelling in a straight path.
- Left turn collision [Verid00], [Fuers05] is a collision that involves a SV that is turning left (in Australia, it is turning right due to the right side driving) with a POV that is going straight (see Figure 4.2.a and Figure 4.3.c). This collision type encompasses 23.8% of all crash problems in USA.

In Germany and France, more than 60% of all collisions consist of side collisions and left turn collision patterns [Fuers05].

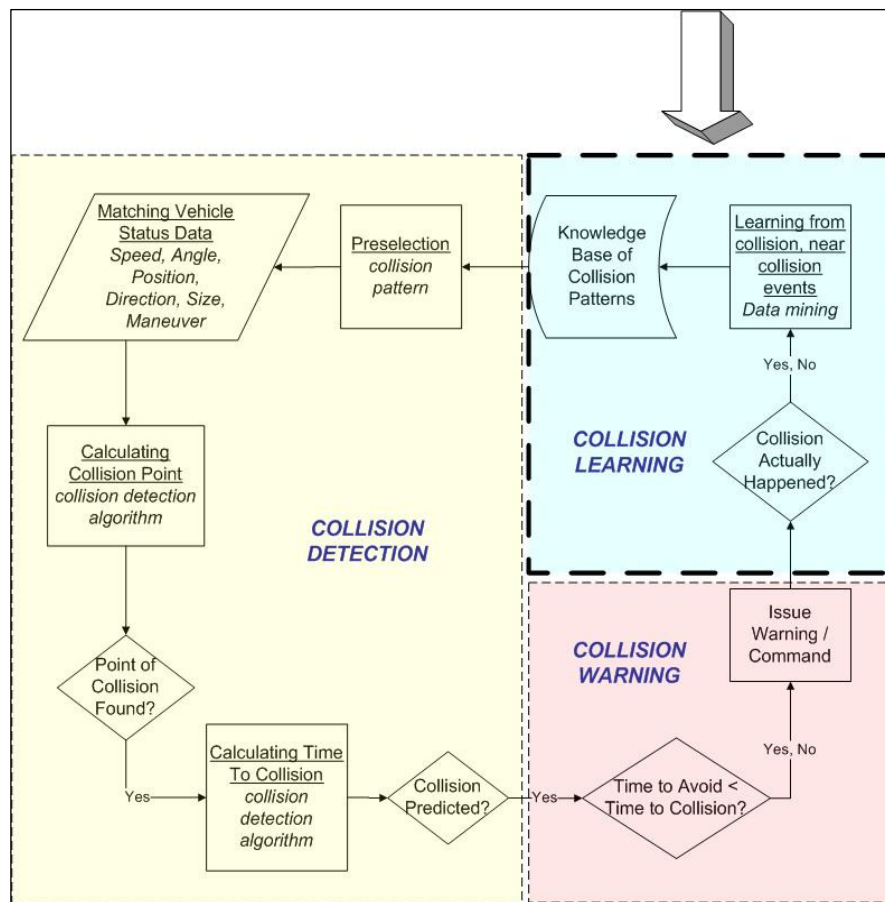


Figure 4.1. Collision Learning in the U & I Aware Framework

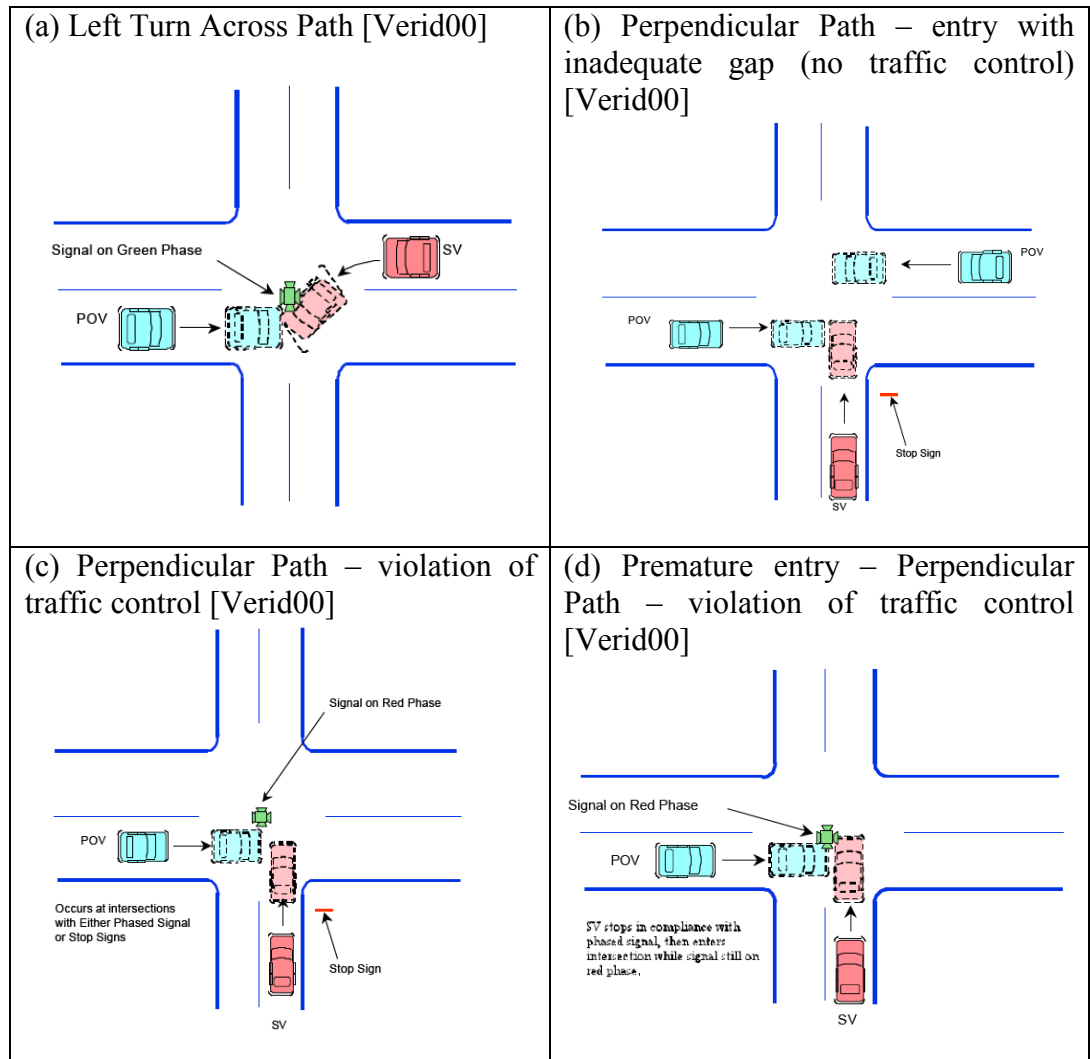


Figure 4.2. Intersection Crash Scenarios [Verid00]

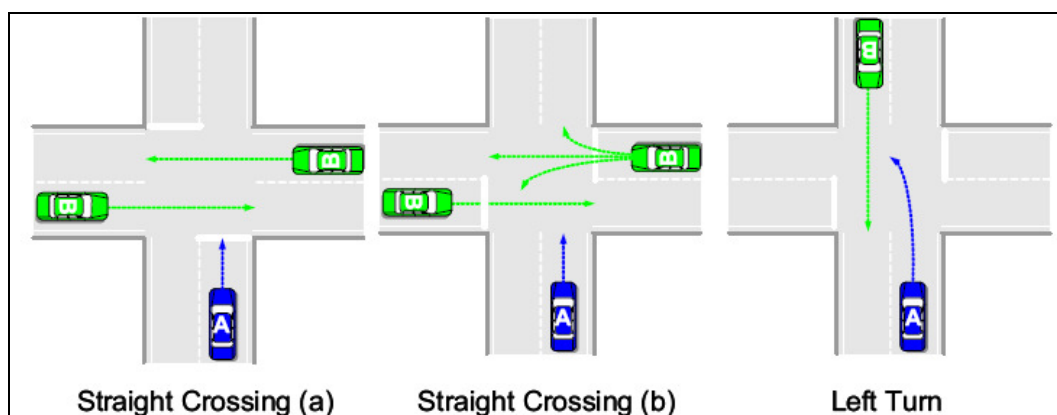


Figure 4.3. The Highest Occurring Scenarios that Encompass More than 60% of Intersections Collisions [Fuers05]

On the other hand, a collision pattern refers to a collision type that occurs in a particular intersection, which is affected by the characteristics of the intersection. Due to different and varying intersection characteristics, there can be numerous collision patterns for a collision type. A collision pattern consists of common attributes of a collision type that have occurred repeatedly (see Section 2.1 and Table 2.1). A collision pattern is characterised by collision type, manoeuvres, direction, location, and angle of the pair of vehicles involved in the collision. An example of a collision pattern that may occur in a four-legged cross intersection is a side collision that involves a vehicle that is travelling southward straight from the north leg with a vehicle that is travelling eastward turning right from the south leg. In Figure 4.2, some patterns of side collisions are shown as follows:

- At an intersection without traffic control (Figure 4.2.b), the SV can collide with the POV because of inadequate gap entry. This collision pattern comprises 30.2% of all crash problems in USA.
- At an intersection with traffic controls, a SV can collide with a POV as the SV violates the traffic control either by red light running (Figure 4.2.c) or premature entry (Figure 4.2.d). The number of side collisions that occur due to traffic control violation encompasses 43.9% of all crash problems in USA.

Since each intersection varies, it is necessary to learn collision patterns that pertain to the intersection for earlier identification of vehicles that exhibit the attributes as described in the collision patterns for that intersection. In order to perform learning, we need to have data. There are three stages of collisions, which are pre-collision, collision, and post-collision. The first issue we encounter is the non-availability of real-world pre-collision and collision data. Collision databases in Australia only record factors that are pertinent to post-collision events, such as fatalities, number of injuries, day and time, type of vehicle, and so on. However, we need real-time data that are recorded within seconds or even milliseconds before collisions occur, such as current speed, manoeuvre, intended

driving direction, acceleration or deceleration, and so on. This data can be collected using existing sensor technology. Due to constraints of resources and the nature of this research, performing real data collection is beyond our scope. However, we view that computer based simulation is a viable mechanism for proof of concept and data collection. Moreover, simulation should precede any real-world trial or deployment in road safety field [Sicking00]. Computer based simulations, which are accurately modelled based on the real-world scenarios, are able to yield accurate results and provide a great amount of information that are not available from a full-scale crash test the in the real world [Sicking00]. Therefore, we develop an intersection simulation that resembles real-world situations to generate traffic and collision data.

This chapter focuses on the Collision Learning component of the U & I Aware Framework (as shown by the arrow in the Figure 4.1). The information about occurrence of collisions (whether collisions have actually happened) at the intersection is archived into the historical collision data files. Learning is then performed on collision data and near collision events and traffic trends using data mining techniques. The results of learning are stored in the knowledge base. This chapter covers a discussion on our intersection simulation through which data collection is performed (Section 4.1) and mining that data in various scenarios along with the integration of the learning results into the knowledge base (Section 4.2). This chapter is then concluded in Section 4.3. The work in this chapter has been previously published in [Salim07b], [Salim07c], [Salim08a].

4.1. Intersection Simulation

Traffic simulators have been developed to replicate real-world traffic situations and test various applications before deployment and evaluation in the real world

can take place. Simulation models are mathematical/logical representations of real-world systems, which are designed to “mimic” the behaviours of complex systems and executed on a computer system [Lieb05]. Each simulation model consists of multiple components and simultaneous interactions among the components that form an abstraction of the real-world [Lieb05].

According to [Lieb05], traffic simulation models can be classified as:

- *continuous* (elements of the system alter their state continuously over time in response to a constant stimuli) or *discrete* (state changes occur at points of time, e.g. periodic changes based on intervals, or due to an event, e.g. traffic light control changes its signal to yellow);
- *microscopic* (system entities and interactions are represented at a high level of detail) , *mesoscopic* (most entities are represented at a high level of detail but activities and interactions are represented at a lower level of detail), or *macroscopic* (entities, interactions, and activities are represented at a low level of detail);
- *deterministic* (entities and interactions are defined by exact relationships of mathematical, statistical, or logical; hence there is no random variables, only constant values are used) or *stochastic* (probabilities functions are used to determine entities and interactions) [Lieb05], [Medina05].

In [Miller07], the presented list of taxonomy of simulations is more exhaustive. It contains the following classifications:

- platform (which operating system it runs on);
- source code availability (whether it is open source or closed);
- cost (whether we need to pay to license it);
- transportation network (whether it is *free-flowing* – such as freeways, or *regulated* with traffic controls);

- underlying method of use (whether it is a *simulator* or driving *emulator*);
- vehicle model (*microscopic* – bird’s eye view of the traffic, or *macroscopic* – vehicle’s individual view);
- data input manner (whether vehicle’s location and speed data are changing *continuously* or being determined in *discrete* location or time period);
- data gathering manner (whether vehicle’s data are *communicated* back from each vehicle to the application or the *traditional* manner where no communications are assumed and data about vehicles are sensed using roadside sensors, such as loop detectors).

Popular traffic simulators that have widespread use worldwide [Miller07], such as CORSIM/TSIS [Owen00], MITSIM [Yang96], Paramics [Camer94], RENAISSANCE [Wang06], VATSIM [Redm99], and VISSIM [Fellen94], are evaluated in [Miller07] along with their proposed simulation, FreeSim. Many of the existing simulators are not free to own and license [Miller07]. Only FreeSim and MITSIM are free and have made their source code open to the public. MITSIM can only be used on Linux platform, whereas FreeSim can be used on any platform [Miller07].

Apart from the cost, platform, and the source code availability, in order to provide vehicle and collision data from vehicle sensors and traffic controls (such as mentioned in Section 3.1) for the learning component of the U&I Aware Framework, we need an application that can simulate an intersection with the following characteristics:

- Simulate both free-flowing (no traffic light) and regulated traffic (with traffic lights). Only CORSIM/TSIS [Owen00], MITSIM [Yang96], Paramics [Camer94], VATSIM [Redm99], and VISSIM [Fellen94] have both free-

flowing and regulated simulations. FreeSim [Miller07] and RENAISSANCE [Wang06] only support free-flowing traffic simulation.

- Model both microscopic (as users and the intersection agent need to see the global view of the intersection) and macroscopic (as we should be able to track each vehicle's status). Most of the existing simulations can only accommodate either macroscopic or microscopic view. Only FreeSim [Miller07] and VISSIM [Fellen94] can represent both views.
- The simulation must be able to support both continuous and discrete input. Vehicle data should be changing continuously, each vehicle should have independent behaviours in terms of the speed, acceleration, manoeuvres, etc. There can be random number of naughty vehicles that will disobey the rules, speed limit, and so on. A certain degree of discrete input needs also to be simulated, for example, speed changes when the vehicle is entering the centre of the intersection and the traffic light controller is turning to yellow or red. Most traffic simulators (CORSIM/TSIS [Owen00], MITSIM [Yang96], Paramics [Camer94], RENAISSANCE [Wang06], and VISSIM [Fellen94]) cannot simulate dynamic variation of vehicle data continuously. The vehicle speed data are inputted at certain discrete location and time in the simulation. Only FreeSim [Miller07] and VATSIM [Redm99] can support continuous data input.
- Data should be gathered from individual vehicles via communication, instead of adopting the traditional manner (i.e. where there is no communication assumed and data are gathered from roadside sensors), since we rely on vehicle sensors to gain information about each of the vehicles in the vicinity. At this stage, only FreeSim [Miller07] can support the non-traditional data gathering manner (directly communicated from vehicles).
- Both stochastic and deterministic models need to be incorporated in order to cater both the regular nature of some intersection components (e.g. traffic

light interval changes) and the unpredictability that can occur in an intersection (e.g. vehicle speed changes, vehicle congestion, etc.)

There is no existing simulator that fulfils all the combination of the above requirements. The only application that can simulate both free flowing and regulated traffic can only model the microscopic view and the traditional way of data gathering (no communication between vehicles and the central system). FreeSim, although it is free and open source, cannot fulfil all the above requirements, since it only simulates free-flow traffic and was released in 2007 (after this research commenced). Therefore, we have developed our own four-leg cross intersection simulation, which is further explained in the following subsections.

4.1.1 An Overview of the Simulation Environment

The purposes of the development of this simulation are as follows:

- i. to generate collision and traffic data that resemble real-world sensor data;
- ii. as a test-bed for collision detection and evaluation of the U&I Aware Framework.

Since this chapter mainly focuses on data collection and mining, we are not going to discuss how the simulation is instrumented for collision detection as this is covered in Chapter 5. This section focuses only on the first objective, which is the development of the simulation to generate traffic and collision data.

In order to generate traffic and collision data, it is necessary to design and implement a simulation that can meet the requirements stated in Section 4.1:

- We simulate both free-flowing (no traffic light) and regulated traffic (with traffic lights). We have created a simulation with two different scenarios:

intersection with traffic lights (Figure 4.4) and without traffic lights (Figure 4.5).

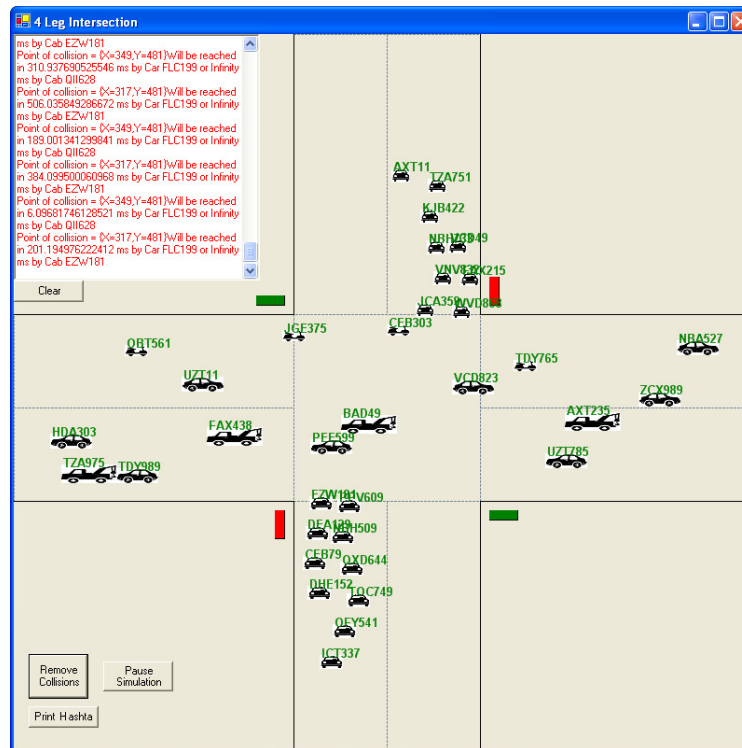


Figure 4.4. Intersection Simulation with Traffic Lights

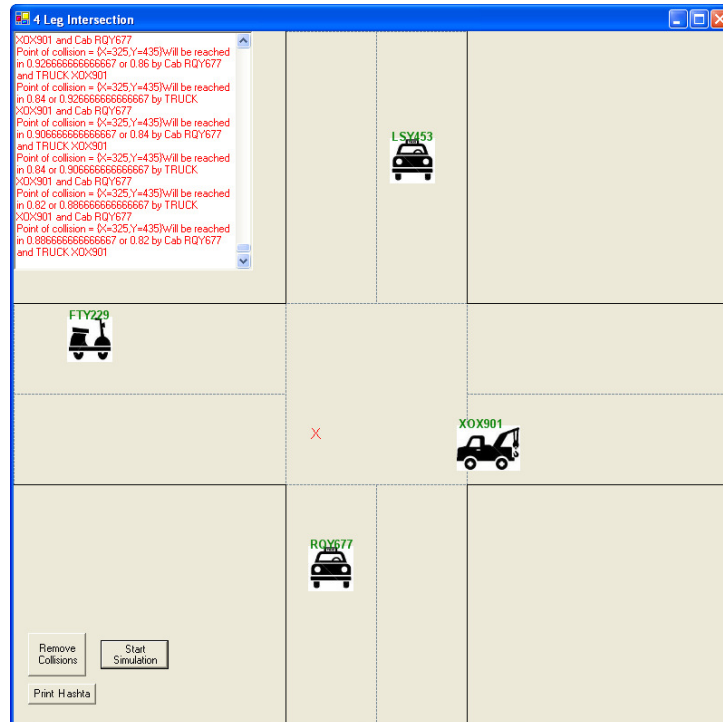


Figure 4.5. Intersection Simulation without Traffic Lights

- We model each vehicle agent as a separate entity that has attributes of its own (i.e. to achieve macroscopic view), yet, a collective global view (microscopic) of the whole intersection can also be obtained through the intersection agent.
- Since each vehicle agent is autonomous, it can change its behaviour independently (i.e. speed change, manoeuvre change). Randomly, “naughty” vehicles, which have the tendency to violate traffic rules, are created at run time.
- Since data should be gathered from individual vehicles via communication, we simulate data being sensed by vehicle sensors, transmitted by vehicle agents and deciphered by the intersection agent.
- Lastly, we consider both stochastic and deterministic properties in modelling the simulation. For example, in terms of vehicle generation, the simulation should be stochastic as vehicles should be generated at different legs of the intersection in various times. However, the frequency of the vehicle generation should be deterministic based on the varying peak and off-peak hours. The car following model and the speed changes are also both stochastic and deterministic.

In real-world situations, an intersection actually consists of a collection of various components nested within one another, hence, it is necessary to capture and simulate these components in our simulation. An intersection consists of intersection *legs*; each leg consists of *leg parts*: one is an *approach* (where vehicles are advancing to the intersection centre) and another is *outgoing* (where vehicles are moving away from the intersection centre); each leg part consists of multiple lane groups and each lane group may consist of multiple lanes. A lane group is a collection of lanes that possess the same rule of manoeuvres and turns (e.g. straight lanes, or right turn lanes). Each lane group has different allowable

manoeuvres, for example, in one approach leg, there can be three lane groups, one for turning right, one for going straight, and the last one for turning left. At a regulated intersection leg, when there are more than one lane groups, there can be multiple traffic controls, one for each lane group.

Apart from the intersection, the other main components of the simulation are vehicles and traffic controls (only at regulated intersections). Driver was considered as part of the simulation when driver behaviours and profiling are to be represented and learnt. However, at this stage, we have not fully developed the simulation of a driver. Hence, this is part of our future work and will not be further discussed here. The next subsection presents the design of the simulation model.

4.1.2 Designing the Simulation Model

Three major steps were taken in designing our simulation models:

- i. classification of the components and their parameters;
- ii. establishing the calibration requirements;
- iii. determining the degree of randomness in the simulation (to obtain both stochastic and deterministic natures of the real-world traffic in an intersection).

Firstly, we determine the main components of the simulation and their parameters, which are as follows:

- Intersection: intersection type, leg (size, count, angle, lane group), lane group (lanes, traffic control), lane (size, vehicle occupation);
- Traffic control: signal time, period, timer, rules of execution;
- Vehicle: speed, acceleration, size, type, position, angle, manoeuvre.

The next major step in implementing a traffic simulation is to determine the calibration of the model [Lieb05]. The parameters of the simulation components have been calibrated to mirror real-world situations so that prediction and learning may yield accurate results that reflect real situations. The calibration is necessary for measuring length, time, and hence, speed and acceleration. One unit in the simulation represents 0.1 metre in the real-world. One second in the simulation is the same as in the real-world. Since the simulation is graphical, it has a graphic refresh rate set on an interval. Hence, the interval value is considered in the calculation of speed, distance travelled, and acceleration of vehicles. Each vehicle that is generated has a proportionate width, length, and size in comparison to the parameters of the intersection.

Apart from having a proper calibration, in order to resemble real-world situations, it is also necessary to determine the degree of randomness of the simulation. The combination of the stochastic nature of the simulation (where random variables are applied) and the deterministic nature need to be implemented as follows:

- *Vehicle Generation*: the density of vehicles generated in the simulation is deterministic, but the distribution of the vehicles (the location where generated vehicles are placed in the simulation) is stochastic. The simulation needs to be able to simulate various vehicle densities based on different time of the day and peak or off-peak hours. The density of vehicles is simulated deterministically as it is based on four different time schemes: morning, afternoon, evening, and dawn that are recorded in our intersection configuration file (see Table 4.1). During peak hours, more vehicles should be generated in the simulation, and vice versa. Hence, this is simulated by varying the interval of the timer used to prompt vehicle generation periodically. When more vehicles are to be generated (e.g. during peak hours), the interval is set to be smaller (calculated by the modulus of the

current timer period counter divided by *CarRegenerate*, a constant value provided in each time scheme recorded in the intersection configuration file). If stochastic behaviour is to be incorporated to the interval of the vehicle generation, a random number generator generates a number between the upper and lower limit of the *CarRegenerate* value. Based on the selected time scheme, the vehicles are randomly generated at a random time period (stochastic traffic distribution) with different speeds, manoeuvres, position and trajectory at the end of each intersection leg.

- *Car Following*: the speed, acceleration, and deceleration of the car following model (between a leader-follower pair in the same lane) is stochastic, however, the following distance is deterministic. The recommended safe following distance and safe stopping distance are three seconds from the vehicle ahead, as a general written rule stated in [ATSB06b] and [Auburn05]. Hence, those rules are followed in the simulation. The speed of the vehicle depends on the current traffic light colour. If it is green, a random number between the upper and the lower bounds of the normal speed of the vehicle type (e.g. scooter, sedan, truck) is generated. A vehicle can only speed up to the speed limit within the safe following distance behind the vehicle ahead, except if it is a naughty vehicle (which is generated randomly and has the chance to exceed the speed limit of the intersection). When the traffic light is yellow, a vehicle can increase its speed (using the random number generator to return a speed value higher than the normal speed threshold) in order to beat the red light if there is no vehicle ahead within the safe following distance; otherwise, smooth braking is applied. When the traffic light is red, a vehicle runs in the normal speed until before it reaches the safe stopping distance and then smooth braking is applied.
- *Smooth Braking*: the deceleration value of smooth braking is stochastic, as it uses the random number generator to create a fraction to be calculated against the current speed. The smooth braking is applied when a vehicle is reaching

the intersection centre, within the safe following distance or the safe stopping distance.

- *Traffic Light*: the interval of the traffic light is deterministic. It is important for traffic light controls in a simulation to follow a specific interval and sequence. For example, in our simulation, the green light period is set to a constant value of 13 seconds, the yellow light period is 2 seconds, and then the traffic light colour changes to red at the same time as the other set of the traffic lights change to green. Hence, the red period of a traffic light is 15 seconds.

Table 4.1. Intersection Configuration File

Parameter Name	Description	Example of values
<i>Mode</i>	indicates the time scheme	Morning, Afternoon, Evening, Dawn
<i>Time</i>	time period in the simulation	6am-12pm, 12pm- 6pm, 6 pm-12 am, 12am-6am
<i>Peak</i>	signifies if the intersection is on peak or non-peak hours mode	Yes, No
<i>CarRegenerate</i>	an integer as a division value of the timer's counter; the smaller the number, the more vehicles are generated, hence the intersection is more crowded.	30 (when modulus of the current timer period value divided by 30 is 0, new vehicles are generated)

The implementation details of the intersection components are further discussed in the next subsection 4.1.3.

4.1.3 Implementation of the Intersection Simulation

The intersection simulation is developed in the Windows environment with Microsoft Visual Studio .NET and the C# language. Each intersection object itself maintains a number of different hash tables, each for a different object collection. We would need to randomly access the object in the collection most of

the time rather than sequential access. Hence, hash tables are used for collections to speed up the random access to the objects in the collection as opposed to other means of collections (e.g. array, linked list, etc), using the key-value pair mechanism. The hash tables that are further discussed in this section are: *LegBuffer*, *Vehicles*, and *TrafficLights*.

The first hash table is called *LegBuffer*, which is a collection of all the Leg objects within an intersection. The leg object maintains information such as the position and size of itself, a textual name attached to it (for example: LEFT), and object references to the parts of the leg (namely *LegPart*): the approach leg (where vehicles are entering the intersection or travelling towards the intersection centre) and the outgoing leg (where vehicles are leaving the intersection or travelling away from the intersection centre). The *LegPart* object holds references to lane groups.

Since we mirror the simulation to the real-world situations as close as possible, we follow the calibration rule in our simulation (i.e. 1 unit in the simulation is equal to 0.1 metre in the real world). In our cross intersection simulation, the length of each intersection leg is 30 meters (300 units in the simulation) and the width of each intersection leg is 20 meters, with 15 meters width for each of the two leg parts.

The *LegBuffer* hash table's keys are the leg's textual names (e.g. LEFT) and the values are inner / nested hash tables (i.e. leg part hash tables), which have keys that contain either "Approach" (indicates an approach leg part) or "Outgoing" (indicates an outgoing leg part) and values that contain nested hash tables. These hash tables inside each of the leg part hash tables store references to the vehicles that are currently located in that particular intersection's approach / outgoing leg part. The key of the hash table is vehicle registration number as the key needs to

be unique; the value is the object of that vehicle. The structure of this three-tiered nested hash tables are illustrated in Figure 4.4. These hash tables are used most of the time to track individual vehicle's movement and overall traffic around the intersection.

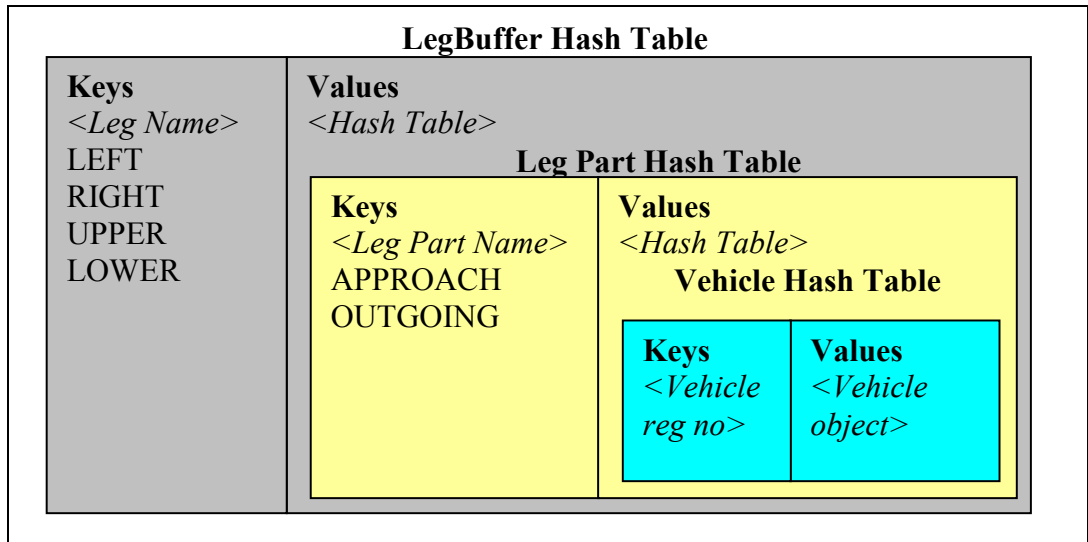


Figure 4.6. Content of LegBuffer Hash Table

Another main hash table is *Vehicles*, which is similar to the vehicle hash table that is nested within the leg part hash table of the leg buffer hash table. Vehicles hash table stores all vehicle object references that are currently at the intersection. The reason why Vehicles hash table is needed is because a quick retrieval of vehicle information is necessary, such as for drawing all the vehicles at the intersection at every 5 milliseconds (the graphic refresh rate). Whenever a new vehicle is created by the simulation, it is added to the Vehicles hash tables and the vehicle hash table nested inside the LegBuffer hash table.

Vehicles are created based on the vehicle configuration file. There are four different vehicle types that are recorded in the vehicle configuration file: scooter, small sedan, large sedan, and truck. Each of the types has different sizes and

range of speed that are scaled to real-world measurements. The parameters of each vehicle type in the configuration file are listed in the Table 4.2.

Table 4.2. Vehicle Configuration File

Parameter Name	Description	Example of values
<i>Type</i>	a textual name of a vehicle type	Cab, Truck, Sedan
<i>Position</i>	indicates the leg name	LEFT, RIGHT, UPPER
<i>XSize</i>	length of the car	25 (equal to 2.5 metres in the real world)
<i>YSize</i>	width of the car	15
<i>Angle</i>	angle of the car in relevance to 0° straight horizontal line	90
<i>Normal Speed</i>	the speed of the car when entering the intersection, measured in km/h	50
<i>Approaching Intersection</i>	indicates whether the vehicle is in the approach leg or outgoing leg	True, False
<i>Current Intersect Name</i>	refers to the name of the intersection where the vehicle is initially created	CrossIntersection
<i>Moving Direction</i>	signifies the planned travel direction of the vehicle expressed in the series of intersection leg names	BOTTOM CENTRE UPPER
<i>Image</i>	the file name of the image to be used to draw the vehicle	cab_from_front.jpg

The vehicles should follow several traffic rules, e.g. the traffic light signals and the speed limit. Therefore, in order to generate collision events, we simulate “naughty vehicles”. Random “naughty” vehicles are generated in the simulation so that its impact on road safety can be analysed and to test the ability of the collision detection and learning algorithms. The probability of naughty vehicles at the intersection is 1:5. When a naughty vehicle is generated, its speed will be a random number up to 40 km/h above the speed limit. Other natural and naughty driving behaviours are also simulated at the intersection. For example, when a vehicle is located at the front line of the intersection leg and the traffic control turns to yellow, the vehicle will attempt to beat the red light. When a vehicle is passing the intersection centre during yellow light, it will increase its speed.

All the traffic light controllers at an intersection are stored in the *TrafficLights* hash table. A traffic light control does not control the whole approach leg. Instead it controls a lane group. Therefore, there can be more than one traffic light in an approach leg if there is more than one lane group. Whenever a new traffic light controller is created, the reference is added in the *TrafficLights* hash table and also in the relevant lane group object. Each traffic light has a reference to the *TrafficControlRule* object, which holds and manages the *TrafficLights* hash table. Each *TrafficLight* is run by the *TrafficControlRule*. When it is the time for a traffic light to turn to green, the *TrafficControlRule* enables the timer of that traffic light, and the green period starts, and the timer keeps ticking until the green period is over, then the timer is disabled and the traffic light turns to red. Just before the traffic light turns to red, it will notify *TrafficControlRule*, which will then execute the other traffic lights that should turn to green, enable their timers, and so on.

Using an existing method in the Visual Studio .NET to check if one rectangle intersects with another, the simulation is able to identify collisions that exist in the intersection simulation. Once a collision is identified, the vehicles involved in the collision are disabled, and then removed from the simulation in few milliseconds after data about the collision has been recorded.

When the simulation is run (Figure 4.4), data from traffic and collision events generated from the simulation are recorded in log files. Vehicle data that consist of speed, angle, position, direction, size, and manoeuvre that are required for collision detection calculation (see Figure 4.1) can be easily obtained from in-vehicle sensors (as discussed in Section 3.1).

The following figures (Figure 4.7, 4.8, and 4.9) are samples of data that can be generated from our intersection simulation. Each data set is collected for each

case of learning analysis. Different combinations of attributes are taken to feed the data mining algorithms. For example, whenever there is an event of collision at the intersection, it is recorded as shown in Figure 4.8 and Figure 4.9. In Figure 4.8, speed, distance to intersection, traffic light colour, and collision point data are recorded as those attributes may allude to traffic rule violations associated with collisions that occur at the intersection. In Figure 4.9, a collision event with attributes of manoeuvre, direction, angle, and collision type are recorded as these attributes may describe a collision pattern. In addition, apart from collision event data, aggregate traffic and collision data (Figure 4.7) are collected periodically. At this stage, we have up to six different scenarios where different sets of sensor data are simulated and collected every 5 milliseconds in our simulation, which produces up to 6.78 MB of data per minute. The frequency of the readings can be adjusted; however, we set 5 milliseconds for the purpose of measuring the scalability and performance of the system. The log files are in comma separated values (.csv) format, which can be used in many data mining applications. The data collected in our simulation can be useful for Road Safety Analysis (RSA).

Avg Traffic Volume	Avg Speed	Total Collisions	Total Side Collisions	Total Rear End Collisions
7	48	0	0	0
13	48	1	0	1
16	33	1	0	1
18	38	0	0	0
20	31	1	0	1
19	35	1	0	1
18	34	1	0	1
18	31	0	0	0
19	24	2	1	1

Figure 4.7. Periodic Traffic Data

SpeedCar 1	Distance ToInters Car1	TLCOLOR1	SpeedCar 2	Distance ToInters Car2	TLCOLOR2	CollisionP ointX	CollisionP ointY
30	-145	0	0	-8	1	436	306
50	-298	0	55	-273	0	568	359
37	-117	0	30	-145	0	415	321
42	-94	0	37	-117	0	387	332
44	-253	0	55	-273	0	543	359
57	-288	0	55	-337	0	212	433
43	-70	0	42	-94	0	364	334
50	-269	0	54	-235	0	430	534
54	-209	0	54	-235	0	433	500

Figure 4.8. Collision Event Data with Attributes of Speed, Distance, Traffic Light Colour, and Collision Point

Veh1_ Manoeuvre	Veh1_ Direction	Veh1_ angle	Veh2_ Manoeuvre	Veh2_ Direction	Veh2_ angle	Coll_Type
STRAIGHT	DOWN	90	STOPPED	LEFT	0	SideCollision
STRAIGHT	DOWN	90	STRAIGHT	DOWN	90	RearEndCollision
STRAIGHT	UP	90	STRAIGHT	UP	90	RearEndCollision
STRAIGHT	DOWN	90	STRAIGHT	DOWN	90	RearEndCollision
STRAIGHT	DOWN	90	STOPPED	LEFT	0	SideCollision
STRAIGHT	UP	90	STRAIGHT	UP	90	RearEndCollision
STRAIGHT	DOWN	90	STOPPED	LEFT	0	SideCollision
STRAIGHT	DOWN	90	STOPPED	LEFT	0	SideCollision

Figure 4.9. Collision Event Data with Attributes of Manoeuvre, Direction, Angle, and Type.

We have assumed the implementation of manoeuvre prediction in our simulation based on [Oliver00b] by enumerating the manoeuvres that can be predicted by [Oliver00b], which include: passing, turning right, turning left, changing lane right, changing lane left, starting, and stopping. Since we currently only simulate straight vehicle movement on a single lane in each leg (i.e. there is no lane change capability incorporated in the simulation yet), only three manoeuvres are practically in use: STRAIGHT, STOPPING, STOPPED (Figure 4.9). The values of direction generated by the simulation can be: LEFT, RIGHT, UP, and DOWN (Figure 4.9). These values exhibit the intersection leg destination of the vehicle. In combination with the vehicle manoeuvre data, we can infer the trajectory information. LEFT direction with STRAIGHT manoeuvre signifies that a vehicle is travelling from right (west) leg of the intersection to the left (east) leg. UP

direction with STRAIGHT manoeuvre signifies that a vehicle is travelling from the bottom (south) leg of the intersection to the upper (north) leg of the intersection. Consequently, the simulation can only generate rear-end collisions and side collisions, which are the only collision types recorded in the collision event data (Figure 4.9). Although the collision data generated from the simulation denotes the collision types in the intersection, the collision patterns that pertain to the intersection (i.e. the combination between intersection characteristics and collision types) need to be learnt using data mining.

In order to derive more meaningful information for the knowledge base of the U&I Aware Framework and to facilitate collision detection, it is necessary to mine for trends and patterns in the intersection, such as follows:

- As seen in Figure 4.7, the simulation is able to output a periodical traffic data. In the sample data, each row is recorded every four seconds (i.e. the average traffic volume and average traffic speed is accumulated and calculated every four seconds). The average traffic volume and average speed of traffic are compared with the total number of collisions and total collisions of each collision type. This data is collected from various period (i.e. peak/off-peak hours, morning/afternoon/evening/dawn). The data can be more meaningful if we can learn the correlation among the varying traffic volume, varying average traffic speed, and the increasing or decreasing number of collisions. Hence, through this data, we can learn and identify the changes of traffic conditions and traffic hazard levels at the intersection.
- In Figure 4.8, each collision event in the simulation is recorded with attributes of speed of each vehicle in the colliding pair, distance to intersection of each vehicle in the colliding pair, traffic light colour, and collision point of each vehicle in the colliding pair. By analysing this data, we can learn the correlation between dangerous driver behaviours and traffic violations that may lead to collisions. This particular collision event log files contains data

about the current traffic light colour, the current speed of the vehicle before collision, the location of the collision, and the distance to intersection centre (whether the vehicle is located at the intersection centre, before crossing the intersection centre, or after crossing the intersection centre). As those attributes may well be related with traffic rules violation, applying mining techniques to such data may extract trends of dangerous driver behaviours, red light running behaviours, and violation of speed limit.

- In Figure 4.9, each collision event in the simulation is recorded with attributes of manoeuvres, vehicle direction, and angle of each vehicle in the colliding pair, and the collision type. It is essential to learn collision patterns from these data. By mining this data, the collision patterns learnt at the intersection (that include the combination of various manoeuvres, direction, and angle of each vehicle in the colliding pair in a particular collision type) can be extracted to be used as the basis for collision detection. It is also necessary to learn and identify the collision pattern with the highest occurrence. Hence, in the event of occurring vehicle pairs that match this pattern, collision detection can take precedence.

Note that the frequency of collisions in the simulation does not correspond to the frequency of collisions in the real world. In this simulation, the number of collisions is much higher in comparison with the real-world situations. This is due to simulated (simplified) vehicles, which when in the path of collision, as has been previously detected, will eventually collide. This is because our simulation is designed to focus on generating collision data at this stage. Note also that, in a real-world setting, it is necessary that not only data about actual collisions should be used for analysis, but also data about near-misses (i.e. collisions that were likely to happen but avoided due to braking or steering action of the drivers) can also be included in the analysis to provide indicative trends. There is no minimum threshold of data required in order to commence data mining. Once

collision or near-miss data exists, mining can commence. Nevertheless, if there is more data collected, it may entail higher confidence/support of the rules extracted from the mining results.

As such, collisions and near-misses will constitute only a small segment/percentage of real-world data. In such cases, outlier analysis (i.e. a data mining technique used to focus on exceptions) may also be undertaken. Outlier analysis is widely used in applications such as credit card fraud detection, where the focus is on a small percentage of fraudulent transactions [Aggarw05].

We apply data mining techniques to the data collected in the simulation. This is presented in Section 4.2, where each learning scenario is discussed further with the data set and learning algorithm used.

4.2. Mining Intersection Traffic and Collision Data

Data mining is a powerful means of extracting valuable patterns from traffic and collision data. Given real-time or historical and traffic or collision data of an intersection, data mining can be used to characterise information that is pertinent to a particular intersection, such as:

- collision patterns;
- patterns of intersection's conditions or behaviours during non-collision-free periods (to determine dangerous traffic trends);
- patterns of driver's conditions or behaviours during non-collision-free periods (to determine dangerous driver behaviours).

Note that the above list is not exhaustive. It only encompasses the subjects that are considered in this thesis. There can be other application areas where data mining can be found useful to improve intersection safety.

Given the various characteristics and collision patterns in intersections, an intersection collision warning and avoidance system that is applicable for all types of intersections is required. Since each intersection is unique and has different characteristics from other intersections, it also has different sets of collision patterns. Understanding collision patterns of an intersection is essential in order to identify dangerous situations in an intersection and foresee similar situations through the patterns that are already known. Thus, to achieve the goal of this research, which is to develop a generic and adaptive intersection collision warning and avoidance framework, it is necessary to learn collision patterns of the particular intersection where the system is located. It is also essential to monitor the occurrences of the collision patterns learnt at the intersection for threat detection and safety enhancements. When collision patterns of an intersection are identified, those collision patterns are maintained in the knowledge base of the particular intersection and such knowledge can be used as the basis for threat assessment, collision detection, warning and avoidance, and intersection site maintenance. In road safety research, Road Site Analyses (RSA) normally includes learning of collision patterns (as discussed in Chapter 2, Section 2.1). Collision patterns learning in RSA research is customarily performed manually through human observations. Furthermore, although research projects that develop intersection collision warning and avoidance systems also include learning of collision patterns ([Verid00], [Fuers05]), those research projects do not employ automated learning techniques. This section discusses how data mining can be used to extract useful information from intersection traffic and collision data and how the information can be used in the knowledge base.

Traffic data captures information about average speed, average traffic volume, total throughput, total number of collisions, etc in a period of time (for example,

see Figure 4.7), which are useful for analysing the efficiency and safety of a road segment for that period. Collision data captures details about a collision event that occurs at a particular road segment. There can be various details recorded, such as speed, distance to intersection, traffic light presence, traffic control rule, collision point, angle, direction, type, etc, which can be maintained for different learning purposes (samples of such data can be seen in Figure 4.8 and Figure 4.9). Both traffic and collision data can be mined offline using historical data or online using online data captured from sensors in real-time [Gaber05].

By having a knowledge base in the system that maintains the traffic or collision patterns of the particular intersection, thus, characteristics that are specific to a particular intersection can also be learnt and incorporated into the knowledge base of that intersection. Note that data mining is not suggested to replace existing procedures. The knowledge base can initially be filled with expert knowledge or rules learnt through existing process or manual observation. However, the usage of data mining can also supplement and enhance existing procedures for learning of collision and traffic patterns. The patterns learnt as a result of data mining can be consolidated in the knowledge base.

In addition, the efficiency of the conventional collision warning system that is based on the brute force approach can be improved by utilising collision patterns as the criteria for identifying or selecting the vehicle pairs that are candidate for potential collision. Collision patterns that contain definitions of possible traffic conflicts (a traffic conflict is a relationship between two road users on a collision route [Sauni07]) at the intersection are maintained in the knowledge base to be used as the basis for preselection. We will present our strategy for preselection (i.e. a mechanism that increases the efficiency of collision detection algorithms) and its performance implication in Chapter 5.

We propose a general methodology for mining traffic and collision data, which is as follows:

- i. *Identify the nature of the problem and the goal of learning.* We need to first identify the issues to be addressed as well as the goal of applying data mining techniques in each scenario. The expected learning outcome of the scenario must first be decided. For example, in the case of traffic data collection, a sudden change in the figures of traffic flow and volume may indicate the occurrence of a traffic incident. Hence, the expected outcome of such learning scenario may be to extract patterns of normal traffic characteristics that can be used to detect changes or anomaly.
- ii. *Identify the method to be used.* It is necessary to establish the correct data mining method (e.g. clustering or classification) in dealing with the issues depicted in each scenario. If there is no existing class labels attached to the data, classification cannot be performed [Witten05]. Therefore, clustering should be performed prior to classification in order to view how the data spread across various cluster groupings and to extract the appropriate class labels for each cluster. Otherwise, classification can be performed directly when class labels exist. However, Witten and Frank also suggested that clustering can improve the accuracy of classification when there is an existing pool of both labelled and unlabelled data [Witten05].
- iii. *Identify the technique to be used.* There are many existing data mining techniques in each method. However, each technique has various input requirements and output models. Therefore, it is necessary to assess the input data that can be accepted by the learning algorithm. Some learning algorithms can only accept numeric values, some can accept only nominal values, and only a few can accept both. The output models can also vary, such as probabilistic models (i.e. Bayesian Network), tree structure (i.e. decision tree), or formula (i.e. regression techniques). Some techniques also required specific input parameters. For example, k-means clustering algorithm requires the

number of clusters to be specified. Therefore, it is essential to consider those aspects in choosing the most appropriate technique to be used to analyse data in a specific case or scenario.

- iv. *Identify the technique for validation.* Since this research is vital in terms of considering its impact on road safety and reducing fatalities, it is necessary to identify another technique to validate the outcome of the first learning technique.
- v. *Identify implementation strategy.* Once the previous steps have been established, we need to decide on the data mining tools, platform, and devices that are to be used and perform implementation.
- vi. *Compare, analyse, and evaluate results.* The results of data mining need to be analysed and interpreted by the users. Consequently, the results need to be visualised or presented in a way that can be understood.
- vii. *Integrate with the knowledge base.* Finally, how the rules or trends (acquired through data mining) are represented in the knowledge base needs to be decided. Interesting and useful patterns retrieved from the data mining process can supplement existing patterns or rules in the knowledge base of the intersection.

Each stage of the methodology needs to be dealt specifically for each learning scenario. However, in the light of the aim of this thesis, the main purpose of applying data mining is for real-time collision detection and the adaptability of the U&I Aware Framework to various intersections. In order to facilitate generality of the framework to various intersections, a knowledge base is employed along with data mining. The knowledge base is utilised as the basis of the preselection method (i.e. search mechanism to identify vehicle pairs that have the likelihood to collide). For the purpose of preselection, the knowledge base of the U&I Aware Framework can be set on two different system modes, which are *optimistic* setting and *pessimistic* setting. This is described as follows:

- If it is set to be pessimistic, it takes into account all the collisions patterns stored in the knowledge base, including those with low probability of occurrence, and uses them to identify vehicle pairs that are likely to collide;
- If it is set to be optimistic, it only considers the most frequently occurring collision patterns at the intersection and ignores the rest. Thus, the system identifies vehicle pairs that are likely to collide based on the collision patterns that have high probability of occurrences at the intersection.

Hence, it is essential to retrieve data mining results that can be used to populate the knowledge base. Data mining is applied to extract collision patterns that pertain to the intersection as well as to identify the most frequently occurring collision patterns. There are two categories of collision patterns in the knowledge base: *generic* and *specific* collision patterns. A *specific collision pattern* is made of a collision pattern name, the manoeuvre, leg position and direction of the first vehicle, the manoeuvre, leg position and direction of the second vehicle, and the collision type. It is used to signify a unique characteristic (e.g. the most frequently occurring collision pattern in the intersection). For example, when vehicles located on the left leg with straight manoeuvre and are travelling to the right are most likely to collide with vehicles located on the upper leg travelling down with straight manoeuvre, but not with vehicles from other directions or vehicles that entail other manoeuvres. Hence, a specific collision pattern should be created to describe such situation. On the other hand, a *generic collision pattern* is described by the geometry of the conflict path and the manoeuvre of each vehicle in the vehicle pair. Since it does not involve a description about a particular leg location or direction, the pattern depicts that the conflict path may occur anywhere at the intersection. Every pair of vehicles that are travelling with the same manoeuvre pair set and form the geometry as portrayed in the generic collision pattern is to be identified as potentially conflicting vehicles. A generic collision pattern generalises a specific collision pattern by assuming that a

particular pattern can generally occur in any leg location with any direction pair as long as it has the same manoeuvre pair combination and the same collision geometry. A generic collision pattern is not the same as a collision type since a collision type only considers the geometry of a collision and does not include any manoeuvre combination.

When using collision and traffic data, it is necessary to mine both trends and rules that can be consolidated into specific and generic collision patterns for the knowledge base. We commence learning by applying unsupervised learning to the collision and traffic data. This is particularly useful when there is no expert knowledge about existing collision patterns that pertain to the intersection stored in the knowledge base. Exploratory analysis is performed using a range of techniques. We use existing classification and clustering algorithms that have been previously developed. The Weka library of data mining algorithms [Witten05] is used for learning from historical data. Since we have only performed offline learning of collision patterns and dangerous traffic trends, the other scenarios are not addressed in this thesis. The description, motivation, algorithms used and results in the following learning scenarios: (i) learning collision patterns and trends is discussed in subsection 4.2.1, (ii) learning dangerous traffic trends is discussed in subsection 4.2.2, and (iii) learning dangerous driving trends is discussed in 4.2.3.

4.2.1 Collision Patterns Learning

The purpose of learning collision patterns is to extract specific trends of existing collision types in a particular intersection. As previously discussed, collision patterns vary from one intersection to another due to variations of intersection characteristics and collision types (e.g. side collision) that may occur in the

intersection. A collision pattern involves data about the collision type and attributes of a colliding vehicle pair, such as manoeuvre, direction, and location. This data can be obtained from sensors in the real world. For evaluation purposes, this data is generated by our traffic and collision simulation. To learn collision patterns and trends, the simulated data (Figure 4.9) has seven attributes, which are *direction*, *manoeuvre*, and *angle* from each of the colliding vehicle pair (i.e. vehicle 1 and vehicle 2), and *collision type*. Whenever there is a collision or near-collision event in our intersection simulation, data from the colliding (or near-colliding) pair of vehicles are collected and mined. Near-collision events are set by a threshold value of distance between two vehicles that almost collide with each other.

During preselection, each SV is paired up with one or more POV based on the current directions and manoeuvres (e.g. straight, stopped, and stopping) of both vehicles. When a Subject Vehicle (SV) is travelling from one particular intersection leg with a certain direction, manoeuvre and angle, it is necessary to assess the pattern that exhibits the directions and leg locations of Principal Other Vehicles (POVs) that have the possibility to collide with the SV. When such information is known, we can eliminate the process of checking the SV with each and every other vehicle at the intersection for possibility of collision. Instead, the SV is only compared with the POVs that exhibit the travel direction, location, and manoeuvre that collide with SV's travel direction, location and manoeuvre according to existing collision patterns in the knowledge base.

In our exploration to discover patterns from the collision data, we are interested to find clusters of collision patterns and observe the distribution of the collision data across the clusters. Thus, unsupervised learning needs to be performed to find clusters of collision patterns. The collision event data (Figure 4.9) contain seven attributes, i.e. *direction*, *manoeuvre*, and *angle* from each vehicle in a

colliding pair, and *collision type* (side collision or rear-end collision). Using collision event data, we applied unsupervised clustering algorithms, such as the prevalent k-means and *EM (Expectation-Maximization)*. EM is an unsupervised clustering algorithm, where the expected class values or the cluster probabilities is firstly calculated, which is then followed by the calculation of the distribution parameters that maximize the likelihood of the distributions based on the data [Witten05]. The unsupervised learning using k-means algorithm only found the rear-end collision clusters. However, no side collision clusters are correctly shown, since the percentage of side collisions in the training data is much smaller than rear-end collisions. And there are also some unique instances do not belong to any discovered clusters. The discovery of such instances is not trivial. If k-means clustering technique is used, such unique instances are merged into the closest cluster centres. When EM Clustering technique is used, some of side collisions data are inaccurately merged into the closest cluster centres and some are merged into a separate cluster of side collisions. Such collision data should be considered as outliers or noise due to the uniqueness and small occurrences in the training data, however, both k-means and EM cannot deal particularly well with outliers or noise.

Therefore, we need to find a suitable unsupervised learning algorithm that can handle outliers well. Hence, we use *DBScan (Density Based Spatial Clustering of Applications with Noise)* to find clusters of collision patterns that pertain to the intersection from the collision event data since DBScan can recognise noise [Ester96]. DBScan performs much better than the K-means and EM algorithms implemented in Weka. In Figure 4.10, clusters of intersection collision data are visualised in the matrix of vehicle direction pair (veh1_direction and veh2_direction). The visualisation of DBScan clustering results shows six clusters in total (Figure 4.10) and regards few data items as noise. There are seven attributes in each data, which are veh1_manoeuvre (the manoeuvre of SV),

veh1_direction (the direction of SV), veh1_angle (the angle of SV), veh2_manoeuvre (the manoeuvre of POV), veh2_direction (the direction of POV), veh1_angle (the angle of POV), and coll_type (the type of collision). The clusters are listed in Table 4.3.

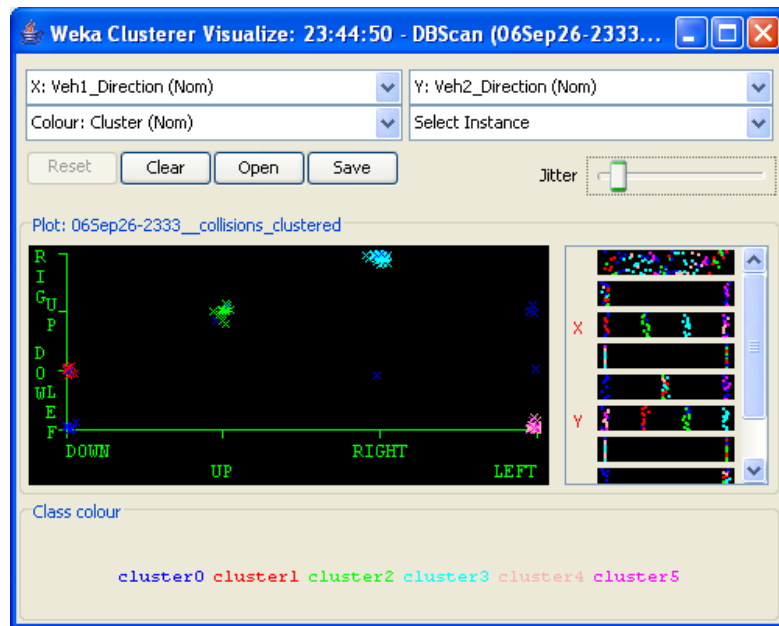


Figure 4.10. Collision Patterns Clustered by DBScan Algorithm with Vehicle Direction as Visualisation Category

Table 4.3. Clusters of Collision Event Data as Clustered by DBScan

Cluster No	Veh1_manoeuvre	Veh1_direction	Veh1_angle	Veh2_manoeuvre	Veh2_direction	Veh2_angle	Coll_Type
0	STRAIGHT	DOWN	90	STOPPED	LEFT	0	Side
1	STRAIGHT	DOWN	90	STRAIGHT	DOWN	90	Rear-end
2	STRAIGHT	UP	90	STRAIGHT	UP	90	Rear-end
3	STRAIGHT	RIGHT	0	STRAIGHT	RIGHT	0	Rear-end
4	STRAIGHT	LEFT	0	STRAIGHT	LEFT	0	Rear-end
5	STOPPING	LEFT	0	STOPPING	LEFT	0	Rear-end

Based on table 4.3, we can see a number of specific collision patterns which are derived from two collision types and learnt from approximately 120 collision

instances that occur in the intersection. It shows a cluster of a side collision with STRAIGHT-STOPPED vehicle manoeuvre pair and DOWN-LEFT vehicle direction pair. The other five clusters depict collision patterns that are derived from rear-end collisions.

As stated in the methodology of mining traffic or collision data, a second mining technique needs to be applied for validation of results. A number of issues that require validation based on the results are as follows:

- i. Firstly, we need to classify the side collision patterns based on the vehicles' direction pairs. Using this technique, side collision instances are either regarded as member of cluster 0 or noise/outliers. This is because side collisions occur rarely in this particular intersection. This result by DBScan is better compared with k-means or EM that simply disregards side collision instances or inaccurately cluster side collision instances together with rear-end collisions. Thus, it is important to validate these results by performing classification on side collision instances.
- ii. Secondly, it is also necessary to learn the probability of occurrences of the collision patterns at the intersection. There are collision instances that are less frequent (or may only occur once) but still noteworthy to be learnt since a potential collision may be derived from learning those instances. However, there are also collision patterns that tend to occur more frequently in the intersection. When a pair of vehicles travelling in the intersection exhibit characteristics of the more frequently occurring collision pattern, the pair of potentially colliding vehicles should be prioritised for checking.
- iii. Thirdly, the clustering result also reveals the trends in vehicles' manoeuvre pairs. The common manoeuvre pairs of SV-POV in rear-end collisions are STRAIGHT-STRAIGHT and STOPPING-STOPPING. Whereas in side collisions, the common manoeuvre pairs of SV-POV are STRAIGHT-STOPPED and STRAIGHT-STOPPING. This trend needs to be validated by

applying classification techniques to find pairs of colliding vehicle's manoeuvres of each collision type.

Hence, for the purpose of validation, the next steps to be taken are to identify appropriate techniques and compare, analyse and evaluate the results.

(i) Classification of side collision patterns based on vehicle directions

For the purpose of this scenario, which is to learn the classification of side collisions, we generate only the side collision data (Figure 4.11) from the simulation, which has around approximately 60 side collision records when a simulation without traffic control is run for two to three minutes (since the intersection has no traffic control and the vehicles are not yet equipped with collision avoidance capabilities, there are more collisions expected than normal). A side collision involves vehicles that travel in two paths that intersect at a point. Hence, we exclude collisions that involve any pair of vehicles that travel in the same direction (parallel paths) or rear-end collisions. Six attributes are included (direction, manoeuvre, and angle from each vehicle in the colliding pair), as collision type attribute is excluded from the data (since all the data are about side collisions).

In order to perform classification of side collisions, and the vehicle directions, manoeuvres, and angles involved in intersection collisions, we propose that decision tree learning is to be applied. A decision tree is to be constructed based on the vehicle direction of the SV as the predefined input classes and the vehicle direction of the POV as the output values. A decision tree represents a simple structure of the input root nodes that can traverse to different branches (based on attribute value groupings) and corresponds to one or more leaf or terminal nodes (as output values) [Quinlan86]. The classification rules can be derived from the decision tree by traversing the tree nodes from one of the root nodes until a leaf node is reached. This data is used for the decision tree construction.

	A	B	C	D	E	F	
1	Veh1_Manoeuvre	Veh1_Direction	Veh1_angle	Veh2_Manoeuvre	Veh2_Direction	Veh2_angle	
2	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
3	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
4	STRAIGHT	RIGHT	0	STRAIGHT	DOWN	90	
5	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
6	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
7	STRAIGHT	UP	90	STRAIGHT	LEFT	0	
8	STRAIGHT	RIGHT	0	STRAIGHT	DOWN	90	
9	STRAIGHT	UP	90	STRAIGHT	RIGHT	0	
10	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
11	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
12	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
13	STRAIGHT	DOWN	90	STRAIGHT	RIGHT	0	
14	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
15	STRAIGHT	RIGHT	0	STRAIGHT	DOWN	90	
16	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
17	STRAIGHT	RIGHT	0	STRAIGHT	DOWN	90	
18	STRAIGHT	LEFT	0	STRAIGHT	UP	90	
19	STRAIGHT	RIGHT	0	STRAIGHT	DOWN	90	
20	STRAIGHT	RIGHT	0	STRAIGHT	DOWN	90	

Figure 4.11. Side Collision Event Data with Attributes of Manoeuvre, Direction, and Angle of Each Vehicle in a Pair

We have successfully classified types of side collisions or perpendicular crashes in a cross intersection using data mining. We applied the *C4.5* decision tree learning (implemented as *J48* classifier in Weka [Witten05]) and the second vehicle direction (*Veh2_Direction*) attribute is nominated as the class label. Classification with *C4.5* displays the most frequent vehicles' direction pairs given the *veh1_direction* as the nominated decision attribute. The implementation results (Figure 4.12) show the most common vehicle's direction pairs that exist within the particular intersection where the traffic data was acquired:

- If *veh1_direction* (direction of vehicle 1) = LEFT: *veh2_direction* = UP
- If *veh1_direction* = RIGHT: *veh2_direction* = DOWN
- If *veh1_direction* = UP: *veh2_direction* = RIGHT
- If *veh1_direction* = DOWN: *veh2_direction* = RIGHT.

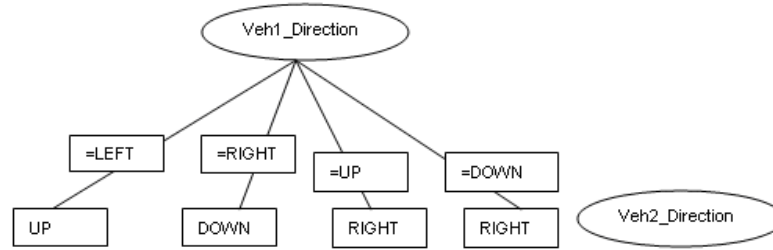


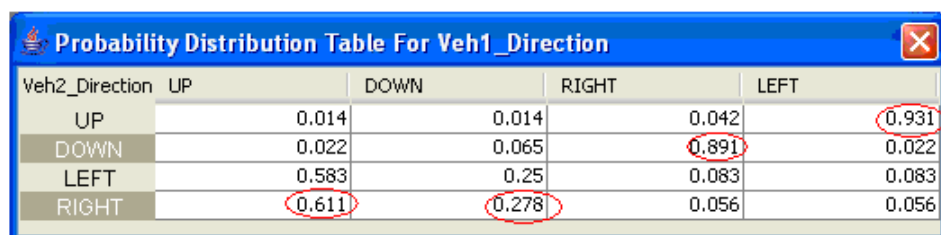
Figure 4.12. Side Collision Patterns Based on Vehicle Direction as Classified by C4.5

For example, our results, using randomly seeded data, show that a Subject Vehicle (SV) or vehicle 1 that travels with a straight manoeuvre from the right leg to the left leg of the intersection (veh1_direction = LEFT) tends to collide with a Principal Other Vehicle (POV) or vehicle 2 that travel with a straight manoeuvre from the lower leg to the upper leg (veh2_direction = UP). The result of this classification technique can be used in the following scenario. An SV is travelling on a high speed, hence there is not much time is available to compute collision detection. When the SV is to be assessed for collision detection, instead of pairing SV with every other vehicle at the intersection for collision detection computation, only POVs that exhibit the most common direction that collide with SV's direction are to be paired up with SV and computed for collision detection.

In order to assess the validity and consistency of the C4.5 classification result of side collisions, it is necessary to mine the probability distribution table of all the occurrences of side collision. A Bayesian network is a probabilistic graphical model of a direct acyclic graph form, which represents a joint probability distribution over a set of variables [Pearl88]. A Bayesian network includes all the possible nodes, variations of dependency between nodes and the probability values of each dependency set. Hence, a Bayesian network never excludes any possible inference of a node and dependency set. Therefore, it is very appropriate to build a Bayesian network in order to learn for all the possible side collision patterns and the probability of their occurrences.

A Bayesian classifier, *BayesNet*, an algorithm to learn Bayesian Networks using nominal attributes and with no missing values [Witten05], is used to classify the same data using vehicle direction as the class category. Bayesian Network yields probability estimation for each given instance in each class [Witten05], therefore the results of the C4.5 decision tree learning can be compared with the instances of the resulting BayesNet learning that possess the highest probability based on the class.

In our scenario, we enumerate four possible straight driving directions in a four legs cross intersection, which are left, right, up, and down. The classification shows the matrix of vehicle's direction pairs with the probability rate of each direction pair (Figure 4.13). The highest probability of a crash pattern in each direction is circled in red in Figure 4.13. Out of all the collisions that occur to vehicles that travel from the right leg to the left leg (i.e. "LEFT" direction), 93.1% of the collisions occur with vehicles from the lower leg to the upper leg (i.e. "UP" direction). This result conforms to the result of classification with C4.5 decision tree (Figure 4.12).



Veh2_Direction \ Veh1_Direction	UP	DOWN	RIGHT	LEFT
UP	0.014	0.014	0.042	0.931
DOWN	0.022	0.065	0.891	0.022
LEFT	0.583	0.25	0.083	0.083
RIGHT	0.611	0.278	0.056	0.056

Figure 4.13. The Probability of Side Collision Patterns Based on Vehicle Direction as Classified by Bayesian Network

In conclusion, the most frequently occurring vehicles' direction pairs as learnt with C4.5 and BayesNet classification techniques are listed as follows (format: SV_direction–POV_direction):

- UP-RIGHT: if an SV travels from the south leg to the north leg (UP direction), it will most likely collide with a POV that travels from the west leg to the east leg (RIGHT direction).
- DOWN-RIGHT: if an SV travels from the north leg to the south leg (DOWN direction), it will most likely collide with a POV that travels from the west leg to the east leg (RIGHT direction).
- RIGHT-DOWN: if an SV travels from the west leg to the east leg (RIGHT direction), it will most likely collide with a POV that travels from the north leg to the south leg (DOWN direction).
- LEFT-UP: if an SV travels from the east leg to the west leg (LEFT direction), it will most likely to collide with a POV that travels from the south leg to the north leg (UP direction).

The above results lead to composing the collision patterns that pertain to the intersection. Since a collision pattern includes not only the direction pairs of vehicles and collision types but also the manoeuvre pairs and optionally the leg location pairs, the side collision patterns listed in Table 4.4 are still partial.

Table 4.4. Partial Side Collision Patterns Based on the Direction Pairs

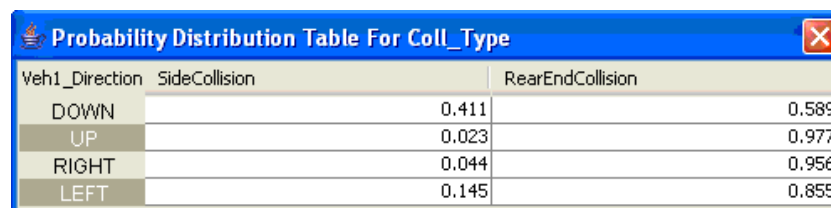
CollisionType	SV Direction	POV Direction
Side	UP	RIGHT
Side	DOWN	RIGHT
Side	RIGHT	DOWN
Side	LEFT	UP

(ii) Probability Distribution of Collisions

Since there can be numerous vehicles in the intersection, it is essential to identify the vehicles that should be prioritised for preselection. There can also be multiple collision types that have previously occurred in an intersection. It is also necessary to identify the most common or frequent collision. Hence, in monitoring the intersection, the priority should be given to check the potential occurrence of such collision that may occur again. For instance, vehicles

travelling eastward (i.e. RIGHT direction) from the left leg of the intersection to the right leg has the highest probability of side collision. Hence, in checking for future side collisions, the vehicles that are located on the left leg of the intersection are to be prioritised.

In order to set the right priorities for preselection, it is necessary to learn the probability of certain collision types and also the probability of various vehicles' direction pairs. As in the case of learning the probability of side collisions, to generate the probability distribution of the patterns of vehicles' direction pairs of any collision types, a Bayesian network classifier is appropriate. This is because the learning output of Bayesian classifiers are probability inference of the classes of data. *BayesNet* [Witten05] can be applied to mine the collision event data (Figure 4.9). To obtain the probability of collisions that involve vehicle pairs that travel either in parallel paths (rear-end collisions) or traversing paths (side collisions), we included both data of rear-end collision and side collision events that occur in the simulation in the data (Figure 4.9). In this particular intersection, when BayesNet is applied with collision type nominated as the class, the visualisation of the result shows that rear-end collision occurs much more often than side collisions in this particular intersection (Figure 4.14). Hence, it is appropriate to prioritise preselection and performing collision detection of rear end collisions over side collisions.



Veh1_Direction	SideCollision	RearEndCollision
DOWN	0.411	0.589
UP	0.023	0.977
RIGHT	0.044	0.956
LEFT	0.145	0.855

Figure 4.14. The Probability of All Collision Patterns Based on Collision Types as Classified by Bayesian Network

BayesNet is applied on the same data again and the directions of SV and POV are nominated as the class labels. The result shown in Figure 4.15 displays the probabilistic distribution of each possible direction pair in the intersection. From the Figure 4.15, we exclude all the collision patterns with the probability value less than or than or equal to 0.022 (e.g. UP-DOWN, LEFT-RIGHT) since these vehicle pair combinations do not exist in the collision event data. Hence, collision patterns learnt at the intersection based on the direction of the SV (Veh1_direction) are listed as follows (format: SV_direction-POV_direction): DOWN-DOWN, DOWN-LEFT, UP-UP, RIGHT-RIGHT, RIGHT-DOWN, LEFT-LEFT, LEFT-UP, LEFT-DOWN. Based on the results displayed in Figure 4.14 and Figure 4.15, partial collision patterns that consist of collision types, SV direction, and POV direction are constructed as listed in Table 4.5.

Veh1_Direction	LEFT	DOWN	UP	RIGHT
DOWN	0.397	0.569	0.017	0.017
UP	0.022	0.022	0.935	0.022
RIGHT	0.014	0.043	0.014	0.929
LEFT	0.828	0.047	0.109	0.016

Figure 4.15. The Probability of All Collision Patterns Based on Vehicle Direction as Classified by Bayesian Network

Table 4.5. Partial Collision Patterns Based on the Direction Pairs

CollisionType	SV Direction	POV Direction
RearEnd	UP	UP
RearEnd	DOWN	DOWN
RearEnd	LEFT	LEFT
RearEnd	RIGHT	RIGHT
Side	UP	RIGHT
Side	DOWN	RIGHT
Side	DOWN	LEFT
Side	RIGHT	DOWN
Side	LEFT	UP
Side	LEFT	DOWN

(iii) Classification of collision patterns based on vehicles' manoeuvres

There is also a need to learn the variations of vehicle manoeuvre pairs that can lead to collisions. For example, an SV travels from the north leg to the south leg (DOWN direction) is matched with a POV that travels from the west leg to the east leg (LEFT direction) based on the DOWN-LEFT rule learnt previously. However, the collision detection computation is executed on the SV-POV pair and no future collision is detected. Apparently, the SV and POV are currently “stopping” (i.e. it will not collide eventually). Hence, this vehicle pair should not be identified as a potential colliding vehicle pair. Therefore, it is necessary to find all the clusters of collisions and analyse the correlations between each manoeuvre with a collision type in the collision data.

The knowledge base needs the information to pair up SV-POV based on the current manoeuvre of SV during preselection process and thus it is necessary to perform classification of collision data (Figure 4.9) to extract decision tree rules that represents classes of vehicle manoeuvres. Therefore, C4.5 decision tree learning (implemented as J48 algorithm in Weka [Witten05]) is applied on the collision data. In the first training, vehicle1_manoeuvre is nominated as the class label. The decision tree shows the following result:

- If veh2_manoeuvre = STOPPED: veh1_manoeuvre = STRAIGHT
- If veh2_manoeuvre = STRAIGHT: veh1_manoeuvre = STRAIGHT
- If veh2_manoeuvre = STOPPING: veh1_manoeuvre = STOPPING

In the second training, vehicle2_manoeuvre is nominated as the class label. The decision tree shows the following result:

- If collision_type = SideCollision: veh2_manoeuvre = STOPPED
- If collision_type = RearEndCollision and veh1_manoeuvre = STRAIGHT: veh2_manoeuvre = STRAIGHT

- If collision_type = RearEndCollision and veh1_manoeuvre = STOPPING:
veh2_manoeuvre = STOPPING

We conclude that in this particular intersection, most side collisions occur when one of the vehicle pair is stopped and another one is with straight manoeuvre. Furthermore, rear-end collisions happen mostly when vehicles are on the move with straight manoeuvre and secondly when vehicles are stopping. The result of this learning helps in prioritising the vehicle-pairs selected as possible candidates for collisions. For example, when a vehicle's manoeuvre is stopped, then side collision detection algorithm is performed first before rear-end collision detection. Table 4.6 displays the partial collision patterns made of collision types, manoeuvre of SV, and manoeuvre of POV.

Table 4.6. Partial Collision Patterns Based on the Manoeuvre Pairs

CollisionType	SV Manoeuvre	POV Manoeuvre
RearEnd	Straight	Straight
RearEnd	Stopping	Stopping
Side	Straight	Stopped

Note that the collision patterns were learnt simulated data from a specific intersection. Applying the same technique to a different intersection (with different data) could lead to different likely situations for collisions – the point is that applying such learning techniques would enable collision situations specific to a particular intersection to be recognized automatically and identified as “dangerous” patterns. The results of the collision patterns learning are used to update the knowledge base of the collision detection in that particular intersection. The above results are beneficial for the preselection process since there is no need to apply collision detection computation for every possible pair in the intersection to predict for collision, but only to the vehicle pairs that satisfy the rules in the knowledge base. Also, the collision patterns with higher

occurrences are placed on a higher priority for checking whenever there are situations that lead to such patterns. As a result, the intersection collision warning system can detect threats faster, as explained further in Chapter 5. Moreover, this knowledge can be submitted to the road traffic authority for further assessment and follow up.

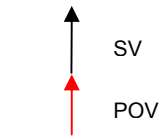
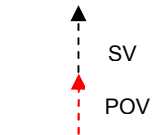
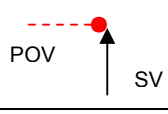
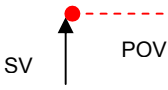
The results of collision patterns learning are summarised into fourteen patterns in the knowledge base as displayed in Table 4.5. These can be entered as specific collision patterns. These patterns are derived from the partial collision patterns learnt from the classification. The collision patterns derived from the clustering exploration (e.g. Table 4.3) can be used as a comparison against the list of specific collision patterns (Table 4.7). Based on our exploration, we found that the result of clustering collision data can help in finding initial collision patterns. However, further data analysis is necessary since a cluster may actually contain several specific collision patterns which are treated as one pattern/cluster by the clustering algorithm. Thus, classification techniques are useful to find finer details that compose specific collision patterns.

Based on the specific collision patterns in our exploration, we can deduce that there are two side generic collision patterns, which are “Perpendicular Left Straight Stopped” and “Perpendicular Right Straight Stopped” (Table 4.8). And there are two rear-end generic collision patterns, which are “Rear End Straight Straight” and “Rear End Stopping Stopping” (Table 4.8). For example, specific collision patterns no. 9, 11, 12, and 13 signify a side collision with perpendicular collision angle and the POV on the left hand side of the SV. Similarly, specific collision patterns no. 10 and 14 signify a side collision with perpendicular collision angle and the POV on the right hand side of the SV.

Table 4.7. Specific Collision Patterns

N o	Collision Type	SV Manoeuvre	POV Manoeuvre	SV Direction	POV Direction	SV Leg Location	POV Leg Location
1	RearEnd	Straight	Straight	UP	UP	Any	Any
2	RearEnd	Straight	Straight	DOWN	DOWN	Any	Any
3	RearEnd	Straight	Straight	LEFT	LEFT	Any	Any
4	RearEnd	Straight	Straight	RIGHT	RIGHT	Any	Any
5	RearEnd	Stopping	Stopping	UP	UP	Any	Any
6	RearEnd	Stopping	Stopping	DOWN	DOWN	Any	Any
7	RearEnd	Stopping	Stopping	LEFT	LEFT	Any	Any
8	RearEnd	Stopping	Stopping	RIGHT	RIGHT	Any	Any
9	Side	Straight	Stopped	UP	RIGHT	SOUTH, CENTRE	WEST, CENTRE
10	Side	Straight	Stopped	DOWN	RIGHT	NORTH, CENTRE	WEST, CENTRE
11	Side	Straight	Stopped	DOWN	LEFT	NORTH, CENTRE	EAST, CENTRE
12	Side	Straight	Stopped	RIGHT	DOWN	WEST, CENTRE	NORTH, CENTRE
13	Side	Straight	Stopped	LEFT	UP	EAST, CENTRE	SOUTH, CENTRE
14	Side	Straight	Stopped	LEFT	DOWN	EAST, CENTRE	NORTH, CENTRE

Table 4.8. Generic Collision Patterns in the Knowledge Base

Pattern Name	Collision Type	Subject Vehicle (SV) Manoeuvre	Principal Vehicle (POV) Manoeuvre	Geometry
Rear End Straight Straight	Rear End	Straight	Straight	
Rear End Stopping Stopping	Rear End	Stopping	Stopping	
Perpendicular Left Straight Stopped	Side	Straight	Stopped	
Perpendicular Right Straight Stopped	Side	Straight	Stopped	

The advantage of implementing specific collision patterns are that since it is purely based on the result of mining historical collision data, it is useful for the optimistic system mode, as only the patterns that have historical existence in the intersection are checked against future collision prediction. However, since the

number of specific collision patterns is higher than the number of generic collision patterns, it takes longer to iterate through specific collision patterns than through generic collision. It will be faster processing time to match an SV with a POV in a specific collision pattern, since all the required parameters to find a matching POV (manoeuvre pairs, direction pairs, leg location pairs) are stated. In contrast, in a generic collision pattern, an iterative process to match a POV based on the collision geometry and the manoeuvre pair would be necessary.

Given the high number of specific collision patterns, we implement four generic collision patterns, which encompass only the most frequently occurring manoeuvre pairs of each collision type in the intersection. These patterns allow vehicles to be matched based on the criteria coded in the pattern, which are manoeuvres and the collision geometry (implemented as a delegate function). In this way, the system is not set to be moderate. It is not very optimistic, since no matter where is the subject vehicle's location, it will be matched with other vehicles based on the colliding manoeuvre pairs and the geometry. It is not pessimistic either since not all possible manoeuvre pairs are considered. If the system is set to be very optimistic, specific collision patterns must be used as only certain direction pairs that exhibit the highest probability of a collision type is entered into the knowledge base.

All the patterns stored in the knowledge base are deduced from the result of learning from collision event data generated from the simulation, which is a cross intersection with a traffic light and no turning vehicle movements. Hence, these patterns are applicable only to the particular intersection simulated. In the real-world implementation, equivalent methodologies are applicable. All the collision event data and real-time traffic data need to be collected and learnt from. Collision patterns and traffic trends extracted from the data mining process are entered into the knowledge base for the generality of the intersection collision

warning and avoidance system and are used for the basis of detecting imminent and future collisions in the particular intersection. When such a system is transferred to a new intersection location, new knowledge about the intersection is learnt and entered into the knowledge base for that intersection. Therefore, due to the collision learning component in the U&I Aware Framework, the framework itself becomes adaptable to various and varying characteristics of an intersection.

Given the positive results yielded from the data mining techniques employed on the intersection simulation, it can be explicitly extrapolated that employing same or similar techniques on various intersections will also yield positive results as that is a property of data mining techniques. The intersection model has covered two major collision patterns as described in page 130-131 as rear-end collisions and side collisions, which in some literatures are described as the collision patterns with the highest occurrence [Mitre99], [USDOT04], [Lages04].

Next, we discuss learning of traffic trends in an intersection that may lead to collision.

4.2.2 Traffic Trends during Non-Collision-Free Periods

Apart from learning collision patterns, various characteristics of traffic may determine the risk for collisions to occur. We are interested to know whether variations in traffic attributes in different periods of a day can contribute to a higher or lower number of collisions. Attributes of intersection's traffic that can be monitored for a period of time are as follows: period, number of collisions, traffic volume, time of day, peak hours/non-peak hours, average speed of vehicles, safe ranges of vehicle speeds. In the U&I Aware Framework, the data is

gathered periodically from our simulation, where different parameters of time of day (morning/afternoon/evening/dawn) and peak/off-peak hours are applied to produce different behaviours in speed and traffic volume as in the real-world situations (e.g. during peak hours, there are higher traffic volume but the average traffic speed is lower, and during off-peak hours, the traffic volume is lower but the average traffic speed is higher). The main purpose of this learning is to determine whether the variation of speed and traffic volume may affect the number of collisions and different kinds of collisions in an intersection. To find such correlations, there is a need to apply an unsupervised clustering algorithm, which is useful to find trends and patterns in data when no attribute is specified for classifying or segregating data. This is because we cannot choose a particular class label that determines a traffic trend, as there can be various factors that contribute to a safer (or more dangerous) traffic trend.

Since there is no outlier to be considered such as in collision pattern learning, DBScan technique is inappropriate for clustering this data. Although either k-means or EM (Expectation Maximization, a clustering algorithm discussed previously in 4.2.1) can be used for this scenario, k-means is less accurate than EM. Hence, we propose to apply EM to the periodic collision data (Figure 4.5), which is generated by our simulation. Each record in the data is generated in every interval (which is four seconds in our simulation) with the following attribute values: *average traffic volume, average speed, total number of collisions, total number of side collisions, and total number of rear-end collisions in the last interval*. The Pantheon Gateway Project [Gross05] uses a similar set of attributes of real-world sensor data (speed, volume, occupancy) to learn changes in highway traffic. The initial results of applying EM on the data generated from our simulation (with the size of 50 – 80 records per file) are as follows:

- The higher the traffic volume and speed, the higher is the risk of collision.

The exact figures cannot be quantified since collisions may happen at any

traffic situation. However, this can be used as an indication. When the current figures of traffic volume and speed are increasing, the system should be more pessimistic.

- The number of rear-end collisions is heavily affected by traffic volume. The higher the traffic volume, the higher the possibility of rear end collision. Speed also contributes to rear end collisions.
- Side collision is not much correlated with traffic volume but more so with higher speeds, especially when the speed limit is violated.

Based on the results of the EM learning on traffic data on this particular intersection, we can deduce that the presence of collision (particularly rear-end collisions) during peak hours (morning peak hours when people commute to work and evening peak hours when people return home), are more likely than during non-peak hours. More side collisions occur during off-peak hours when average speed is higher than usual. Although these traffic patterns vary throughout the day, there is no correlation between the traffic patterns and collision patterns, since collision patterns are not affected by variations in traffic volume and average traffic speed. However, the number and type of collisions that occur can be affected by those attributes. Therefore, these traffic patterns can be useful in determining whether the intersection collision warning and avoidance system should be more pessimistic or optimistic. For example, during off-peak hours, we can set the system to be pessimistic for side collision detection, so that all possible side collision patterns are considered.

The above results are only applicable at the intersection where learning is performed. In another intersection, results may vary. This is why data mining can contribute to a generic model of intersection safety system that can self-adapt to different types of intersections by learning from the data specific to the intersection.

Future works of traffic trends learning include: characterisation of incident free behaviours at intersections (the antithesis of the learning described in this section), safe thresholds (i.e. the attributes' values under safe or normal situations), and identification of hazardous situations at the intersections when possibilities of collisions are present.

4.2.3 Normal Behaviours of Drivers during Non-Collision-Free Periods

Another set of data generated by our simulation is used to identify dangerous driving trends and it consists of attributes that are collected from pairs of vehicles involved in a collision, which are *speed and distance to intersection of each vehicle, traffic light colour faced by each vehicle, and the collision point* (Figure 4.6) characterise ideal or dangerous driver behaviours, are as follows: above speed limit, above average speed, collision presence, approaching intersection, increasing speed (yes / no). Once we know the acceptable threshold of those attributes for a collision free drive in an intersection, we would be able to easily identify abnormality in intersection or abnormal behaviours of drivers that are using the intersection if they exhibit any attribute that exceeds the acceptable threshold.

The purpose of this learning is to determine the boundaries of safe and dangerous driving behaviours. Data is recorded when a collision occurs. There are 20 – 30 records in the collision data. After applying Expectation-Maximization (EM) unsupervised clustering to extract dangerous driving trends, the result shows that in this particular intersection, most of the collisions occur when one of the vehicles in the collision pair have speed over 49. Again, this is merely an indication of what can be learnt from such data. When we apply this knowledge

to a collision warning system, an earlier prediction and extra precautions can be taken to vehicles that speed above 49. Another result of learning driver behaviour data of this particular intersection using C4.5 is that most cases of speed limit violation occur if the vehicle increases its speed when leaving the approach leg (any intersection leg where there is incoming traffic to the intersection centre) and enters the centre of the intersection. Future work in this area includes learning driver's distractions and its relationship to collisions and driver's safest manoeuvres for avoidance.

In terms of the time statistics of learning (or time taken to perform a data mining algorithm on a dataset), we have recorded that the time taken to build each model is around 0 to 0.03 seconds for each scenario described in Section 4.2.1 – 4.2.3. Since it is necessary to use learning algorithms that can work in real time and future works may include online incremental learning, our approach is proven efficient.

As the preceding discussion and analysis show the potential scope for data mining in ITS and road safety is very significant, and there are still many issues to be explored where data mining can be found useful, e.g. driver distractions, drowsy drivers' behaviours, red light running behaviours, etc. One of the aims of this thesis is to show and demonstrate the usage of data mining for development of a generic and adaptable intersection collision warning and avoidance systems, and therefore, the investigation of these issues is not in the primary concern of this research. However, the same principles and methodologies proposed in this thesis are applicable for various problem areas in road safety and ITS.

4.3. Summary

Along with the detection and warning components that are generally found in a collision warning and avoidance systems, the U&I Aware Framework comprises of a key feature, namely the learning component. Learning collision and traffic data to facilitate a generic and adaptable framework is a novel and a distinguishing feature that contributes to ITS and road safety. Collision learning is not aimed at replacing current methods or processes of analysing collision data, but it is aimed to serve as a supplement or enhancement to current methods and processes. This novel approach also facilitates faster collision detection. With the results of learning attributes contributing to collisions being integrated with the knowledge base, the U&I Aware Framework becomes context-aware, as only patterns and trends that pertain to the intersection are kept and utilised.

Since collision and traffic data from sensors data need to be gathered for collision learning, a traffic simulator is required to generate such data. We do not use existing traffic simulators since current existing simulators do not provide the following requirements: free flow and regulated traffic, macroscopic and microscopic view of traffic, continual and discrete data input by vehicles, and communication between vehicles and road infrastructure or base station. Moreover, the high cost of existing simulators makes them inaccessible. Hence, we have developed our own traffic simulation using four-leg cross intersection. The simulation generates intersection traffic and collision data and enables us to perform collision detection.

Once we obtain the traffic and collision data from the simulation, we apply data mining techniques in order to learn collision patterns, traffic trends during non-collision-free periods, and driver behaviours during non-collision-free periods. We apply different algorithms in different scenarios to achieve comprehensive

and correct results. A general approach for performing data mining on collision and traffic data are proposed and used:

- Identify the nature of the problem and the goal of learning;
- Identify the method to be used;
- Identify the technique to be used;
- Identify the technique for validation;
- Identify implementation strategy;
- Compare, analyse, and evaluate results;
- Integrate with the knowledge base.

Finally, the results of collision learning are interpreted and represented in the knowledge base. The collision patterns are stored either as specific or generic patterns. A specific collision pattern comprises of a pair of leg position, vehicle direction, and manoeuvre of both vehicles. Specific collision patterns are used when the mode of the intersection collision warning and avoidance system is set to optimistic. A generic collision pattern is used when the mode is either moderate or pessimistic. It has a delegate function that defines the geometry of a set of specific collision patterns.

Once all the useful patterns and trends for collision detection are entered into the knowledge base, they can be used as the criteria for filtering vehicle pairs, or namely “preselection”, before collision detection computation. The implementation of knowledge base is then utilised by the collision detection component of the U&I Aware Framework. Preselection is discussed in the next chapter.

COLLISION DETECTION

To recapitulate, one of the objectives of this thesis is to address the issue of how a future collision at an intersection can be detected before the collision actually occurs in real-time. Hence, the collision detection component is part of the U&I Aware Framework (Figure 5.1). In ITS, the term *collision prediction* is commonly used, such as in [Stubbs03] and [Kwon06] to refer to the activity of foreseeing a future and imminent collision. Sometimes, the term *collision* or *threat detection* is used instead, such as in [Miller02]. In robotic collision avoidance, the term *collision detection* is used either to state whether two objects that are moving will come into a collision over a given time span [Camer90]. In computer graphic or geometry research, *collision detection* is used to refer to a process of finding graphical objects that are currently colliding or intersecting with each other. In this thesis, the term *collision detection* and *collision prediction* are used interchangeably, as both refer to the definition of recognising a potential future collision before it actually takes place.

Collision detection and avoidance in road safety field are different from collision detection and avoidance in robotic collision avoidance. Studies in robotic collision avoidance have existed for many years [Fayad99], [Fox97], [Mani93].

Robots need to be able to find their own way to their destination as well as to avoid obstacles on their path. Although it seems that a robotic collision avoidance system has much resemblance to the problem of a road collision avoidance system, those two subjects differ in many aspects, which are as follows:

- A robotic collision avoidance system mostly focuses on static obstacles, such as walls [Mani93];
- A robotic collision avoidance system focuses on the goal of the robotic tasks such as to find a way out of a room, whereby a road collision avoidance system focuses on getting to the destination safely;
- Collision avoidance is a component in the path finding in a robotic collision avoidance system [Fayad99], [Fox97], which is not applicable in a road collision avoidance system;
- A robotic collision avoidance system does not need a human user, whereby a road collision avoidance system serves to assist a driver.

Due to the above differences, we need to approach road collision avoidance issues differently from robotic collision avoidance. However, the dynamic knowledge base technique introduced by [Mani93] can be used in conjunction with collision detection. Therefore, the incorporation of collision learning and knowledge base (discussed in Chapter 4) with the collision detection algorithms (discussed in this chapter) facilitate effective and efficient collision avoidance.

To facilitate collision avoidance, every incoming collision must be detected early. Therefore, fast and accurate techniques are needed such that collision warnings can be issued in time and the number of false alarms can be reduced. In the U&I Aware Framework, the results of the collision learning are stored in the knowledge base and become the basis for the preselection method for the collision detection component (Figure 5.1). As vehicle status data is received by

the intersection agent, the possibility of collision of each vehicle with other vehicles is assessed based on the known collision patterns (that have been obtained through mining, field observation, etc). The preselection method yields only the pairs of vehicles that have the potential for collision to the collision detection algorithm. The collision detection algorithm computes a future collision point of potentially colliding vehicle pairs as identified by the preselection. When a collision point exists, the time of each vehicle to reach the collision point is calculated. The timings for each of the vehicles to reach the collision point are compared and if the figures are almost equivalent, then an incoming collision is predicted.

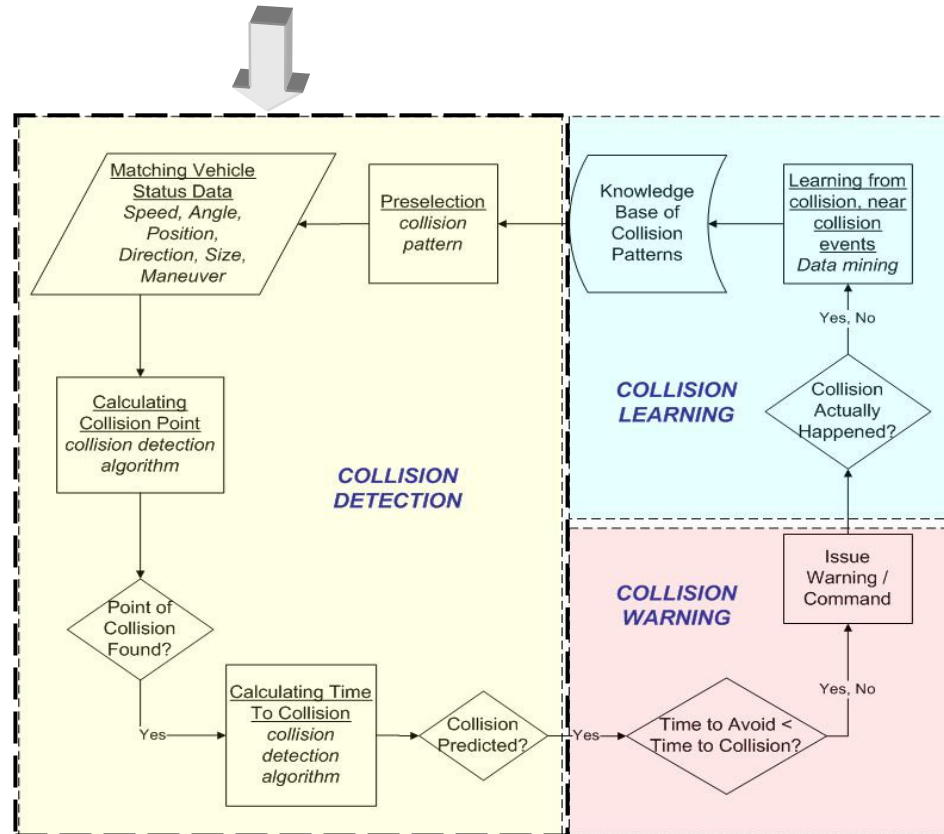


Figure 5.1. Collision Detection in the U&I Aware Framework

This chapter covers discussion on the collision detection component of the U&I Aware Framework. The work in this chapter has been previously published in [Salim07a], [Salim07c], [Salim08b]. The conventional method of collision

detection (the pair wise collision detection algorithm, as adapted from [Miller02]) is reviewed in Section 5.1. Then, our proposed preselection method that is designed to increase the performance of collision detection with regards to the conventional method of collision detection is discussed in Section 5.2. The implementation details of collision detection and preselection are discussed in Section 5.3. Section 5.4 presents evaluation results of our approach which is implemented through our simulation. This chapter is concluded in Section 5.5.

5.1. Improving Existing Collision Detection and Warning Algorithms by Preselection

The basic of calculating collision detection is the well-known speed formula, which is calculated by:

$$v = \frac{s}{t} \quad (5.1)$$

where v is speed, s is distance and t is travel time within the distance.

Based on the formula (5.1), collision detection can be calculated by the following steps:

- Calculate future collision point, which is by finding route contention of a pair of vehicles;
- Calculate time for each vehicle to reach future collision point (Time-To-Collision) based on the above speed formula;
- If Time-To-Collision (TTC) of one vehicle is equal or nearly equal with TTC of another vehicle to reach the same collision point, then collision is detected.

The peer to peer collision warning system by Miller and Huang [Miller02], as discussed previously in section 2.3.4, consists of a pair-wise collision detection

algorithm that computes the point of collision, Time-To-Collision (TTC) and Time-To-Avoidance (TTA). Their proposed algorithms to calculate the future collision point (x_+, y_+) [Miller02] are stated in (5.2) and (5.3) and the symbols used in those formulas are illustrated in the Figure 5.2 [Miller02].

$$x_+ = \frac{(y_2 - y_1) - (x_2 \tan \theta_2 - x_1 \tan \theta_1)}{\tan \theta_1 - \tan \theta_2} \quad (5.2)$$

$$y_+ = \frac{(x_2 - x_1) - (y_2 \cot \theta_2 - y_1 \cot \theta_1)}{\cot \theta_1 - \cot \theta_2} \quad (5.3)$$

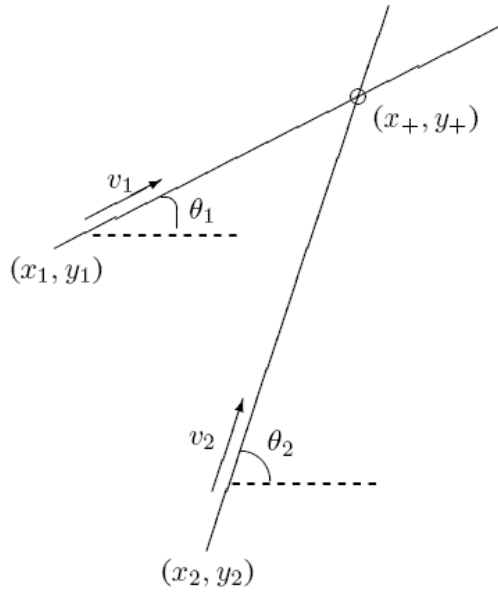


Figure 5.2. Collision Detection Algorithm [Miller02]

The x and y coordinate represents the location of the vehicle. In the real world, this coordinate can be obtained from GPS. The θ represents the angle between the line drawn from the same orientation or point of reference used by both vehicles and the trajectory of the vehicle. Using the coordinates and angle of the pair of vehicles, the future collision point (x_+, y_+) is calculated.

After a collision point is found, Time-To-Collision is then calculated. Time-To-Collision (TTC) is a common term used in Traffic Conflict Studies and Collision Warning Systems to measure elapsed time before an accident or collision occurs.

TTC is defined by Hayward [Hayw72] as the time needed for two vehicles to collide if they continue at their current speed and on the same path. Time-To-Collision (TTC) has been used as an effective measure to assess the severity of traffic conflicts and to distinguish dangerous from normal behaviour [Horst93]. TTC is calculated as a finite number that keeps decreasing in time as the future collision event goes unnoticed and vehicle speed and angle are constant.

The time for each car to reach the future collision point (TTX) [Miller02] is calculated by:

$$TTX_1 = \frac{|\vec{r}_+ - \vec{r}_1|}{|\vec{v}_1|} \text{sign}((\vec{r}_+ - \vec{r}_1) \cdot \vec{v}_1) \quad (5.4)$$

$$TTX_2 = \frac{|\vec{r}_+ - \vec{r}_2|}{|\vec{v}_2|} \text{sign}((\vec{r}_+ - \vec{r}_2) \cdot \vec{v}_2) \quad (5.5)$$

where v is velocity of each car and r is the vector of the coordinate (x, y) [Miller02].

As vehicles have variation in size, collision can no longer be expressed as a point; instead as a region. The α parameter is used to represent the size of the region (which can be determined by various factors, e.g. vehicle size). Therefore, a future collision is detected if time for both vehicles to reach the collision point is the equal or nearly equal, that is expressed in formula 5.6 [Miller02]. In this case, TTC is equal to TTX.

$$|TTX_1 - TTX_2| < \alpha \quad (5.6)$$

In some cases, when both of the intersection angles are perfectly perpendicular, future collision point cannot be directly calculated using Miller and Huang's formula (5.2) and (5.3). For example: if $\theta_1 = 0^\circ$ and $\theta_2 = 90^\circ$, then $\tan(\theta_1) = 0$, $\cot(\theta_1) = \infty$, $\tan(\theta_2) = \infty$, $\cot(\theta_2) = 0$. Therefore, the formula simply becomes:

$$X_+ = X_2 \quad (5.7)$$

$$Y_+ = Y_1 \quad (5.8)$$

By analysing the algorithm, we found that there are still a number of limitations. The algorithm may incur high communication overheads. The algorithms require very frequently updated information due to split second velocity and location changes. This is because firstly, the proposed collision system by Miller and Huang is a peer-to-peer vehicle based collision system [Miller02]. Therefore each vehicle needs to know the status of every other vehicle. Every time the car moves, all other vehicles must be informed. Hence, there should be a message queuing procedure that needs to be incorporated in each vehicle. However, since our proposed system applies a centralised computation approach, the message queuing procedure only needs to be applied on the central intersection agent. Secondly, collision detection should be computed again to find out possible collisions with any other vehicles in the vicinity using the current position. Changes in velocity in terms of acceleration and deceleration of vehicles in calculating collision time prediction are not considered. Average acceleration or deceleration a can be calculated by:

$$a = \frac{\Delta v}{t} \approx a = \frac{v_t - v_0}{t} \quad (5.9)$$

where Δv is velocity difference in a given time interval, v_t is future velocity, v_0 is current velocity, and t is time interval. When acceleration or deceleration is taken into account, communication cost can be reduced. Update of information, in particular velocity, can be less frequent as near future velocity can be predicted with acceleration or deceleration. Therefore, we have taken acceleration into account in designing the protocol of status message sent from vehicles. The communication model and protocol were discussed and presented in Chapter 3.

This algorithm also incurs high computational cost because the algorithm requires calculation for each possible pair of vehicles in the intersection (brute force). Therefore, real time detection is challenging when the number of vehicles increases at the intersection. As centralised approach for collision detection computation is adopted in the U&I Aware Framework, the formula to calculate the number of vehicle pairs to be monitored for collision detection is as follows:

$$\sum_{i=1}^n (i-1) \quad (5.10)$$

where n is the number of vehicles. Hence, the number of vehicle pairs grows in a linear square as the number of vehicles in the intersection grows. In order to sustain the performance and scalability of collision detection of vehicle pairs in an intersection, there is a need for reducing the number of vehicle pairs for which collision detection points need to be calculated.

Furthermore, mere application of the algorithm only enables the system to react to threat. There is a need for analysing collision, near collision or near miss data to enhance collision detection. Therefore, applying data mining techniques, as discussed in Chapter 4, along with implementation of the pair-wise collision detection algorithm help better situation recognition. In addition, with the results gained from mining collision patterns in Chapter 4, the number of vehicle pairs to be calculated for collision detection can be reduced by applying the preselection method. Therefore, the following section presents the preselection method as a proposed solution to deal with the issue.

5.2. Preselection

Preselection is a method to improve the performance of the conventional collision detection by reducing the number of vehicle pairs in the intersection to be

calculated for collision detection. Every Subject Vehicle (SV) is paired up with the potential Principal Other Vehicle (POV) based on the collision patterns learnt at the intersection. This pair is then added into the pool of matching vehicle pairs. Other vehicles that do not match with the SV based on the characteristics of any collision pattern are not included in the pool.

An SV is paired up with a POV based on the direction pair, manoeuvre pair, and/or location pair in an existing collision pattern. Whenever a pair of vehicles for potential collision is found, the matching collision pattern yields the collision type (i.e. side collision or rear-end collision) and the relevant collision detection computation based on the collision type is applied to assess the possibility of an imminent collision.

An example scenario of how a preselection algorithm can improve computational time is as follows (as pictured in Figure 5.3):

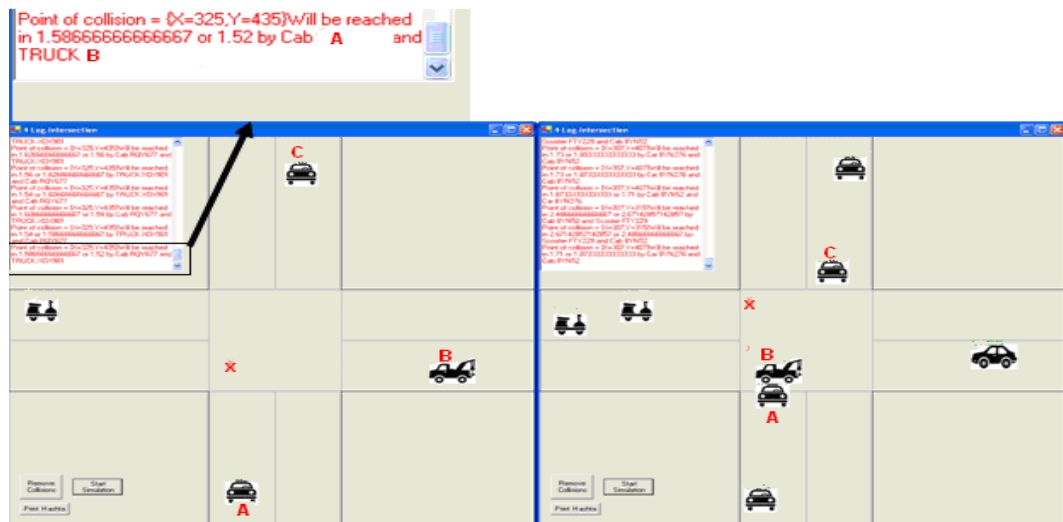


Figure 5.3. Cross Intersection without Traffic Lights Implementation

- i. Figure 5.3 – left shows four vehicles in a four leg cross intersection. Without preselection and the knowledge base with patterns (i.e. when brute force

approach is adopted), there are six possible pairs to be calculated for collision detection at every computation interval. For example, if the collision detection algorithm is to be executed at every 10 milliseconds, then at every 10 milliseconds, these vehicle pairs should be tested for possibility of collision. This is ineffective, since there are at least two vehicle pairs that will never collide, i.e. the pair of vehicle A and vehicle C.

- ii. The knowledge base of the intersection in this example records two types of side collision patterns: perpendicular left with straight manoeuvre and perpendicular right with straight manoeuvre.
- iii. In the brute force approach, each vehicle that moves needs to be checked for side collision prediction. However, we will not compare each vehicle to every other vehicle in the intersection. Only vehicles that are located within a certain area and exhibiting certain manoeuvres are selected. As for the truck B located at the right leg of the intersection in Figure 5.3, the algorithms will only be applied on vehicles on the upper and bottom legs that are exhibiting straight manoeuvre, based on perpendicular left with straight manoeuvre and perpendicular right with straight manoeuvre patterns. Those vehicles are vehicle A at the bottom leg and vehicle C at the upper leg.
- iv. After preselection is executed, only then the pair-wise collision detection algorithm is applied.

Hence, in dealing with the issue of the high computational cost of a conventional collision detection algorithm, we propose a preselection strategy, so that collision detection is only performed on pairs of vehicles that have the possibility of collisions based on the known intersection collision patterns. Preselection is implemented by selecting merely the vehicles in the vicinity based on matching the behaviours, location, and driving manoeuvres exhibited by each SV and POV pair with the collision patterns in the knowledge base. The methodology of preselection is described as follows:

- i. *Selection of SV.* Whenever status data is received from an SV, preselection is commenced for the particular SV. Based on the previous discussion in section 3.6.1, a variable interval time is used based on the speed of vehicles and how far the vehicles have travelled since the last status update. Hence, preselection is not performed by scanning all vehicles in the intersection at a constant interval time. Thus, it increases the efficiency of collision detection computation.
- ii. *Finding the Relevant Collision Pattern.* After the intersection agent receives the status data from the vehicle agent, it extracts the information about the manoeuvre, direction and/or leg location of the SV. These information is used to find the matching collision pattern in the knowledge base. These patterns are stored in form of hashtables. As the knowledge base may consist of generic collision patterns and specific collision patterns, depending on the pattern type, the access method of finding the pattern varies. This is further elaborated in the next section.
- iii. *Finding the POVs.* When a matching collision pattern is found based on the SV's manoeuvre, direction and/or leg location, then the manoeuvre, direction and/or leg location of POVs are retrieved from the collision pattern. The vehicles that are currently located in the relevant leg location as specified by the collision pattern are retrieved by directly accessing the LegPart hashtable (see Figure 4.6). Then, each vehicle is compared with the POV's manoeuvre as specified by the collision pattern. Only the vehicles that match with the specified manoeuvre are considered as the potentially colliding POVs and thus passed on to the pair-wise collision detection algorithm to be computed as shown in section 5.1.

As previously discussed in Chapter 4, the preselection can use two different modes as employed by the knowledge base, which are optimistic or pessimistic. In order to preselect pairs of vehicles that belong to the most frequently occurring

intersection collision patterns (i.e. optimistic system mode), it is necessary to be able to pair up one vehicle with other vehicle based on one of the attributes of the collision pattern, such as vehicle manoeuvre, direction, or leg location. For example, based on the given direction and manoeuvre of a Subject Vehicle (SV), we should be able to identify the opponent vehicles (Principal Other Vehicle – POV) that are most likely to collide with the SV. Each SV is then paired up with POV for collision detection computation. If the system is set to be pessimistic (an SV is paired up with POV based on an existing collision pattern, which is not necessarily the most frequently occurring pattern), all the existing collision patterns in the knowledge base of the intersection are to be used to identify potentially colliding vehicle pairs. A threshold can be set to determine the minimum probability of the occurrences required for a collision pattern to be included in preselection. Any collision patterns with any occurrence probability higher than the set threshold are to be considered in the preselection.

The next section discusses further how the collision detection and preselection are implemented in our simulation.

5.3. Collision Detection Evaluation

As stated in Section 4.1, the simulation is also developed as a test-bed for intersection collision detection. Hence, in this section, components in the simulation that are used to evaluate collision detection are discussed.

The knowledge base is coded as a class named *CollisionPatterns*. The *CollisionPatterns* class allow new patterns (each pattern is coded as *CollPattern*) to be added and provide methods to check for traffic conflicts. Specific collision patterns are coded as a class with five attributes (Figure 5.4): *patternName* (a

textual description for the pattern), *drivingManoeuvre* (of the SV), *currentLeg* (of the SV), *collidingLeg* (of the POV), and *collidingManoeuvre* (of the POV).

```
public CollPattern(string patternName, string drivingManoeuvre,
Leg currentLeg, Leg collidingLeg, string collidingManoeuvre);
```

Figure 5.4. Implementation of Specific Collision Pattern

A generic collision pattern is implemented as an object with a delegate function (Figure 5.5). The geometry of the collision is coded into the delegate function of the generic collision pattern. For example, “straight perpendicular left” collision pattern signifies that vehicles that are travelling with straight manoeuvre have possibilities of conflicts with vehicles approaching from their left hand side, no matter which leg they are currently located. To find conflicting vehicles with a generic collision pattern, a delegate function is used. The delegate function determines the conflicting vehicles based on the location of the subject vehicle and the attributes of the collision pattern coded into the delegate.

```
public delegate Leg findCollidingLeg(Leg currentLeg, string
direction, bool outgoing);

CollPattern(string patternName, string drivingManeuver,
findCollidingLeg collidingLeg, string collidingManeuver);
```

Figure 5.5. Implementation of Generic Collision Pattern

We have implemented the preselection and the pair-based collision detection algorithm. Preselection is implemented in the intersection agent. The implementation details are as follows:

- i. As mentioned in the Section 4.2, the knowledge base class *CollisionPatterns* maintains all the existing collision patterns learnt at the intersection. *CollisionPatterns* has a method named *getConflictingLegsAndManuevers*

(Figure 5.6), which is the main method that implements the preselection. The method requires input of the SV's manoeuvre, current leg position, direction, and information about the SV's movement (whether it is about to enter the intersection, at the intersection centre, has left the intersection centre or is leaving the intersection).

- ii. If generic collision patterns are used, this method compares those parameters of the SV with the generic collision patterns in the knowledge base using the *findCollidingLeg* delegate in the collision pattern. Otherwise, if specific collision patterns are used, string comparison between the parameters of the SV and the attributes of the collision patterns is performed.
- iii. When conflicts are found, the sets of conflicting manoeuvres, leg, and direction are recorded and returned in a hashtable.
- iv. *getConflictingLegsAndManuevers* method is called and executed whenever a new status update from a vehicle is received.
- v. If the hashtable returned from executing *getConflictingLegsAndManuevers* is not null, then we get the leg parts of the *Legs* hashtable, which match the values of the conflicting leg locations.
- vi. Following this, we retrieve all the POV in those legs that match the conflicting manoeuvres and directions.

```
Hashtable getConflictingLegsAndManuevers(string currentManeuver,  
Leg currentLeg, string currentDirection, bool outgoing);
```

Figure 5.6. getConflictingLegsAndManuevers Method

After all the conflicting POV have been retrieved, we instantiate *CarState* object (Figure 5.7) for each vehicle. Subsequently, we perform pair-wise collision detection by executing the *predictCollision* method (Figure 5.8). This method firstly calculates the future collision point. If future collision point exists, then the

value of α as required in the formula 5.6 (see section 5.1) is calculated (α is the size of the larger vehicle divided by the speed of the larger vehicle).

```
public CarState(Point carPos, int carAngle, double speed, string
travelDirection, Size size);
```

Figure 5.7. CarState Class Constructor

```
public static Collision predictCollision(CarState car1, CarState
car2, int intersectionWidth, int intersectionHeight);
```

Figure 5.8. Pair Wise Collision Detection Algorithm Implementation

The time for each vehicle to reach the future collision point (TTC) is calculated. If the difference between both TTCs is smaller than the value of α , it means a future collision has been detected, and a *Collision* object (Figure 5.9) is created and returned by *predictCollision* method, otherwise null is returned.

```
public Collision(Point collisionPoint, double timeToCollCar1,
double timeToCollCar2);
```

Figure 5.9. Collision Object

As seen in the simulation screen images of intersection without traffic lights (Figure 5.3), the red X mark is the collision point and the note on the upper left tells us the time to collision. The red X mark is displayed when the *Collision* object is not null, which means a collision is certain in near future if there is no manoeuvre or trajectory change. In Figure 5.3 – left screen, the collision point will be reached in 1.586 seconds by the cab A or truck B, as shown by the last sentence in the collision detection box. In the right screen of Figure 5.3, the collision actually happens in time of the predicted TTC.

To sum up, the pseudocode of the preselection method is as listed in Figure 5.10. Currently, the algorithm implemented for pair-wise collision detection is only for side collision detection, which comes from [Miller02], as has previously been discussed. The algorithm for side collision detection cannot be used for rear-end collision detection as rear-end collisions can be caused by multiple chain reactions, where there can be a number of cars following a collision, especially a rear-end collision. At this stage, we have not yet found an effective algorithm to detect multiple rear-end collisions, which are results of the chain reactions, as this is not the main focus of the research.

```

if (status_update)
    for each vehicle
        get the current leg location, manoeuvre and direction
        getConflictingLegsAndManuevers - return Hashtable object
        if Hashtable is not null
            retrieve leg parts that match the Hashtable values
            if leg part is not null
                then get vehicles that match the Hashtable values
                for each conflicting vehicle
                    instantiate CarState
                    predictCollision - return Collision object
                    if Collision is not null
                        send / display warning

```

Figure 5.10. Preselection and Pair-Wise Collision Detection Pseudocode

The preselection method has been evaluated and the results show that preselection optimises the performance of conventional collision detection algorithm. The next section discusses how the collision detection is evaluated and presents the evaluation results.

5.4. Collision Detection Evaluation

In order to show the performance improvement of collision detection made by the preelection method, we evaluate our approach using the following methods:

- Speed of detection;
- Performance/accuracy: precision and coverage.

The efficiency of the U&I Aware Framework needs to be evaluated by both speed and accuracy of collision detection. It is stated in the beginning of Chapter 3 that it is necessary to increase the speed of detection for a collision to be avoided. However, the accuracy of the detection should not be compromised to compensate for speed. Both evaluation methods are equally important to demonstrate the efficiency of the framework.

Each of these methods is performed in our system in two ways:

- i. the side collision detection is performed without using knowledge base and preselection (i.e. pure implementation of pair-wise collision algorithm [Miller02] where each possible pair of all the vehicles in the intersection is calculated);
- ii. the side collision detection is performed after applying preselection criteria from the knowledge base.

These methods are further discussed in the following subsections.

5.4.1 Speed of Detection

Whenever a future collision event is detected for the first time, it is recorded in a log file, with attributes as follows: *registration number of both vehicles, collision point, time to collision, leg location of both vehicles, and collision type*. Afterwards, the average of detection time (*time to collision*) for each run is

calculated. In each execution, the average time to collision is calculated. At the evening peak vehicle distribution model (average traffic volume 37-42 vehicles), if preselection is ignored in collision detection, the average time to collision is 5.6 seconds. However, when preselection is used, the average time to collision is 10.7 second, which is around 5 seconds earlier than the previous method. In each distribution model, preselection yields faster detection result. Therefore, preselection is shown to speed up the process of collision detection. The greater the number of vehicles in an intersection, the more preselection is useful and effective. This is shown in Figure 5.11. When collision detection is performed with preselection, the collision is detected faster (as TTC is greater).

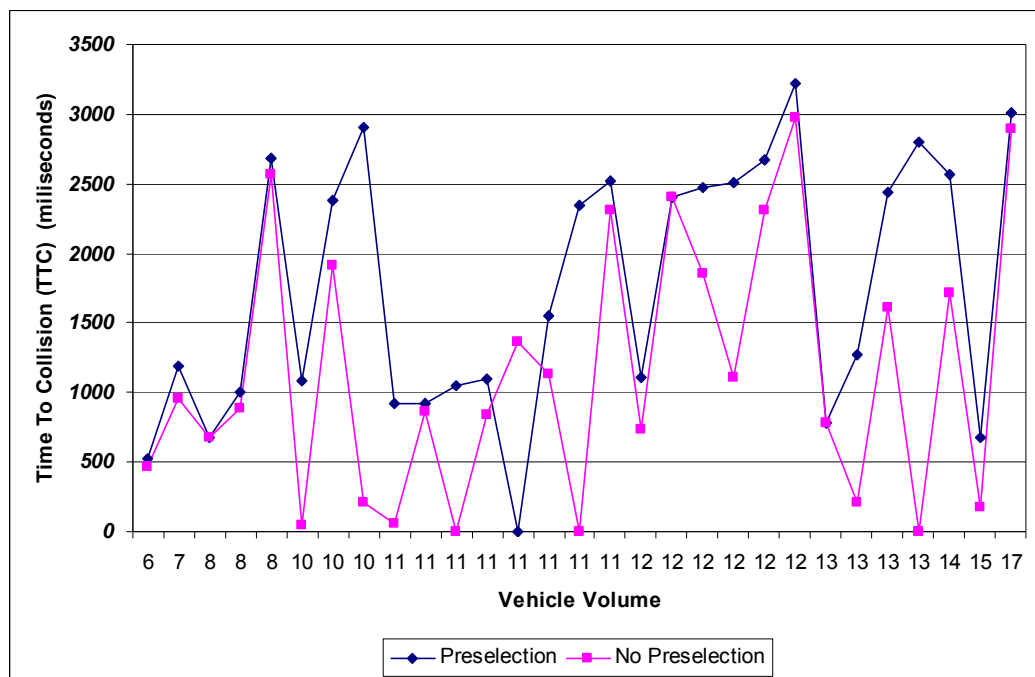


Figure 5.11. Speed Evaluation of Collision Detection

5.4.2 Accuracy: Precision and Coverage

This evaluation focuses on the accuracy of using preselection for collision detection and avoidance. Whenever a prediction of a future collision event is

issued, it is evaluated on whether the collision really happens. If it does, it is counted as a *true positive* (valid detection). However, when a predicted collision does not happen, it is counted as a *false positive* (invalid detection). When a collision occurs, and it is not previously predicted, then it is counted as *false negative* (undetected collision). The terms are described in Fig. 5.12.

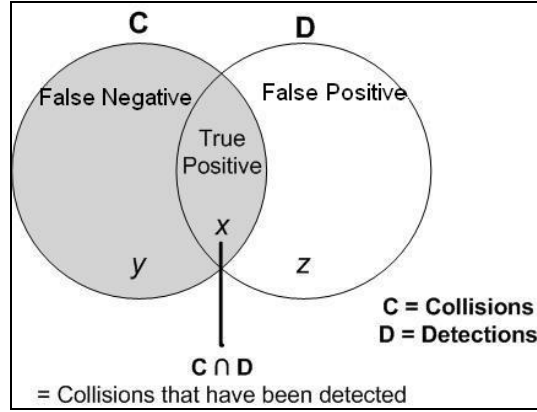


Figure 5.12. Evaluation Terms

We determine performance based on the terms of *precision* (of all the detections) and *coverage* (of all the collisions), respectively:

$$precision = \frac{\text{no_of_valid_Detections}}{\text{total_collision_Detections}} = \frac{\text{true_positive}}{(\text{true_positive} + \text{false_positive})} = \frac{x}{(x + z)} \quad (5.11)$$

$$Coverage = \frac{\text{no_of_valid_Detections}}{\text{total_Collisions}} = \frac{\text{true_positive}}{(\text{true_positive} + \text{false_negative})} = \frac{x}{(x + y)} \quad (5.12)$$

Both precision and coverage are evaluated through the simulation. When an incoming collision is predicted, the registration numbers of both vehicles are entered into the *CollPrediction* hashtable as a new object of key and value pair. The time for each vehicle to reach the collision point is entered into the *CollPredictionTime* hashtable, which is updated throughout the course of the collision. When a collision occurs, the collision details are entered into the *trueCollisions* hashtable. Using a periodic timer, the method to calculate

precision and coverage are invoked periodically. The values of the *CollPrediction* hashtable is compared with the values of the *trueCollisions* hashtable. The matching values are considered as true positive events. The values in the *trueCollisions* hashtable that are not included in the *CollPrediction* hashtable are considered as false negative events. In order to calculate the false positive events, the *CollPrediction* hashtable is compared with the *Vehicles* hashtable that contains references to all the vehicles in the intersection. If a collision is predicted for a certain vehicle that no longer exists in the *Vehicles* hashtable and it is not included in the *trueCollisions* hashtable as a collision that actually happens, then the collision prediction is obsolete and considered as a false positive event.

Based on the accuracy evaluation on side collision detection in our simulation, we achieve 100% precision when side collision detections are present and 100% coverage when side collisions are present. This level of 100% precision and coverage is valid in the simulation. This result shows that the collision detection algorithm is correctly implemented and effective. Furthermore, it reveals that the preselection algorithm has successfully identified potential collisions using collision patterns learnt at the intersection. When there is a false negative, it may indicate a new collision pattern that has not been included in the knowledge base. The collision learning component that continuously learn for collision patterns can identify this new collision pattern, which has to be added into the knowledge base. Thus, having a generic and adaptable framework for intersection collision avoidance serves the requirements of the dynamic and changing situations of an intersection. The integration of collision learning, detection, and warning components of the U&I Aware Framework produces a powerful and effective solution for intersection collision avoidance.

We also remark that the speed and accuracy results obtained from the evaluation are limited to computer based simulations. The following facts need to be considered when a full scale real-world evaluation is performed:

- Sensor accuracy is probabilistic. Since each sensor has a range of error rate (as mentioned in Section 1.1), when multiple sensors are used, the error rates are accumulated. This affects the accuracy of status data that are typically based on vehicle sensors). Hence, in the real-world deployment, 100% accuracy cannot be guaranteed.
- Computation time and workload is uncertain for various machines. Evaluation on various machines, platforms, and mobile devices has not been performed.
- The tradeoffs between performance (i.e. speed and accuracy) and cost of computation. Given the availability of higher resources and computing power, the performance rate can be higher. However, when small mobile devices are used and only limited resources are available, there should be a threshold allowed for lower performance rate.

In the next chapter, we discuss how to address the above issues in the real-world evaluation as part of the future directions of this research.

5.5. Summary

This chapter has presented methods and algorithms for collision detection in intersection collision avoidance systems. Mere application of the existing conventional pair-wise collision detection algorithm such as proposed by Miller and Huang [Miller02] can pose several issues: the performance and scalability when the number of vehicle pairs in the intersection grows exponentially, and the inability of the algorithm to adjust to the collision patterns that pertain to the

intersection for better situation recognition and faster detection. Given those challenges in existing collision detection algorithms, it is necessary to develop a method to reduce the number of vehicle pairs to be computed for collision prediction. The dynamic knowledge base that contains the collision patterns of the U&I Aware Framework can be used in combination with the collision detection algorithm for that purpose.

Hence, the collision detection component in the U&I Aware Framework is coupled with the collision learning component. The preselection method is proposed to reduce the number of vehicle pairs to be computed for collision prediction. The learning results that are stored in the knowledge base are utilised as the basis for the preselection method. Each vehicle in the intersection is only paired up with another vehicle in the intersection if it matches the preselection criteria, which are the collision patterns. Only the pairs that are selected by the preselection method are used for collision prediction computation, which uses the conventional collision detection algorithm.

The collision prediction has been implemented and evaluated in the intersection simulation. The performance and accuracy of the collision detection are evaluated based on the speed and the coverage of detection (which evaluates both precision and recall of collision detection). The speed of the collision detection is improved when preselection is used. The precision of the collision detection is 100%, and the recall of side collision detection is 100% in the context of this evaluation.

CHAPTER 6

CONCLUSION

Road intersections have claimed and injured many lives worldwide. The costs of intersection collisions financially are also not trivial. Initiatives and efforts to increase safety for road users have resulted in new sensor technologies installed in vehicles and on the road, increased safety measures in vehicles, and intersection collision warning and avoidance systems being designed and developed. Nevertheless, the existing intersection collision warning and avoidance systems are mainly infrastructure-only. They typically rely only on infrastructure sensors as the data source and roadside LED signs for issuing warning. The implications here are that these systems do not leverage the available data sources adequately. They are also limited in their models for communicating warnings effectively. Furthermore, they are designed only for a particular type of intersection and are not capable of learning and adapting to different and varying characteristics of the intersection. Therefore, this thesis has investigated the features required in a cooperative intersection collision warning and avoidance systems that can adapt to the varied characteristics of intersections. The next section presents the contributions of this thesis.

6.1. Research Contributions

This research has contributed novel findings for the pervasive computing community as well as the road safety and Intelligent Transportation Systems research, as discussed below.

- **The U&I Aware (Ubiquitous Intersection Awareness) Framework**

First of all, this thesis has proposed a generic and adaptive framework for real-time collision detection (or prediction) and warning at road intersections, namely the U&I Aware (Ubiquitous Intersection Awareness) Framework. The following qualities have been incorporated into the U&I Aware Framework: *adaptability* of the framework to various intersections, improvement of *performance and scalability* of the collision detection (or prediction) process, usage of appropriate *real-time data sources*, and a *real-time communication model and protocol* between vehicles and the system infrastructure with an *effective warning delivery* based on the available time before collision is predicted to occur.

The pervasive computing techniques – data mining, knowledge based systems, and context-awareness, which enable learning and adaptability have inspired the work of this thesis and are integrated as components of the framework, which are collision learning, collision detection, and collision warning. Current intersection collision warning and avoidance systems do not encompass collision learning, which is the capability for the system to learn collision patterns and other trends at the intersection. Through the learning of collision patterns, the U&I Aware Framework can be tailored for operation in any given intersection. Thus, the ability of the framework to adapt to various intersections is one of its key contributions. Furthermore, the patterns learnt at the intersection can be used as the basis for preselection, which identifies vehicle pairs that are likely to collide.

This approach improves the performance of the collision detection component in the U&I Aware Framework. An evaluation of the framework has been carried out using our custom-built intersection traffic simulation.

For the purpose of collision avoidance, it is necessary to know the cost model of Time-To-Avoidance (TTA), since TTA must be lesser than Time-To-Collision (TTC). A comprehensive cost model of TTA, that considers all the cost components from existing research and also the U&I Aware Framework, has been proposed. TTC is known from computing the possibility of future collision between two vehicles with the collision detection algorithm. Given the need for effective warning delivery, we used two warning delivery types, which are collision warning message (intended for the driver, the cost of issuance is expressed as $TTA_{warning}$) and collision command message (intended for the vehicle braking system, the cost of issuance is expressed as $TTA_{command}$). If TTC is greater than $TTA_{warning}$, collision warning message is issued, otherwise collision command message is sent. The cost model for calculating $TTA_{warning}$ and $TTA_{command}$ are presented in this thesis.

There are two major positive characteristics given by the U&I Aware Framework in improving safety at intersections. The first major impact is *adaptability*, as the framework is able to adapt to different and varying intersection characteristics. The second is the improvement in the performance and scalability of collision detection at intersections. In addition, the intersection simulation is also a contribution of this research. These features are discussed further.

- **Enabling Adaptability of the Framework to Different and Varying Intersection Characteristics**

Due to different and varying characteristics of intersections, it is necessary to

enable adaptability of an intersection collision warning and avoidance system to various intersections. A generic and adaptable intersection collision warning and avoidance system has been enabled through the U&I Aware Framework using intersection-specific collision pattern learning and its dynamic knowledge base.

Collision learning in the U&I Aware Framework is performed to enable new patterns to be learnt and added into the knowledge base and thereby enhance the knowledge base for better collision detection. Offline mining is performed to extract collision patterns, dangerous traffic trends during various times of the day, and dangerous driver behaviours in the intersection from intersection data, which include collision and near collision events, driving behaviour, and real-time traffic data. The appropriate data mining algorithms for each learning scenario are suggested and applied in this thesis.

The knowledge base is populated with results from mining intersection data. Information learnt at the intersection, such as collision patterns and traffic trends, is stored in the knowledge base to be used as the basis for identifying vehicle pairs that are likely to collide. Given the features of the collision learning component, which consists of data mining and a knowledge base, the U&I Aware Framework is applicable to various intersections. Since learning is performed using the traffic and collision data from the intersection vicinity, the knowledge base gets updated with collision patterns and traffic trends that pertain to that particular intersection.

- **Improvement of Performance and Scalability**

An intersection collision warning and avoidance system needs to perform efficiently and be scalable for increasing number of vehicles travelling through the intersection. The existing pair wise route contention (or collision detection)

algorithm relies on calculating point of collision of a vehicle pair and travel time of each vehicle in the vehicle pair to the collision point. Hence, such computation requires every possible pair of vehicles in the intersection to be calculated for possibility of collision. This thesis has proposed a preselection method, which performs better than the brute force method. The preselection method reduces the number of vehicle pairs in the intersection by only selecting the vehicle pairs that corresponds to one of the existing collision patterns in the knowledge base. The preselection method then passes the list of the vehicle pairs to the collision detection algorithm.

In order to perform preselection, a global bird's eye view of the intersection is needed, therefore, the U&I Aware Framework uses a central component that is located in the intersection's vicinity, namely the intersection agent, which manages the tasks of communication, data mining, predicting potential collisions, and issuing warning to relevant vehicles. The dynamic knowledge base that is required for adaptability and preselection is located in the intersection agent [Salim08a]. The patterns learnt at the intersection are maintained as rules in the knowledge base and are used for the preselection technique. The mining results can be used to determine whether a pair of vehicles travelling in the intersection is possibly due for an imminent collision [Salim07a].

The performance and scalability of the intersection collision detection are evaluated based on the speed and the accuracy of the detection. We measure the speed of the detection by comparing the collision detection that is equipped with preselection, with collision detection that requires calculation of each possible vehicle pair in the intersection. The evaluation shows that preselection increases the speed of collision detection. The higher the number of vehicles in the intersection, the more effective preselection becomes. The accuracy of the detection is measured on the precision (rated by the number of collisions that are

detected correctly divided by all the collision detections issued) and the coverage (rated by the number of collisions that are detected correctly divided by all the actual collisions that occur in the intersection). The accuracy of the side collision detection in the intersection simulation is 100% precision and 100% coverage. Rear-end collisions, which mostly happen due to the chain effect after a side collision, are not detected at this stage since we have not found an effective rear-end collision detection algorithms that can deal with chain collision effect.

- **Intersection Simulation**

In this thesis, we have proposed and developed an intersection simulation that is equipped with and without traffic light operation, able to represent microscopic and macroscopic view of the traffic, able to accept both continuous and discrete input, able to simulate vehicle sensor data (as if data are communicated wirelessly from vehicles to the intersection agent), and consider both stochastic and deterministic models for vehicle distribution, vehicle speed change, and car following model. None of the existing simulations can provide all the above characteristics. The intersection simulation is used to generate collision events and traffic data, so the learning component of the U&I Aware Framework can mine the data. The simulation is also used as a test-bed for the implementation of the preselection and intersection collision detection. The data generated in the intersection resembles the real-world representations and yields interesting and useful patterns when learning is applied.

The next section presents further investigations and extensions that are proposed as future directions for this research.

6.2. Research Directions

The thrust of the work of this thesis has been focused on the pre-collision stage. Collision learning has been focused on finding the characteristics of driving (e.g. manoeuvre, trajectory, speed, etc) and traffic conditions just before a collision takes place. However, it is also important to learn the safest driver behaviours and manoeuvres during collision and post-collision stage on various conditions that will alleviate impacts, reduce severity of the collision, or avoid a collision completely on a given situation.

Ideally, those processes are to be executed in a mobile and small resource-constrained device (such as a Personal Digital Assistant (PDA)) for easy deployment in vehicles. Thus, data mining on resource-constrained devices in vehicles should be considered as a future work. It is also desirable to learn collision pattern and dangerous driver behaviours using online stream data from vehicle sensors and incrementally add the learning results into an evolving dynamic knowledge base. Although the framework is now able to adapt to various types of intersections, threat and collision learning is still performed offline on historical data using data mining. The current state-of-the-art of data mining research, which is ubiquitous data stream mining, is a significant area to explore, given the increasing number of sensors and small devices that are available in vehicles.

We also see the need for personalisation of warning since every driver is different. One driver might have a tendency to drive cautiously, while others might have a history of reckless driving. Young probationary drivers have the tendency to drive faster than middle-aged probationary drivers. A proficient driver might not need a very early warning for incoming threats, as it can be a nuisance to him. Therefore, it is also necessary to adjust the collision warning

based on the driver's profile. With advances of ubiquitous data stream mining, learning can be done onboard the vehicle utilising driver's profile and vehicle sensor data, thus making the vehicle agent that sits in the vehicle to be aware of the vehicle and the driver behavioural contexts. As a result, the intersection collision warning system can be more informed when a driver exhibits dangerous driving behaviours. Considering the tradeoffs between performance and computational cost, there is also a need to learn probabilistic model of the correlation between performance and computing resources. This is useful particularly in dealing with various road users that have various requirements. For example, an elderly may need a warning system that has a higher accuracy and thus a higher resource machine should be used.

A full-scale real world deployment should be considered. The messaging cost model and protocols in the U&I Aware Framework [Salim08a] are proposed in this thesis without a real-world performance evaluation. It is necessary to implement the messaging protocols of the U&I Aware Framework at the intersection agent (in the server) and vehicle agents (in small devices) and evaluate the performance and accuracy of the collision warning and avoidance systems with input from real-world sensor data. Based on the given context (known from the sensor data), a specific contextual warning or manoeuvre should be suggested to completely avoid a collision or alleviate the impact. In a real-world deployment, we need to setup a virtual time base where each part of the distributed application refers to the same timestamp. Furthermore, a real-world deployment also requires more complex manoeuvre modelling. This can be achieved by integrating a simple vehicle model with a filter (e.g. Kalman filter) in order to assess the vehicle state. Such filter needs to consider different types of noise that may arise in communication/messaging, vehicle model data, and sensor data.

Furthermore, the communication model in the U&I Aware Framework can be extended to include knowledge sharing capability among the intersection agent and vehicle agents. Since learning can also be done in each vehicle using mobile/small devices, the knowledge learnt can also be shared. In order to maintain the awareness of the system with the up to date situations on the road, knowledge sharing needs to be applied. After a vehicle is registered in an intersection administration zone and if the option of knowledge sharing is enabled in the vehicle agent (for privacy concerns, knowledge sharing can be disabled), the vehicle agent can also communicate the knowledge learnt about the driver. Patterns of dangerous driving behaviours can be utilised by the intersection agent to detect the presence of certain behaviour and activity that may lead to collisions in the intersection. Patterns of driver's avoidance manoeuvres can be used by the intersection agent to correlate the collision pattern with certain manoeuvres, so that the best manoeuvre to avoid a foreseen collision in the intersection can be suggested to the relevant vehicles. However, when considering such scenario, security and privacy issues should also be dealt with.

Given the contributions and directions of this research, we have demonstrated the potential of pervasive computing (i.e., the combination of situated sensing and computation) when applied to road intersection safety. There are still other areas of Intelligent Transportation Systems that are not discussed in this thesis where pervasive computing research can be applicable and useful.

In summary, this thesis has made a novel and signification contribution to intersection safety through the proposal and development of the U&I Aware Framework.

REFERENCES

- [Abdel06] Abdel-Aty, M. and Pemmanaboina, R., (2006), “Calibrating a real-time traffic accident prediction model using archived weather and ITS traffic data”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, June, IEEE Intelligent Transportation Systems Society, pp. 167-174.
- [Aggarw05] Aggarwal, C., Yu, S., (2005), “An effective and efficient algorithm for high-dimensional outlier detection”, *The VLDB Journal — The International Journal on Very Large Data Bases*, vol.14, no.2, April, Springer-Verlag New York, Inc, pp. 211-221.
- [AIDE04] AIDE, (2004), *AIDE Project 1st Newsletter*, October, Retrieved 21 March 2005 from AIDE website: http://www.aide-eu.org/pdf/aide_1st_newsletter.pdf.
- [Arem06] Arem, B. van., Driel, C. J. G. van, and Visser, R., (2006), “The impact of cooperative adaptive cruise control on traffic-flow characteristics”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, December, IEEE Intelligent Transportation Systems Society, pp. 429–436.
- [Arndt03] Arndt, O. K., (2003), *Relationship Between Unsignalised Intersection Geometry and Accident Rates*, PhD Thesis, March 2003, School of Civil Engineering, Queensland University of Technology, Australia.
- [ATSB05] Australian Transport Safety Bureau (ATSB), (2006), *Road Deaths Australia: 2005 Statistical Summary*, Retrieved 23 April 2007 from ATSB website: http://www.atsb.gov.au/publications/2006/pdf/rda_ss_2005.pdf.

- [ATSB06a] Australian Transport Safety Bureau (ATSB), (2006), *International Road Safety Comparisons: The 2004 Report. A comparison of road safety statistics in OECD nations and Australia*, Retrieved 23 April 2007 from ATSB website: http://www.atsb.gov.au/publications/2006/pdf/Int_Comp_03.pdf.
- [ATSB06b] Australian Transport Safety Bureau (ATSB), (2006), *Key Facts for New drivers*, Retrieved 23 April 2007 from ATSB website: http://www.atsb.gov.au/pdfs/key_facts.pdf.
- [Auburn05] Auburn Council – Road Safety Officer, (2005), *Stopping Distance*, Retrieved 23 April 2007 from Auburn Council website: <http://www.auburn.nsw.gov.au/page.aspx?id=790&>.
- [Bazzan05] Bazzan, A. L. C., (2005), “A distributed approach for coordination of traffic signal agents”, *Autonomous Agents and Multi-Agent Systems*, vol. 10, no. 2, March, Springer, pp. 131-164.
- [Bhargava06] Bhargava, V. K., (2006), “State of the art and future trends in wireless communications: advances in the physical layer”, *Proceedings of the 4th Annual Communication Networks and Services Research Conference (CNSR'06)*, 24-25 May, New Brunswick, Canada, IEEE Computer Society Press, pp. 3.
- [Boury00] Boury-Brisset, A. C. and Tourigny, N., (2000), “Knowledge capitalisation through case bases and knowledge engineering for road safety analysis”, *Knowledge-Based Systems*, vol. 13, no. 5, October, Elsevier, pp. 297-305.
- [Bruin04] Bruin, D. de, Kroon, J., Klaveren, R. van, and Nelisse, M., (2004), “Design and test of a cooperative adaptive cruise control system”, *Proceedings of Intelligent Vehicles Symposium*, 14-17 June, Parma, Italy, IEEE Intelligent Transportation Systems (ITS) Council, pp. 392–396.
- [BITRE00] Bureau of Infrastructure, Transport and Regional Economics, Department of Infrastructure, Transport, Regional Development and Local Government, Australia, (2000), *Road Crash Costs in Australia – Report 102*, May, Retrieved 18 July 2008 from BITRE website: <http://www.bitre.gov.au/publications/47/Files/r102.pdf>.
- [Camer90] Cameron, S., (1990), “Collision detection by four-dimensional intersection testing”, *IEEE Transaction on Robotics and Automation*, vol. 6, no. 3, IEEE Robotics and Automation Society, pp. 291-302.
- [Camer94] Cameron, G., Wylie, B. J.N. and McArthur, D., (1994), “Paramics: moving vehicles on the connection machine”, *Proceedings of the 1994 ACM/IEEE Conference on Supercomputing*, 14-19 November, Washington DC, USA, ACM Press, pp. 291-300.

- [Chan03] Chan, C-Y., Misener, J. and Lins, J., (2003) , *Smart Buses, Smart Intersection Shine at Washington IVI Meeting*, Presented at the National Intelligent Vehicle Initiative Meeting, 24-26 June, Washington, D.C., USA, Retrieved 31 March 2005 from California Partners for Advanced Transit and Highways (PATH) website:
<http://www.path.berkeley.edu/PATH/Research/Featured/102803/washington.html>.
- [Chan04] Chan, C-Y., and Marco, D., (2004), "Traffic monitoring at signal-controlled intersections and data mining for safety applications", *Proceedings of IEEE Intelligent Transportation System Conference*, 3-6 October, Washington, D.C., USA, IEEE Intelligent Transportation Systems Society, pp. 355-360.
- [Charles03] Charles, P. and Sayeg, P., (2003), *Handbook on Intelligent Transport Systems*, ITS Australia, Melbourne, <http://www.its-australia.com.au/KMXServer3/Portals/0/ITSAHanbook.pdf>.
- [Chass05] Chassiakos, A.P., Panagolia,C., and Theodorakopoulos, D. D., (2005), "Development of decision-Support System for Managing Highway Safety", *Journal of Transportation Engineering*, vol. 131, no. 5, May, American Society of Civil Engineers, pp. 364-373.
- [Chen00] Chen, G. and Kotz, D., (2000), *A Survey of Context-Aware Mobile Computing Research*, Technical Report TR2000-381, November, Department of Computer Science, Dartmouth College, USA, Retrieved 19 October 2004 from:
<http://citeseer.ist.psu.edu/390713.html>.
- [Chong04] Chong, M., Abraham, A., Paprzycki, M., (2004), "Traffic accident data mining using machine learning paradigms", *Proceedings of the Fourth International Conference on Intelligent Systems Design and Applications (ISDA'04)*, 26-28 August, Budapest, Hungary, Springer Verlag, pp. 415-420.
- [Dey99] Dey, A. K. and Abowd, G. D., (1999), *Towards a Better Understanding of Context and Context-Awareness*, GVU Technical Report GIT-GVU-99-22, June 1999, Georgia Institute of Technology, presented at the 1st International Symposium on Handheld and Ubiquitous Computing, Retrieved 27 April 2005 from: <ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf>.
- [Duben03] Dubendorf, V. A., (2003), *Wireless Data Technologies*, John Wiley & Sons, Ltd.
- [Ester96] Ester, M., Kriegel, H-P., Sander, J., Xu, X., (1996), "A density based algorithm for discovering clusters in large spatial databases with noise", *Proceedings of the Second International Conference*

on *Knowledge Discovery and Data Mining*, AAAI Press, pp. 226-231.

- [Farkas06] Farkas, K. I., Heidemann, J. S., Iftode, L., Kosch, T., Strassberger, M., Laberteaux, K. P., Caminiti, L., Caveney, D., and Hideki, H. (2006), "Vehicular communication", *IEEE Pervasive Computing*, volume 5, no. 4, Oct-Dec, IEEE Computer Society Press, pp. 55-62.
- [Fayad99] Fayad, C. and Webb, P., (1999), "Optimized fuzzy logic based on algorithm for a mobile robot collision avoidance in an unknown environment", *Proceedings of 7th European Congress on Intelligent Techniques & Soft Computing*, Aachen, Germany, September 13-16, EUFIT, Retrieved 11 December 2006 from: http://www.cs.nott.ac.uk/~cxf/Papers/Optimised_Fuzzy_Logic.pdf
- [Fayyad96] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., (1996), "From data mining to knowledge discovery in databases", *AI Magazine*, vol. 17, no. 3, Fall, AAAI Press, pp. 37-54.
- [Fellen94] Fellendorf, M., (1994), "VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority", *64th Institute of Transportation Engineers Annual Meeting*, Session 32, October, Dallas, Texas, USA, Institute of Transportation Engineers, Retrieved 14 August 2007 from: <http://www.english.ptv.de/download/traffic/library/1994%20ITE%20VISSIM%20Bus%20Priority.pdf>.
- [Ferlis01] Ferlis, R. A., (2001), "Infrastructure intersection collision avoidance", *Intersection Safety Conference*, 14-16 November, Milwaukee, WI, USA, Institute of Transportation Engineers, Retrieved 12 January 2005 from ITE website: <http://www.ite.org/library/IntersectionSafety/Ferlis.pdf>.
- [Fletch03] Fletcher, L., Apostoloff, N., Petersson, L., and Zelinsky, A., (2003), "Vision in and out of vehicles", *IEEE Intelligent Systems*, vol. 18, no. 3, May-June, IEEE Computer Society Press, pp. 12-17.
- [Fox97] Fox, D., Burgard, W., and Thrun, S., (1997), "The dynamic window approach to collision avoidance", *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, March, IEEE Computer Society Press, pp. 23-33.
- [Frye01] Frye, C., (2001), "International Cooperation to Prevent Collisions at Intersections", *Public Roads Magazine*, vol. 65, no. 1, July-August, Turner-Fairbank Highway Research Center, United States Department of Transportation - Federal Highway Administration, Retrieved 1 February 2005 from: <http://www.tfhrc.gov/pubrds/julaug01/preventcollisions.htm>.

- [Fuers05] Fuerstenberg, K. Ch., (2005), "A new European approach for intersection safety - the EC-Project INTERSAFE", *Proceedings of the Intelligent Transportation Systems 2005*, Vienna, Austria, 13-15 Sept, IEEE Intelligent Transportation Systems Society, pp. 432-436.
- [Funder04] Funderburg, K. A., (2004), "Update on intelligent vehicles and intersections", *Public Roads Magazine*, vol. 67, no. 4, January-February, Turner-Fairbank Highway Research Center, United States Department of Transportation - Federal Highway Administration, Retrieved 1 February 2005 from: <http://www.fhrc.gov/pubrds/04jan/08.htm>.
- [Gaber04a] Gaber, M. M., Krishnaswamy, S., and Zaslavsky, A., (2004), "Ubiquitous data stream mining", *Proceedings of Current Research and Future Directions Workshop*, in conjunction with The Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining, 26 May, Sydney, Australia, Springer Verlag, Retrieved 20 September 2004 from: <http://www.csse.monash.edu.au/~mgaber/CameraReadyPAKDD.pdf>.
- [Gaber04b] Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S., (2004), "Resource-aware knowledge discovery in data streams", *Proceedings of First International Workshop on Knowledge Discovery in Data Streams*, in conjunction with the 15th European Conference on Machine Learning (ECML 2004) and the 8th European Conference on the Principles and Practice of Knowledge Discovery in Databases (PKDD 2004), 20-24 September, Pisa, Italy, Springer Verlag, Retrieved 11 August 2004 from: <http://www.lsi.us.es/~aguilar/ecml2004/FP4.PDF>.
- [Gaber05] Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S., (2005), "Mining data streams: a review", *SIGMOD Record*, vol. 34, no. 2, June, Special Interest Group on Management of Data (SIGMOD) of the Association for Computing Machinery (ACM), pp. 18-26.
- [Green00] Green, M., (2000), "How long does it take to stop? Methodological analysis of driver perception-brake times", *Transportation Human Factors*, v. 2, Lawrence Erlbaum Associates, Inc., pp.195-216.
- [Gross98] Grossman, R., (1998), *Supporting the Data Mining Process with Next Generation Data Mining Systems*, August, Enterprise Systems, 1105 Media Inc., Retrieved 7 June 2005 from Enterprise Systems website: <http://esj.com/article.aspx?ID=8139813919PM>.
- [Gross05] Grossman, R. L., Sabala, M., Alimohideen, J., Aanand, A., Chaves, J., Dillenburg, J., Eick, S., Leigh, J., Nelson, P., Papka, M., Rorem, D., Stevens, R., Vejcek, S., Wilkinson, L., and Zhang,

- P., (2005), "Real time change detection and alerts from highway traffic data", *Proceedings Of ACM/IEEE Supercomputing Conference 2005*, 12-18 November, Seattle, Washington, USA, IEEE Computer Society Press, pp. 62-69.
- [Gruyer05] Gruyer, D., Rakotonirainy, A., and Vrignon, J., (2005), "Advancement in advanced driving assistance systems tools: integrating vehicle dynamics, environmental perception and drivers' behaviours to assess vigilance", *Proceedings of the Intelligent Vehicles and Road Infrastructure Conference (IVRI '05)*, Melbourne, Australia, The Society of Automotive Engineers, Retrieved 22 August 2005 from: http://www.carrsq.qut.edu.au/documents/publication_083.pdf.
- [Harr04] Harrington, A., and Cahill, V., (2004), "Route profiling - putting context to work", *Proceedings of the 19th ACM Symposium on Applied Computing (SAC 2004)*, 14-17 March, Nicosia, Cyprus, ACM Press, pp. 1567-1573.
- [Hayw72] Hayward, J. Ch., (1972), *Near Miss Determination Through Use of A Scale of Danger*, Report no. TTSC 7115, The Pennsylvania State University, Pennsylvania, USA.
- [Hents00] Henstridge, J., Soet, W., Hill, D., and Roydhouse, R., (2000), *Analysis of Road Crash Statistics, Western Australia, 1990 to 1999: Report*, December, Office of Road Safety, Road Safety Council, Government of Western Australia, Retrieved 3 September 2007 from Office of Road Safety website: <http://www.officeofroadsafety.wa.gov.au/documents/crashstats1990-1999.pdf>.
- [Horo06] Horovitz, O., Krishnaswamy, S., and Gaber, M, M., (2006), "A fuzzy approach for interpretation of ubiquitous data stream clustering and its application in road safety", *Intelligent Data Analysis - Special Issue on Knowledge Discovery from Data Streams*, vol. 11, no. 1, IOS Press, pp.89-108.
- [Horst93] Horst, R. v. d. and Hogema, J., (1993), "Time-To-Collision and collision avoidance systems", *Proceedings of the 6th International Cooperation on Theories and Concepts in Traffic Safety (ICTCT) Workshop*, Salzburg, Austria, Retrieved 16 June 2007 from <http://www.ictct.org/workshops/93-Salzburg/Horst.pdf>.
- [Hsu02] Hsu, J., (2002), "Data mining trends and developments: the key data mining technologies and applications for the 21st century", *Proceedings of ISECON 2002*, vol. 19, 31 October to 3 November, San Antonio, Texas, USA, AITP Foundation for Information Technology Education, Retrieved 26 May 2005 from: <http://www.csse.monash.edu.au/~mgaber/DATA%20MINING%20TRENDS%20AND%20DEVELOPMENTS%20The%20Key%20Data%20Mining%20Technologies%20and%20Applications%20fo>

r%2021%20century.pdf.

- [Huang04] Huang, Q. and Miller, R., (2004), "Reliable Wireless Traffic Signal Protocols for Smart Intersections", *ITS America Annual Meeting*, Retrieved 10 May 2005 from http://www2.parc.com/spl/members/qhuang/papers/tlights_itsa.pdf
- [INTER05] INTERSAFE, (2005), *D40.4 Requirements for intersection safety applications*, Preventive and Active Safety Applications Integrated Project (PReVENT IP), European Commission, Retrieved 15 December 2005 from PReVENT IP website: http://www.prevent-ip.org/download/deliverables/INTERSAFE/PR-40430-SPD-050131-v23_VW_Intersafe_Requirements.pdf.
- [ISO85] International Organisation for Standardisation (ISO), (1985), *ISO 5807 Information processing - Documentation symbols and conventions; program and system flowcharts*, ISO website: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=11955
- [IVI02] Intelligent Vehicle Initiative, (2002), *IVI 8 Major Problem Areas*, Retrieved 17 August 2004 from IVI website: <http://www.its.dot.gov/ivi/ivi.htm>.
- [Jensen94] Jensen, P., (1994), *In Marconi's Footsteps 1894 to 1920 Early Radio*, Kangaroo Press Pty Ltd, New South Wales, Australia.
- [Jha04] Jha, S. and Mukherjee, A., (2004), "Advances in future mobile/wireless networks and services", *Computer Communications*, vol. 27, no. 8, May, Elsevier B.V., pp. 695-696.
- [Jones02] Jones, W. D., (2002), "Building safer cars", *IEEE Spectrum*, vol. 39, no. 1, January, IEEE Computer Society Press, pp. 82-85.
- [Julien02] Julien, C. and Roman, G-C., (2002), "Egocentric context-aware programming in ad hoc mobile environments," *Proceedings of the 10th International Symposium on the Foundations of Software Engineering (FSE-10)*, 18-22 November, Charleston, South Carolina, USA, ACM Press, pp. 21-30.
- [Kajiya04] Kajiya, Y., Yamagiwa, Y., Masaoka, H., (2004), "Road web markup language and its application examples", *Proceedings of 11th ITS World Congress 2004*, 18-22 October, Nagoya, Japan, Intelligent Transportation Society of America, Retrieved 12 January 2005 from: <http://rwml.its-win.gr.jp/papers-pdf/RWML-ITSWC2004AichiNagoya.pdf>.
- [Kargup04] Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., Dull, J., Sarkar, K., Klein, M., Vasa, M. and Handy, D., (2004), "VEDAS: A mobile and distributed data stream mining system for real-time vehicle monitoring", *Proceedings of the SIAM*

International Data Mining Conference, 22-24 April, Florida, USA, Society for Industrial and Applied Mathematics (SIAM), pp. 300-311.

- [Kerry08] Kerry, S. J. and Armstrong, L., (2008), *Status of Project IEEE 802.11 Task Group p: Wireless Access in Vehicular Environments (WAVE)*, Retrieved 29 May 2008 from IEEE P802 website: http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm.
- [Knoll06] Knoll, T., (2006), *Robert Bosch GmbH – Business / Information – Press information*, Retrieved 1 May 2008 from BOSCH website: <http://www.bosch-presse.de/TBWebDB/en-US/PressText.cfm?id=2567>.
- [Krish05] Krishnaswamy, S., Loke, S.W., Rakotonirainy A., Horovitz, O., and Gaber, M. M., (2005), “Towards situation-awareness and ubiquitous data mining for road safety: rationale and architecture for a compelling application”, *Proceedings of the Intelligent Vehicles and Road Infrastructure Conference*, 16 February, Melbourne, Australia, Retrieved 22 August 2005 from <http://www.csse.monash.edu.au/~mgaber/CameraReadyIVRI05.pdf>.
- [Kwas07] Kwasniak, A. and Tarko, A. P., (2007), “Tool for Supporting High-Crash Site Investigation”, *Proceedings of Transportation Research Board Annual Meeting 2007 Paper #07-1545*, Transportation Research Board, Retrieved 6 March 2007 from: <http://web.ics.purdue.edu/~akwasnia/last.pdf>.
- [Kwon06] Kwon, O., Lee, S-H., Kim, J-S., Kim, M-S., Li, and K-J., (2006), “Collision prediction at intersection in sensor network environment”, *Proceedings of IEEE Intelligent Transportation System Conference*, 17-20 September, Toronto, Canada, IEEE Intelligent Transportation Systems Society, pp. 982 – 987.
- [Lages04] Lages, U., (2004), “INTERSAFE – new European approach for intersection safety, funded by the European Commission in 6th framework program”, *11th ITS World Congress 2004*, 18-22 October, Nagoya, Japan, Intelligent Transportation Society of America, Retrieved 21 March 2005 from: http://www.prevent-ip.org/download/Events/20041018-22%20ITS%20World%20Congress_Nagoya/20041024%20ITS2004_EU-FP6_INTERSAFE.pdf.
- [Lieb05] Lieberman, E. and Rathi, A. K., (2005), “Traffic simulation”, *Revised Monograph on Traffic Flow Theory*, Chapter 10, Turner-Fairbank Highway Research Center, United States Department of Transportation - Federal Highway Administration, Retrieved 21 January 2008 from <http://www.tfhrc.gov/its/tft/tft.htm>.

- [Light02] Lightman, A. and Rojas, W., (2002), *Brave New Unwired World*, John Wiley & Sons, Inc, New York, USA.
- [Mani93] Manivannan, S., (1993), "Robotic collision avoidance in a flexible assembly cell using a dynamic knowledge base", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, May-June, IEEE Computer Society Press, pp. 766-782.
- [Miller02] Miller, R. and Huang, Q., (2002), "An adaptive peer-to-peer collision warning system", *IEEE 55th Vehicular Technology Conference (VTC) Spring 2002*, 6-9 May, Birmingham, Alabama, USA, IEEE Computer Society Press, pp. 317-321.
- [Miller07] Miller, J. and Horowitz, E., (2007), "FreeSim – a free real-time freeway traffic simulator", *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*, 30 September – 3 October, Seattle, WA, USA, IEEE Intelligent Transportation Systems Society, pp. 18-23.
- [Mitch97] Mitchell, T. M., (1997), *Machine Learning*, McGraw-Hill, Inc., New York, USA.
- [Mitre99] Mitretek, R. R., (1999), *Problem Area Descriptions: Motor Vehicle Crashes - Data Analysis and IVI Program Emphasis*, Intelligent Vehicle Initiative, U.S. Department of Transportation, ITS Joint Program Office, Retrieved 1 February 2005 from: <http://www.itsdocs.fhwa.dot.gov/jpodocs/EDLBrow/9101!.pdf>.
- [Mitro05] Mitrovic, D., (2005), "Reliable method for driving events recognition", *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, June, IEEE Intelligent Transportation Systems Society, pp.198-205.
- [Moran01] Moran, T. P. and Dourish, P., (2001), "Introduction to this special issue on context-aware computing", *Human-Computer Interaction: A Journal of Theoretical, Empirical, and Methodological Issues of User Science and of System Design*, Vol. 16, No. 2-4, Taylor and Francis Group, London, UK, pp. 87-95.
- [Moriar98] Moriarty, D., & Langley, P., (1998), "Learning cooperative lane selection strategies for highways", *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 26-30 July, Madison, Wisconsin, USA, AAAI Press, pp. 684-691.
- [Morr02] Morrow, R., (2002), *Bluetooth Operation and Use*, McGraw-Hill Inc., New York, USA.
- [Nakata04] Nakata, T. and Takeuchi, J., (2004), "Mining traffic data from probe-car system for travel time prediction," *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 22-25 August, Seattle, Washington,

USA, ACM Press, pp. 817-822.

- [Ogden94] Ogden, K. W. and Newstead, S. V., (1994), *Analysis of Crash Patterns at Victorian Signalised Intersections*, Monash University Accident Research Centre Report No. 60, February, Australian Road Research Board, Retrieved 1 February 2005 from: <http://www.monash.edu.au/muarc/reports/muarc060.pdf>.
- [Oliver00] Oliver, N. and Pentland, A., (2000), "Graphical models for driver behavior recognition in a smart car", *Proceedings of IEEE Intelligent Vehicles Symposium*, 3-5 October, Dearborn, Michigan, IEEE Intelligent Transportation Systems (ITS) Council, pp. 7-12.
- [Onken94] Onken, R., (1994), "Daisy, an adaptive knowledge-based driver monitoring and warning system", *Proceedings of 1994 Vehicle Navigation and Information Systems Conference*, 31 August - 2 September, Yokohama, Japan, IEEE Computer Society Press, pp. 3-10.
- [Owen00] Owen, L. E., Zhang, Y., Rao, L., McHale, G., (2000), "Traffic flow simulation using CORSIM", *Proceedings of the 2000 Winter Simulation Conference*, 10-13 December, Orlando, Florida, USA, IEEE Computer Society Press, pp. 1143-1147.
- [Medina05] Medina, J. S., Moreno, M. G., and Royo, E. R., (2005), "Stochastic vs deterministic traffic simulator. Comparative study for its use within a traffic light cycles optimization architecture", *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*, Lecture Notes in Computer Science (LNCS), vol. 3562/2005, Springer Verlag, pp. 622-631.
- [Pearl88] Pearl, J., (1988), *Probabilistic Reasoning with Intelligent Systems*, Morgan & Kaufman, San Mateo, USA.
- [Queens07] Queensland Transport – Queensland Government, (2007), *Web Crash*, Retrieved 5 February 2008 from Web Crash website: <https://www.webcrash.transport.qld.gov.au/webcrash2/>.
- [Quinlan86] Quinlan, J. R., "Induction of Decision Trees", (1986), *Machine Learning*, vol. 1, no. 1, Springer Verlag, pp. 81-106.
- [Rabin89] Rabiner, L., (1989), "A tutorial on hidden markov models and selected applications in speech recognition", *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286.
- [Redm99] Redmill, K.A. and Ozguner, U., (1999), "VATSIM: a vehicle and traffic simulator", *IEEE/IEE/JSAI 1st International Intelligent Transportation Systems Conference*, October 5-8, Tokyo, Japan, IEEE Intelligent Transportation System Society, pp. 656-661.
- [Salim05] Salim, F.D., Krishnaswamy, S., Loke, S. W. and Rakotonirainy,

- A., (2005), "Context-aware ubiquitous data mining based agent model for intersection safety", *Proceedings of the 2005 IFIP Conference on Embedded and Ubiquitous Computing Workshops (EUCW 2005)*, in conjunction with EUC 2005, 6-9 December, Nagasaki, Japan, Lecture Notes in Computer Science, Springer-Verlag, pp. 61-70.
- [Salim06] Salim, F. D., Loke, S. W., Rakotonirainy, A. and Krishnaswamy, S., "U & I Aware (Ubiquitous Intersection Awareness): A framework for intersection safety", accepted for publication in November 2006 as a book chapter in *Handbook on Mobile and Ubiquitous Computing: Innovations and Perspectives*, to be published by American Scientific Publishers.
- [Salim07a] Salim, F. D., Loke, S. W., Rakotonirainy, A. and Krishnaswamy, S., (2007), "U&I Aware: a framework using data mining and collision detection to increase awareness for intersection users", *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, in conjunction with AINA-2007, 21-23 May, Niagara Falls, Canada, IEEE Computer Society Press, pp. 530-535.
- [Salim07b] Salim, F. D., Loke, S. W., Rakotonirainy, A. and Krishnaswamy, S., (2007), "Simulated intersection environment and learning of collision and traffic data in the U&I Aware framework", *In Proceedings of The 4th International Conference on Ubiquitous Intelligence and Computing (UIC-07)*, 11-13 July, Hong Kong, China, Springer Berlin/Heidelberg, pp. 153-162.
- [Salim07c] Salim, F. D., Loke, S. W., Rakotonirainy, A., Srinivasan, B. and Krishnaswamy, S., (2007), "Collision pattern modeling and real-time collision detection at road intersections", *Proceedings of The 10th International IEEE Conference on Intelligent Transportation Systems*, 30 September - 3 October, Seattle, Washington, USA, IEEE Intelligent Transportation Systems Society, pp. 161-166.
- [Salim08a] Salim, F. D., Cai, L., Indrawan, M. and Loke, S. W., (2008), "Road intersections as pervasive computing environments: towards a multiagent real-time collision warning system", *In Proceedings of the 1st IEEE Workshop on Agent Technologies for Pervasive Communities*, in conjunction with the Sixth IEEE International Conference on Pervasive Computing and Communications (Percom '08), 17-21 March, Hong Kong, China, IEEE Computer Society Press, pp. 621-626.
- [Salim08b] Salim, F. D., (2008), "A generic and real-time collision warning and avoidance system in a ubiquitous intersection environment", *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008): Google PhD Forum*, 17-21 March, Hong Kong, China, pp. 37-40.

- [Sara04] Sara, V., (2004), "Down to earth", *Friday Magazine*, May, Retrieved 4 November 2004 from Australian Research Council website: http://www.arc.gov.au/pdf/features/smart_car.pdf.
- [Satake07] Satake, M., Takatori, Y. and Hasegawa, T., (2007), "Performance evaluation of driving assistance systems that uses sensors and/or communications", *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*, 30 September - 3 October, Seattle, IEEE Intelligent Transportation Systems Society, pp. 974-979.
- [Sauni07] Saunier, N., Sayed, T. and Lim, C., (2007), "Probabilistic collision prediction for vision-based automated road safety analysis", *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems*, 30 September - 3 October, Seattle, Washington, USA, IEEE Intelligent Transportation Systems Society, pp. 872-878.
- [Schil94] Schilit, B., Adams, N. and Want, R. (1994), "Context-aware computing applications", *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 8-9 December, Santa Cruz, California, USA, IEEE Computer Society Press, pp. 85-90.
- [Schlei02] Schleiffer, R., (2002), "Intelligent agents in traffic and transportation", (editorial), *Transportation Research C*, vol. 10, no. 5-6, 2002, Elsevier, Kidlington, ROYAUME-UNI, pp. 325-329.
- [Seeing05] Seeing Machines Pty Ltd, (2005), *Seeing Machines Website*, Retrieved 10 March 2005 from Seeing Machines Website: <http://www.seeingmachines.com>.
- [Sharke03] Sharke, P., (2003), "Smart cars", *Mechanical Engineering*, March, Retrieved 24 August 2004 from Mechanical Engineering website: <http://www.memagazine.org/backissues/mar03/features/smartcar/smartcar.html>.
- [Shea00] Shea, J., (2000), *Brief History of Wireless Communications*, Retrieved 3 September 2007, from: <http://www.wireless.ece.ufl.edu/jshea/eel6509/misc/history.html>.
- [Sicking00] Sicking, D. L and Mak, K. K., (2000), "Improving roadside safety by computer simulation", *Transportation in the New Millennium , State of the Art and Future Directions, Perspectives From Transportation Research Board Standing Committees*, January, Transportation Research Board, Retrieved 19 July 2008 from: <http://onlinepubs.trb.org/Onlinepubs/millennium/00053.pdf>.
- [Singh03] Singh, S., (2003), "Identification of driver and vehicle characteristics through data mining the highway crash data",

Federal Committee on Statistical Methodology 2003 Conference, November, Arlington, Virginia, Retrieved 9 August 2004 from <http://www.fcsm.gov/03papers/Singh8c.pdf>.

- [Singhal01] Singhal, A., (2001), "Modern information retrieval: a brief overview", *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 24 no. 4, IEEE Computer Society Press, pp. 35-43.
- [Sivaha04] Sivaharan, T., Blair, G., Friday, A., Wu, M., Duran-Limon, H., Okanda, P. and Sørensen, C-F., (2004), "Cooperating sentient vehicles for next generation automobiles", *In Proceedings of ACM/USENIX MobiSys 2004 International Workshop on Applications of Mobile Embedded Systems*, 6 June, Boston, USA, ACM Press, Retrieved 10 August 2004 from: <http://www.comp.lancs.ac.uk/computing/research/mpg/reflection/papers/MobiSys04.pdf>.
- [Strob04] Strobel, T., Servel, A., Coue, C. and Tatschke, T., (2004), *Compendium on Sensors - State-of-the-art of Sensors and Sensor Data Fusion for Automotive Preventive Safety Applications*, ProFusion IP Deliverable, PReVENT IP, European Commission, Retrieved 21 March 2005 from from PReVENT IP website: http://prevent.ertico.webhouse.net/download/pdf/deliverables/PR-13400-IPD-040531-v10-Compendium_on_Sensors.pdf.
- [Stubbs03] Stubbs, K., Arumugam, H., Masoud, O., McMillen, C., Veeraraghavan, H., Janardan, R. and Papanikolopoulos, N., (2003), "A real-time collision warning system for intersections", *Proceedings of Intelligent Transportation Systems America*, May, Minneapolis, USA, Retrieved 1 February 2005 from: <http://colinm.org/papers/Stubbs-2003-ITSA-final.pdf>.
- [Sun07] Sun Microsystems, (2007), *Mobile Information Device Profile (MIDP) Overview*, Retrieved 5 May 2007 from Sun Java website: <http://java.sun.com/products/midp/overview.html>.
- [USDOT99] U.S. Department of Transportation – Federal Highway Administration, (1999), *Intersection Collision Warning System*, Retrieved 1 February 2005 from: <http://www.tfhrc.gov/safety/pubs/99103.pdf>.
- [USDOT00] U.S. Department of Transportation, (2000), *Intelligent Vehicle Initiative: Business Plan*, Intelligent Transportation Systems Joint Program Office, Retrieved 1 February 2005 from: <http://www.its.dot.gov/ivi/docs/BP701.pdf>.
- [USDOT04] U.S. Department of Transportation – Federal Highway Administration, Institute of Transportation Engineers, (2004), *Intersection Safety Briefing Sheet*, Retrieved 12 January 2005 from:

<http://www.ite.org/library/IntersectionSafety/BreifingSheets.pdf>.

- [USDOT07] U. S. Department of Transportation, (2007), *Cooperative Intersection Collision Avoidance Systems*, Retrieved 18 Sep 2007 from: <http://www.its.dot.gov/cicas/index.htm>.
- [UQ06] UQ News Online, (2006), *\$17 billion annual bill for road trauma*, June, The University of Queensland, Brisbane, Australia, Retrieved 26 July 2008 from The University of Queensland website: <http://www.uq.edu.au/news/index.html?article=9863>.
- [Veera02] Veeraraghavan, H., Masoud, O. and Papanikolopoulos, N., (2002), "Vision-based monitoring of intersections", *Proceedings of Intelligent Transportation Systems Conference*, 3-6 September, Singapore, IEEE Intelligent Transportation Systems Society, pp. 7-12.
- [Verid00] Pieroxicz, J., Jocoy, E., Lloyd, M., Bittner, A. and Pirson, B., (2000), "Intersection collision avoidance using ITS countermeasures", *Final Report: Performance Guidelines*, No. DOT HS 809 171, National Highway Traffic Safety Administration, Retrieved 1 February 2005 from: http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/@@L01!.pdf
- [Verís02] Veríssimo, P., Cahill, V., Casimiro, A., Cheverst, K., Friday, A. and Kaiser, J., (2002), "CORTEX: towards supporting autonomous and cooperating sentient entities", *Proceedings of European Wireless 2002*, February, Florence, Italy, pp. 595-601.
- [Vidal02] Vidales, P. and Stajano, F., (2002), "The sentient car: context-aware automotive telematics", *First European Workshop on Location Based Services*, 16 September, London, UK, Retrieved 10 August 2004 from: <http://www.cl.cam.ac.uk/~fms27/papers/2002-VidalesSta-car--lbs.pdf>.
- [Wang06] Wang, Y., Papageorgiou, M., and Messmer, A., (2006), "RENAISSANCE: a real-time freeway network traffic surveillance tool", *Proceedings IEEE 9th International Intelligent Transportation Systems Conference*, 17-20 September, Toronto, Canada, IEEE Intelligent Transportation Systems Society, pp. 839-844.
- [Weev03] Weevers, I., Kuipers, J., Zwiers, J., Dijk, E.M.A.G. van., and Nijholt, A., (2003), "The Virtual Driving Instructor: A Multi-agent Based System for Driving Instruction", *CTIT technical reports series (13813625)*, Vol. 03, No.17, CTIT, pp. 3-17.
- [Weiser91] Weiser, M., (1991), "The computer for the 21st century," *Scientific American* 265, No.3, September, pp. 94-104; reprinted in IEEE

Pervasive Computing, (2002), Jan-Mar, pp. 19-25.

- [Werner03] Werner, J., (2003), *Inside the USDOT's "Intelligent Intersection" Test Facility*, ITS Cooperative Deployment Network, Retrieved 1 February 2005 from: <http://ntlsearch.bts.gov/tris/record/tris/00961269.html>.
- [Wiki07a] Wikipedia, (2007), *Automatic Number Plate Recognition*, Retrieved 7 June 2007, from Wikimedia Foundation, Inc., http://en.wikipedia.org/wiki/Automatic_number_plate_recognition
- [Wiki07b] Wikipedia, (2007), *Global Positioning System*, Retrieved 7 June 2007, 2007, from Wikipedia Foundation Inc: http://en.wikipedia.org/wiki/Global_Positioning_System.
- [Wimax07] Wimax Forum, (2007), *Wimax Forum – Home*, Retrieved 3 September 2007 from Wimax Forum: <http://www.wimaxforum.org/home/>.
- [Witten05] Witten, I.H. and Frank, E., (2005), *Data Mining: Practical machine learning tools and techniques*, 2nd edition, Elsevier, San Francisco, USA.
- [Woold02] Wooldridge, M., (2002), *An Introduction to Multiagent Systems*, John Wiley & Sons, Chichester, England.
- [Xuan06] Xuan, Y. and Coifman, B., (2006), "Lane change maneuver detection from probe vehicle DGPS data", *Proceedings of the Intelligent Transportation Systems Conference 2006 (ITSC '06)*, Toronto, Canada, IEEE Intelligent Transportation Systems Society, pp. 624 – 629.
- [Yang96] Yang, Q. and Koutsopoulos, H. N., (1996), "A microscopic traffic simulator for evaluation of dynamic traffic management systems", *Transportation Research Part C: Emerging Technologies*, Vol. 4, No. 3, June, Elsevier Science Ltd, pp. 113-129.
- [Zhang05] Zhang, K. and Taylor, M. A. P., (2005), "Simulation of freeway incident detection using Bayesian networks", *Proceedings of the Intelligent Vehicles and Road Infrastructure Conference (IVRI '05)*, 7 February, Melbourne, Australia, The Society of Automotive Engineers, Retrieved 22 August 2005 from: <http://arrow.unisa.edu.au:8080/vital/access/manager/Repository/unisa:27527>.