An Automated Co-driver for Advanced Driver Assistance Systems: The next step in road safety.

Luke Sebastian Fletcher

A thesis submitted for the degree of
Doctor of Philosophy
at the Australian National University

Department of Information Engineering
Research School of Information Sciences and Engineering
Australian National University

April 2008

Statement of originality

These doctoral studies were conducted under the supervision of Professor Alexander Zelinsky. The work submitted in this thesis is a result of original research carried out by myself, except where duly acknowledged, while enrolled as a PhD student in the Department of Information Engineering at the Australian National University. It has not been submitted for any other degree or award.

Luke Fletcher

Acknowledgements

First, I wish to thank the Centre for Accident Research and Road Safety Queensland (CARRS-Q), my scholarship sponsor, for giving me the opportunity to undertake this work.

I would like to thank Alex for keeping me on track, his understanding and ready advice when things didn't go to plan and for always having his eye on the big picture. I would also like to thank Prof. Dickmanns for taking the time to come and share his vast experience with us.

Thanks to Lars Petersson, who has always been a great sounding board, not to mention a dig-in helper to get the experiments done. Similarly may I thank all the test drivers, particularly Niklas Pettersson, for their enduring patience during the interminable tedium of debugging on the run. Also thanks to Raj Gore for the positive encouragement to complete the work.

I am, of course, obliged to thank the other members of "Order of the Green Couch": namely Gareth Loy and Nicholas Apostoloff, with whom I collaborated in the development of the "Distillation algorithm" computer vision tracking framework and the initial applications of face tracking and lane tracking.

Finally, I would like to thank my long suffering family: Leah, Alice, Matilda, the Habgoods and the Twitts for their tireless support. In memory of The Pern.

Luke Fletcher.

Abstract

Road vehicles offer unique challenges in human-machine interaction. Road vehicles are becoming, in effect, robotic systems that collaborate with the driver. As the automated systems become more capable, how best to manage the on-board human resources becomes an intriguing question. Combining the strengths of machines and humans while mitigating their shortcomings is the goal of this intelligent-vehicle research.

Almost every driver has avoided an accident thanks to a warning from a vigilant passenger. In this work we develop the computerized equivalents of the core competencies of a vigilant passenger. The developed systems are then integrated to create a new kind of Advanced Driver Assistance System (ADAS) an *Automated Co-driver*. We show that the *Automated Co-driver* is a powerful concept, that could be the next significant step in road safety.

Our work has concentrated on road scene computer vision and the scope for improvement on two fronts. First, looking outside the vehicle, we investigated and developed road scene monitoring systems. The systems track the lane, obstacles, road signs and the "visual monotony" of the scene ahead of the vehicle. A visual-ambiguity tolerant framework was developed to extract information about the road scene from noisy sensor data. The algorithm was used for robust lane tracking and obstacle detection. A fast and effective symbolic sign reading system was also developed, as was a road scene visual monotony and clutter estimator. Visual monotony, a likely key contributor to fatigue, was estimated by measuring the variability in the road scene over time.

Secondly, these developed components were then combined with the vehicle state, and existing pedestrian detection and a driver eye-gaze monitoring system, to form a comprehensive Advanced Driver Assistance System. In the integrated system the measured driver eye-gaze was correlated with detected road scene features to create a new class of Advanced Driver Assistance Systems. Systems with

the potential to detect driver inattention by monitoring the driver's *observations*, not just the driver's actions. The essential combination of driver, vehicle and road scene monitoring enables us to obtain the missing driver-state information required to contextualise driver behaviour. Finally, we conducted a series of trials on the developed *Automated Co-driver* ADAS. Through our analysis and these trials we show that it is feasible to detect live in-vehicle correspondences between driver eye-gaze and road scene features to estimate the driver's observations and potentially detect driver inattention. The correlation between eye-gaze and road scene features is shown to be particularly useful in the detection of unobserved road events.

Publications resulting from this thesis

Book chapters

- Lars Petersson, Luke Fletcher, Nick Barnes and Alexander Zelinsky, Towards Safer Roads by Integration of Road Scene Monitoring and Vehicle Control, *Advances in Applied Artificial Intelligence (Computational Intelligence and Its Applications)*, edited by John Fulcher, Idea group, IBSN:159140827X, March 2006.

Journal publications

- Luke Fletcher and Alexander Zelinsky, An Automated Co-driver Advanced Driver Assistance System the next step in road safety, *International Journal of Robotics Research*, 2008.

- Nick Barnes, Luke Fletcher and Alexander Zelinsky, Real time speed sign detection using the radial symmetry detector, *IEEE Transactions on Intelligent Transport Systems*, 2008.

- Lars Petersson, Luke Fletcher, Nick Barnes and Alexander Zelinsky, Towards Safer Roads by Integration of Road Scene Monitoring and Vehicle Control, *International Journal of Robotics Research*, vol. 25, no.1, January 2006.

- Luke Fletcher, Gareth Loy, Nick Barnes and Alexander Zelinsky, Correlating driver gaze with the road scene for Driver Assistance Systems, *Robotics and Autonomous Systems*, vol. 52, iss. 1, 2005.

- Luke Fletcher, Lars Petersson, Nicholas Apostoloff and Alexander Zelinsky, Vision in and our of Vehicles, *IEEE Intelligent Systems magazine*, June 2003.

Conference publications

- Luke Fletcher and Alexander Zelinsky, Context Sensitive Driver Assistance based on Gaze - Road Scene Correlation, *International Symposium on Experimental Robotics (ISER)*, Rio De Janeiro, Brazil, July 2006.
- Lars Petersson, Luke Fletcher and Alexander Zelinsky, A Framework for a Driver-In-The-Loop Driver Assistive System, *International IEEE Conference on Intelligent Transport Systems*, Vienna, Austria, September 2005.
- Luke Fletcher and Alexander Zelinsky, BEST STUDENT PAPER: Driver State Monitoring to Mitigate Distraction, *The Australasian College of Road Safety (ACRS), STAYSAFE Committee NSW Parliament International Conference on Driver Distraction*, 2-3 June 2005, Parliament House, Sydney.
- Luke Fletcher, Lars Petersson, and Alexander Zelinsky, Road Scene Monotony Detection in a Fatigue Management Driver Assistance System, *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2005)*, Las Vegas USA, June 2005.
- Luke Fletcher, Lars Petersson, Nick Barnes , David Austin and Alexander Zelinsky, A Sign Reading Driver Assistance System Using Eye Gaze, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Automation (ICRA2005)*, Barcelona Spain, April 2005.
- Luke Fletcher, Nick Barnes and Gareth Loy, Robot Vision for Driver Support Systems, *Proceedings of of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2004) Workshop on Advances in Robot Vision*, Sendai Japan, September 2004.
- Luke Fletcher and Alexander Zelinsky, Super-resolving Signs for Classification, *Proceedings of the Australasian Conference on Robotics and Automation (ACRA2003)*, Canberra, Australia, December 2004.
- Lars Petersson, Luke Fletcher, Nick Barnes and Alexander Zelinsky, An interactive Driver Assistance System monitoring the scene in and out of

the vehicle, *Proceedings of the International Conference on Robotics and Automation (ICRA2004)*, New Orleans, USA, April 2004.

- Andrew Dankers, Luke Fletcher, Lars Petersson and Alex Zelinsky, Driver Assistance: Contemporary Road Safety, *Proceedings of the Australasian Conference on Robotics and Automation (ACRA2003)*, Brisbane, Australia, December 2003.

- Lars Petersson, Luke Fletcher, Nick Barnes and Alexander Zelinsky, Towards Safer Roads by Integration of Road Scene Monitoring and Vehicle Control, *Proceedings of the International Conference on Field and Service Robotics (FSR03)*, Mt. Fuji, Japan, July 2003.

- Luke Fletcher, Lars Petersson and Alexander Zelinsky, Driver Assistance Systems based on Vision In and Out of Vehicles, *Proceedings of the IEEE Intelligent Vehicles Symposium (IV2003)*, Columbus, Ohio, USA, June 2003.

- Gareth Loy, Luke Fletcher, Nicholas Apostoloff and Alexander Zelinsky, An Adaptive Fusion Architecture for Target Tracking, *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition (FG2002)*, Washington D.C., USA, May 2002.

- Luke Fletcher, N. Apostoloff, J. Chen and Alexander Zelinsky, Computer Vision for Vehicle Monitoring and Control, *Proceedings of the Australasian Conference on Robotics and Automation (ACRA2001)*, Sydney, Australia, November 2001.

- David Austin, Luke Fletcher and Alexander Zelinsky, Mobile Robotics in the Long Term, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2001)*, Hawaii, USA, October 2001.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The daily occurrence of traffic accidents has become the horrific price of modern life. Complacency about the dangers of driving contributes to the death of more than one million people worldwide in traffic accidents each year (WHO, 2001). Fifty million more are seriously injured (WHO, 2001). In OECD countries, road accidents are the primary cause of death for males under the age of 25(OECD, 2006).

Law enforcement, improved vehicle design and public awareness campaigns have had a marked effect on accident rates from the 1970s (ATSB, 2004b; OECD, 2006). Little, however, has been achieved on the hard cases of road safety, such as fatigue, distraction and inattention (Stutts *et al.*, 2001; Neale *et al.*, 2005; Treat *et al.*, 1979). In Australia in recent years growing concern has generated a national inquiry into fatigue (Australian Government, 2000) and two state enquiries into driver distraction (VicRoadSafety, 2006; NSWStaySafe, 2005).

Although the terms "distraction" and "inattention" seem the same, they have a distinctive definition in this thesis. A *distraction* is an event such as a phone call or the Sun in moving into view, whereas *inattention* is a lack of attention to a given task. A distraction can cause a driver to be inattentive, but other factors - such as fatigue - can also give rise to inattention. Although we may see distraction as the cause and inattention as its effect, in many ways it is inattention - and its consequences - that casts the widest net when we look at the causes of traffic accidents.

Almost every driver has experienced a warning from a passenger, perhaps alerting him or her to an obscured car while merging lanes, or a jaywalking pedestrian.

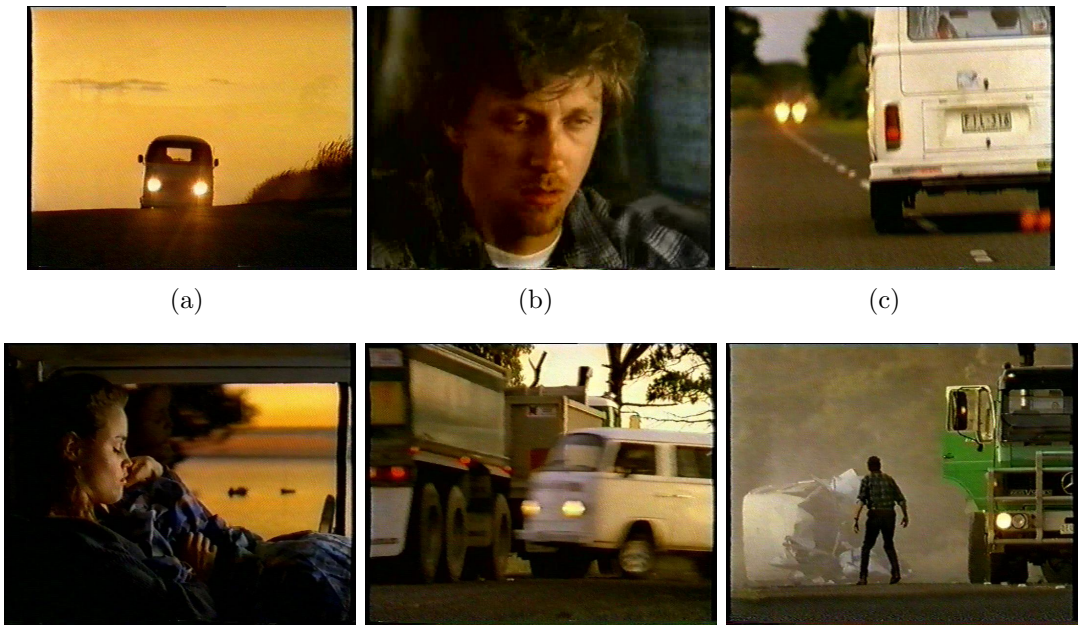
These warnings of inattention save countless lives every day. In a recent keynote address (Regan, 2005), Professor Michael Regan recently appointed Director of Road Safety Research at INRETS (The French National. Institute for Transport and Safety Research), highlighted the fact that, unlike other complex, potentially dangerous vehicles like planes and ships, road vehicles are operated by a single person. That single person is prone to error and may be, due to the day-to-day nature of driving, slow to recognise potential hazards. Professor Regan speculated that the unlikely introduction of co-drivers into road vehicles could be the key to road accident prevention.

Road vehicles offer unique challenges in human-machine interaction. Road vehicles are becoming, in effect, automated systems that collaborate with the driver. As the computerized systems in vehicles become more capable, the question of how best to manage the on-board human resources becomes an intriguing question. Combining the strengths of machines and humans, and mitigating their shortcomings, is the goal of our intelligent-vehicle research.

Road vehicles also provide unique challenges for computer vision. Vehicles operate in a dynamic environment with fast moving objects and extremes of illumination. In contrast the road infrastructure is highly regulated, predictable and designed for easy visual perception. Road vehicles are driven by rule and convention, lanes and signs are engineered to a standard.

To illustrate the aim of this research, consider the example featured as part of an award-winning Victorian Transport Accident Commission advertising campaign (TAC, 1994). The advertisement 'Night-shift' portrayed a young couple who set off on a long drive overnight to their weekend destination (see Figure 1.1). Without intervention the driver succumbs to fatigue, veers into the oncoming traffic lane and collides with a truck. The full video clip is included on the Appendix DVD-ROM (Page 257).

No reasonable policing or awareness-raising initiative is guaranteed to save this couple. But what if there were someone or something else watching out for them? Could a computer system act as their guardian angel?



(a) (b) (c)

(d) (e) (f)

Figure 1.1: Victorian transport accident commission advertisement (TAC, 1994) of dangerously drowsy driver with no support. (a) Dawn after driving all night. (b) The driver is succumbing to fatigue. (c) The vehicle veers onto the wrong side of the road. (d) A possible second set of eyes, the passenger, rests. (e) The driver rouses but cannot correct in time. (e) Vehicle crashes. The full video of the advertisement is in the Appendix DVD-ROM (Page 257).

As this thesis will show, we are coming to an age when we can construct a computer system in a vehicle that can collaborate with the driver to manage the driving task. In this thesis we define the concept of an "Automated Co-driver" as an intelligent agent with the core competencies of a vigilant passenger.

Let's imagine the same accident scenario with an Automated Co-driver on board.

In the first instance, an Automated Co-driver could have intervened and advised the driver to stop before he reached the point of exhaustion. The system could have noted how long the driver had been driving, the time of day, what breaks had been taken and estimated time till arrival at the destination. An Automated Co-driver capable of computer vision could have also noticed that the driver looked tired and even noted the monotonous appearance of the stretch of road. The system would have noted the deterioration in the driving performance, as indicated by the increased swing in the driver's lane position (Thiffault and Bergeron, 2003) or by the drowsy eye-gaze pattern (Wierwille and Ellsworth, 1994). Finally, the system could have detected the impending danger of the lane de-

parture and the oncoming obstacle, determined that the driver was inattentive and generated increasingly assertive interventions. This system - which brings together the strengths of the driver, a vigilant passenger and the endurance of an automated system - has the very real potential to prevent the death of these two people.

Intelligent vehicle research has already developed systems to relieve some of the tedious aspects of driving, such as adaptive cruise control (ACC) and lane keeping. However, we believe the key to a substantial reduction in road accidents lies in addressing the primary contributing factor in road accidents, driver error (Treat *et al.*, 1979). Advanced Driver Assistance Systems using direct driver monitoring provide the opportunity to tackle the hard cases in road safety: fatigue, distraction and inattention.

We use the term "Advanced Driver Assistance Systems" to denote systems that assist the driver through some intelligent agent as opposed to driver assistance systems currently on the market such as: blind spot cameras, ABS, cruise control or rain sensing wipers which operate in a purely mechanistic way.

1.1 Principal objectives

The goal of this research is to develop an in-vehicle computerized system able to operate and interact with the driver as an *Automated Co-driver* with the equivalent core competencies of a vigilant human passenger.

We define an Automated Co-driver as an intelligent agent capable of:

- Alerting the driver to a unseen threat (by the driver).
- Suppressing alarms when the driver is already monitoring the event.
- Assessing the driver's fitness regarding fatigue or distraction based on the prevailing conditions.
- Assisting the driver in a manner natural to the task of driving.
- In some cases, taking control of the vehicle if the driver is incapacitated.

Though it is compatible with our model of an Automated Co-driver, autonomy per se is not the goal of our experiments. There has been considerable research in this area (see Chapter 2), but little has been done to interface autonomous systems technology with the driver. Our interest is to improve road safety via observation and interaction with the driver, whereas actuated systems often have to usurp the driver to achieve their goals.

The key to the development of an Automated Co-driver is the creation of an Advanced Driver Assistance System able to understand the behaviour of the driver and intervene while there is time for a modest correction - significantly before the imminent "pre-crash" scenario.

To put the behaviour of the driver in context, the system must receive the same cues from the road environment as the driver. The system must locate the road ahead of the vehicle, identify threats and understand road signs - all of which influence the driver's behaviour.

While there have been many successful systems developed in road scene computer vision, there are several aspects of the road scene computer vision problem that plague many systems. The first objective of this work is the development of a computer vision algorithm designed explicitly:

- To cope with temporary visual ambiguities characteristic of the visual projection of the real-world scene.
- To self tune to the current road conditions across the diverse variance of weather and road environments.
- To be scalable so that best use can be made of the available computational resources.

The developed algorithm is to be used for lane tracking, particularly to combine previous vision algorithms proved to be effective in the road environment to operate over a wide range of road conditions. The same algorithm is also to be used for robust obstacle detection and tracking.

The next objective is to design a road-sign reading system capable of reliably detecting and understanding road signs presented to the driver as they are encountered.

The road scene also can affect the driver's behaviour in another, more subtle, way. Thiffault and Bergeron (2003) demonstrated that sparse road environments over time are a contributing factor to driver fatigue. One objective of this research will be to develop an attainable metric to quantify the road scene variability over time. Such a metric of visual monotony[1] has the potential to be useful determinant for assessing the driver's fitness. At the other end of the monotony scale is the case of too much detail in the road scene overwhelming the driver (Mourant *et al.*, 1969). We will investigate road scene clutter measures for driver workload management.

The final research objective is to demonstrate that the potential Automated Co-driver capabilities listed above could be realised using the developed Advanced Driver Assistance System. Road scene events must be detected and correlated with the driver's eye-gaze direction to potentially determine if the driver has witnessed a given event. This way the system could potentially infer the driver's behaviour and intervene only if required.

The achievement of the these objectives has produced a number of contributions to intelligent vehicle, road safety and computer vision systems research.

1.2 Contributions

The primary contributions of this work are:

- The invention of the first Advanced Driver Assistance System modelled on the concept of an Automated Co-driver. The developed Advanced Driver Assistance System demonstrated the value of road-scene analysis combined with driver eye-gaze monitoring to directly address driver inattention. The system was shown to be capable of detecting events "missed" by the driver, highlighting context-relevant information, suppressing redundant warnings and acting with a human-machine interface natural to the driving task. This ability is achieved by a new concept in Advanced Driver Assistance Systems, *Driver observation monitoring*. Figure 1.2 illustrates the concept of driver observation monitoring. Instead of simply sensing the actions of the driver through the vehicle, driver eye-gaze tracking and critical road scene feature detection are combined to determine what the driver is seeing

[1] In this work we define "visual monotony" as visual environments consistent with roads which are periodic, monotonous, or uneventful in accord with the well known "Highway hypnosis" phenomenon of Williams (1963).

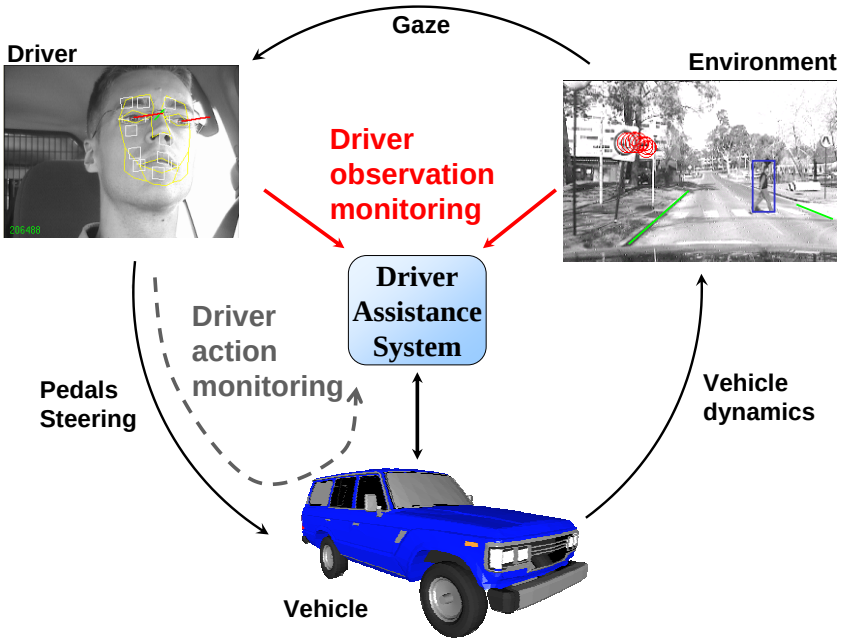


Figure 1.2: Driver observation monitoring: Driver eye-gaze tracking and critical road scene feature detection are combined to determine what the driver is seeing (or not seeing).

(or not seeing). Determining what the driver has definitely seen is impossible, however our experiments demonstrate the value in knowing what the driver definitely hasn't seen.

- The development of a general tracking algorithm capable of supporting ambiguities and combining multiple visual cues for robust tracking. The algorithm could dynamically control visual cue execution rates and incorporate "late" sensor data to tune visual cue selection to suit the current conditions. The system was applied to robust lane tracking. The lane tracking system was shown to accurately estimate road curvature, include supplementary camera views and adapt to uncertain conditions.

- The development of a real-time obstacle detection and tracking system also based on the developed visual-cue fusion architecture. The system is shown to use "bottom up" detection and "top down" hypothesis verification to attain and track road obstacles using multiple visual cues and hypotheses.

- The development of a fast and effective real-time symbolic sign reading system, including online road sign image enhancement capable of reliably recognising the sign at a greater distance from the same image sequence

data.

- The development of a real-time metrics of visual monotony and road scene clutter. The visual monotony metric was shown to consistently estimate the road scene variability over time relevant to the driver. This visual monotony metric was used as a determinant for fatigue detection and could be used by road makers to assess the fatigue risk of roads. The road scene clutter metric could be used in a driver workload manager.

- The development of an extensive hardware and software framework for Advanced Driver Assistance Systems. The framework permitted the modular construction of Advanced Driver Assistance Systems applications across a distributed computer architecture.

1.3 Thesis outline

The thesis begins with an analysis of the next potential steps in road safety in light of previous research and current technological and road-safety research trends. Next, a description of computer vision algorithms developed for road scene monitoring is given, beginning with robust lane tracking. Principal components of the algorithm are then used for robust road obstacle detection. We then present an effective method for detecting and interpreting road signs. This is followed by an investigation of methods for evaluating the road appearance to aid in drive-safety issues such as fatigue and distraction. Finally, we integrate the developed components into a demonstration Advanced Driver Assistance System based on the Automated Co-driver concept. The thesis concludes with a discussion of an Automated Co-driver as the next step in road safety in the light of the technologies developed in this thesis.

Next, we present a synopsis of each chapter.

1.3.1 The next step in road safety

Chapter 2 provides an analysis of the next steps in road safety in the light human behaviour while driving, emerging mobile technologies, computer vision and previous intelligent transport-systems research. The chapter concludes with set

of requirements for an Advanced Driver Assistance System designed to emulate an Automated Co-driver.

1.3.2 Lane tracking

A key challenge for computer vision is that although many effective techniques exist to find and track objects in video footage, these techniques do not operate across the highly variable circumstances encountered in most real-world environments. Computer vision algorithms require an adaptive framework which is capable of switching to another technique when a particular approach fails. To combat this issue, the "Distillation algorithm" was developed. Chapter 3 describes the Distillation algorithm. The algorithm is a tracking algorithm designed to track objects in a noisy environment using multiple visual cues within a visual-ambiguity tolerant framework. The developed algorithm was demonstrated first for people tracking, then applied to lane tracking. The initial version of the lane tracking system was extended to estimate road curvature and vehicle pitch, multiple views and variable lookahead based on lane estimate confidence. The lane tracking system was tested on 1800km of roads. We can use lane tracking to determine the orientation of the vehicle. A key skill of an Automated Co-driver is to warn of unexpected lane departure. The developed tracking algorithm found another application as part of our obstacle detection and tracking system.

1.3.3 Obstacle detection

In chapter 4 we develop an algorithm to detect and track obstacles in the road scene. Unlike in lane tracking, in obstacle detection an unknown number of objects may be present in the road scene and visual ambiguities are common. In this case we use image analysis to explicitly segment obstacle candidates. The segmented candidates are injected into the Distillation algorithm. The Distillation algorithm then condenses obstacle candidates into tracked targets. With obstacles and lanes identified, one further essential cue used by the driver is road signs. Next we develop a fast and effective technique to read road signs.

1.3.4 Road sign recognition

In Chapter 5 we describe a technique to detect and interpret road signs. We focus on speed signs, but our work is also readily extendable to other symbolic signs and signals. Road signs and signals with a closely regulated geometry like Stop, Give-way and Round-about signs as well as traffic lights are all candidates for this technique. Under the common constraint of low resolution imagery, an on-line image enhancement was developed to improve the sign classification performance.

Although we have developed methods to find the major components of interest in the road scene, there is one final aspect that needs consideration. A key role of an Automated Co-driver would be to assess the fitness of the driver. Driver fitness is heavily dependent on the driving environment over time.

1.3.5 Road scene complexity assessment

To complete the capabilities of an Automated Co-driver, in Chapter 6 we will examine the road ahead as a whole to derive some useful measures of visual stimulation. These metrics can feed in to driver behaviour diagnostic models to enhance tools such as driver fatigue detectors and driver information load managers.

The previous four chapters detail how we have developed a detailed model of the road environment. Finally, in the following chapter, we are ready to integrate the road scene assessment with the observed driver behaviour to implement an Advanced Driver Assistance System based on the Automated Co-driver concept.

1.3.6 Automated Co-driver experiments

In Chapter 7 we prototype a number of Advanced Driver Assistance Systems. The systems are developed to investigate the viability the Automated Co-driver concept for road safety. The systems integrate driver-gaze monitoring with road-scene analysis to achieve driver observation monitoring, a crucial skill of any co-pilot. A speed sign assistance co-pilot combining vehicle monitoring, speed-sign detection and driver gaze monitoring is demonstrated. Then finally, we use an integration of vehicle monitoring, road object detection, lane tracking, sign recognition, pedestrian detection and visual monotony detection along with

driver monitoring to demonstrate an integrated, comprehensive, context-sensitive Advanced Driver Assistance System approximating the full functionality of an Automated Co-driver.

We conclude by returning to the question of whether an Automated Co-driver could be the next step in road safety.

1.3.7 Conclusion

Chapter 8 brings together the insights and conclusions that can be drawn from the research we have conducted. The developed systems provide a useful demonstration of the potential of intelligent vehicle technologies not only to enhance comfort, but crucially, to take a substantial step forward in road safety by finally addressing the key component in road safety: The driver. We conclude with suggestions for the next further steps to advance the Automated Co-driver concept.

1.3.8 Appendices

Appendix A provides a detailed description of the experimental vehicle and hardware used in all of the systems. Just as an experimental vehicle is a necessity, an extensive software infrastructure is needed to give structure to the highly complex, multi-sensor, multi-purpose systems. The chapter lays out the distributed modular software infrastructure that was developed. The software provides the flexibility to combine the various systems of systems required across a network of computers.

Appendix B contains multimedia content to illustrate developed components in action. The Appendix also contains source code from our software infrastructure to demonstrate how the components were implemented.

Chapter 2

The next step in road safety

Law enforcement, safer vehicle design, road layout and public awareness campaigns have had a marked effect on accident rates from the 1970s (ATSB, 2004a; OECD, 2006). However, some contributing factors to road fatalities, namely fatigue, distraction and inattention remain practically unaddressed by current initiatives (Stutts *et al.*, 2001; Neale *et al.*, 2005; Treat *et al.*, 1979). What should be the next step in road safety?

We begin the chapter by looking at positive options to answer this question. In Section 2.1 we generate a set of alternate future directions for road safety. In Section 2.2 we conduct a brief analysis of the problem of death and injury on the road. Then in Section 2.3 we review past intelligent vehicle research. By Section 2.4 we are ready to we evaluate the alternative steps in road safety and we state the case for an Automated Co-driver. In Section 2.5 we examine the role of a vigilant human passenger to define a set of the core capabilities for an Advanced Driver Assistance System to implement an Automated Co-driver. The chapter concludes with a discussion on how to implement the Advanced Driver Assistance System, identifying the outstanding research questions and a set of design requirements for a potential system.

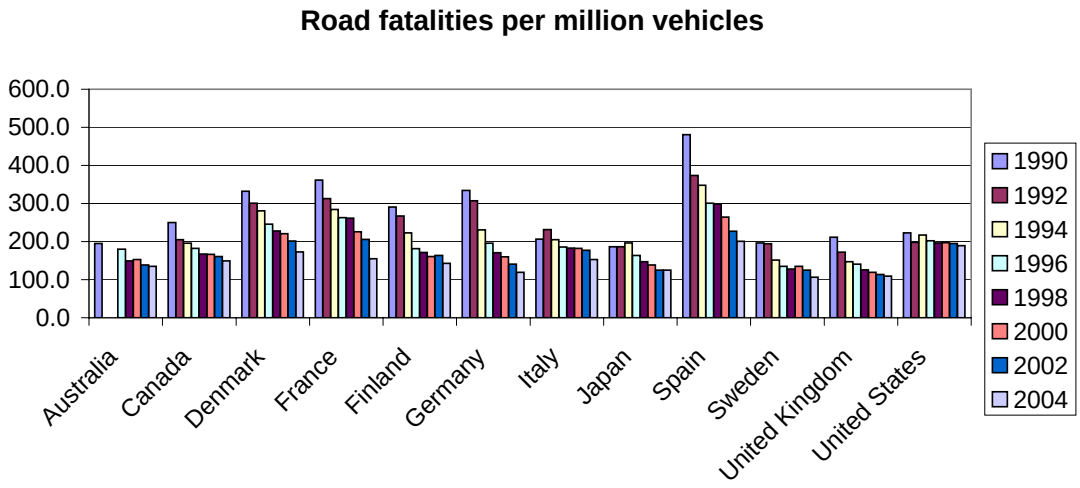


Figure 2.1: Road toll from 1990 to 2004 per million vehicles across selected OECD countries. Data from (OECD, 2006).

2.1 The problem

There has been a substantial reduction in road fatalities in many OECD countries over the past 15 years (OECD/ECMT, 2006). The trend is shown in Figure 2.1.

The reduction can be attributed to a number of successful road safety initiatives (OECD/ECMT, 2006):

- Improvements to roads, such as divided highways, road bypasses, road shoulder sealing, use of rumble strips and removal of roadside hazards.
- Minimum standards for vehicles, including: child restraints, head restraints, airbags, and crumple zones and roll-over strength.
- High visibility law enforcement with speed and red light cameras as well as random breath [alcohol] testing.
- Road safety public education campaigns for drink driving and speeding.

However, Figure 2.1 also shows that while many countries starting from a high base like France and Spain have steady declines, countries with the lowest fatality rates, Sweden and the UK, no longer have a constant decline. Instead, fatality rates in these countries appear to be leveling off (OECD/ECMT, 2006). In fact most OECD countries are failing to progress toward the OECD aim of a 50%

Country	Fatalities in 2000	Fatalities in 2004	Average annual reduction (or increase) achieved in 2000-04	Average annual reduction required during 2005-2012 to reduce fatalities by -50% by 2012
Australia	1824	1590	-3.4%	-6.7%
Canada	2927	2730	-1.7%	-7.5%
Japan	10 403	8 492	-4.9%	-5.9%
Korea	10 236	6 563	-10.5%	-3.1%
Mexico				
New Zealand	462	436	-1.4%	-7.6%
United States	41 945	42 636	0.4%	-8.5%

Table 2.1: Progress of non-European OECD countries toward a 50% reduction in fatalities from 2000 by 2012. From (OECD/ECMT, 2006).

reduction in fatalities from 2000 by 2012(OECD/ECMT, 2006). The trend is world wide and highly correlated with the penetration of current road safety initiatives. Table 2.1 shows the progress of non-European OECD countries. Korea has made leaps and bounds as the road and vehicle standards approach best practice. Note that even Japan, a world leader in driver assistance systems due to open vehicle technology regulations still fails to hit the mark.

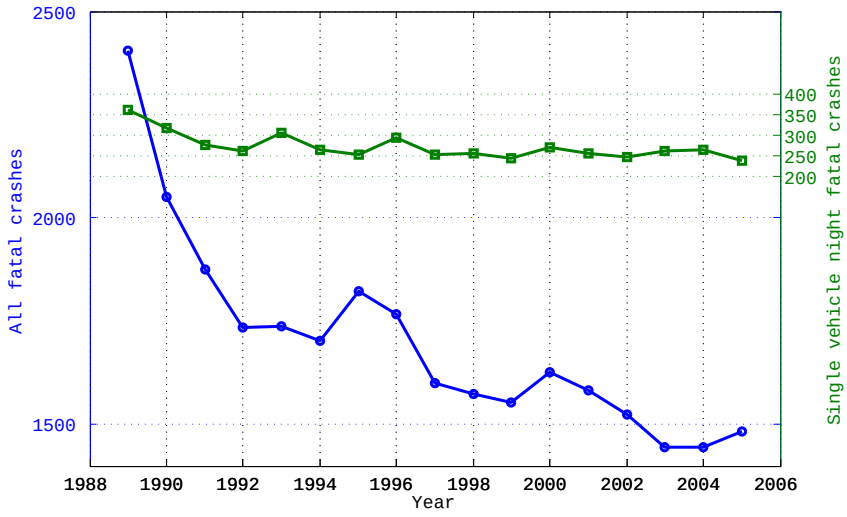


Figure 2.2: The fatigue toll unfortunately doesn't mirror the trend of road accidents in general. Data from (ATSB, 2006a).

We believe that, as road fatalities from speeding and drink driving drop down, the difficult cases in road safety of fatigue, distraction and inattention have become more prominent. To gauge this trend we extracted data from the Australian

Road Crash Database (ATSB, 2006a). The data extracted was the number of fatal single vehicle crashes during the night (22:00-7:00) compared with the total number of fatal crashes. The data is plotted in Figure 2.2. The plot shows that there is no significant downward trend in [likely] fatigue related crashes despite a clear downward trend in fatal crashes per se. Clearly a new approach is required.

We will now review the challenges in road safety in more detail and define and evaluate potential next steps in road safety.

2.2 Review of road safety

The top contributing factors in road accidents are speed, alcohol and drug abuse, fatigue, distraction (ATSB, 2004a).

2.2.1 Speeding

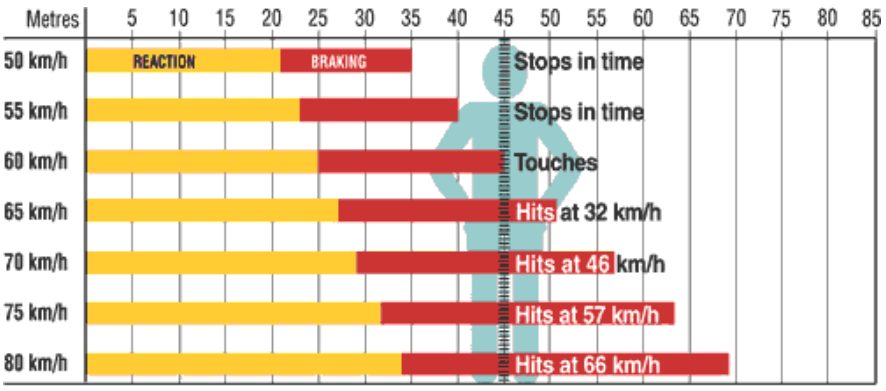


Figure 2.3: Typical stopping distance for a modern vehicle with good brakes and tyres on a dry road. From (ATSB, 2006b)

Accident statistics show that the risk of a crash doubles when travelling just 5km/h above the designated speed limit (ATSB, 2006b). Figure 2.3 plots the stopping distance in a modern car in fine weather. In suboptimal conditions at least 5-10 metres is required. Any intervention that can reduce speeding even by an average of 5km/h will have a large benefit for road safety.

Often when entering urban regions or travelling on unfamiliar roads, speed changes can be missed. These cases are particularly dangerous because road vehicles can be travelling 20km/h or more over the speed limit, substantially increasing the

crash risk (Draskczy and Mocsri, 1997; Oesch, 2005). In-vehicle systems capable of monitoring the vehicle speed and knowing the current speed-limit have a large potential to reduce road fatalities by notifying the driver about a missed speed change or by reducing the speed in excess of the speed-limit. A good example system was developed by Carsten and Tate (2001). This group used force feedback through the accelerator pedal for speed control. As the driver exceeds the speed-limit the resistive force of the pedal increases. The driver can maintain the correct speed just 'by feel' through the pedal. If necessary, the driver can override the resistive force by pushing harder to accelerate to safety. The group found that drivers tend to maintain speeds close to the limit because it is most comfortable to drive that way. This system is a good example of the kind of integrated, context relevant driver assistance we intend to implement with our Advanced Driver Assistance System.

Speeding per se does not cause accidents, accidents are caused by the insufficient reaction time available when driving in a road environment designed for a slower speed. The speed-limitless sections of German highways (Autobahns) have been shown to be no more dangerous than highways of other countries or other sections of the highway were speed-limits are set (Yowell, 2005). Great care is taken to engineer these roads with gentle curvature and sufficient look-ahead distances to provide safe reaction times even at these high speeds. The harm in speeding comes from an inability to react in time, that is, from inattention (Yowell, 2005). An Automated Co-driver able to intervene to alert the driver sooner when speeding or to provide earlier warning to road events has the potential to reduce speed-related road fatalities.

2.2.2 Alcohol, drugs and impairment

Accident statistics show that even at a blood alcohol concentration (BAC) equal to the legal limit of 0.05 doubles a driver's crash risk (Howat *et al.*, 1991).

Amphetamines and caffeine are often used by long-haul drivers to reduce the effects of fatigue, though other mental processes have been shown to be are compromised by them. Caffeine, often promoted to improve the arousal of fatigued drivers, has been shown to substantially impair his or her ability to execute procedural tasks and to temper impulsive decisions (Balkin, 2004). Neither impairments are good news for road safety. Even less is known about the effects of

illicit drugs on driving performance (Chesher, 1995).

Even with in-built breath alcohol testing in vehicles there is no shortage of other causes, even illness or anxiety, that can afflict the driver. This means that there is no way to ensure the competence of a driver behind the wheel.

Again, in-vehicle systems have the potential to improve road safety by providing earlier warning of critical traffic events providing a longer time for the reaction of an impaired driver. An in-vehicle system could also use driver monitoring and eye-gaze scan pattern statistics to detect impairment (Victor, 2005). Our system could leverage off these results to estimate the driver's state and intervene if impairment is detected. Our concept of an Automated Co-driver is 100% compatible with this kind of metric. A final system could easily incorporate this functionality, however since interventions based on this metric are already under active development (Victor, 2005) we will not make research contributions using this metric. Instead we will go further and attempt real-time road event inattention detection.

2.2.3 Fatigue

When considering the potential of driver safety systems it is hard to go past fatigue. The most horrific road accidents often involve driver fatigue (Haworth *et al.*, 1988).

Fatigue at the wheel often results in accidents (Haworth *et al.*, 1988):

- at high speed
- involve stationary objects or oncoming traffic
- without braking or other crash preparedness (e.g. swerving, posture correction)

The frequency of fatigue accidents is also alarming, though thought to be under reported, at least 30% of all accidents involve driver fatigue (Howarth *et al.*, 1989). A particularly high risk group is long haul semi-trailer trucks. One out of two hundred trucks on Australian roads are involved in a fatal accident each year (Howarth *et al.*, 1989).

Fatigue is often misdiagnosed or undetectable even to experienced drivers. Short periods of the first stage of sleep, often termed "micro-" or "nano-sleeps", lasting up to several seconds, have been demonstrated to occur without the conscious recollection by the driver (Torsvall and aAkerstedt, 1987).

Fatigue can afflict the driver at the start as well as into a journey. For example, more fatigue related crashes occur the morning after a working week (Haworth *et al.*, 1988). Although the driver could well have had a sound night of sleep, the sleep debt accumulated from the previous busy week may not have been eliminated (Balkin *et al.*, 2000). For professional drivers regulated hours of service recorded in log books have been the most effective fatigue management strategy to date. The under reporting of driving hours in log entries remains rife within the industry (Braver *et al.*, 1992).

(Haworth *et al.*, 1988) found continuous in-vehicle monitoring provided the only reliable method for detecting fatigue. A system capable of intervening to detecting fatigue by direct driver observation has the potential to reduce fatalities. (Wierwille and Ellsworth, 1994) developed the Percentage Eye Closure (PERCLOS) driver eye-blink monitoring metric. The metric computes the percentage of time the driver has his or her eye closed over a moving time window. The metric has been found to be highly correlated with fatigue-like deteriorations in driving performance. Computer vision systems to automatically calculate this metric are in active development (Seeing Machines, 2001). This kind of information about the driver is critical in mimicking the ability of a vigilant passenger to assess the competence of the driver. However, conducting experiments where drivers exhibit fatigue behaviour is beyond the scope of this project. Our research will concentrate on the inattention aspect of fatigue as a contributing factor to road fatalities. The detection of fatigue in the driver's face is again an active area of ongoing research and our Automated Co-driver is designed to be compatible with this emerging system. One aspect we will investigate is road environment assessment contributing to driver fatigue. Desmond and Matthews (1997) and Nilsson *et al.* (1997) others have noted disappointing performance of automated fatigue detectors due to false positives caused by the absence of road environmental cues. Cause to speculate as to the driver's alertness is likely to occur to a vigilant passenger from a perception of boredom or monotony regarding the road travelled. Thiffault and Bergeron (2003) confirmed that this is a valid intuition. Driving in monotonous road environments, even for short periods (significantly less than 20 minutes) can induce fatigue. To achieve our aim of mimicking the

human vigilant passenger we are challenged to build-in this kind of intuition into our Advanced Driver Assistance System.

Just like speeding and alcohol, fatigue is lethal because it leads to the inability of the driver to react to critical road events. An in-vehicle system capable of providing interventions regarding missed events has the potential to provide the driver with the critical seconds needed to recover and prevent an accident and stop safely.

2.2.4 Distraction

The Information and Communications Technology (ICT) revolution has brought waves of additional information to the driver. However, together with the potential for making road vehicles safer is the danger that mobile devices will distract the driver. There is no doubt that distractions caused by recent in-vehicle devices such as GPS maps, entertainment systems and mobile telephones increase a driver's crash risk (Stutts *et al.*, 2001). Talking on a mobile phone, for example, is thought to increase crash risk up to four times, and hands-free usage is no safer (Redelmeiser and Tibshirani, 1997).

One quarter of all accidents in the United States are estimated to be due directly to distraction (Stutts *et al.*, 2001). Distraction is not only the result of the use of new technologies however, Stutts *et al.* (2001) found that 30% of accidents involved distractions from outside the vehicle, with a further 11% due to a distraction caused by another vehicle occupant. Surveys and in-vehicle monitoring indicate that leading contributing distractions in accidents are: distractions outside the car, other driving tasks, adjusting console (radio, air conditioning), passengers/children in the vehicle, eating/drinking/smoking, and mobile phone use (Stutts *et al.*, 2001).

Victor (2005) used automated driver eye-gaze tracking to demonstrate that driver eye-gaze fixation patterns narrow with secondary cognitive demands on the driver (see Figure2.4). This finding indicates that the driver eye-gaze direction is likely be effective for estimating the cognitive state of the driver. This group defined a metric on driver eye-gaze patterns named Percentage Road Centre (PRC). As the name suggests, the metric estimates the percentage of time the driver is looking at the centre of the road scene. Overly low percentages represent that the driver

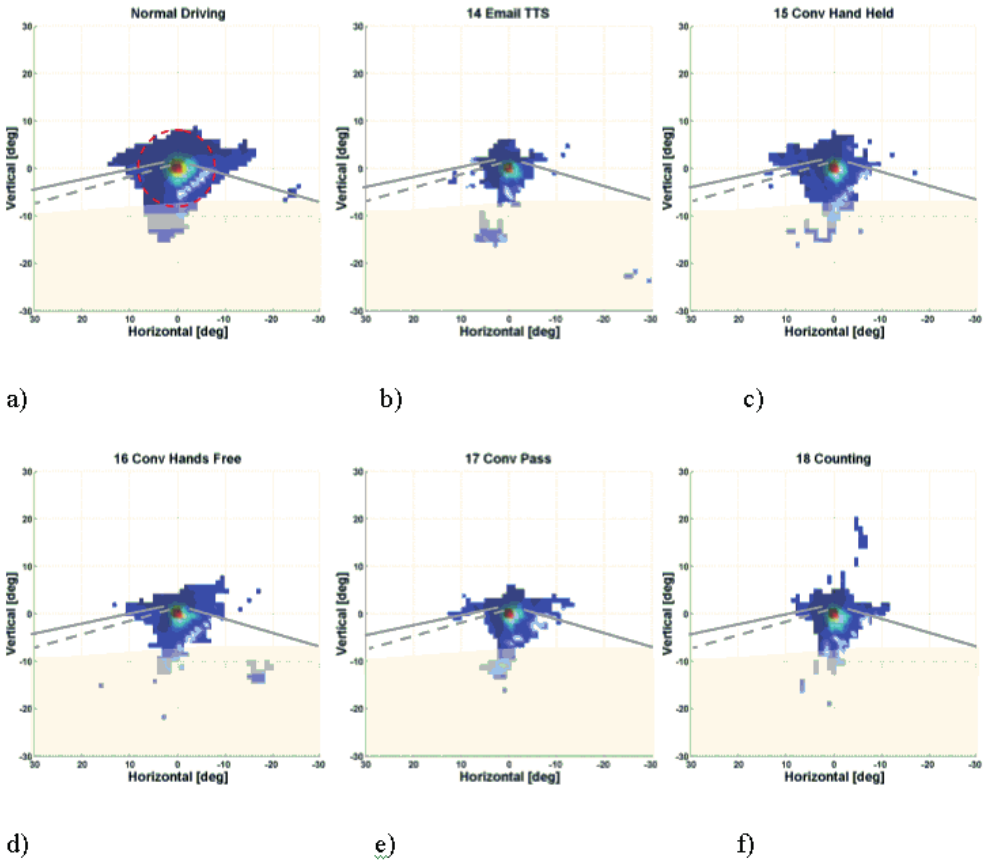


Figure 2.4: Histogram of driver eye-gaze direction. Gaze fixation narrows under distraction from secondary tasks. **a:** Normal driving. **b:** Listening to synthesised speech. **c:** Hand held phone answering questions. **d:** Hands free phone answering questions. **e:** Experimenter as passenger, driver answering questions. **f:** Counting task. Reproduced from (Victor, 2005)

is visually distracted (looking somewhere else too much for safe driving), while overly high percentages also represent a dangerous driving scenario as the driver's eye-gaze fixation patterns indicate a cognitive distraction Victor (2005) causing the driver to simply stare at the centre of the road. This metric is another useful indicator of the driver's state that could be incorporated into the "intuitions" of an Automated Co-driver.

Holding a conversation on a telephone has been found to be significantly more distracting than holding a conversation with a passenger. Passengers tend to pause or simplify conversations during more challenging driving conditions. It is thought that this attribute is the significant difference between the disruptive effect of telephone conversations over passenger conversations (Young *et al.*, 2003). Again this intuition of a human passenger is just the kind of behaviour we would

like to mimic in our Automated Co-driver. This concept is related to an emerging research area known as driver workload management (Green, 2000). The aim of driver workload management is to postpone secondary tasks when the driving task is judged to be too demanding. This functionality is a natural fit with the Automated Co-driver concept.

2.2.5 Inattention

As stated in Chapter 1, inattention is not necessarily the result of a distraction. Inattention can simply be the result of misdirected attention or a too demanding driving environment such that there are too many driving critical events to attend to (such as when speeding). Treat *et al.* (1979) concluded that over 92% of road accidents were due to driver error, most could be classified as inattention. Recently, Neale *et al.* (2005) used 100 vehicles equipped with video and sensor logging equipment in a comprehensive study of road safety. They found that 78% of accidents and 67% of near accidents involved momentary inattention (within 3 seconds) before the incident.

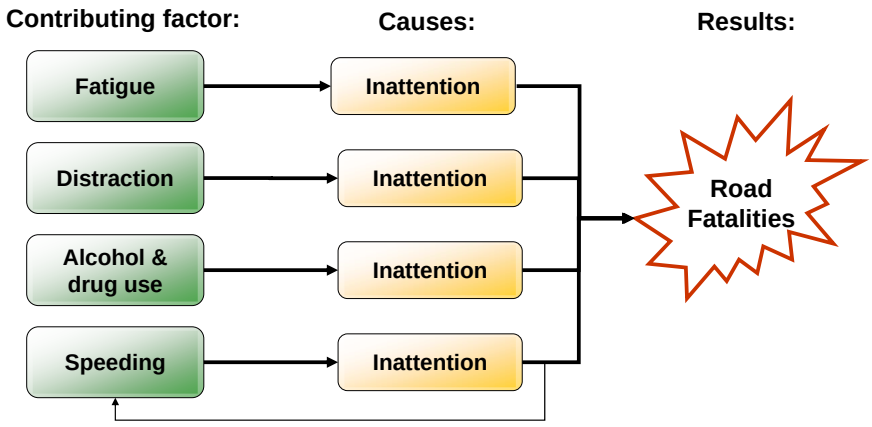


Figure 2.5: Accident contributing factors to lead to inattention which leads to road fatalities.

It is our developing argument that the primary contributing factors to road fatalities: speed, drink driving, fatigue and distractions, are actually causes of inattention. The contributing factors all impair the driver's ability to react in time to critical road events. Figure 2.5 illustrates this point. The primary causes of accidents induce moments of driver critical inattention the consequences of which can lead to road fatalities.

By concentrating our efforts on interventions at the inattention phase we can break these critical event chains permitting a modest correction before the incident is out of control. Interventions at the inattention phase have potential to substantially reduce of road fatalities across many road scenarios: intersection crashes, passing, merging, road departure, pedestrian collisions, the list goes on.

2.3 Review of intelligent vehicle research

Roads offer an excellent compromise between the rigid constraints available in the laboratory and the unbridled complexity of the outside world. Roads present a structured world for robotic systems, less structured than the laboratory but better defined than the home.

We will first look at research efforts in autonomous vehicles before looking at autonomous technologies applied to driver assistance.

2.3.1 Autonomous vehicles

In the mid 1980s, as soon as the computer technology became readily accessible, control and robotics researchers began experimenting with the automation of road vehicles.

The first era of intelligent vehicles research was dominated by three groups: Universität der Bundeswehr Münich (UBM), Carnegie Mellon University (CMU) and Universita di Parma (UParma). There were, of course, many other groups which conducted important research into intelligent vehicles over this period (Kenue, 1989; Krüger *et al.*, 1995). (Bertozzi *et al.*, 2002) and (Sun *et al.*, 2006) contain good reviews of this era. These three groups all conducted impressive autonomous drives of at least 2000 kilometres each, so they have important lessons to teach about successful automated systems in cars. The groups also took quite different approaches to achieving autonomous driving, which is also of interest.

Finally, we will look at the renewed efforts in autonomous driving fostered by the DARPA Grand Challenges.

Universität der Bundeswehr Münich (UBM): VaMoRs, VAMP

The pioneers of automated vehicle research were the Universität der Bundeswehr Münich (UBM), Germany. For the past 20 years this group has investigated lane keeping, vehicle following, vehicle avoidance and automatic overtaking in highway environments. More recently the group has investigated automated intersection negotiation and off-road negative obstacle (ditch) avoidance. The cornerstone of the group's success was it's "4D approach" which uses physical-world based modelling with optimal filtering to track the vehicle and road state over (3 dimensional) space and time (+1 dimension). The group managed to demonstrate visual lane keeping on a very modest transputer system in 1987(Dickmanns and Graefe, 1988b,a). The group has undergone three iterations of hardware development but has stuck to it's "4D" philosophy, now using a four-camera system on an active platform with PC-based processing technology (Behringer and Müller, 1998). The final system developed by the group, the "MarVEye" Multi-focal, active/reactive Vehicle Eye system, consisted of an active camera platform to stabilise the video images against vibrations of the vehicle. The application software named the "Expectation-based Multi-focal Saccadic (EMS)" vision system combined optimal filtering with cameras of overlapping fields of view but different focal lengths. The multiple cameras were required to attain a sufficient look-ahead range and resolution to allow for horizontal road curvature estimation on German highways(Autobahns) while travelling at high speed. For safe operation the look-ahead distance was required to be in excess of 150 metres (Gregor *et al.*, 2002).

In 1995 the UBM team demonstrated the VaMP prototype vehicle driving from Munich to Odense. During the 1600 kilometre journey the vehicle performed lane keeping, speed control, collision avoidance as well as driver initiated overtaking manoeuvres. The automated systems were estimated to be in control 95% of the time with over 400 overtaking manoeuvres conducted. The primary failure points were reported to be glare from sunlight and the false detections or missed obstacles due to failure of the simplifying assumptions built into the object detection system (Dickmanns, 1999a).

The UBM group championed physical-world based models. Their systems were designed with sufficient states to adequately describe the true 3D geometry of the scene (Dickmanns, 1999a; Gregor *et al.*, 2002). Lanes, obstacles, the vehicle state, Sun position, even the fuel level was represented in carefully constructed

real world models (Dickmanns and Graefe, 1988b,a). The group argued that it was ill-posed to attempt to model or explain the behaviour of 3D objects moving over time using 2D representations in the image. A simple linear translation or angular motion in 4D (3D space plus 1D time) can have a highly non-linear path when projected into 2D image space. What is gained initially by avoiding the effort of making the physically grounded models is quickly lost by the challenge of dealing with unpredictable higher-order image motion. Modelling the higher order system in the image space also leads to nonsensical degrees of freedom with variables attempting to explain the projection of simple but hidden states of the true physical system (Dickmanns, 1999a).

The counter argument levelled at the UBM group was that strict model-based techniques lose their flexibility. Any case not considered during the modelling and noise estimation phase can potentially derail the system. For example, the same model-based approach that prevents the lane tracking from being susceptible to shadows on the road may also prevent the system from adapting to non-standard lane geometries made by temporary road works.

We support the physically grounded model argument, but in our work use recent tracking approaches (specifically particle filtering discussed in Chapter 3) to permit a much higher tolerance to unmodelled system dynamics and disturbances (Isard and Blake, 1998).

The following group is a good example of the counter case to the UBM approach.

Carnegie Mellon University (CMU): NavLab

Research on intelligent vehicles started in earnest at Carnegie Mellon University (CMU) Robotics Institute when the US Defence Advanced Research Projects Authority (DARPA) sponsored the Autonomous Land Vehicle project and the birth of CMU's Navlab (Thorpe, 1990a). A fleet of experimental vehicles, Navlab 1 (1986) through to Navlab 11 (2004), was developed over the next 20 years (Batavia *et al.*, 1998). The Navlab group's work diverged from UBM's approach early on as the Navlab group concentrated on machine learning based techniques such as neural networks to follow the road. The first road following algorithms: Supervised Classification Applied to Road Following (SCARF) and the UNSupervised Classification Applied to Road Following (UNSCARF) used learned colour models to segment road (Crisman and Thorpe, 1993b). The developed systems were

very effective at following roads yet made no explicit assumptions about the road appearance such as a search for lane markings. During the late 1980s the Autonomous Land Vehicle In Neural Network (ALVINN) road follower was developed (Baluja, 1996). This road follower attempted to learn a mapping between a low resolution image of the road and a steering actuator. This remarkably simple system was able to steer the vehicle. The system could, with retraining, follow a multi-lane highway or a dirt track (Baluja, 1996). In 1995 the next generation system named "RALPH" (the Rapidly Adapting Lateral Position Handler (Pomerleau, 1995)) successfully completed a test drive "No hands across America", from Pittsburgh, Pennsylvania to San Diego, California. An over 4500 kilometre trip. During the drive the RALPH system steered the vehicle with obstacle detection supplemented by a radar-based system. In the RALPH system the original camera view was projected on to a virtual "aerial view" of the road ahead. This view was then "straightened". Starting from the bottom of the virtual aerial view image (which is nearest to the vehicle) each scan-line was correlated with the line above, all the way up the image. The set of required shifts between each scan-line was then used to estimate the lane direction and required steering angle. The system was reported to be active 98% of the time during the road trial, failure points were noted to be due to: rain obscuring vision, sun reflections, strong shadows, featureless roads, degraded roads and road works (Pomerleau and Jochem T., 1996).

The ingenuity of this group is impressive. Many difficult phases of road following problem were simply sidestepped by their approaches. The approach of this group meant that lane tracking was possible without explicitly detecting lane markings. The system could instead use any feature parallel to the lane direction to follow the lane. However, the point raised by the UBM group was still valid, the benefits of the image based techniques come with limitations. Strong shadows on the road, close to parallel to the lane direction, caused significant problems for the system. Crucially, there was no obvious way to overcome these problems other than the redesign of the entire system or the introduction of some hand made exception processing for this and any similar case encountered. Due to these final points this approach does not support the generic tracking approach envisaged for our work. Though the results of the CMU group are impressive it is hard to build on the work without replicating a similar system with the same limitations. e.g. Trying to implement only the vision component of the ALVINN system (without the neural network) or the RALPH system (without the "aerial view" projection).

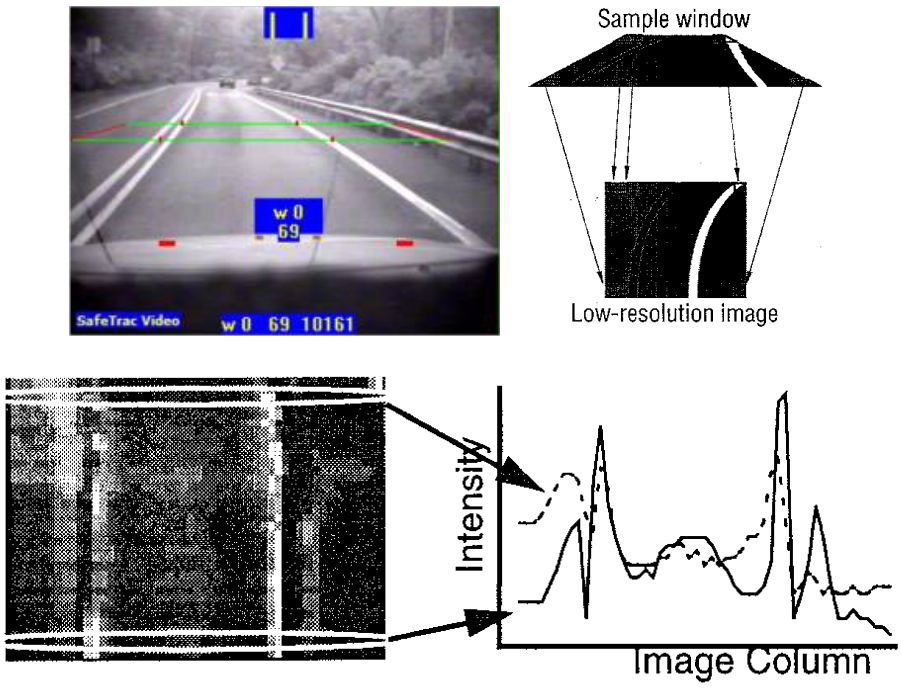


Figure 2.6: CMU's RALPH lane tracking technique. **(a):** The system works with a trapezoidal region of interest in the original camera view. **(b):** The region of interest is transformed into a virtual "aerial view". **(c):** Each scan-line of the aerial view is compared and "straightened" to determine the road geometry. From (Pomerleau and Jochem T., 1996).

The approach of the third group falls in-between the UBM and the CMU groups' solutions.

Universita di Parma: GOLD

Another highly successful group is the Universita di Parma, Italy's "ARGO" vehicle project. The group developed the "GOLD" Generic Obstacle and Lane Detection system. The group used edge detection to locate and track lane markings, while a wide baseline stereo vision system was used to detect obstacles (Bertozzi and Broggi, 1998). The camera view, similar to CMU's algorithms, was mapped into a virtual "aerial view" for lane tracking and obstacle detection (Bertozzi and Broggi, 1998). This group was also able to demonstrate their system on a road trial. The vehicle completed a 2000 kilometre circuit of northern Italy. During the test drive the system steered the car and identified potential obstacles. The system also completed manually initiated lane changes (Bertozzi *et al.*, 2000).

The group estimate the vehicle drove autonomously for over 94% of the journey. Failure points were often due to worn lane markings, road works, strong sun reflections and dramatic changes in illumination such as the start and end of tunnels (Bertozzi *et al.*, 2002).

The group developed a procedure for extracting lane markings using image kernels and adaptive image thresholding from the virtual aerial view images. Though we do not intend on using the aerial view projection, we will use a similar approach to detect lane markings as one of several cues for lane tracking.

Stereo matching for obstacle detection was also done in the virtual aerial view images. An accurate calibration was crucial to the reconciliation of the two views, otherwise artifacts would appear in the segmentation. The group developed an online self calibration method to maintain an accurate calibration under the effects of vehicle vibration (Broggi *et al.*, 2001). This wide baseline stereo system is similar to (Williamson and Thorpe, 1999)'s approach. We will reserve comment to Section 2.3.2 when we discuss (Williamson and Thorpe, 1999)'s work.

DARPA Grand Challenge

Renewed interest in fully autonomous vehicles has developed recently when the US Defence Advanced Research Projects Agency (DARPA) elected to hold the DARPA Grand Challenge in 2004(DARPA, 2004). The DARPA Grand Challenge is a race to complete a specified course with fully automated road vehicles. The fact that DARPA began the competitions testifies to their assessment that the related research community was up to the challenge. Regardless of the outcome of the competition, DARPA has already succeeded in porting a substantial amount of research (and researchers) from the mobile robotics and robotic vision community into intelligent vehicle research (Thrun *et al.*, 2006). In this way they are already achieving their aim of kick starting the development of autonomous vehicle technologies required to achieve the Congressional mandate to automate one third of military vehicles by 2015 (DARPA, 2004).

The first Grand Challenge held in 2004 was an on-road, off-road contest to drive an autonomous vehicle around a course specified in GPS coordinates. In the first attempt all teams failed within 7 kilometres from the starting line (DARPA, 2004). Many lessons were learned and a very similar challenge was held a year later. The result was quite different. In 2005, five autonomous vehicles managed

to finish the 200 kilometre course (DARPA, 2005). The scenario of the first challenge represented the classic mobile robotics problem of traversing unknown terrain between specified way points. The challenge was mainly an obstacle field traversal problem with minimal requirements for local perception to complete the race. Thrun *et al.* (2006) noted that the DARPA Grand Challenge is very much a software engineering challenge. The Stanford group put its 2005 victory down to adaptive speed control. The speed control system adjusted the maximum speed along sections of the course based on vibrations detected using the inertial management unit. Too large high frequency vibrations indicated that the vehicle was going too fast on uneven ground. The speed adaptive system made the vehicle slow down in these circumstances and navigate safely through the rougher terrain (Thrun *et al.*, 2006). The 2007 Grand Challenge was a contest in an urban scenario. In this challenge for the first time the automated vehicles were required to obey traffic laws including urban lane keeping, obstacle avoidance, merging, passing, and parking with other traffic. This contest was the first attempt to make a significant advance past the first era of autonomous vehicles (DARPA, 2007). DARPA Grand Challenge contestants were heavy users of lidar sensors. Lidar sensors became very popular in mobile robotics in the 1990s due to the superior quality of the sensor data compared with infrared or sonar data (Thrun *et al.*, 2000). The lidar sensors that are commonly available, the same used on "Stanley" the winning vehicle, only have a useful range of 40 metres (SICK AG, 2003). This limited range, only a second if travelling at 100km/h, limits their use for driver assistance systems. While capable of winning the race, the "Stanley" vehicle is not safe to operate around pedestrians due to this limited sensing range. The perception for road vehicles is very much an open problem for safe operation in populated environments. Again as the Stanford group acknowledged the course during the 2005 DARPA Grand Challenge while challenging to autonomous road vehicles, would not have challenged to an experienced human driver (Thrun *et al.*, 2006). Similarly, the 2007 Urban Challenge while much more realistic to an urban driving environment still lacked much of the complexity faced by most road commuters (Leonard *et al.*, 2008). So while autonomous vehicles have and will progress in capability, an autonomous vehicle based solution to road fatalities is still a distant future hope.

2.3.2 Autonomous technologies for driver support

After the success of the 1990s autonomous vehicles, interest in full autonomy waned. In the wake of the dramatic achievements of these groups further research on incremental improvements to autonomous control lacked lustre. These original autonomous vehicle successes were a decade ahead of their time. Only now, though the DARPA Grand Challenge, are these achievements beginning to be surpassed. Suitable sensors and processing power to implement these systems in production vehicles has also been prohibitively expensive. Video sensors capable of the extreme dynamic range required to work in all road environments (estimated to be around $\times 10^5$ intensity variation (Bertozzi *et al.*, 2002)) have only recently become available (Photon Focus, 2006). Capable long range lidar systems are still on the cusp of car industry viability (Ibeo AS, 2007). Fundamental legal issues, such as the car makers' liability if the vehicle crashes, are not easily addressed. Most crucially however, a seemingly endless list of exceptions to the "rules of thumb" used to achieve road vehicle autonomy (such as planar roads, clear lane markings, dark road appearance, vehicle and pedestrian appearance etc.) stand in the way of fully autonomous vehicles.

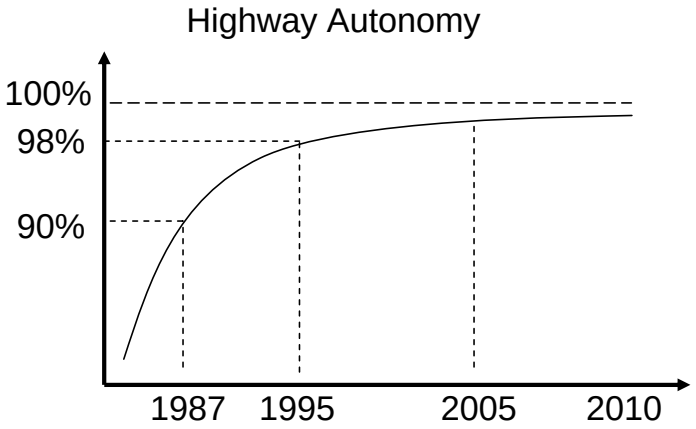


Figure 2.7: Diminishing returns of vehicle autonomy. On highways: UBM's "VaMORS" in 1987, CMU's "Navlab 5" in 1995.

Autonomous vehicles are not likely to be the next step in road safety. Figure 2.7 illustrates progress in autonomous vehicles for highway environments. Urban scenes with substantially more scene complexity and variance are significantly further behind (Franke *et al.*, 1999).

Fully autonomous vehicles are not yet ready for the real world. The long distance

demonstrations of the 1990s demonstrated impressive feats of innovation but no system then or since is capable of working all of the time. i.e. Even if the vehicle operates correctly 99.9% of the time, there is 0.1% where the human driver would be required. Research has drained away from autonomous vehicles because there was a diminishing return on the effort of solving the problems remaining in the field. Eventually, autonomous modes in vehicles may emerge in the low hanging fruit of traffic scenarios like "from on-ramp to off-ramp" highway systems on dedicated highways. These vehicles may prove convenient for commuters and improve highway infrastructure utilization. Hopefully there will be a modest win for road safety in these systems, but, as these systems only address part of the road network their benefits will only address a small part of road fatalities.

Research is now turning instead to the application of autonomous technologies to driver assistance systems. In stark contrast to the autonomous case, Maltz and Shinar (2004) showed that even imperfect driver assistance systems, complete with missed warnings and false alarms, still improved the overall driving performance of the human drivers. Similarly, Zheng and McDonald (2005) found that while preferred settings by drivers and the optimal safety settings of adaptive cruise control devices often diverge, drivers were able to adapt their driving to derive benefit and safety improvements from the devices. This indicates that an Automated Co-driver, while not providing full autonomy, should provide a significant benefit to road safety across the whole road network.

Lane departure warning

CMU's RALPH lane tracker was made into a stand-alone lane departure warning system called SafeTrac^{TM}, developed by (Assistware, 2006). Iteris (Iteris, 2002) has also developed a lane tracking product. MobileEye (MobilEye Vision Technologies Ltd, 2002) went further still and implemented lane tracking in a chip (ASIC device). Due to the economics and market size of the car industry this was a bold move. A move that is unlikely to have paid off for the company. There is little research using particle filtering for lane tracking. A notable exception is (Southall and Taylor, 2001) whose approach parallels ours in this regard. Particle filtering enables the support of a multi-modal distribution in the tracked state space. (Southall and Taylor, 2001) used this characteristic to track lane changes, where as we use this property for incorporating "late" sensor data to fuse multiple visual cues and to tolerate visual ambiguities due to varied road conditions.

Each of these systems is configured to generate warnings when the vehicle is about to cross or has crossed the lane boundary. These systems produce visual, vibration or auditory warnings when lane boundaries are crossed. Although many of these systems can be interfaced with the turning indicators on the vehicle to prevent warnings due to intended lane crossings (NHTSA, 1999), there are still many occasions where lane markings are crossed (such as passing straight through an intersection with turning lane marks or merging) where false warnings may occur. To be truly effective, these systems need to be capable of sensing whether the driver intends to turn. Just as a vigilant passenger would not warn the driver about a straightforward passage through an intersection, future assistance systems need to model the driver's intent not just observe the driver's actions. Further information cues are required to determine if lane departures are intentional, we will monitor the driver to make this determination.

Obstacle detection

Franke and Heinrich (2002) combined stereo and optical flow data to improve obstacle detection in cluttered urban environments. The work is impressive in that it is one of few examples of obstacle detection applied in the urban environment. We intend on using stereo vision and optical flow in our detection system. The group mention further work of using image pyramids to enable faster vehicle speeds to be supported. Though our algorithm will be quite different the use of stereo and optical flow reinforces our decision to use multiple visual cues for improved robustness as a cornerstone to our target tracking framework.

(Williamson and Thorpe, 1999) developed a large baseline tri-camera obstacle detection system. The system could detect objects the size of a drink can up to 100 metres in front of the vehicle. This system used a technique called ground plane stereo which is similar to the virtual aerial view approach, named inverse perspective mapping (IPM), of (Bertozzi and Broggi, 1998). With a baseline of $1.5 metres$ the road scene viewed between the cameras can appear quite different. Calibration, stereo correspondence and occlusions are more challenging with this configuration. The group demonstrated the need for Laplace of Gaussian (LOG) pre-filtering to enhance the image so that dense stereo depth-map algorithms would work reliably. In our research we use similar image pre-filtering.

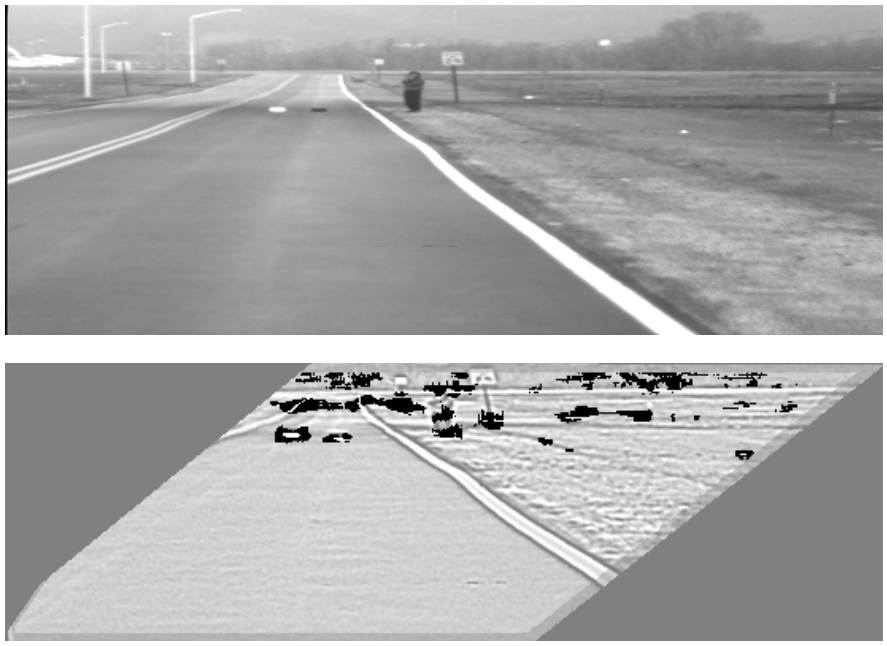


Figure 2.8: Ground plane stereo obstacle detection. **(a):** Original image. **(b):** Detected obstacles. From (Williamson and Thorpe, 1999).

Although quite a few groups have used camera images transformed into a virtual aerial view for lane tracking and obstacle detection (Bertozzi and Broggi, 1998; Pomerleau and Jochem T., 1996; Williamson and Thorpe, 1999; Thrun *et al.*, 2006), we do not intend on using this approach. Small errors in the estimated transformation from the camera to the "aerial view" generate large differences in the "aerial view". The transformation must be based on the estimated relationship between the camera and the road scene. Vehicle pitch, vibrations and varying road geometries (such as non-planar roads and road camber) all perturb the true transformation. While some groups have shown that their algorithms still work on non-planar roads (Pomerleau and Jochem T., 1996; Bertozzi *et al.*, 2000), these sources of error are effectively hidden by the image transformation from further stages of the algorithm. In computer vision research there has been a movement away from algorithms using transformations like these, instead the preferred technique has been to project estimated real world features back into the original image and measure the error in the sensor space. For example, preferred methods for estimating homographies, stereo geometries and 3D structure estimation attempt to minimise the reprojected error in the original image due to the estimated variables (Hartley and Zissermann, 2000). This argument is similar to the physically grounded models argument of the UBM group and is possibly why the UBM group used the reprojected error in the camera image for their own

work. We will use an approach that minimises the reprojected errors.

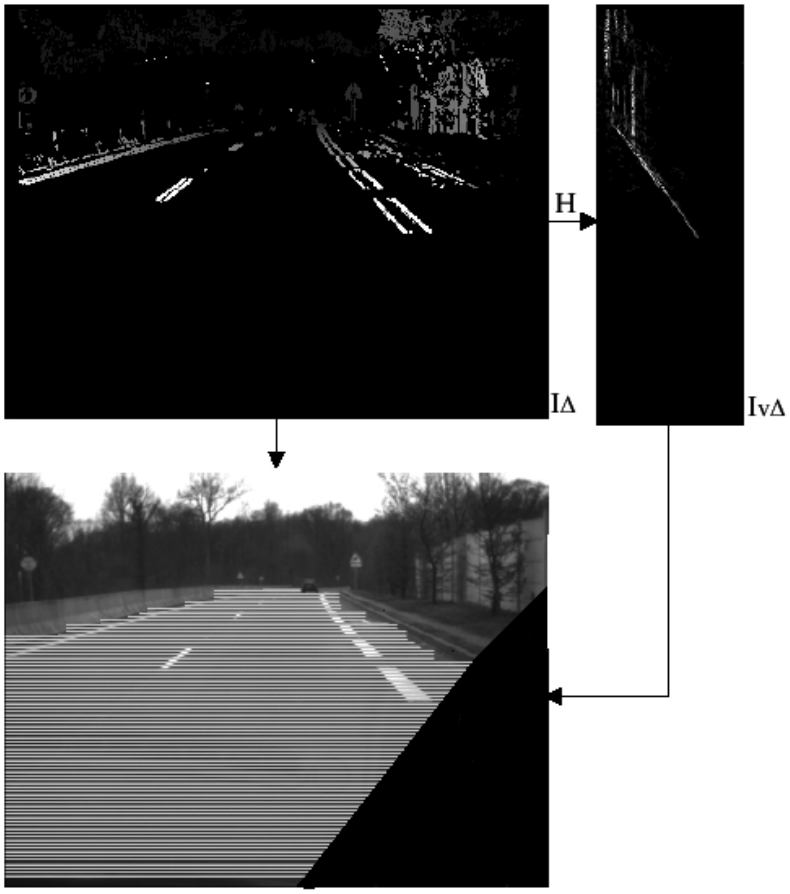


Figure 2.9: **top left:** Source disparity map. Scanlines are accumulated by disparity to form the V-disparity image. **right:** The resultant V-disparity image featuring a line representing the ground plane. **bottom:** Detected road surface. From (Labayrade *et al.*, 2002)

The ground plane is a strong and useful constraint for road scene understanding and segmentation. Labayrade *et al.* (2002) found a way to estimate the ground plane and use the ground plane constraint without requiring a planar model such as an aerial view. In this work, the traditional stereo disparity is plotted as a 2D histogram called a V-disparity image. For each scan-line in the original disparity map, the V-disparity image accumulates points with the same disparity. The image plots scan-lines against stereo disparity. A point in the V-disparity image represents the number of pixels in that scan-line at that disparity. The governing principle is that features of interest in the disparity map will constitute a signif-

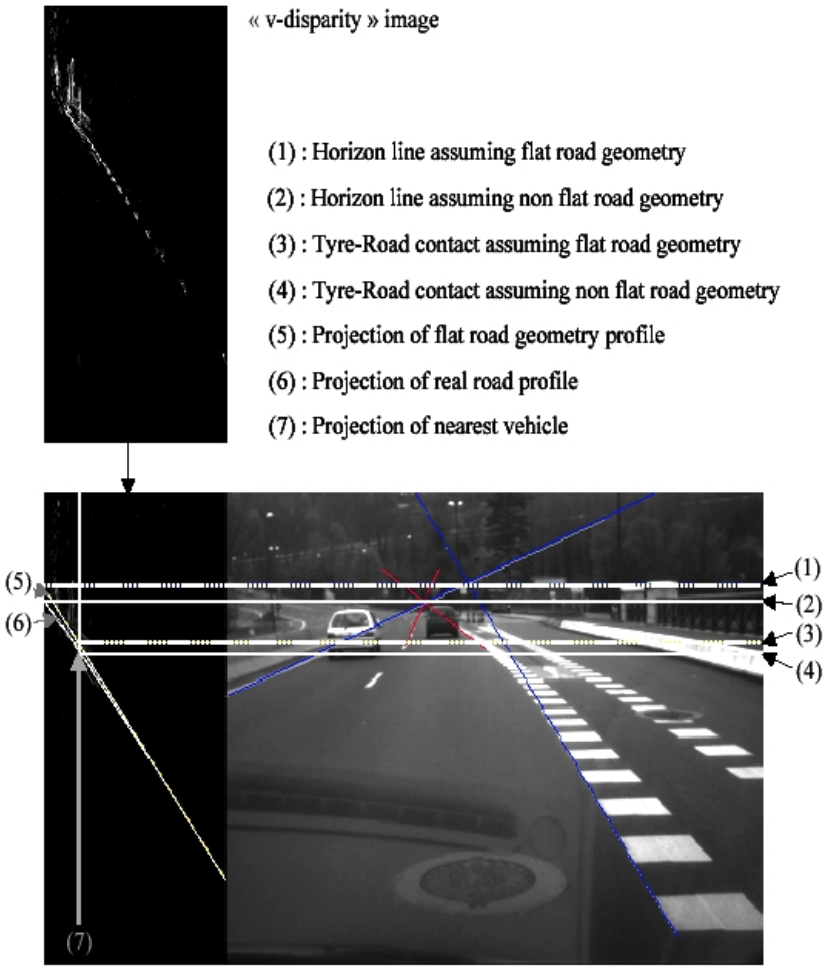


Figure 2.10: **top:** V-disparity image. **bottom:** Features segmented in the image. The red (v-disparity) and blue (planar) lines illustrate the discrepancy if a flat road model was fitted. From (Labayrade *et al.*, 2002)

icant number of pixels to be prominent on the V-disparity image. Figure 2.10 shows how this method can be used to extract useful road scene features such as the road profile and obstacles from the stereo data without having to create an aerial view or explicitly fitting a plane. Since this method relies on a voting space, it has a good tolerance to uncorrelated noise in the disparity data. One sensitivity of the algorithm not mentioned by the authors is that objects in the road scene running along the road sides that are not at the road height, such as guard rails or dense traffic, can introduce a competing phantom path into the V-disparity image. Since the road surface is notoriously untextured the strong edges of such an objects can cause an incorrect road plane to be selected. This algorithm offers an effective way of using the ground "plane" constraint without

requiring the road ahead to be planar. We will take advantage of this algorithm in our obstacle detection system.

Viola *et al.* (2005) used the Ada-boost machine learning approach to detect pedestrians. The algorithm is very flexible and powerful and has since been applied to many computer vision classification problems. This approach was built on by our local project collaborators the "National ICT Australia (NICTA) Smart Cars" group. Since their work is well advanced in the development of their pedestrian spotting system, we use their system as opposed to developing our own system for pedestrian detection. The NICTA pedestrian detection algorithm is used to classify objects detected by our algorithms.

Road sign recognition

Colour segmentation is a popular method for road sign detection (Priese *et al.*, 1994; Piccioli *et al.*, 1996; Paclik *et al.*, 2000; Johansson, 2002; Fang *et al.*, 2003; Shaposhnikov *et al.*, 2002; Hsu and Huang, 2001). The dominance of the technique is likely due to the conspicuousness of the coloured signs to human viewers. Colour segmentation is actually a poor choice of technique for the outdoor environment. Sign colour is composed as a product of the incident light and the sign colour plus specularities. Any time when the incident light colour is different (e.g. dawn or dusk, street lighting or strong shadows) the colour segmentation algorithms may potentially fail (Austin and Barnes, 2003). Road sign shape is a much more invariant feature. Franke *et al.* (1999) originally used colour segmentation for traffic light detection. A newer road sign detection system was then developed using a shape segmentation technique (Borgefors, 1986). We use the less variant road sign shape information as well. As noted in Section 2.2, speeding is a serious road safety problem likely to be substantially improved by the in-vehicle speed monitoring function of an Automated Co-driver. To achieve speed monitoring in vehicles, the system must be capable of reading speed signs. Loy and Zelinsky (2003) developed a fast technique for detecting circles in images, the same technique will form the basis of our robust circular road-sign recognition system.

2.3.3 Driver modeling

In addition to vehicle intelligence for assessing the road environment the idea of using intelligent systems to interpret driver behaviour has also received significant research. Many groups have conducted driver behaviour monitoring and modeling using vehicle simulators and some times in vehicle systems.

Driver actions such as steering wheel movements, have been used to assess the driver's intent to pass as well as a data source to assess fatigue (Kuge *et al.*, 1998; Thiffault and Bergeron, 2003).

(Kuge *et al.*, 1998) used steering data and hidden markov models to detect the driver's intention to overtake. While steering data and passing manuevers are highly correlated they conclude that it was imperative that future work use road environmental information in order to acheive satisfactory performance for an advanced driver assistance systems. Road scene monitoring as well as driver monitoring is required to differentiate actions such as merging and following high road curvature from overtaking.

Recently unintentional cues such as posture, facial expression and eye gaze has opened up new opportunities in driver state assessment. These earlier cues in the sense-think-act paradigm provide information on the driver's intent as opposed to their actions.

Matsumoto *et al.* (1999) demonstrated head pose and eye-gaze tracking as a potential human machine interface. The research has been developed by Seeing Machines (2001) into an in-vehicle system. The system can accurately track head pose and eye-gaze direction down to $\pm 1^{\circ}$. As mentioned earlier the Percentage Eye Closure (PERCLOS)(Wierwille and Ellsworth, 1994) driver eye-blink metric and Percentage Road Center (PRC) (Victor, 2005) metric have been used to detect fatigue and prolonged inattention respectively. These metrics can both be computed by the in-vehicle gaze tracking system. The real potential of real time in-vehicle eye-gaze tracking could be far greater. We will use this system to determine the correlation between the driver eye gaze and road-scene events.

Oliver and Pentland (2000) used manually annotated road scene video data then combined head pose, pedal position data and hidden markov modeling to implement a driver action predictor (passing, turning, starting and stopping actions). The system was developed in a generative fashion enabling them to draw some

interesting conclusions. They found that there was a plateau of performance preventing better prediction from vehicle instrumentation based models alone. Driver head pose over time seemed especially highly correlated with the driver's mental state. The resultant model was able to anticipate the driver behaviour approximately one second before the action was initiated.

Driver action predictors can substantially reduce accident severity. McCall and Trivedi (2006a) was able to anticipate the driver's intention to brake by using a camera to watch the foot pedals in a vehicle. Such a system is another example of how we can combine the strengths of humans and machines to attain a better outcome.

Baker *et al.* (2004) used deformable models to track features on the driver's face to estimate the driver's mood (happy, sad, angry etc.). It is an interesting idea that an automated system would find use for a metric of the driver's mood. Determining how to use this metric in our Automated Co-driver is unclear to us. We will not use this metric, though it could be a subject for future work. Another metric of the driver's mood, however, that can readily be applied is their on-task attentiveness due to fatigue or high workload.

Road scene variance measurement

Thiffault and Bergeron (2003) found that monotonous visual environments can induce fatigue in drivers. However, little or no research has been conducted into the measurement of road scene variability for fatigue. Sethi and Patel (1995) found that scene changes could be detected in video data based on substantial image changes encoded by an MPEG movie encoder. A method similar to this may be capable of measuring variability in the road scene over time. We use MPEG encoding to define a metric of visual monotony of the road scene. The developed metric significantly enhanced the ability of our Advanced Driver Assistance System to estimate the alertness, and thereby the behaviour, of the driver.

The converse problem of measuring the complexity the current driving task also has important implications for estimating driver workload. Mourant *et al.* (1969) found that the visual workload of direction sign reading can have a significant effect on driving performance. (Green, 2000) concluded that driver workload management devices should be technically possible as many of the required technologies to monitor road features relevant to driving are in active development,

including: road geometry, traffic, speed, signs, weather, time of day measurement. Though they note that how to combine these features into a single measure of driver workload is as yet completely unknown. Salvucci and Liu (2002) suggested a system with situational awareness capable of "gazing" at its environment with simulated "eyes" mimicking the gaze patterns found in driver monitoring. The system complete with forgetfulness and blind spots would simulate driver behaviour in order to judge the current cognitive demand of the driving task.
This approach sounds similar to the current concept of road threat assessment. Broadhurst *et al.* (2005) used behaviour models and statistical sampling to estimate possible threats in a given (simulated) road scenario. Perhaps the quantity of significant threats from a system like this could be used as a workload metric. This approach would require a substantial amount of modelling and road scene understanding.
Another approach may be to look for image features that are generally highly correlated with driving related features of the road scene. Image features such as the number of edges or bilateral symmetry may be applicable. Once the degree of difficulty of the driving task can be measured, an Advanced Driver Assistance System could take charge of low urgency tasks, defer phone calls and other electronic disruptions until safer stretches of a busy route. We will investigate using image compressibility to measure image complexity as a measure the road scene complexity. This metric could be used to manage the demands on the driver.

2.4 Discussion

From the above review we can summarise next potential steps in road safety under the following categories:

- Further use of the existing initiatives.
- Autonomous vehicles.
- Vehicles fitted with road scene monitoring based systems such as: ACC, lane keeping, obstacle detection, pedestrian detection.
- Vehicles fitted with driver monitoring based system such as fatigue monitoring.

- Vehicles fitted with road scene and driver monitoring combined into an Advanced Driver Assistance System.

We will now analyse each of these alternate directions for addressing road fatalities.

Further use of the existing initiatives

Previous safety initiatives have harvested the low hanging fruit of road safety. This approach is challenged by the diminishing returns for investment problem. A redoubling of efforts in existing initiatives will not make a significant impact on the issues of fatigue, distraction and inattention because these contributing factors induce dangerous scenarios that develop in an instant. In-vehicle interventions are the only way to achieve effective gains in road safety. A paradigm shift is required to take the next significant step in road safety. An initiative is needed that will address the key contributing factor in road fatalities, the driver.

Autonomous vehicles

By removing the driver, the autonomous vehicles approach certainly does address the key contributing factor to road accidents. However, removing the driver from road vehicles is not feasible yet. Fully autonomous vehicles are not the next step in road safety. There remain the significant problems of liability and reliability. They have the potential to take a very large step forward in road safety when the time comes, but the age of autonomous road vehicles is not yet here. The achievements of autonomous vehicle researchers are impressive but a flawed human driver is still a much safer driver than any existent autonomous vehicle (Thrun *et al.*, 2006; Leonard *et al.*, 2008).

Vehicles fitted with road scene monitoring based systems

In-vehicle monitoring of road scene and vehicle events will permit constant road event monitoring. This intervention has the potential to reduce road fatalities due to lateral and longitudinal collisions and warnings regarding speed and obstacles. Earlier warnings provide the driver with a longer time to evaluate a danger and react. This approach does not address the problem of fatigue, distraction and

inattention. Also, since it does not put the driver in the loop, this approach is vulnerable to false alarms as it can make no judgement of driver intent.

Vehicles fitted with driver monitoring based systems

Fatigue and distraction can be detected with in-vehicle driver monitoring. Fatigue has been shown to be detected with PERCLOS (Wierwille and Ellsworth, 1994) and distraction with the PRC (Victor, 2005).

However, inattention is not so easily detected without knowledge of current road events - highlighting the weaknesses of these systems when they operate in isolation from one another.

Crucially for this discussion, the research in driver modeling has clearly lamented the lack of road scene context information to adequately model the otherwise hidden states of the driver's behaviour (Desmond and Matthews, 1997; Nilsson *et al.*, 1997; Green, 2000; Kuge *et al.*, 1998; Oliver and Pentland, 2000; Salvucci and Liu, 2002; Thiffault and Bergeron, 2003). Without the realtime road context information the power of predictive driver monitoring simply plateaus (Oliver and Pentland, 2000).

Vehicles fitted with road scene and driver monitoring combined

Fatigue, distraction and inattention can only be properly detected with a combination of in-vehicle driver, road and vehicle monitoring. Combined systems don't only warn of upcoming road events (as vehicles fitted with road and vehicle monitoring based systems do), nor do they simply detect fatigue or distractions (as vehicles fitted with a driver monitoring based systems do). By the correlation of driver eye-gaze monitoring and road scene feature extraction, these systems can detect driver inattention to key, life-critical, road events. Road events such as missed pedestrians, speed changes, unintentional lane departure and workload management, can be detected without introducing further distraction dangers. An Automated Co-driver could provide advanced warning of crucial road events, suppress redundant warnings and interact with the driver in the context of the current driving task.

The addition of driver monitoring to road scene feature extraction also supports the introduction of new autonomous technologies to road vehicles.

Human driver	**Automated driver**
Competent to drive 100% of time.	Competent to drive (highways) 98% of time.
Distractable, inattentive.	Vigilant.
Performance degrades over time. Susceptible to fatigue.	Tireless.
Subject to boredom, tedium.	Consistent, multitasking, servile.
Highly adaptable.	Limited programmed adaptability.
Online problem solving.	Limited programmed problem solving ability.
High ambiguity and uncertainty tolerance.	Limited programmed behaviour change.
Highly evolved qualitative sensory perception system, developed for outdoor environment.	Limited but upgradable quantitative sensory system, not confined to range of human senses.
Limited reaction time.	Near instantaneous reaction time.

Table 2.2: Competencies of human and automated drivers.

Although autonomous technologies in vehicles work very well in their element (over 98% of the time), these systems are highly susceptible to being "tricked" by extreme or unlikely circumstances due to the rules of thumb they are built on. Humans on the other hand are remarkably flexible. They are able to problem solve on the fly, even when prompted by the most improbable situations (for example, flash flooding completely obscuring the road or an earthquake leading to sudden cracks or large peaks in the bitumen).

The answer we propose in this thesis is to combine the complementary competencies of the human and the autonomous systems. Table 2.2 summarises the complementary strengths and weaknesses of the human and the automated driver. Instead of the traditional model of a driver applying the control signal to the vehicle to move through the road environment, we can consider two drivers: the human driver and the autonomous driver, collaborating to control the vehicle.

The crucial question is how such a collaboration could be implemented. There is a ready analogy where two drivers are in charge of the one vehicle: an aircraft pilot and a co-pilot.

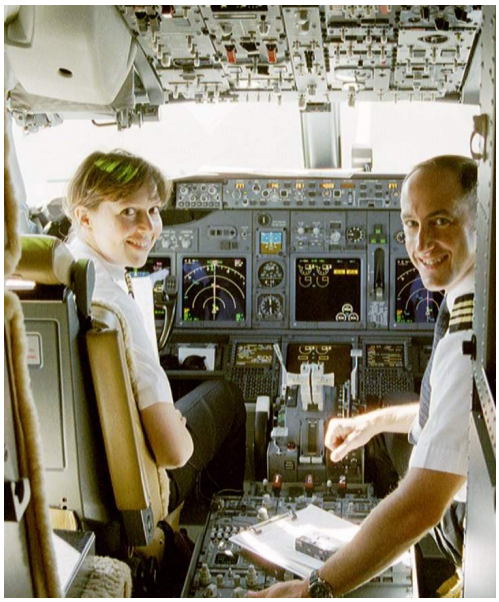


Figure 2.11: Aircraft Co-pilot

A human aircraft co-pilot (Figure 2.11) provides relief to the pilot by assuming control of the vehicle, but the co-pilot also double checks life-critical actions and shares the burden of administration. We envisage that an Automated Co-driver would provide a role in road vehicles which is an amalgam of a vigilant passenger, driver aid and a safety system. An Automated Co-driver could: double check life-critical actions; relieve the driver of tedious activities; warn about upcoming or missed events, and minimise harm in an imminent accident.

We subscribe to the "Hippocratic" like opinion of Goodrich and Boer (2000) that the burden of proof is on the assistance system to demonstrate a safety enhancement compelling enough to justify the cost of the driver's autonomy/disruption. The importance or benefit of an intervention must out weight the potential cost of the disruption. If the benefit is not there a system should take no action. Studies of driver gaze show that the driver looks in the direction of his or her intended path down the road (Land and Lee, 1994; Salvucci and Liu, 2002; Apostoloff and Zelinsky, 2004). For example, our potential system will use driver eye-gaze monitoring in addition to lane curvature estimation to determine whether the driver is intending to depart the lane. If the driver eye-gaze pattern is consistent with a lane departure, no warning will be generated. If lane departure is determined to be unintentional, then the system will generate a warning.

Figure 2.12 shows a list of key problems remaining in road safety as identified

by a survey of traffic authorities of OECD member countries. We envisage an Automated Co-driver could potentially make a contribution to the (blue bold) highlighted problems. The contributions range from inattention detection, earlier warning to crucial road events and driver workload management.

Accidents with animals	Infrastructure aspects: conflict potential, condition of roads: black spots; safety barriers; separation, **obstacle on roadside**, inadequate maintenance; small investment in infrastructure	Railway crossing
Bus safety	Institutional problem / Lack of coordination / Lack of political will / developing a strategy	Rural roads / Narrow roads / overtaking in rural roads/ **head-on collisions on rural roads**
Children	**Inter vehicle Distance**	Seatbelt (front and rear); child restrain system; seatbelt in buses
Drink Driving	International co-ordination	**Single vehicle accidents / roadway departure crashes / roadside hazards/ run off crash**
Drugs	Intersection; left turn at junctions	**Slower rate of reduction of fatal casualties**
Education / training / road safety awareness / Long life education	Investment (lack of) infrastructure	**Speed, speeding, speed limits**
Elderly drivers	License (driving without)	**Traffic signal violations**
Enforcement: Non compliance of rules / low level of enforcement / **implementation of new tech for enforcement**, serious offenders	**Making use of scientific potential**	Under reporting of injury accidents
Evaluation / Monitoring of road safety	Media not used as they should	**Urban areas**
Fatigue	Medical Care / trauma	Vehicle inspection / **safety of vehicle / safety equipment of vehicle** / no ESP in smaller vehicles
Foreign drivers	management	**Vulnerable road users, cyclist**
Frontal accidents	**Mobile phone**	Weather conditions
Hazardous driving, **poor attention while driving**, aggressive driving	Motorcycles / mopeds / helmet and protecting gears	**Young Driver / novice drivers/lack of driving experience / unsafe behaviour / negligent driving**
HGV / commercial vehicles	Motorways	
Improvised rule making	**Pedestrian**	

Figure 2.12: List of key problems remaining in road safety. Highlighted problems (blue bold) addressed by the proposed approach. Data from (OECD/ECMT, 2006).

To implement an Automated Co-driver that mimics the supporting role of a passenger in road vehicles, we need to identify the key qualities that allow people to achieve this task.

2.5 Reverse engineering a human driver to create an Automated Co-driver

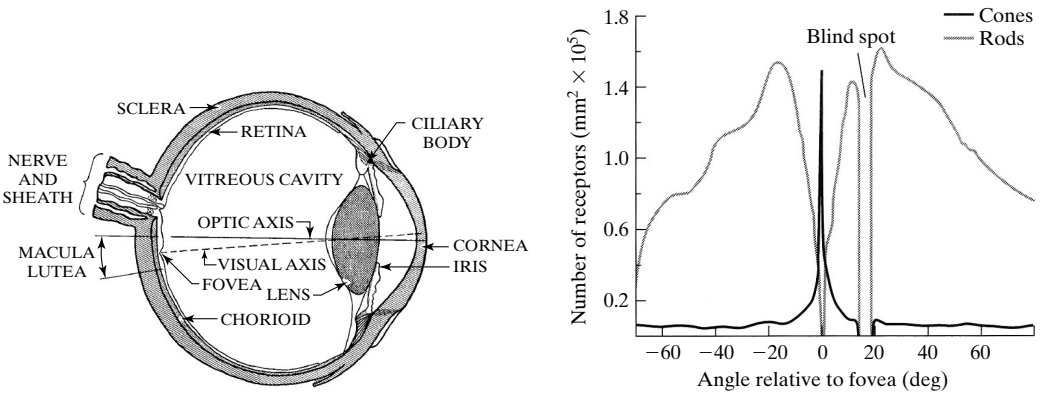
As noted in Chapter 1, road vehicles - unlike other complex, potentially dangerous vehicles like planes and ships - are operated by a single person. In aircraft, a human co-pilot understands the environment outside of the vehicle, the vehicle itself and the behaviour of the pilot - operating as an essential safety measure. In road vehicles, a vigilant passenger can fulfil many of these roles - and an

Automated Co-driver could take the best of these human elements to create a robust safety system tailored to the environment both inside and outside the vehicle.

In this work, we take a driver inspired approach for several reasons. First, a human driver (or aircraft co-pilot) is the only example we have of an intelligent system capable of the task. Second, our system attempts to estimate the intentions of the driver and this is best done using an experimental set-up which is as close to the driver's capabilities as possible. Finally, we take the view that failing to take full account of the driver's strengths and weaknesses is to fail to properly understand the real-world nature of road fatalities.

Next, we investigate how human drivers accomplish the task of driving. Although some insights and the parallels in automated systems are well known, we will briefly review this literature in terms of the application of driver eye gaze monitoring to understand the driver's intent.

The task of driving is often described as a tracking task (Gordon A. D., 1966; Land and Lee, 1994). The driver interprets the road scene primarily by vision and then applies the required control signal to the vehicle to maintain the correct relative position.



(a) (b)

Figure 2.13: **(a):** Principal components of the human eye. **(a):** Distribution of rods and cones across the field of view. From (Forsyth and Ponce, 2002).

The human vision system, the primary sense for driving, consists of two photosensitive organs each containing about one hundred and twenty million intensity-sensitive rods and about six million colour-sensitive cones (see Figure 2.13). The

cones are principally in a foveated region in the centre of the field of view, decreasing sharply away from the centre region. The rods have the reverse distribution, being non-existent at the centre of the foveated region and rising steeply in density until around 20° from the optical axis (see Figure 2.13(b)). The result of this geometry is a high resolution colour sensitive central region of vision of around 2.6° diameter (Wandell, 1995) and dropping off to a lower resolution monochrome view which extends to form a 120° peripheral field of view. The foveated region of vision permits high acuity tasks such as reading text and target tracking. The peripheral field of view is very effective at detecting changes in image motion and is used for directing attention. To generate the perception of a wide detailed view of the world, the eyes are able to move in the order of 500° per second (Wandell, 1995). Given this vision sensing geometry, driver eye-gaze tracking has the potential to be effective in determining what regions of the road scene have the driver's attention.

Gordon A. D. (1966) reported on a mathematical analysis of the motion of the road scene to the driver. He concluded that, while driving, the road appears in steady state to driver. Driving then becomes a lateral tracking (lane keeping) problem. Road boundaries, not focus of expansion (as often thought), is the dominant cue for aligning the vehicle in the lanes. If the vehicle is misaligned laterally the whole view field moves as a result. Summala and Nicminen (1996) demonstrated that peripheral vision was sufficient for short term lane keeping. When the driver is lane keeping the peripheral vision is capable of verifying that small motions in the road scene indicate that the vehicle is on track. When the vehicle starts to diverge from the lane direction, the induced whole view field motion alerts the driver to take direct notice. In correlating the driver eye-gaze with the road scene, this means that for lane keeping the driver does not always need to fixate on the road for safe driving. However, when the vehicle is misaligned with the road ahead, such as during lane departures, the driver would receive a strong visual cue to monitor the road position. If the driver is not observing the road direction, then this is a significant indicator of inattention.

This leads to the question of where the driver would look in the road scene for attentive lane tracking. In a clinical trial Land and Lee (1994) investigated the correlation between eye-gaze direction and road curvature. The group found that the driver tended to fixate on the tangent of the road ahead. This is a useful result for lane-keeping attentiveness. Since peripheral vision is sufficient for short-term lane keeping, as long as the driver maintains lane position our system would need

only to check for periodic attention. When lane departure events are predicted, our system can determine if the driver is attentive by verifying whether he or she is monitoring the road tangent.

Road scene features such as signs, pedestrians and obstacles require foveated vision. Maltz and Shinar (2004) proved that peripheral vision is insufficient for road hazard detection. For our system, this means that to have seen a critical road object the driver must have, at some stage, directed their eye-gaze directly at the object.

Understanding the driver-gaze pattern provides essential cues for how we can evaluate the driver's behaviour based on the driver's eye-gaze pattern in our Automated Co-driver system. From this analysis we have found that: lane keeping can be achieved by the driver for short periods with peripheral vision, while lane changes and departures require foveated vision. Road objects can be detected by peripheral vision, though recognising road objects requires foveated vision. With in-built driver eye-gaze monitoring coupled with road scene monitoring, we can detect these cases to validate the behaviour of the driver against these criteria of expected behaviours.

2.6 An Automated Co-driver approach to road safety

As highlighted in Section 2.2, inattention is a key cause of road fatalities. Inattention is the common element in accidents caused by fatigue, distraction, alcohol and drug use, and speeding. Addressing inattention will break the chain of events which leads to road fatalities. Our experiments will concentrate on combining driver monitoring with road-scene feature extraction to attempt driver observation monitoring. Driver observation monitoring is the key to detecting driver inattention (See Figure 2.14).

Several groups have examined driver monitoring alone to interpret driver behaviour (Wierwille and Ellsworth, 1994; Victor, 2005; Baker *et al.*, 2004). A system suitable for our research has been previously developed in our laboratory (Seeing Machines, 2001). The system estimates head pose and eye-gaze direction of the driver in real-time. The implementation and analysis of such a system could easily be a research thesis in itself. The commercially available

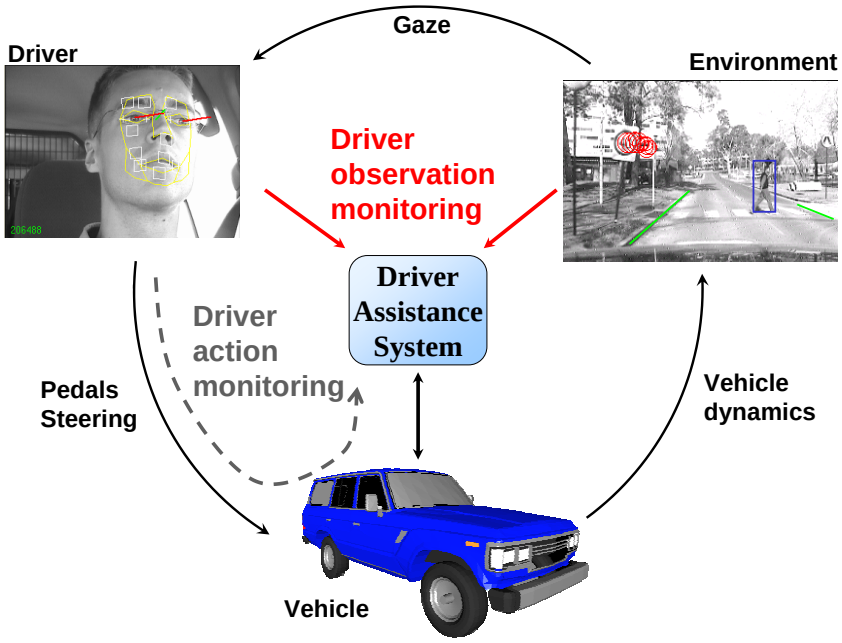


Figure 2.14: Driver observation monitoring: Driver eye-gaze tracking and critical road scene feature detection are combined to determine what the driver is seeing (or not seeing).

system is sufficient for our experiments.

The group of Trivedi *et al.* (2005), a particularly strong group due to the inclusion of human factors and intelligent vehicle researchers used a sensor rich vehicle for driver behaviour modeling. A kindred spirit of our research, the group used a heavily instrumented vehicle (McCall *et al.*, 2004) using vehicle data, head pose monitoring, foot pedal monitoring and panoramic vision of the road scene to estimate driver behaviour. Moving from driver behaviour analysis to assistance systems the group has recently been correlating driver head pose with road scene perception (Trivedi *et al.*, 2005). They show an improved classification of the driver intent to change lane when head pose data is included in the modeling.

Gerdes and Rossetter (2001) used force feedback through the steering wheel as a natural human machine interface for lane keeping. The driver feels a gentle force through the steering wheel to encourage the driver to maintain a central position in the lane. The perception to the driver is that the car is being guided down the lane as if it is on rails, or driving along a trough. The applied force is weak enough that the driver can override the force to overtake or depart the lane. As mentioned in Section 2.2, Carsten and Tate (2001) developed a similar

natural accelerator force feedback mechanism for speed control. In addition to developing the Percentage Road Centre (PRC) metric, Victor (2005) developed an application for the metric. They used a trail of coloured lights to lead the driver's attention back to the road when the PRC metric indicated a visual or cognitive distraction was preoccupying the driver. This interface could be compatible with our Automated Co-driver, but our aim is to use a communication channel in-line with what a human would use, so we will use auditory messages.

Much research has also been devoted to the development of individual driver aids such as adaptive cruise control and lane keeping. While a final Automated Co-driver system would incorporate such functions, there is little to be gained in re-implementing these functions in our work since the research questions regarding machine-driver interaction can be addressed without automation. Also the experimental vehicle is not permitted by the road-traffic authorities to use actuated systems on public roads. We chose to avoid the final actuation stage in our experiments to permit us to experiment on a diverse range of public roads instead of being constrained to a small atypical road environment available for actuated experiments. Instead we concentrate on experiments which implement novel combinations of autonomous systems technologies for the machine-driver interaction.

The key to our Automated Co-driver system is the combination of road-scene monitoring, driver eye-gaze and vehicle monitoring. While driver gaze can detect fatigue and distraction, road-scene monitoring can detect obstacles and vehicle monitoring can detect speeding, in the real-world driving environment all three of these systems are required for inattention interventions.

The proposed Automated Co-driver system is an integration of many significant components of existing and new ideas in intelligent vehicles. Figure 2.15 illustrates where we will leverage off the existing state of the art with new research contributions to create the Automated Co-driver system of systems.

2.6.1 Proposed experiments

To demonstrate the efficacy of an Automated Co-driver to address the key issue of driver inattention in road fatalities, we will show how the proposed system can intervene:

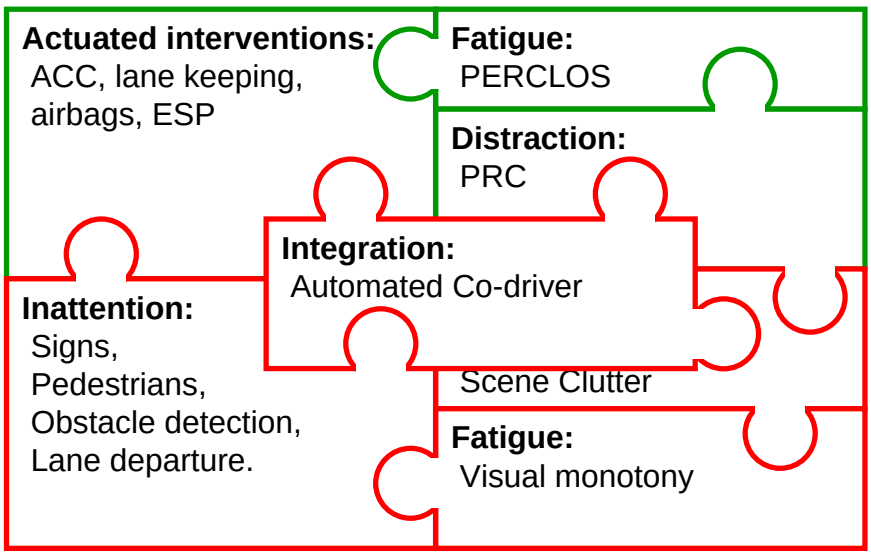


Figure 2.15: Research questions remaining for an Automated Co-driver system. *(green)*: Elements were we are content to use the state of the art. *(red)*: Where our research will contribute.

- To alert the driver to a threat unseen (by the driver).
- To suppress warnings when the driver is already observing the road event.
- To assess the driver's fitness regarding fatigue or distraction based on the prevailing conditions.
- To assist the driver in a manner natural to the task of driving, thereby not introducing additional distraction.

Figure 2.16 shows the Advanced Driver Assistance System components required to implement our Automated Co-driver experiments.

2.6.2 Experimental testbed

To achieve the desired research goals we developed a test vehicle. The vehicle contains sufficient video cameras and processing power to implement the developed algorithms in realtime The vehicle is also equipped with a eye gaze tracking system to provide realtime gaze direction information to our driver assistance applications (see Figure 2.17).

A distributed modular software infrastructure was also developed to implement

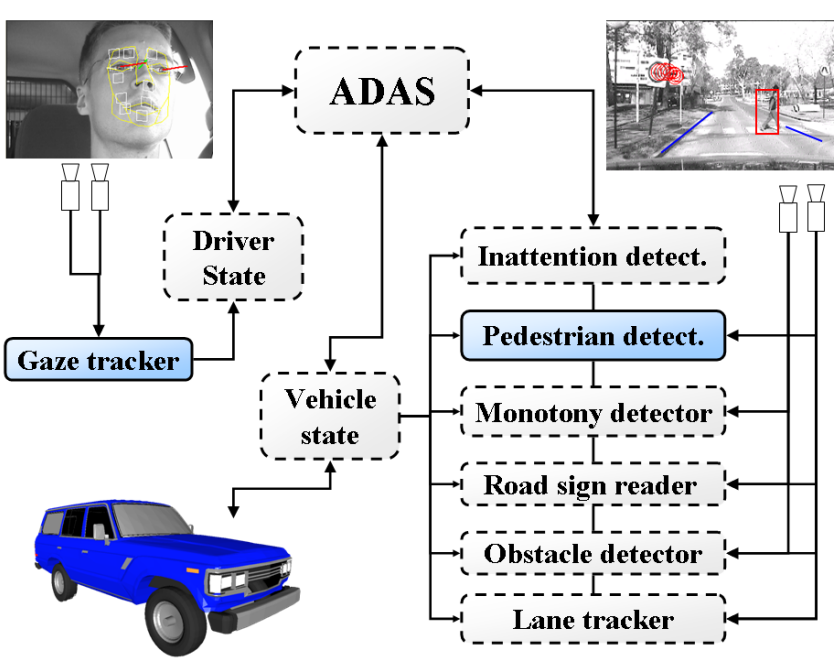


Figure 2.16: Advanced Driver Assistance Systems components. **Dashed:** components requiring development. **Solid:** components available from previous research.

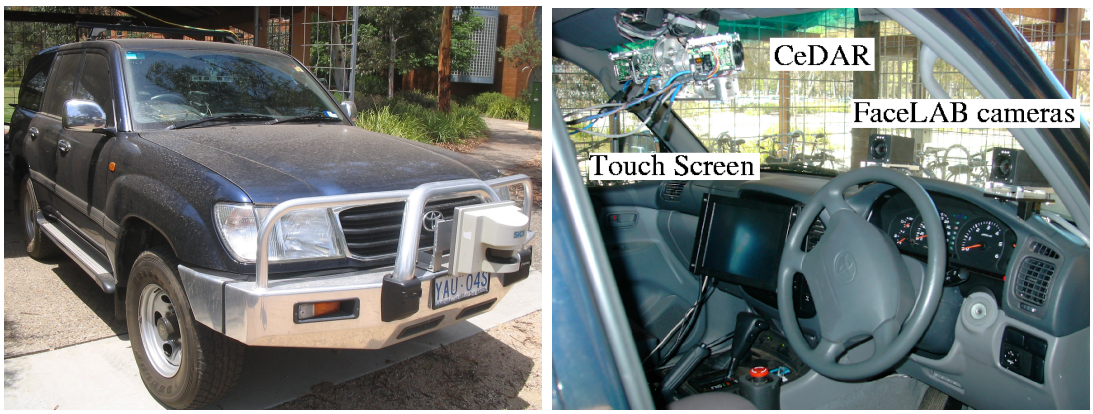


Figure 2.17: Experimental vehicle. The vision platforms in the vehicle. The CeDAR active vision head in place of a central rear vision mirror and faceLAB^{TM} passive stereo cameras on the dashboard facing the driver.

the driver assistance applications on the vehicle. The software infrastructure enabled subsystems operating as many separate processes on different computers to operate in concert to support the envisaged driver assistance system.

Appendix A describes the developed experimental vehicle in detail, the hardware that was used and the distributed modular software infrastructure that was

developed to execute our experiments.

2.7 Summary

Observing the complementary strengths and weaknesses of humans and autonomous systems, it is logical that an integration of both provides the best hope to improve road safety. A useful paradigm is needed to combine the two controlling agents. Our conjecture is that an Automated Co-driver that brings together the strengths of motor vehicle drivers, vigilant passengers, human co-pilots and the tirelessness of automated systems embodies that new paradigm.

By addressing driver inattention - a key underlying factor in road fatalities - an Automated Co-driver could substantially reduce road fatalities and therefore will provide the next significant step in road safety.

In the following chapters, we design an Advanced Driver Assistance System experimental architecture and computer vision algorithms to demonstrate an Automated Co-driver.

Chapter 3

Lane tracking

Despite many impressive results in the past (Behringer and Müller, 1998; Bertozzi *et al.*, 2002; Dellaert *et al.*, 1998a), it is clear that no single algorithm can perform 100% reliably in all road environmental conditions. If we are to build reliable Advanced Driver Assistance Systems, we will require systems that have the highest reliability. Autonomous systems require systems that are almost 100% reliable, which is a difficult challenge. Systems such as the Automated Co-driver will require sub-systems which have high-reliability, and which require occasional intervention. The goal of our research is to create an improved lane tracker for use in an Automated Co-driver.

In this chapter we highlight the need for visual systems to tolerate ambiguity and how such ambiguity can be later resolved by additional information only if multiple possibilities were propagated. We present a computer vision system that adaptively allocates computational resources over multiple cues to robustly track a target in the world state space. We use a particle filter to maintain and propagate multiple hypotheses of the target location. Bayesian probability theory provides the framework for sensor fusion, and resource scheduling is used to allocate limited computational resources across the set of available visual cues. We used a people tracking example to demonstrates the system localising and tracking a person moving in a cluttered environment. This application permitted the development of the vision framework in a test environment that was easy to manipulate. We then show how the algorithm can be applied to combine visual cues often cited for lane tracking into a system which is effective over a range of road appearances, weather and illumination conditions.

The system developed addresses the issues of which cue should be used and when,

how cues should be combined and and how many computational resources should be expended on each cue.

This vision framework addresses these issues in fulfilling the following criteria:

- combining visual cues based on Bayesian theory;
- exploiting reuse of low and medium level "bottom-up" processing steps (such as gradient images) between visual cues;
- employing a top-down, hypothesis-testing approach instead of reconstructive techniques that can often be poorly conditioned;
- efficiently allocating finite computational resources for calculating cues, accounting for the cue's expected usefulness and resource requirements;
- integrating visual-cue performance metrics so that deteriorating performance can be dealt with safely;
- facilitating cues running at different frequencies, and
- allowing tracking of multiple hypotheses.

Section 3.1 introduces some background theory, and then Section 3.2 describes our system in detail. Section 3.3 demonstrates a method of finding and tracking a person's face in a cluttered environment. In Section 3.4 the same algorithm is used to track lanes with an amalgam of visual cues from transport-systems literature. Finally, in Section 3.5 lane tracking is extended to demonstrate the flexibility of the developed framework in supporting a number of features desirable for integration with a larger system. The lane tracking is made robust with variable look-ahead, support for supplementary views as well as road curvature and vehicle pitch estimation. The lane tracker was found to track effectively 96.2% of an 1800 kilometre journey.

3.1 Review of robust target tracking techniques

3.1.1 Lessons learned from human vision

We have identified several concepts that we believe are crucial to the success of a human in the task of robust driving. We will implement analogies to these in our work. Human vision uses parallel visual paths and tolerance of visual ambiguities to achieve robust vision.

Parallel visual paths

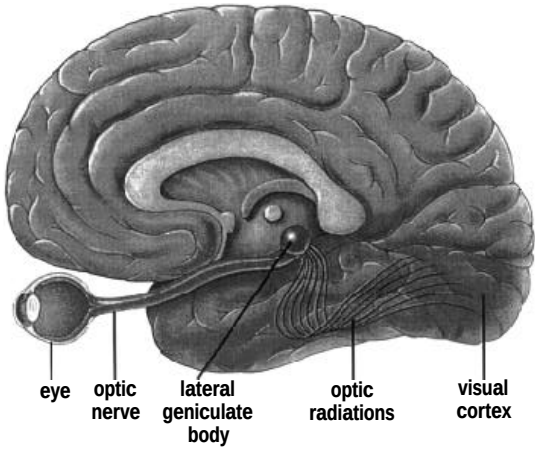


Figure 3.1: The Human visual pathway. Notice in particularly the "optic radiations" parallel visual paths. Reproduced from McEwen and Schmeck (1994).

In Figure 3.1 the general visual path is shown. One feature of the visual path is the parallel paths, named the "optic radiations". These parallel paths are thought to conduct several parallel representations of the sensed image to the centres of higher perception. Research in primates has shown at least twenty "renotopic maps" of the sensed image. These renotopic maps process the raw signals from the photoreceptors in distinctly different ways. In this way it is more likely that instead of a seeing one image of the world from each eye, a person actually has a set of parallel versions of the scene image. Each accentuates different salient features of the scene. We believe that a key component missing from existing road-scene vision systems is the lack of parallel representations or visual cues. We will use several visual cues, tuned to the road conditions and derived from the one sensing device, to achieve robust perception.

Tolerance of visual ambiguities

Vision is the projected view of the 3D world onto a 2D imaging sensor. Not only does this projection create non-linear motions as highlighted by the UBM group, but the projection can also induce cases where the true 3D geometry is ambiguous from the projected image, even when the 3D structure is known. Figure 3.2 illustrates one example where a 3D cube may have two different orientations. The human vision system switches between the two orientations.

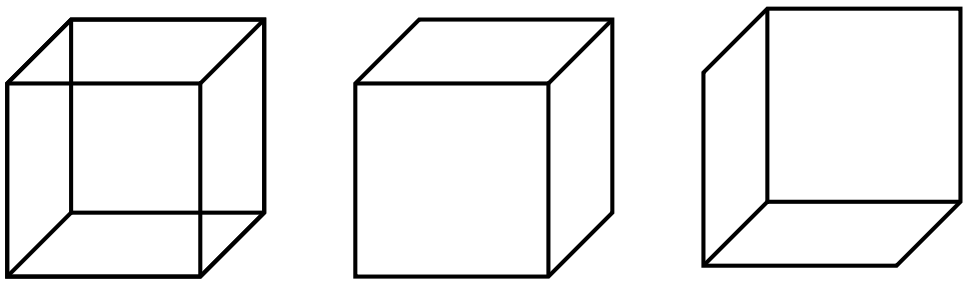


Figure 3.2: Necker Illusion. **Left:** Wire frame 3D cube with an ambiguous orientation. **Centre, Right:** Two alternative orientations of the cube.

The insight here is that the human vision system does not fixate on one orientation, or compromise with a merged view of both orientations. Instead, the human vision system supports both alternate orientations until further information permits one orientation to be identified (see Figure 3.3).

This tolerance of visual ambiguities is another characteristic of the human vision system which enables a level of robustness not yet exploited in road scene computer vision systems.

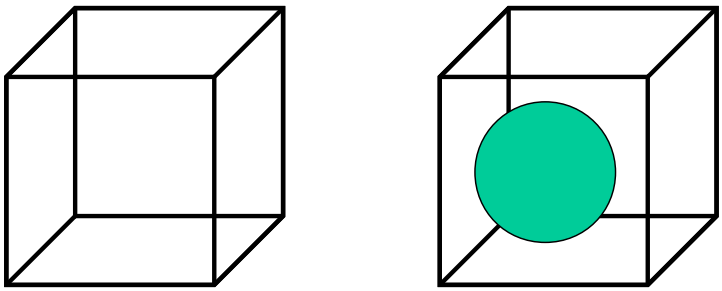


Figure 3.3: Necker illusion. **Left:** Wire frame 3D cube with an ambiguous orientation. **Right:** The orientation is no longer ambiguous with an additional visual cue.

3.1.2 Supporting ambiguity

The ingenious Kalman filtering algorithm (Kalman, 1960) has dominated tracking for the past 45 years (Brown and Hwang, 1997). Kalman filtering combines several

estimates using a weighting derived from a ratio of the variances of the estimates. Kalman filters provide the optimal solution for state estimation of linear systems with Gaussian noise sources (Brown and Hwang, 1997). However in the real world, most systems are not linear and most noise sources are not Gaussian. There has been further research on extending Kalman filtering to treat these cases. Extended Kalman filters, unscented Kalman filtering and multiple filters have gone a long way towards broadening the applicability of this technique to real-world problems (Anderson and Moore, 1979; Brown and Hwang, 1997; Franke and Rabe, 2005). However, for a range of applications with dynamics that are highly nonlinear, multi-modal or just unknown, the assumptions for Kalman filtering are too restrictive.

Since the 1990s, increased computational resources have made trial and error approaches feasible and popular. The particle filter also known as the condensation algorithm (Isard and Blake, 1996), or Sequential Monte Carlo Sampling (Thrun *et al.*, 2000), has been simultaneously developed by several groups. The groups (Isard and Blake, 1996; Gordon *et al.*, 1993; Kitagawa, 1996) are all credited with concurrent development of the algorithm. Isard and Blake (1996) in particular developed and applied the algorithm for tracking applications in computer vision.

Particle filtering is an essentially different approach to Kalman filtering because it comes from the domain of statistical sampling. Both algorithms achieve state estimation, but in particle filtering, state distributions are not explicitly modelled at all. Probability distributions are instead approximated by a sampled representation. The benefits of particle filtering are not explicitly designed in. They are a convenient consequence of using sampled representations.

The well-known condensation approach to contour tracking (Isard and Blake, 1996, 1998) tracked target outlines using particle filtering and active contours. The outline of the target is parameterised using B-splines, and described a point in the world state space. Impressive results have been shown illustrating how particle filter-based contour tracking methods can effectively deal with ambiguity, occlusions and varying lighting conditions.

Since the introduction of the Condensation algorithm, there has been an explosion of applications in robotics and computer vision, notably in robot localisation (Thrun *et al.*, 2000). The particle filter approach to target localisation, also known as Monte Carlo localisation (Thrun *et al.*, 2000), uses a large number of particles to "explore" the state space. Each particle represents a hypothesised

target location in state space. The algorithm cycles through the steps listed in Table 3.1.

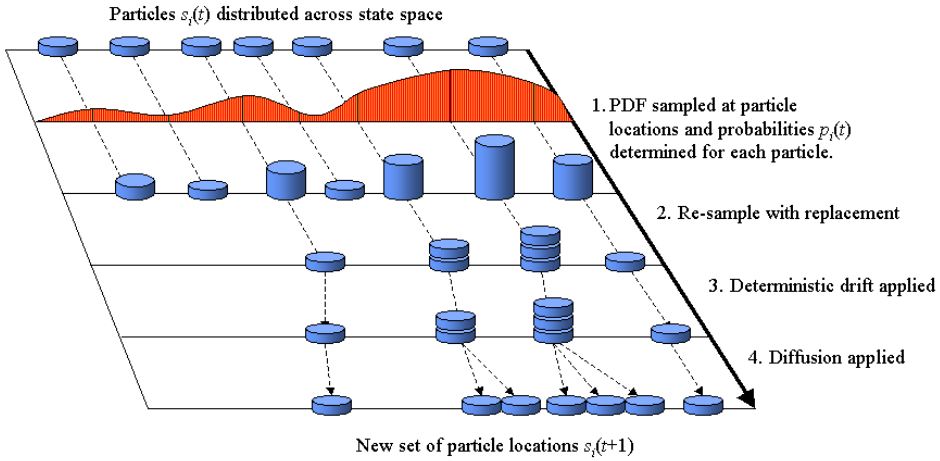


Figure 3.4: Evolution of particles over a single time-step. The unknown PDF is measured only at the particle locations, particles are then re-sampled with replacement, and drift and diffusion are applied to evolve the particles to their new locations.

This cyclic processing results in particles congregating in regions of high probability and dispersing from improbable locations, thus the particle density indicates the most likely target states. Furthermore, the high density of particles in these "target-like" regions means that these regions are effectively searched at a higher resolution than other, more sparsely populated, regions of state space.

The hypothesis-verification approach used by particle filters does not require the probability density function to be calculated across the entire state space, but only at particle locations. Thus locating a target in an image does not require searching the entire image, rather just the evaluation of image regions corresponding to hypothesized target locations.

Figure 3.5 illustrates another useful property of particle filtering - in this case, finding faces with ambiguity. For ambiguous cases such as the Necker illusion (Necker, 1832), competing hypotheses need to be preserved (not averaged as a Kalman filter would do) until additional information or time allows the ambiguity to be resolved.

Particle filtering algorithm:

Each particle represents a hypothesised target location in state space.
Initially the particles are randomly distributed across state space. For each subsequent frame the algorithm cycles through the following steps (illustrated in Figure 3.4):

1. **Measure:** The Probability Density Function (PDF) is measured at (and only at) each particle location, indicating the likelihood that a given particle is the target.
2. **Re-sample particles:** The particles are re-sampled with replacement, where the probability of choosing a particular particle is equal to the PDF at the particle location.
3. **Deterministic drift:** Particles are moved according to a deterministic motion model. Any dynamics known about the system can be modelled in this step.
4. **Diffuse particles:** Particles are moved a small distance in state space under Brownian motion. This step accommodates unmodelled errors in the system.

Table 3.1: Particle filtering algorithm.

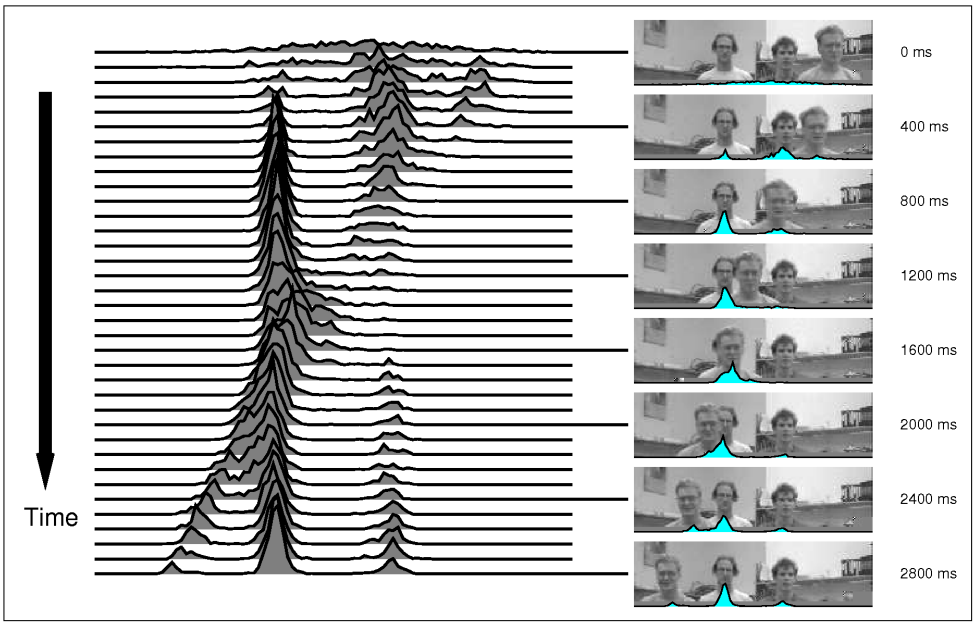


Figure 3.5: Particle filtering enables multiple hypotheses to be tracked. This is a key feature for coping with temporary ambiguities. Reproduced from (Isard and Blake, 1998)

3.1.3 Multiple visual cues

The use of multiple visual cues has been shown to improve the robustness and overall performance of target localization systems (Soto and Khosla, 2001; Crowley and Berard, 1997). While a number of researchers have used multiple cues for detection and tracking, there have been few attempts to develop a system that considers the allocation of computational resources amongst the cues, a notable exception being Crowley and Berard (1997).

Crowley and Berard (1997) used multiple visual processes: blink detection, colour histogram matching, and correlation tracking, together with sound localization, to detect and track faces in video. Each cue was converted to a state vector containing four elements: the x and y co-ordinates of the face centroid, and the face height and width. A confidence measure and covariance matrix were estimated by examining the state vectors of all the cues, and used to combine the state vectors to give the final result. The advantage of this approach is the compact form in which the state vectors represent information. The disadvantage is that only one target could be reported by each cue. Not only is such a system unable to explicitly track multiple targets, but it loses useful additional information by only allowing each cue to report a single target. For instance, if there are two regions of a target-like colour, we would prefer a system to report the presence of both regions and allow the additional cues to determine which is the true target, rather than returning a single result, namely the centre of gravity of the two regions.

Soto and Khosla (2001) presented a system based on *intelligent agents* that could adaptively combine multi-dimensional information sources (*agents*) to estimate the state of a target. A particle filter was used to track the target's state, and metrics used to quantify the performance of the agents. Results for person tracking in 2D show a good deal of promise for a particle filter based approach.

Triesch and von der Malsburg (2000) presented a system suitable for combining an unlimited number of cues. The system was demonstrated using contrast, colour, shape, and two motion cues (intensity change and a predictive motion model) to track a person's head.

For each (i^{th}) cue and (k^{th}) image frame the following quantities are determined:

- an image $\mathbf{A}_i[k]$ describing the probability a given pixel is part of the target,

- a quality measure $q_i[k]$ describing how accurate the sensor was in determining the final result in the previous image frame, and
- a reliability measure $r_i[k]$, which is effectively a running average of the quality measure $q_i[k]$.

The final result was given by the weighted sum $\sum_i r_i[k]\mathbf{A}_i[k]$. The $\mathbf{A}_i[k]$ image is generated by comparing the i^{th} sensor's information with a prototype $\mathbf{P}_i[k]$ describing the target with respect to that sensor. These prototypes were updated dynamically as a running average of the sensor's output at the target locations in previous frames.

The results of this system were impressive and demonstrate how combining multiple cues increases the robustness of a tracking system. This system has been an inspiration for our work with some important differences. First, our goal is to ensure that the object found fits a generic target model, whereas Triesch and von der Malsburg (2000) dynamically adapt their sensor suite to track the target identified in previous frames. Second, we require the system to localize a target in 3D, whereas their system operated only in 2D, and with fixed sized prototypes it could not deal with close or distant targets. Finally, when determining the utility of different cues we wish to take into account not only the tracking performance, but also the computational cost of each cue.

3.1.4 Integration

We have discussed a number of systems that are the quorum of inspiration for our work. The particle filtering approach popularized by Isard and Blake (1996, 1998) offers a solid framework for locating and tracking targets, and as Soto and Khosla (2001) demonstrated it is well suited for use in a multi-cue system. There is no question that multiple cues allow for more robust estimates, however, calculating more cues requires more CPU time and can quickly reach the limits imposed by a real-time system. Few researchers have considered the problem of controlling the allocation of computational resources between cues, in order to allow more effective and efficient cues to operate at the expense of those that are slower or not performing as well.

The algorithm we have developed aims to meld the strongest elements of the systems discussed here. A particle filter is used to maintain multiple hypotheses

of the target's location, and multiple visual cues are applied to test hypotheses. Computational resources are allocated across the cues, taking into account the cue's expected utility and resource requirements. Our approach accommodates cues running at different frequencies. Cues that are currently performing poorly can run slowly in the background for added robustness with minimal additional computation. In this way the system *distills* the available cues so that the most useful cues are favoured at the expense of the under-performing cues. For this reason we title our approach the Distillation Algorithm.

3.2 The Distillation Algorithm

The problem of target localization can be expressed probabilistically as the estimation of the posterior probability density function over the space of possible target states, based on the available data. That is, at time t estimate the posterior probability $P(s_t|e_{0\ldots t})$ of a state s_t given all available evidence $e_{0\ldots t}$ from time 0 to t.

Using Bayesian probability theory and applying the *Markov assumption*[1] the desired probability $P(s_t|e_{0\ldots t})$ can be expressed recursively in terms of the current evidence and knowledge of the previous states. This is referred to as *Markov Localization* (Thrun *et al.*, 2000),

$$P(s_t|e_{0\ldots t}) \quad = \quad \eta_t P(e_t|s_t) \sum_i P(s_t|s_{t-1}^{(i)}) P(s_{t-1}^{(i)}|e_{0\ldots t-1}) \tag{3.1}$$

where (i) denotes the i^{th} discrete value (i.e. particle).

For completeness the derivation due to (Thrun *et al.*, 2000) is included below. The derivation sequentially applies Bayes rule, the Markov assumption, the theorem of total probability and the Markov assumption again, and is as follows:

[1] The *Markov assumption* states that the past is independent of the future given the current state.

$$\begin{aligned} P(s_t|e_{0\ldots t}) &= \eta_t P(e_t|e_{0..t-1}, s_t) P(s_t|e_{0\ldots t-1}) \\ &= \eta_t P(e_t|s_t) P(s_t|e_{0\ldots t-1}) \\ &= \eta_t (e_t|s_t) sum_{s_{t-1}} P(s_t|e_{0\ldots t-1}, s_{t-1}) P(s_{t-1}|e_{0\ldots t-1}) \\ &= \eta_t P(e_t|s_t) \sum_{s_{t-1}} P(s_t|s_{t-1}) P(s_{t-1}|e_{0\ldots t-1}) \end{aligned}$$

This formulation provides a recursive means of estimating the probability of the current state given all the evidence sighted since sensing began.

Our algorithm uses a particle filter to model this equation and track a population of target hypotheses in state space. A number of cues are calculated from image and state information and combined to provide evidence strengthening or attenuating the belief in each hypothesis.

Figure 3.7 shows the structure of the system. It consists of two subsystems: a particle filter and a cue processor, each of which cycle through their loops once per frame. These subsystems interact as shown by the thick arrows in the figure. The particle filter passes the current particle locations to the cue processor. The cue processor determines the probabilities for the particles and passes these back to the particle filter. Each of these subsystems is discussed in further detail below.

3.2.1 Particle filter

The primary appeals of the particle filter approach to localisation and tracking are its scalability (computational requirement varies linearly with the number of particles), and its ability to deal with multiple hypotheses and thus more readily recover from tracking errors. The particle filter was applied for several additional reasons, specifically because it:

- provides an efficient means of searching for a target in a multi-dimensional state space;
- reverses the search problem to a verification problem (e.g. does a given hypothesis appear correct given the sensor information?);
- by sampling in proportion to a posterior distribution (Thrun *et al.*, 2001), concentrates computational resources in areas of the PDF that are most relevant, and
- allows fusion of cues operating at different frequencies.

The last point is especially important for a system operating multiple cues with limited computational resources, as it facilitates running some cues slower than frame-rate (with reduced computational expense) and incorporating the result from these cues when they become available.

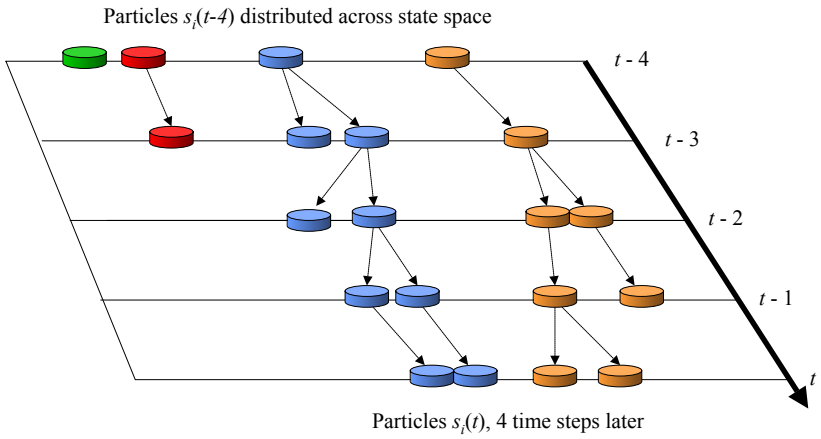


Figure 3.6: Example of particle population ancestry over time.

Particle ancestry to facilitate slow cues

If a cue takes n frames to return a result, then by the time the cue result is ready, the particles will have been re-sampled from where they were n frames ago. To facilitate such cues, the system records the ancestry of each particle over the previous k frames. The cue value determined for a particle $n \leq k$ frames ago is assigned to the descendants of that particle in the current frame, thus propagating forward the cue's response to the current frame.

Figure 3.6 shows a simplified example of the ancestry of particles in a one-dimensional state space, over four time steps. The particles at $s_i(t)$ are descended from the same coloured particles at $t-4$, thus a slow cue that takes four frames to compute is calculated on the particles $s_i(t-4)$ and the result assigned to their respective descendants in frame t. Values that are calculated for particles whose ancestors are not present at frame t are discarded.

Particle probabilities allocated to ancestors are reduced as a function of the time delay. This reduction represents the importance of timely information in the application. A familiar analogy is the way returns on investments are calculated: money received now is worth more than it is in the future. An exponential discount factor $d \in (0,1)$ is introduced that attenuates the cue probability for each frame it is delayed. That is, the probability $P(e_t|s_t^{(i)})$ is attenuated to $d^n P(e_t|s_t^{(i)})$ if it is n frames late. The value of this factor is dependent on the expected dynamics of the application. This discount is also taken into account when judging the expected utility of the cue.

3.2.2 Cue processor

While the particle filter maintains a record of target hypotheses and propagates these in state space, the *cue processor* deals with the calculation and fusion of cues

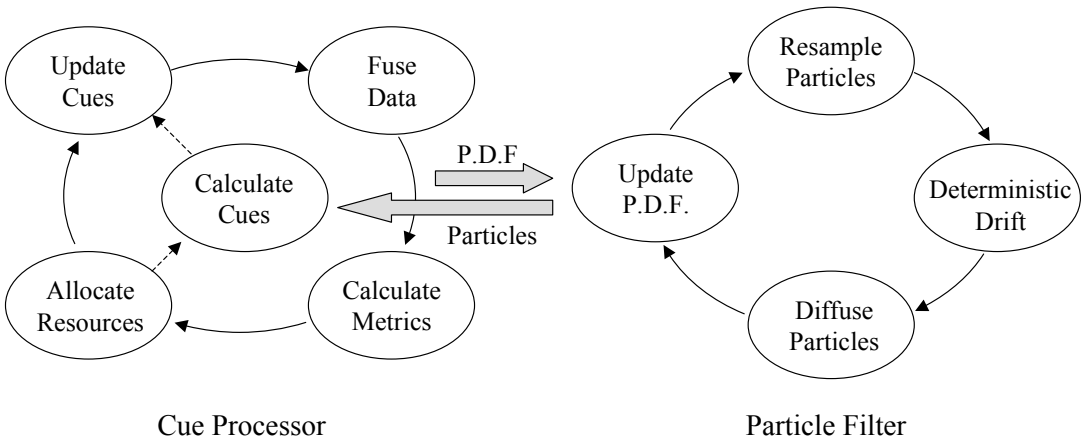


Figure 3.7: The Distillation algorithm is composed of a *Cue processor* and a *Particle filter* running in lock step.

The Distillation algorithm: A particle filter combined with a dynamic cue processing cycle.
The particle filter cycle operates according to Figure 3.1.
The Cue processor cycle iterates through the following steps:

1. **Update cues:** accesses recently calculated cues.
2. **Fuse data:** fuses the results of different cues to estimate the overall probability for each hypothesised target state.
3. **Calculate metrics:** determines metrics for each cue, quantifying how well that cue performed in the last time step.
4. **Allocate resources:** based on the anticipated performance of the individual cues, allocates computational resources to maximise the quality of information obtained.

Table 3.2: Distillation algorithm.

to measure the probability of each hypothesis. The cue processor also determines metrics measuring the performance of each cue, along with the allocation of computational resources to individual cues. For each frame, the cue processor completes the steps listed in Table 3.2.

The *calculate cues* component of the system accepts requests for cue measurements and handles the requests using only the quantity of computational resources allocated to it by the *allocate resources* component.

Calculating and updating cues

Each i^{th} particle from the particle filter represents a hypothesis target location in the state space. A model of this hypothesized target is projected into the image, and each cue returns a set of probabilities $\{P(e_t^{(j)}|s_t^{(i)})$ for $i = 1...N\}$ indicating the j^{th} active cue's belief in the i^{th} hypothesis, where N is the total number of particles.

Calculating some cues may take longer than the time available between sequential frames. When the cue is not available to the *update cues* step it is skipped over until a future frame when the cue result is ready. As discussed in Section 3.2.1, these slow cues are accommodated by the *update PDF* component of the particle filter that is able to propagate their effect through to the probability values in the current frame.

The visual cues applied depend on the target being detected. In the face localization application skin colour, depth and symmetry cues are used for detecting a person's face in clutter, while Section 3.4 uses colour and edge cues for lane tracking.

Fusing cues

A crucial question when fusing sensor information is how to combine the probabilities obtained from different sensor modalities.

To make the fusion calculation tractable, we make the common assumption that the different cues are probabilistically independent. While this assumption is not strictly true across all cues, it is often true, and it allows us to fuse the cues via simple multiplication of probabilities (Kittler *et al.*, 1998; Nageswara, 2001). Subsequently, the probabilities from the cues are fused to determine the overall belief in the i^{th} hypothesis $P(e_t|s_t^{(i)})$ at time t as follows

$$P(e_t|s_t^{(i)}) = \prod_j P(e_t^{(j)}|s_t^{(i)})$$

The probability $P(e_t|s_t^{(i)})$ represents the probability of the evidence given the current hypothesis. A probability of zero implies that the cue knows for certain that the hypothesis is wrong. Since a cue can never be sure that a sensor is functioning correctly, a zero probability should never be returned from measured data. Only when cues encounter constraints such as physical impossibilities in the state space (like states behind the cameras) is a zero probability justified.

Quantifying cue performance

The performance, or *utility*, of each active cue is estimated every frame, and used to decide the distribution of computational resources across the cues.

Fusing the results of all available cues is assumed to give the best estimate of the true PDF $P(e_t|s_t)$ across the state space. So the performance of the j^{th} cue can be quantified by measuring how closely the cue's PDF $P(e_t^{(j)}|s_t)$ matches $P(e_t|s_t)$. This can be done using the relative entropy, or the Kullback-Leibler distance (Kullback and Leibler, 1951), an information theoretic measure of how accurate an approximation one PDF is to another, given by

$$\delta_t\left(P(e_t|s_t), P(e_t^{(j)}|s_t)\right) = \sum_i P(e_t|s_t^{(i)}) \log \frac{P(e_t|s_t^{(i)})}{P(e_t^{(j)}|s_t^{(i)})}$$

where s_t are the particle states at time t. This distance is non-negative. Two PDFs that differ greatly will have a large distance, while two PDFs that are identical will have a distance of zero. Soto and Khosla (2001) used this metric to rate the performance of their cues. Triesch and von der Malsburg (2000) considered it, but opted for a simpler *ad hoc* measure.

We use the Kullback-Leibler distance and define the utility of the j^{th} cue at time t as

$$u_t(j) = \frac{1}{\delta_t(P(e_t|s_t), P(e_t^{(j)}|s_t)) + \epsilon} \tag{3.2}$$

where ϵ is a small constant to ensure the denominator is not zero. A convenient value of ϵ is 1. With this value the Utility metric varies between 0 and 1. A perfect match between PDFs returns a Utility of 1.0. Any discrepancy between the two PDFs results in a Utility of less than 1.0, with a minimum bound of 0.0.

Resource allocation

The resource allocation component of the algorithm seeks to dynamically allocate computational resources to maximize the quality of information obtained per unit of computation. The quality of information is measured as the net *utility* of the cues computed, where the *utility* of each cue is determined using Equation 3.2.

Since our goal is to locate and track targets, and give timely feedback regarding the target's location, it is desirable to have at least some of the cues running at frame-rate. For this reason a certain proportion of the time available for cue processing each frame is devoted exclusively to cues running at frame-rate.

Slow cues are permitted to run once every 2, 4, 8 or 16 frames. However, the longer a cue takes to generate information, the less useful this information is for locating the target in the current frame. As with the cue probabilities, an exponential

discount factor $d \in (0, 1)$ is introduced to attenuate the utility measure of a cue for each frame it is late.

The resource allocation problem is posed as a scheduling problem, which can be solved as an optimisation problem. At each time step an optimisation problem is solved to generate the maximum overall utility given the time available, the cues available and the possible frame-rates. Since the optimisation problem is of a low dimension of discrete variables, an exhaustive search is used. The result is the desired frame-rate for each available cue. Since cues can share resources (such as edge images) the computational cost of each cue is decomposed into the cost of each resource required as well as the cost of evaluating the cue given the resources. Cues that use an expensive resource (such as a stereo disparity map) are likely to be scheduled to start at the same frame in order to share the cost of the computation.

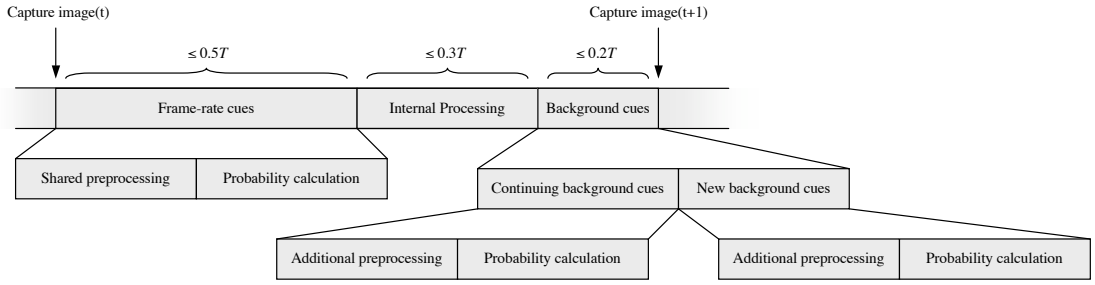


Figure 3.8: Resource time allocation during the period T of one image frame.

Figure 3.3 outlines the resource allocation process developed and shows the time allocation of a single time step. Time is divided between frame-rate cues, background cues and internal processing. Time allocated to the frame-rate cues is further divided into time for performing preprocessing and computing individual cues. This preprocessing is shared among all cues that are initiated in this frame, that is all frame-rate cues and all new background cues. The time allocated for background cues is divided into continuing background cues and newly scheduled background cues. Each of these is then divided into preprocessing and time required for calculating the probability values for the cues. Note that the additional preprocessing is only required if not performed already for the frame-rate cues.

Resource allocation algorithm:

Allocates resources to cues running at frame-rate:

1. Generates all combinations of cues that can be calculated in the time allocated for cues running at frame rate.
2. Chooses the combination with the best overall utility.

Allocates resources to cues running below frame-rate:

1. Calculates the amount of time remaining for computing slow cues in the current frame (taking into account that some resources may already be allocated to background cues that are still being computed from previous frames).
2. Determines all combinations of the remaining cues over all possible slower frame rates such that no combination exceeds the time available for the slower cues.
3. Calculates the net utility for each of these cue combinations using the discount factor to reduce the utility according to how late the cues are.
4. Chooses the combination of background cues offering the best overall utility.

Table 3.3: Resource allocation algorithm.

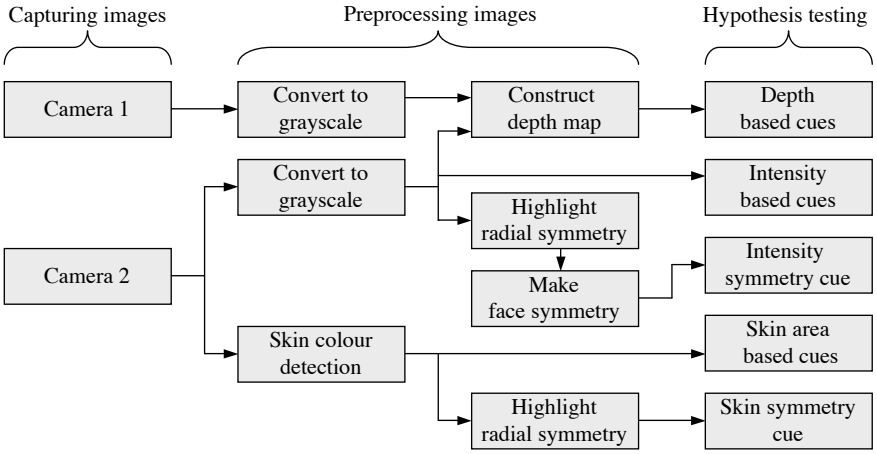


Figure 3.9: Person tracking application with multiple cues.

3.3 Person tracking example

We now apply our distillation method to an example application. Localising and tracking a person's face is a precursor to numerous human-computer interaction and surveillance tasks, such as tracking the head pose, face or expression recognition, and facial feature detection and tracking. Much research has focussed on detecting and tracking faces in images and video (Darrell *et al.*, 2000; Kim and Kim, 2000; Triesch and von der Malsburg, 2000; Crowley and Berard, 1997, e.g.). However, the search for a robust face localising and tracking method is far from over. We apply the vision system developed in this paper to this problem, and demonstrate localisation and tracking of a person's head in a cluttered environment, whilst dealing with changing head pose, occlusion and changing lighting conditions.

An implementation of the system was developed as an object orientated algorithm in Matlab^{TM}. To simulate realtime resource requirements, the computational cost of each cue was estimated from the Matlab^{TM}execution time required. Two uncelebrated colour stereo video cameras were used as sensors. The images from these cameras undergo some preprocessing and are then passed to the cues where each target location hypothesis is tested by computing all active cues. Figure 3.9 shows the sensing process when all cues are active.

3.3.1 Preprocessing

We now discuss preprocessing and hypothesis testing used for this example. Preprocessing is only performed *once* for each new set of images, whereas hypothesis testing requires one test for *every* target hypothesis generated by the particle filter. The preprocessing required for each frame is governed by the cues that are to be computed. These dependencies are illustrated by the network in Figure 3.9.

Figure 3.10: Preprocessing a colour stereo image pair. (a) Image from camera 1, (b) Image from camera 2, (c) Intensity image, (d) Radial symmetry image, (e) Facial symmetry image, (f) Depth map, (g) Skin colour likelihood image, (h) Radial symmetry of skin colour likelihood image searching for a radius of 15 pixels.

Figure 3.10 shows two colour 320×240 stereo images as received by the system's cameras, and the resulting outputs from preprocessing these images.

Depth Map: A dense depth map is generated using the approach of Kagami *et al.* (2000). The same method is used in our developed obstacle detection system. We reserve a detailed description of the algorithm until Chapter 4.

Skin Colour Detection: A skin colour likelihood image is generated from one channel of the stereo image stream (Loy *et al.*, 2002). The value of each pixel in this likelihood image is indicative of the probability that there is skin colour at that location in the original image.

Radial Symmetry: The radial symmetry operator developed by Loy and Zelinsky (2003) was used to highlight possible eye locations in the original grey-scale

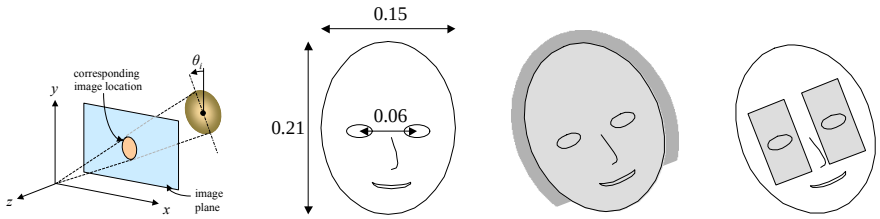


Figure 3.11: Generic head target and associated image regions. (a) Target is projected from state space into the image plane, (b) Dimensions (in metres) of generic head target in image plane, (c) Elliptical face region (light) and face boundary region (dark), (d) Search regions for integral projection.

image, and possible head locations in the skin colour likelihood image. Since this method is also used for road sign recognition we again reserve a detailed description of the algorithm until Chapter 5. The orientation-based variant of the transform was used in both cases. When applied to the skin colour likelihood image the transform highlights light regions that are approximately circular and of a similar diameter to a face, see for example Figure 3.10 (h). The operator was also applied to the intensity image to highlight small dark regions such as the eyes (Figure 3.10 (d)). This output is then convolved with a blurred annulus to highlight the regions between potential eye pairs. This second output is referred to, somewhat arbitrarily, as the *facial symmetry image* (Figure 3.10 (e)).

3.3.2 Visual cues

As stated in Section 3.2.2 at time t each cue returns a set of probabilities

$$\{P(e_t^{(i)}|s_t^{(j)}) \text{ for } j = 1...N\}$$

indicating the i^{th} active cue's belief in the j^{th} hypothesis (N is the total number of particles). The cues were chosen on the grounds of simplicity and efficiency. All cues use the head model dimensions shown in Figure 3.11(b). In the proceeding descriptions the *face region* and *face boundary* refer respectively to the light and dark grey regions in Figure 3.11(c).

Eye Location Cue: This cue uses integral projection to search the regions in Figure 3.11(d) of the intensity image for the darkest bands aligned with the lateral axis of the head.

Radially Symmetric Intensity Cue: The hypothesised depth of the target indicates which radius of facial symmetry should be used.

Radially Symmetric Eye Cue: The generic face model in Figure 3.11 is used

to extrapolate the hypothesised eye locations for the current target hypothesis.

Head Depth Cue: This cue checks to see if the hypothesised face region is at the appropriate depth.

Head Boundary Depth Cue: This cue measures whether the area surrounding the hypothesised head region is at a different depth from that of the hypothesis.

Elliptical Skin Region Cue: This cue indicates the likelihood that the hypothesised target region contains a large proportion of skin-like colour.

Skin Detector Cue: This cue detects targets that contain an instance of highly skin-like colour. It returns 0.5 if any of the pixels sampled in the face region had skin-like colour.

Non-skin Boundary Cue: Returns a high value if there are few skin colour pixels in the face boundary region.

Radially Symmetric Skin Cue: The target is expected to appear in the skin-likelihood image as an approximately round blob of a known radius, and the hypothesised target depth indicates this radius.

3.3.3 Performance

The performance of the system was demonstrated tracking a human face in two image sequences of a person moving around a cluttered environment with occlusions and lighting variation. Video files of the tracking sequences are available in the Appendix DVD-ROM (Page 257). Figure 3.12 shows some frames from the sequences. The blue dots indicate the projected locations of the hypothesised face centres. The hypothesis with the maximum likelihood is indicated as a green ellipse whose size and orientation indicate the hypothesised scale and orientation of the target. Likewise, the expected value calculated across all hypotheses is indicated by a red ellipse. Figure 3.14 shows a sample frame and its associated particle distributions.

Cues were dynamically scheduled to run once every 1, 2, 4 or 8 frames according to their calculated utility and computational cost.

Figure 3.13 shows the performance of two cues over the sequence. The Elliptical Skin Region Cue (SRC) is colour based, so affected by the ambient lighting, while the Eye Location Cue (ELC) is an edge based cue dependent on the contrast of the eyes against the face. In Figures 3.13(a) and 3.13(b) the thin black line represents the mean cue utility. The rate graphs plot the number of frames the cue was executed over. This is either 1,2,4 or 8 frames. Due to their different strengths and weaknesses the performance of the cues vary over the sequence. When a cue's performance drops, the cue is scheduled to run at a slower rate allowing other cues to run quicker. When the relative performance of a cue

(a) (b)

(c) (d)

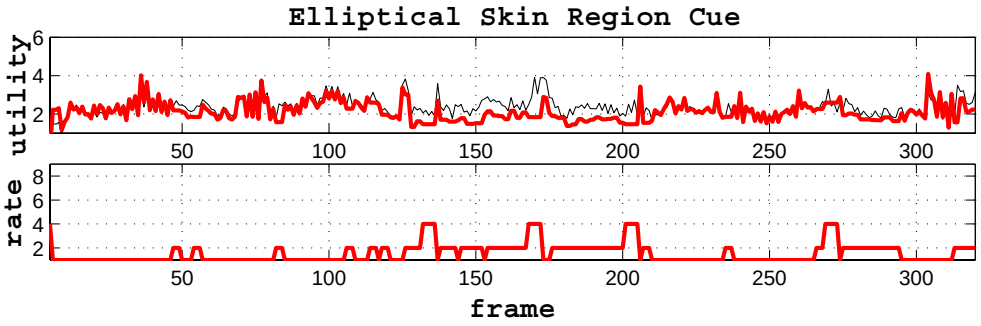
(e) (f)

Figure 3.12: Several frames in tracking sequence. Video files of the sequences are available in the Appendix DVD-ROM (Page 257).

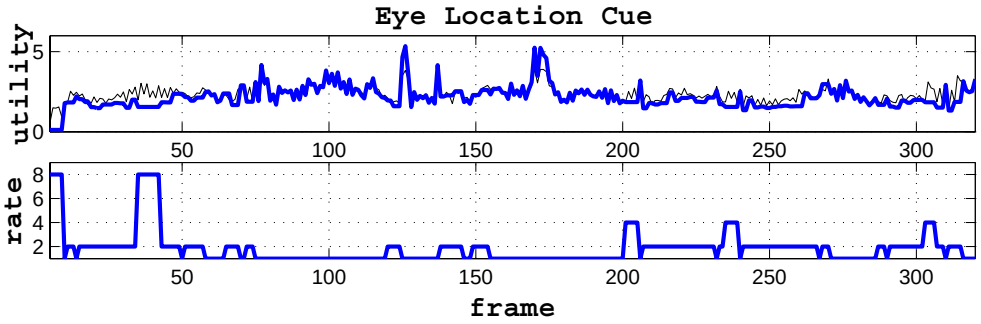
improves, the cue is allowed to run faster.

The simplicity of the cues means no single cue is able to reliably track the head in 3D space. However, by fusing multiple cues the ambiguity in the target location is reduced. Furthermore, by adaptively rescheduling the cues our system was able to enhance the tracking performance possible under a given resource constraint.

All real-world systems have a finite number of computational resources available.



(a)



(b)

Frame	Description
35-45:	Person is side on to the camera so the ELC performance drops and is scheduled to execute over a longer duration (Figure 3.12(b)).
125-200:	The lights are turned off. The SRC is less use and executed slower while the ELC is unaffected (Figure 3.12(c)).
200-260:	Person is drinking from a bottle occluding eyes so the ELC is performs worse while SRC fairs better (Figure 3.12(d)).
260-280:	Person's face is looking down close to the camera in shadow. SRC suffers, ELC unaffected (Figure 3.12(e)).
290-320:	Person is side on to camera walking out of shot. ELC affected, SRC okay (Figure 3.12(f)).

(c)

Figure 3.13: Performance of (a) the Elliptical Skin Region Cue (SRC) and (b) the Eye Location Cue (ELC). In the utility graphs the thin black line represents the mean cue utility. The rate graphs plot the number of frames the cue was executed over (either 1,2,4 or 8 frames). (c) describes what happens during the sequence.

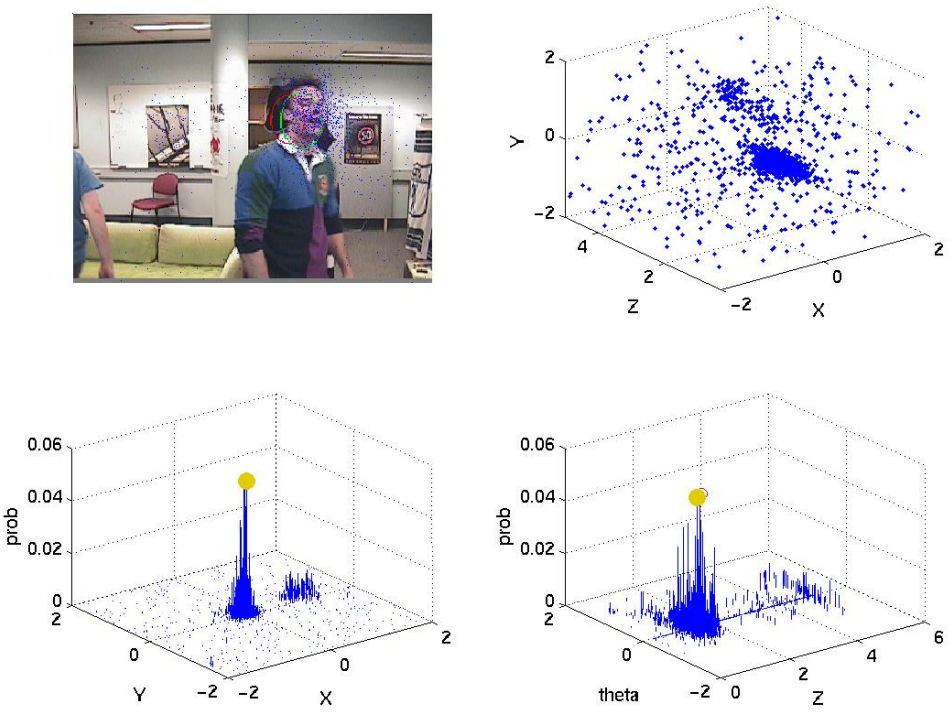


Figure 3.14: Frame in tracking sequence showing (clockwise) particles in image, in 3D space and particle distributions over x, y and z, θ states with the particle with maximum likelihood indicated by a yellow circle.

Scheduling resources to different cues according to their performance enables a system to aim for the best possible return per unit of computational resource. Without resource scheduling a system is still constrained to use only the computational resources available, yet is ignorant of how to alter computational expenditure to improve performance. Resource scheduling aims to increase the amount of useful information obtained per unit of computation, and - since there is only a finite amount of computation available for each image frame - increase the amount of useful information obtained from each frame.

3.4 Lane tracking application

We now apply the distillation algorithm to lane tracking. Lane position is one of the most important features in the road environment, and lane keeping is the key to safe driving. Therefore detection and tracking the lane is a key capability for an intelligent vehicle.

In the face tracking application we learnt that no single cue could track a feature all of the time under all environmental conditions. Using multiple cues was required to continuously and reliably localise the face.

In the lane tracking application, a single visual processing technique can be sufficient to track large sections of particular roads. In fact many successful lane trackers have employed single visual cue, single hypothesis tracking algorithms (Dickmanns and Graefe, 1988b,a; Bertgozzi and Broggi, 1998; Lee and Kwon, 2001).

Using multiple cues can enhance the stability of the tracking estimate, but more importantly for lane tracking, multiple cues allows the system to tolerate substantial changes in lane appearance. That is, multiple cues permit smooth tracking from high contrast lane markings to dirt roads. Multiple cues also permit tracking from stark daylight through to night driving or inclement weather. The tolerance of ambiguities provided by multiple hypothesis tracking is also invaluable for the difficult cases of lane tracking. Since single hypothesis trackers "place all their eggs in one basket", they are likely to eventually make an unfortunate choice, lose the target and require reinitialisation. Though groups have automated the reinitialisation function (Pomerleau, 1995; Bertgozzi and Broggi, 1998; Dickmanns, 1999a), the road context is still lost and tracking is disrupted.

3.4.1 Review of lane tracking techniques

Computer vision based lane tracking for autonomous vehicles and driver assistance has shown much promise. In the late 1980s Dickmanns and Graefe (1988b,a) pioneered the 4D approach to lane tracking in the PROMETHEUS project. A detailed dynamical model of the vehicle was used with edge based feature detectors and an extended Kalman filter to localise and track the road over time. The approach was very effective, however, since edge based feature detection was the primary means of lane extraction, the system was susceptible to dramatic changes in illumination and shadows across the road. The use of extended Kalman filters also limited the tracker to support only a single lane hypothesis, and ambiguities such as new and old lane markings could fool the tracking and require (automated) reinitialisation.

Groups that used algorithms solely based on lane markings include UBM, Ohio State University and Fraunhofer - IITB. These groups either used assumptions such as brighter lane markings on darker road surfaces or edge detection to get

the lane boundaries. The UBM group used lane marking tracking in the near field with image kernels tuned to find a lane edge at a sensible orientation. The search regions for the lane markings were governed by the higher level logic which is estimating the state of the vehicle and the road curvature using a piecewise clothoid model in lateral and longitudinal directions. In the far field, telephoto lenses tracked the road as a whole to refine the road curvature ahead. The vehicle state model consisted of the lateral road offset, yaw drift, change in yaw and change in steering angle (Dickmanns, 1999a).

The NAVLAB project at Carnegie Mellon University (Thorpe, 1990b) explored many different lane detection algorithms (Crisman and Thorpe, 1993a; Kluge and Thorpe, 1992; Baluja, 1996; Jochem *et al.*, 1993; Pomerleau, 1995). Rather than explicitly extracting lane markings, this group favoured whole-image techniques and machine learning to determine the lane direction. These methods were tolerant of many types of road appearance, and capable of tracking marked or unmarked roads with the same systems (although online retraining is needed)(Thorpe, 1990a). The RALPH system transformed a trapezoidal region in front of the vehicle into an aerial view. Hypothetical road curvatures were then applied to the aerial view to "undo" the curvature and a summation of the image columns performed. The correct road curvature would straighten the road image and make the intensity profile the sharpest. Again lane markings were not explicitly required. Any feature parallel to the road would be *sharpened* by the correct hypothetical curvature. The system uses a clothiod model in the lateral direction and assumes a flat plane in the longitudinal direction (Pomerleau, 1995).

The General Obstacle and Lane Detection system (GOLD (Bertgozzi and Broggi, 1998)) used in the ARGO vehicle at the University of Parma transformed a calibrated stereo image pair into a common aerial view and used a pattern matching technique to detect lane markings on the road. Similar to the UBM system, GOLD was shown to be effective on long driving trials on well marked roads, but again suffered from using only one approach to find lane markings. The algorithm used a trapezoidal lane model in birds' eye view of the road scene. The trapezoid accounts for inclines in the longitudinal road profile which would otherwise be parallel lines. Lane markings were recovered by intensity image thresholding.

Recently, McCall and Trivedi (2006b) used steerable filters for lane marking detection of various marking types combined with road texture detection with Kalman filter tracking.

There have been few research efforts into lane tracking that implicitly handles multiple lane position hypotheses. Lane tracking under good conditions is highly unimodal, leading to a tendency toward Kalman filter based implementations. One exception is Southall and Taylor (2001), who used a particle filter to track lane markings using edge detection. The group was able to demonstrate tracking across different modes of the lane position in the state space caused by changing lanes. Multiple lanes is a good example to differentiate between the two tracking algorithms. Kalman filter based lane trackers need distinct logic to handle lane

change events, Particle filters can switch between modes representing each lane with no explicit logic.

We apply the generic Distillation tracking algorithm outlined in Section 3.2 to implement a lane tracking system able to combine multiple visual cues of the style proven to be effective in prior lane tracking research with a framework capable of evaluating and favouring different cues based on the given road appearance. Figure 3.15 shows the substantial variation in the road geometry and appearance, even on arterial routes. Intersections, obstacles and curvature beyond the camera field of view have been excluded yet still there is substantial variation in the images.

3.4.2 Implementation

Our lane tracking system estimates the state of the vehicle relative to the road. The state space and coordinate system used is a subset of the model used by Dickmanns and Mysliwetz (1992). Since we are using particle filtering the core tracking method as opposed to Kalman filtering there is a much greater tolerance of modelling errors and non-Gaussian noise than Kalman filtering. This permits us to use a minimal subset of state variables. Also the UBM group were planning on going very fast ($> 160km/h$) so a large lookahead distance was required. To manage the lookahead distance more parameters were introduced to improve model accuracy.

With reference to Figure 3.17, the lateral offset of the vehicle, y, the yaw of the vehicle, ϕ (with respect to the centre line of the road), and the lane width, w are estimated.

The Distillation algorithm used in the people tracking application was ported to C/C++. The generic algorithm was written on the shared class structure defined in Section A.2. The Lane tracking code inherits abstract Distillation algorithm class objects and extends them as needed for the application. The final application was compiled and executed on one of the processing PCs in our test vehicle (Section A.1.2).

The *drift* and *diffusion* steps in the particle filter in this case must account for the vehicle motion between each iteration of the filter and for the errors inherent in the measurement of the motion. The motion is modelled as a shift in the particles according to an Ackermann motion model (Dixon, 1991) combined with a normal random dispersion. The cues used to track the lanes are primarily image based with some additional cues representing simple state based constraints. The image based cues share a significant amount of preprocessing including edge and colour probability maps, as well as state space to image space line projections.

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 3.15: The wide variety of road conditions. Even with intersections, obstacles and high curvature omitted.

3.4.3 Preprocessing

When preprocessing the raw data from the sensors (i.e., the image streams from the cameras), it is necessary to transform the data into the form that is used for evaluating the sensor model of each cue. The distillation algorithm uses a preprocessing method that ensures that the observations e_t required by the cues are calculated once only, regardless of the number of cues that use the observation. Figure 3.16 shows an example of the type of data that is shared among cues in the lane tracking system.

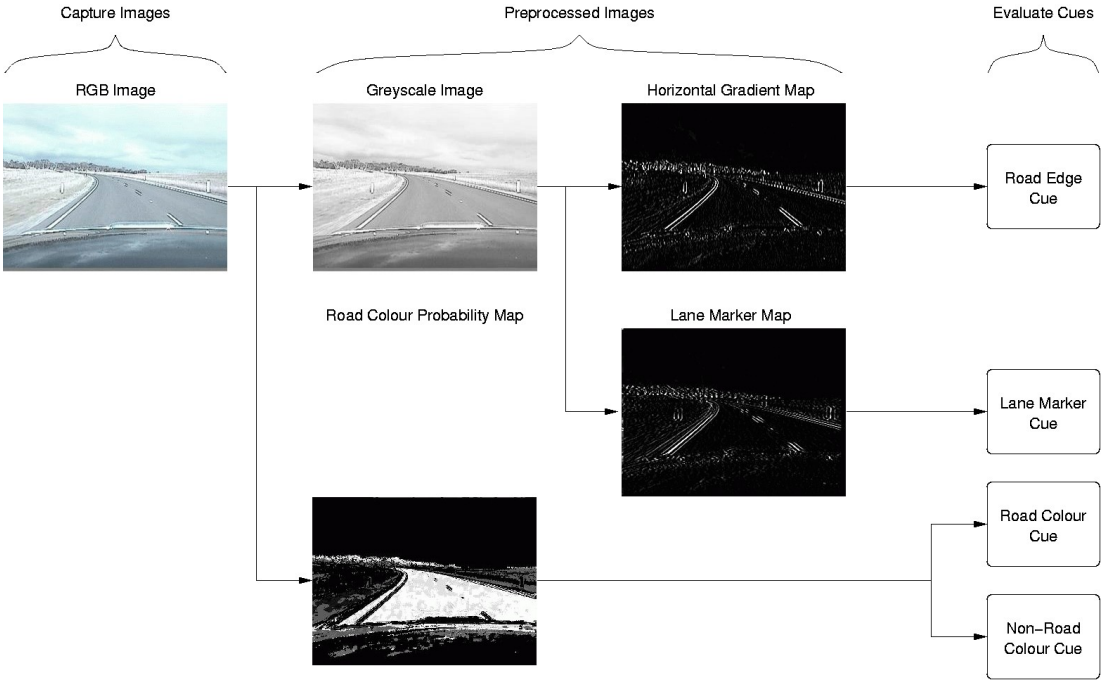


Figure 3.16: An example of the shared observation preprocessing for four cues of the lane tracker. Both the road colour cue and non road colour cue share the road colour probability map while the road edge cue and the lane marker cue share the greyscale image.

3.4.4 Visual cues

The cues chosen for this application were designed to be simple and efficient while covering a variety of road types. The evaluation of the posterior for the j^{th} cue is a direct application of the sensor model, $P(e_t^{(j)}|s_t)$ on the particles passed to it from the particle filter. Two different classes of cues are used in the lane tracker: image based and state based.

Image based cues use the image streams as the observation measurements. These cues depend on the road regions and their transformations into image space as

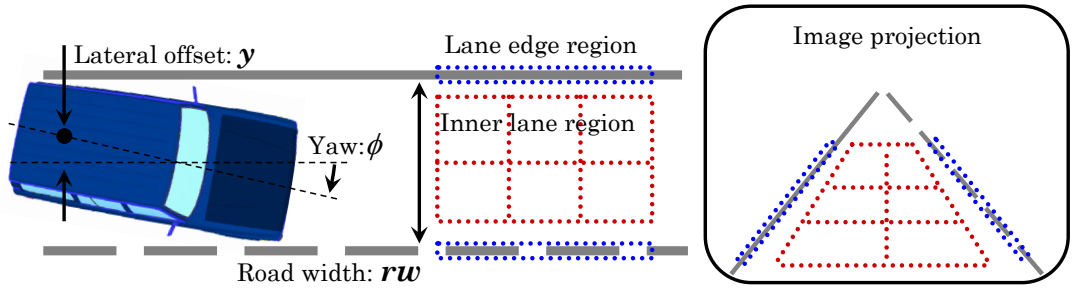


Figure 3.17: State model for lane tracking. The vehicle position is specified by the lateral offset y and the vehicle yaw ϕ. The road geometry is determined by the lane width w. These variables are used to specify a lane edge region and an inner lane region for each particle. The lane edge and inner lane regions are sampled from the preprocessed images for the evaluation of the sensor model for each image based cue. The regions from the road-centric coordinate system are transformed into pixels S in the image plane where they are tested against the preprocessed images of each cue.

shown in Figure 3.17. Each hypothesis from the particle filter produces a road model defined by the pixels S in image space, which are sampled from the preprocessed images to evaluate the sensor model. The following four image based cues use Equation 3.3 to evaluate their sensor model.

$$P(e_t^{(j)}|s_t^i) = \frac{1}{\epsilon + N}\left(\epsilon + \sum_p^N I_t^{(j)}(s_p^{(i)})\right) \tag{3.3}$$

$s_p^{(i)}$ is the p^{th} pixel from the set of pixels S generated by particle i. $I_t^{(j)}(s)$ is the value of pixel s from the observation image I used of the j^{th} cue. N is the number of pixels in the region in image space. ϵ (set to 0.001) is used to support the possibility that the sensor is in error (as discussed in Section 3.2.2).

Lane Marker Cue: This cue is suited to detecting roads that have lane markings. It uses an approximation of a 1D Laplacian of Gaussian (LOG) kernel (see Figure 3.18) correlated across the intensity image of the road to emphasise vertical "bar" like features to produce the lane marker map, I. Figure 3.19(a) shows likely lane markings with the lane hypotheses below.

Road Edge Cue: This cue is suited to roads with lane markings or defined unmarked edges. It uses a Canny edge map (Canny, 1986) to highlight road boundaries and lane marker edges (I) and the road edge region from Figure 3.17 to evaluate the sensor model. Figure 3.19(b) shows potential road edges with the lane hypotheses below.

Road Colour Cue: This cue is useful for roads that have a different colour than

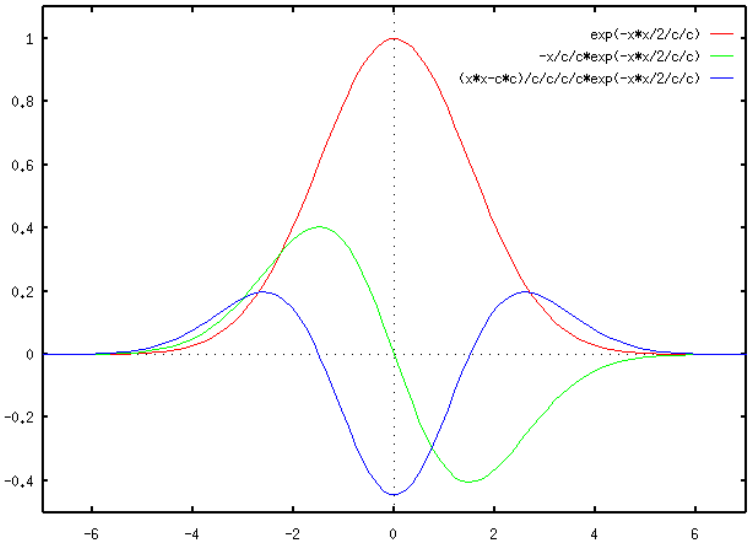


Figure 3.18: 1D Gaussian derivatives. **red:** Gaussian kernel. **green:** 1st derivative of Gaussian kernel. **blue:** 2st derivative of Gaussian kernel known as Laplacian of Gaussian kernel.

their surroundings (both unmarked and marked roads). It returns the average pixel value in the hypothesised inner lane region from a colour probability map, I, that is dynamically generated each iteration using the estimated road parameters and the YUV colour image from the previous iteration. The colour probability map is generated using a method based on Cai and Goshtasby (1999). Figure 3.19(c) shows potential road colour probability map with the lane hypotheses below.

Non-Road Colour Cue: This cue is used to evaluate non-road regions in the road colour probability map described above. It tests that the area just outside of the lane edge region of each hypothesis lie in regions of low road colour probability.

State based cues use the state represented by each particle as the observation measurement. These were introduced as a heuristic to ensure a result concordant with the physical constraints of the model.

Lane Width Cue: This cue is a state based cue that is particularly useful on multi-lane roads where it is possible for the other cues to see two or more lanes as one. It returns a value from a Gaussian function centred at a desired lane width. The mean road width used in this cue was 3.6 m which was empirically determined from previous lane tracking experiments to be the average road width. The standard deviation can be used to control the strength of this cue. A value of 3 m was typically used, allowing the road width cue only a small influence on the posterior.

Elastic Lane Cue: This cue is another state based cue that is used to move particles toward the lane that the vehicle is in. The cue simply returns 1 if the

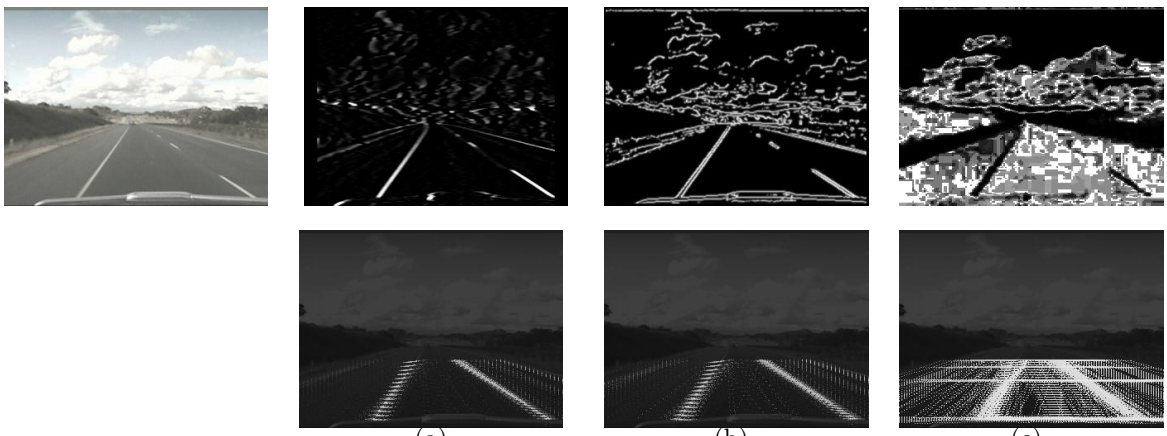


Figure 3.19: Image maps used by the sensor model to evaluate the conditional density. (a) the lane marker map. (b) the lane edge map. (c) the road colour map. *Bottom Row:* Potential lane hypotheses, lighter lines correlate to a higher probability.

lateral offset of the vehicle is less than half of the lane width and 0.5 otherwise. This cue represents our preference for the lane closest to the vehicle to be tracked as opposed to adjacent lanes.

3.4.5 Performance

The lane tracker was tested in a number of different scenarios including varying lighting, road curvature and lane markings. Figure 3.20 shows sample frames from different road conditions. The lane tracker quickly converges to the dominant road hypothesis (see Figure 3.21).

The system successfully handled shadows across the road, departure lanes and miscellaneous markings on the road that can often confuse lane trackers. The lane tracker was able to handle extremely harsh shadows present on the road surface as well as dramatic changes in lighting from overhead bridges. The tracking performance was slightly worse on high curvature roads as they were a violation of the straight road model initially used.

One of the most impressive characteristics of the particle filter is its proficiency for target *detection*. While many methods have separate procedures for bootstrapping the tracking process, the particle filter seamlessly moves from detection to tracking without any additional computation required for the detection phase. The convergence of the dominant mode of the particles to the location of the road takes approximately five iterations while the best estimate of the road location is found within two iterations.

Typical behaviour of the cue scheduling algorithm is presented in Figure 3.22. In

Figure 3.20: Results from the lane tracker on various roads.

this case, the two colour cues are scheduled into the foreground at iteration 1920, while the Lane Edge Cue is scheduled into the background. This occurs as the road shoulder colour becomes more distinct from the road, the combined utility of the two colour cues increases to a value greater than the Lane Edge Cue utility at iteration 1919. All three are not scheduled to run in the foreground due to the processing time constraints outlined in Section 3.2.2.

The lane tracker was found to work robustly with respect to the problems typically associated with lane tracking. This can be attributed to the combination of particle filtering and cue fusion. Because of the particle filter, cues only have to validate a hypothesis and do not have to search for the road. This implicitly incorporates a number of *a priori* constraints into the system (such as road edges meeting at the vanishing point in image space and the edges lying in the road plane) which assist it in its detection task.

Cue fusion was found to dramatically increase the robustness of the algorithm due to the variety of conditions the cues were suited to. The initial set of cues was limited to image based cues that contained no prior information except for the road model described in Section 3.4.4. The particle filter often converged to lane segments that the vehicle was not in or to the whole road instead of a single lane. This was due to the lane markings and edges of the road having stronger signals than the lane markings separating the lane. The addition of the two heuristic cues (Road Width Cue and Elastic Lane Cue) was an effective solution

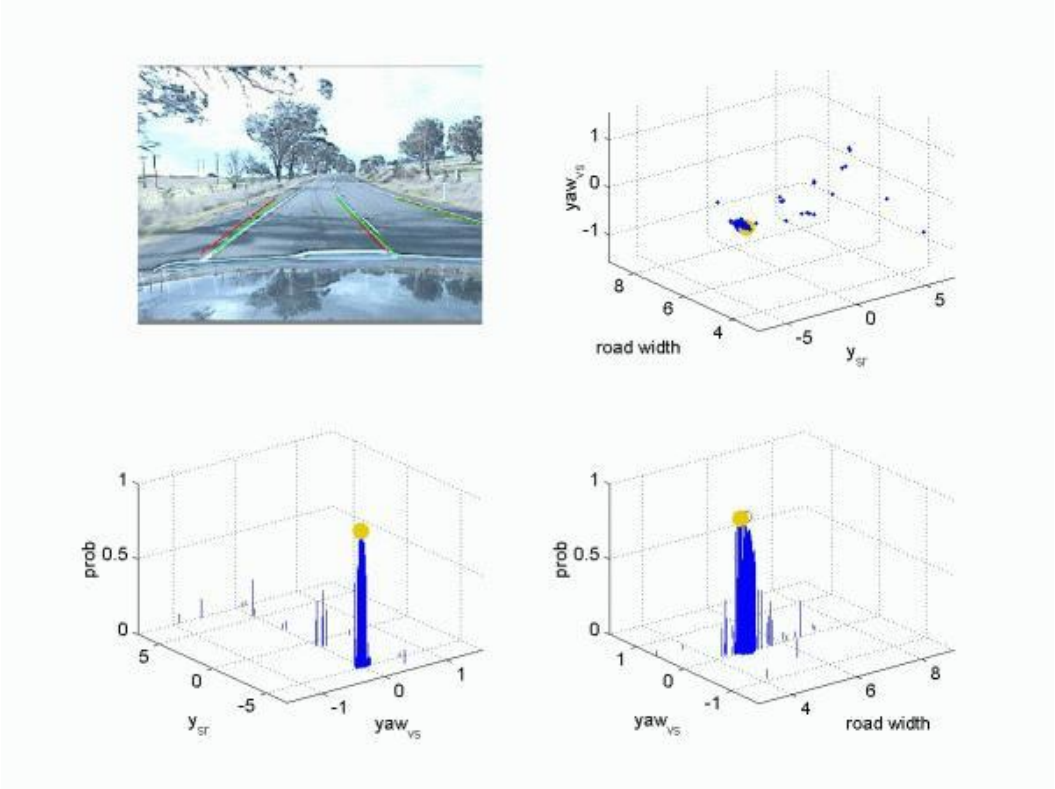


Figure 3.21: Frame in tracking sequence showing (clockwise) lanes in image, particles in 3D space and particle distributions over yaw, lateral offset and lane width with the particle with maximum likelihood indicated by a yellow circle.

to this problem. A video of the straight lane tracking system is on the Appendix DVD-ROM (Page 257).

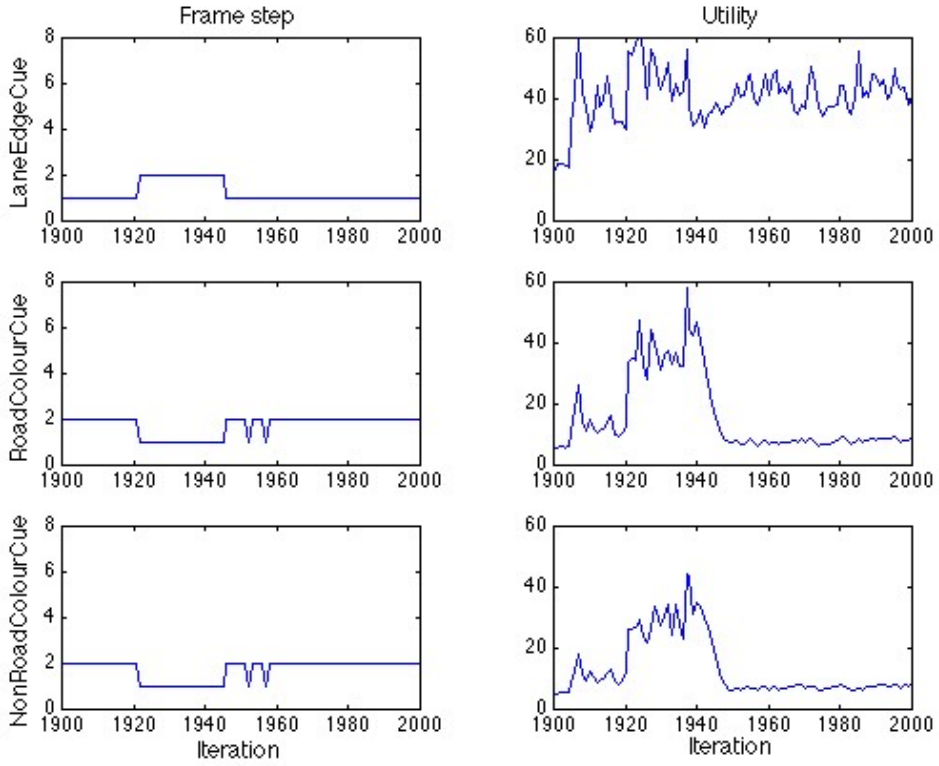


Figure 3.22: The utility of the two colour cues increases between iterations 1920 and 1945 and they are both scheduled into the foreground cue list, while the Lane Edge Cue is removed. The total utility over this range has increased due to the addition of the extra cue into the foreground list.

3.5 Robust lane tracking

The original lane tracking system implemented by Apostoloff and Zelinsky (2004) was based on a simple straight road model and fixed lookahead. Our work extends this system to permit a number of beneficial features. The software implementation was expanded to use multi-threading. Cues were run in parallel across 4 processors (A dual processor machine with hyper-threading). All completed cues were combined during the next particle filter cycle. One of the advantages of this configuration was that no margin was needed in the cue scheduling to allow for cue processing overruns.

3.5.1 Lateral curvature

In our research the lane tracking system has also been augmented to use a 2nd order horizontal clothoid road curvature model again a subset of (Dickmanns and Mysliwetz, 1992). The road model was defined by the following equations which are a reasonable approximation to a clothoid curve for small curvatures:

$$x \approx x_0 + l \tag{3.4}$$

$$y \approx y_0 + \frac{C_0 l^2}{2} + \frac{C_1 l^3}{6} \tag{3.5}$$

Where C_0 is the curvature, C_1 is the change of curvature and l is the arc length.

The drift or motion model is augmented to become:

$$y_{sr}(k+1) = y_{sr}(k) + sin(\phi_{vs(k)}).dl \tag{3.6}$$

$$C_0(k+1) = C_0(k) + C_1(k).dl \tag{3.7}$$

Where $y_{sr}(k)$ is the lateral offset, $\phi_{vs}(k)$ is the vehicle yaw, $C_0(k)$ is the curvature, $C_1(k)$ is the change of curvature and dl is the arc length travelled between filter interactions k to $k+1$ computed from the odometry.

Interestingly, curvature is added to the original state space moving from the original three states (lateral offset, yaw and road width) to five states lateral offset, yaw, road width, instantaneous curvature, change in instantaneous curvature). It was proposed by Apostoloff (2005) that horizonal and vertical road curvature could be estimated with a second distillation algorithm stage building on the straight road estimate. Though feasible, this approach introduces some additional complexities. As the origin of the coordinate system is at the centre of gravity of the vehicle, estimates of the curvature in front of the vehicle affect the estimated lateral offset and yaw between road models. A transformation is required from the lateral offset and yaw from the straight road model to the curved road model. More critically, since the straight lane model is a tangential approximation to the curved road and the road is viewed at a minimum of several metres in front of the vehicle, for roads of non-trivial curvature, there is no guarantee of where

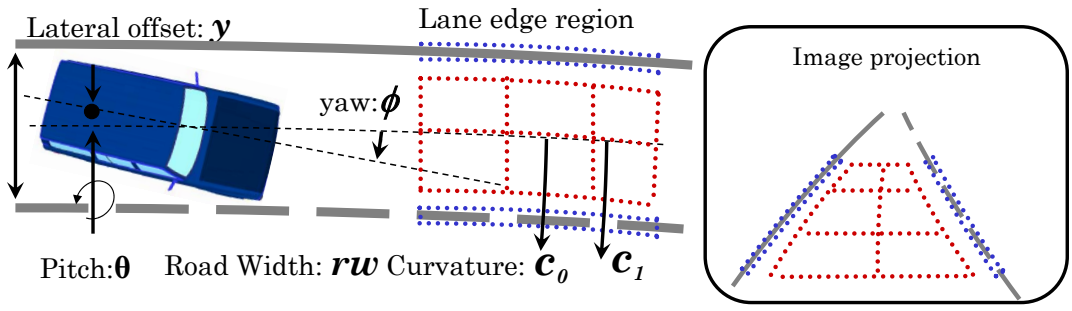


Figure 3.23: Augmented road model. Curvature C_0 and C_1 and pitch estimation added to lateral offset(y), yaw (ϕ) and road width (rw).

exactly the tangent will be taken. The tangent attempts to fit best the projected curved road. That is, due to the road appearance and the position of the vehicle with respect to the road, the straight lane model may approximate the tangent entering a curve, the middle of the curve or the line leading out of the curve.

Figure 3.24 illustrates the benefit of the horizonal clothoidal model over the straight lane model. Since the lane tracking problem is highly modal, the additional dimensions to the state space are accommodated with a moderate increase in particles. Finally, a sixth dimension was added to the state space - namely, pitch. Unlike the clothiod horizontal curves plotted in road design, longitudinally the road way is often a graded version of the natural lie of the landscape (Underwood, 1991). Acceleration and deceleration of the vehicle can also cause significant pitch to occur. By estimating the pitch of the vehicle about the centre of gravity, the shifts due to acceleration changes can be accounted for as well as a bias in the near-field due to a changing grade. Figure 3.23 shows the new lane tracking state space. Referring back to Figure 3.15, it is easy to validate this conclusion. The variance in the vehicle pitch can be detected by comparing the horizons in neighbouring images. Including the vehicle pitch in the state estimation is therefore clearly warranted.

3.5.2 Variable look-ahead

In our research we add a confidence measure to vary the look-ahead distance. When the variance of the primary road state variables (lateral offset, vehicle yaw and lane width) increase beyond a small tolerance the look-ahead distance was reduced to concentrate on robustly tracking the road directly in front of the vehicle at the expense of the far-field. Equation3.5.2 shows how the look-ahead is adjusted. The tolerances (tol_{xx}) are set to 100× the gaussian noise variance used in the particle filter "drift" phase. As the near-field estimate converges the look-ahead distance is incrementally increased. Figure 3.25 illustrates how the

(a)

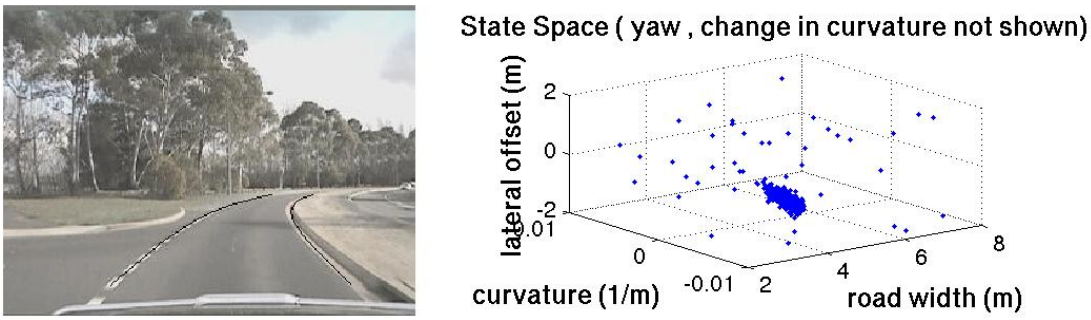
(b)

State:	**Error:**
Image:	*up to* 16*pixels*
Lateral Offset:	0.4*metres*
Heading:	5°
Road Width:	0.2*metres*
Road Tangent Lateral Offset:	1.5*metres*
Road Tangent Longitudinal Offset:	8*metres*

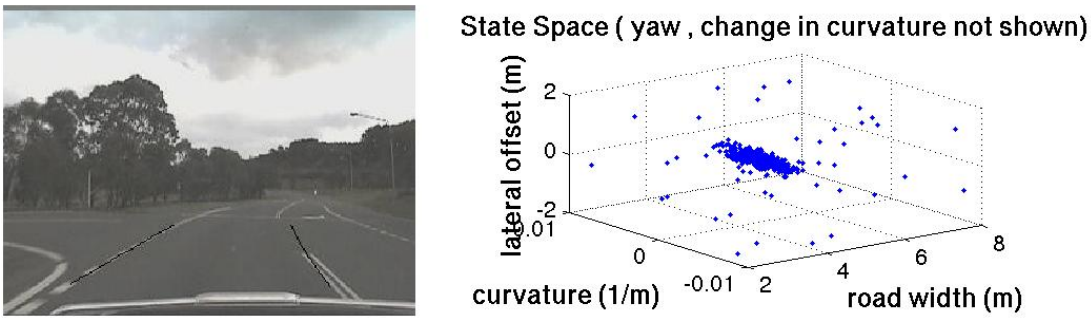
(c)

Figure 3.24: Sample frame comparing straight and curved road model implementations. **(a):** Straight lane model. **(b):** Curved lane model. **(c):** Table listing error in state variables due to unmodelled curvature.

look-ahead distance varies with the spread of the particles. Figure 3.26 demonstrates how an unmodelled change in the road scene is handled. Approaching an intersection the road widens. The state model assumed a constant road width so to accommodate the change in road width the particles "jump" to a better region of the state space. Since the particle spread has expanded, the look-ahead



(a) Long look ahead



(b) Short look ahead

Figure 3.25: Lane tracking look ahead distance varies with certainty measured by the variance or sample spread of the dominant mode.

distance is reduced to concentrate on improving the near field estimate.

$$l_{i+1} = l_i \begin{cases} +k, & var(y_i) < tol_{y_1} \\ & var(yaw_i) < tol_{yaw_1} \\ & var(rw_i) < tol_{rw1} \\ -4k, & var(y_i) > tol_{y_2} \\ & var(yaw_i) > tol_{yaw_2} \\ & var(rw_i) > tol_{rw2} \\ +0, & Otherwise. \end{cases} \tag{3.8}$$

where the look-ahead l_{i+1} is $15 < l_{i+1} < 55$ metres.

3.5.3 Supplementary views

In our research, to demonstrate the ability to integrate cues from different sensors, a second camera of a longer focal length was added. This camera enabled better road curvature estimation by zooming in on the distant road. A second set of the image cues the same as those discussed in section 3.4.4 were added for this second camera but otherwise no changes were made to the lane tracking algorithm. The

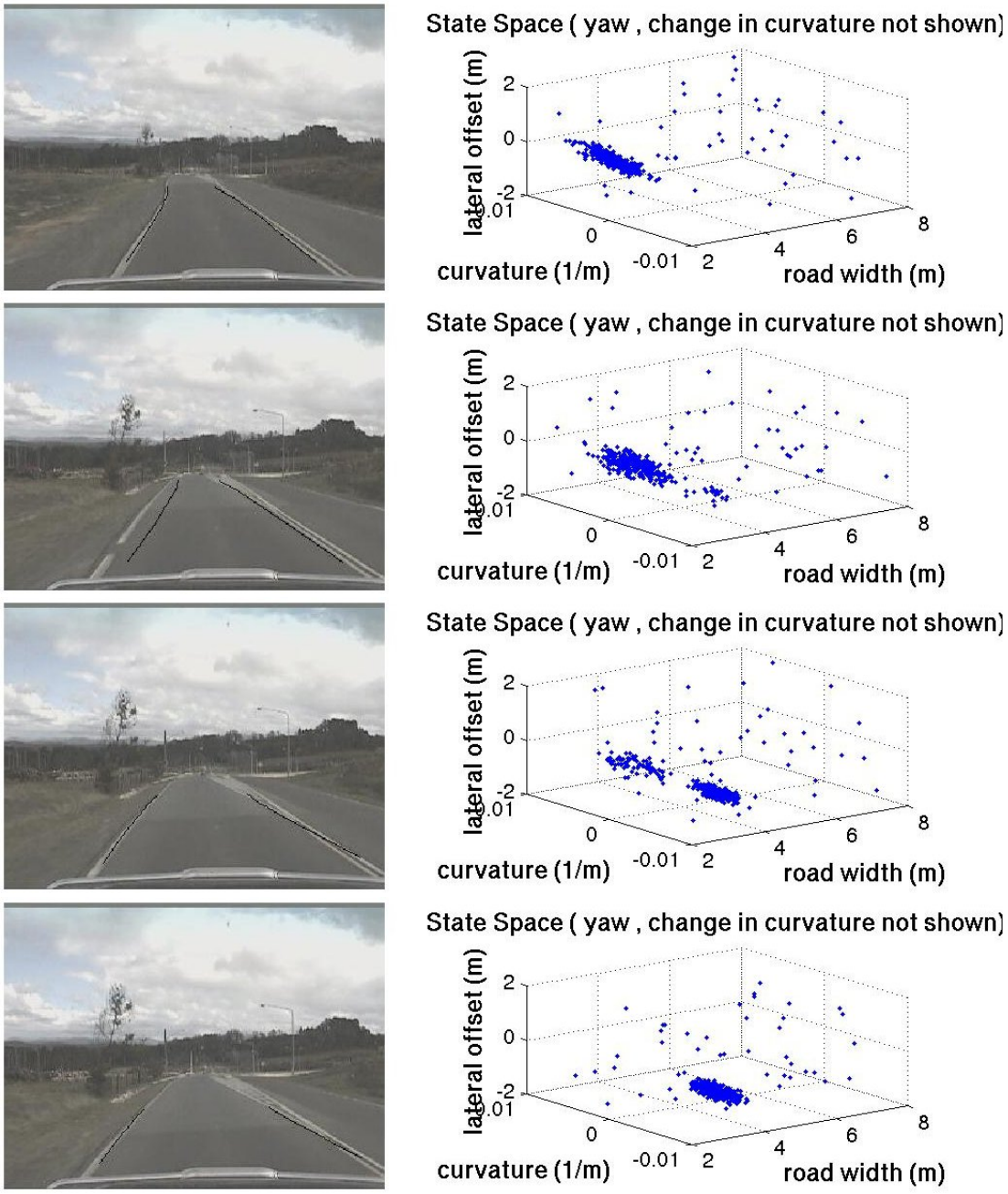


Figure 3.26: A particle mode change and shortened look-ahead to handle a widening road. The video file showing this sequence is available in the Appendix DVD-ROM (Page 257).

second camera improved the lane state estimation, particularly the curvature estimate (see Figure 3.27). There is, of course, an exception to every rule (as shown in the bottom pair of images): strong shadows and faint lane markings can cause a far-field curve misestimate. Though the curvature is wrong, the near-field estimate is still useful. The dominant hypothesis quickly corrects as uncertainty shortens the look-ahead distance and the evidence for this alternate

Figure 3.27: Views from the original and supplementary cameras with current lane hypothesis overlaid. **(a):** Original wide-angle camera. **(b):** Supplementary zoomed camera.

route evaporates.

3.5.4 Curvature verification

As a verification of the lane tracker accuracy in estimating the lane position and curvature, a comparison was conducted between the lane tracker and logged GPS data. Figure 3.28 compares the change in curvature of the GPS data with the lane tracking estimate. In general there is strong agreement between the estimates. Curves and turning points correspond well between the graphs. The lane estimate seems to exhibit a bias toward the end of the graph. From the image data it can be seen that the road undulates toward the end of the sequence, violating the near-field flat road assumption in the road model. The unmodelled property increases uncertainty in the lane estimate, reducing the look-ahead distance and thereby the curvature estimation ability. The video file showing this is available in the Appendix DVD-ROM (Page 257).

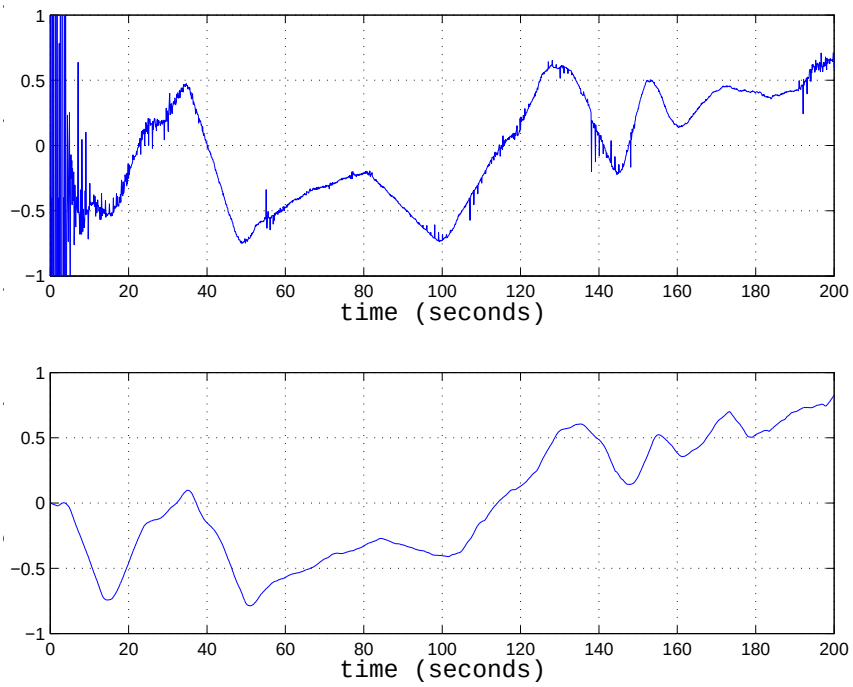
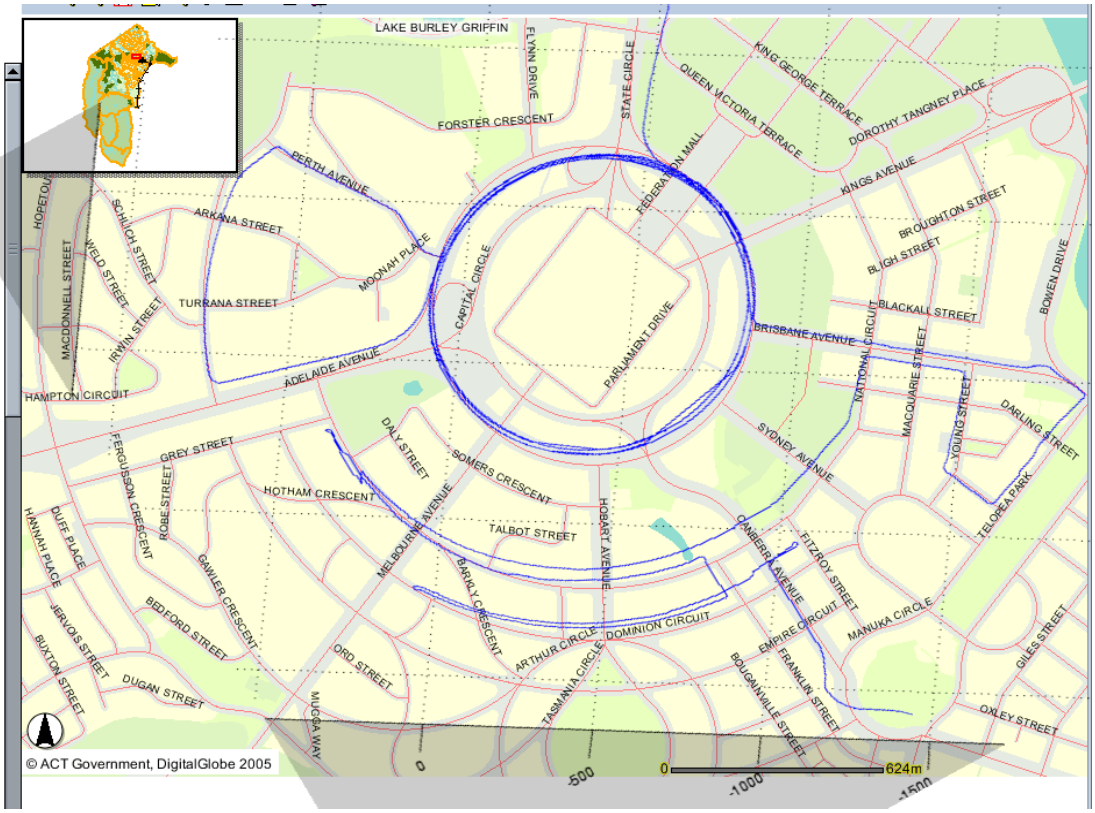
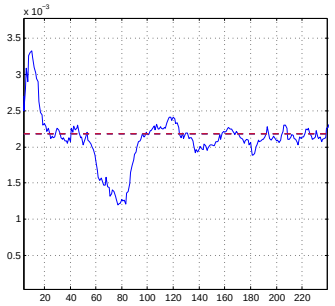


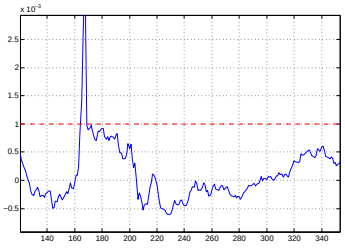
Figure 3.28: Comparison of lane tracking curvature with GPS. *Top:* Change in bearing from GPS (radians/sec). *Bottom:* Estimated change in curvature from lane tracker (radians/sec).



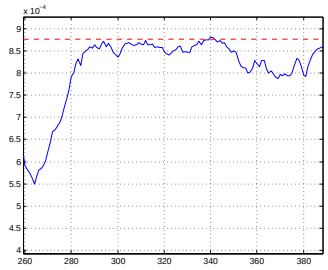
(a)



(b) State Circle



(c) National Circuit



(d) Dominion Circuit

Road:	**Map Curvature (Radius):**	**Mean:**	**Std. dev.:**
State Circle:	0.00218(458m)	0.0021	0.00013
National Circuit:	0.00103(972m)	0.00095	0.0006
Dominion Circuit:	0.00088(1135m)	0.00086	0.00001

(e)

Figure 3.29: Circular road trial. **(a):** Map of circular roads with GPS route overlaid. **(e):** Average curvature estimates for **(b)**, **(c)** and **(d)**.

3.5.5 Road trial

Finally, to demonstrate the robustness of the lane tracker, handicam data from a 1800km round trip from Canberra to Geelong was processed. The route is shown in Figure 3.31, the first leg was a down a coastal highway, the return leg was on an inland motorway. Figure 3.32 shows periodic images from one leg of the trip. The video files from the trip are available in the Appendix DVD-ROM (Page 257). The lane tracker was able to track the road for 96.2% of the journey. A significant portion of the cases were the state estimate in error were sections of road with very tight curvature (see Figure 3.30 mainly the winding roads over the Great Dividing Range) , where the road was out of the camera's field of view. The second significant cause was waiting at intersections where there were no lane markings in the near field (for example obscured by the vehicle ahead). Other issues are those already mentioned and beyond the scope by this work, such as the dynamic range, sun glare and the resolution-field of view trade off of cameras. On all occasions the lane estimate recovered and corrected over time.

(a) (b)

Figure 3.30: Hard to track sections.

Figure 3.31: Route for extended video trial. The first leg of the journey is from Canberra to Geelong via a coastal highway (Route 1). The return trip is via an inland motorway (Route M31).

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 3.32: Periodic lane images from extended road footage.

3.6 Summary

This chapter has presented the Distillation Algorithm as a new approach to target tracking: a vision system that distills multiple visual cues to robustly track a target.We use a particle filter to maintain multiple hypotheses of the target location and facilitate cues running at different rates, while Bayesian probability theory provides the framework for sensor fusion. The uniqueness of our system lies in its ability to schedule resources over the suite of available cues. Cues are run frequently or infrequently depending on the usefulness of the information they are providing and the amount of computational resource they require. Keeping short time histories of each hypothesis in the particle filter enables the system to merge information from cues running at different frequencies.

The versatility of the system was demonstrated by applying it to lane tracking. Our lane tracker was able to combine multiple visual cues of the style proven to be effective in prior lane tracking research with a framework capable of evaluating and favouring different cues based on the given road conditions. Our earlier work in lane tracking was enhanced to include road curvature estimation, multiple views and variable look-ahead without significant changes to the underlying "Distillation algorithm". The efficacy of the lane tracker was validated on 1800 kilometres of road footage and found to track effectively 96.2% of the journey.

Our distillation algorithm is also integrated into an obstacle detection and tracking system. We describe this system in the next chapter.

Chapter 4

Obstacle detection and tracking

In this thesis we are propose that an Automated Co-driver through driver inattention detection can reduce road fatalities. Road obstacles provide a substantial threat to the driver.

In this chapter we develop an algorithm to detect and track road obstacles. An unknown number of obstacles may be present in the road scene, this is unlike lane tracking where a target tracking algorithm was sufficient, obstacle detection requires an explicit detection phase in addition to target tracking algorithms.

Our algorithm explicitly detects potential obstacles by image analysis. The detection phase is designed to robustly detect obstacles (a low false negative rate) the consequence is a high number of phantom detections (high false positive rate). In our system the distillation algorithm serves the role of an obstacle candidate incubator. The results of the detection phase are injected as potential obstacle hypothesises into the Distillation algorithm. The for true obstacles the Distillation algorithm consolidates hypothesises to track the obstacle. For phantom detections the potential obstacle hypothesises dissipate. When an obstacle has been reliably tracked in the Distillation algorithm, the obstacle can be tracked separately with an extended Kalman filter. The system we have developed enables obstacle detection techniques susceptible to high false positives to still provide value in robust obstacle detection and tracking.

The combination of "bottom-up" explicit detection and "top-down" hypothesis testing using the Distillation algorithm has parallels to the human vision system where low-level vision functions such as stereo disparity and image motion estimation happen without significant feedback from the higher brain where as the later stages of the visual path are dominated by higher brain feedback (Mallot, 2000).

This chapter begins, in Section 4.1, with a review of road obstacle detection and the underlying algorithms used. Section 4.2 describes our approach to obstacle detection and tracking. The three phases of our approach are then discussed

in detail in Detecting obstacles (Section 4.3), Distilling obstacles (Section 4.4) and Tracking obstacles (Section 4.5) sections. The system is then evaluated in Section 4.6.

4.1 Review of obstacle detection

In this thesis we refer to obstacles which are often but not always other vehicles in the road scene. They can also be other objects such as pedestrians or barriers. We use the term obstacle detection as opposed to vehicle detection to emphasise the distinction between our approach from many used for vehicle detection. Our system will detect any object of significant size present in the road scene. Our work steers away from algorithms that depend on features particular to vehicles or any single class of object. Instead our system avoids using models describing potential obstacles. In the terms of Sun *et al.* (2006) we avoid knowledge based approaches which search for attributes particular to a class of objects, such as tires on cars. Our work takes a human inspired approach (discussed in Chapter 2). The human vision system is capable of detecting obstacles that it has never previously encountered. We believe our system should also have a similar attribute.

Obstacle detection schemes, while giving some promising results, do not yet provide a level of service comparable to the lane tracking algorithms. Most detection techniques work only in a narrow range of environments or road conditions(Du and Papanikolopoulos, 1997; Krüger *et al.*, 1995). Requiring a flat longitudinal road profile(Du and Papanikolopoulos, 1997; Krüger *et al.*, 1995; Pomerleau and Jochem T., 1996) or that all obstacles are vehicles(Hoffmann, 2006; Schweiger *et al.*, 2005). Highway results are common, systems that work on secondary or urban roads are significantly more scarce, (Franke and Heinrich, 2002) is a notable exception.

Smith and Brady (1995) describes ASSET-2 (A Scene Segmenter Establishing Tracking Version 2), a feature-based optical flow algorithm that doesn't require explicit removal of ego-motion. Instead vehicles are segmented in the image space based on grouping flow vectors that move consistently over time. In each frame features and edges identified. A feature list is maintained by tracking engine using a 2D feature model. The vector field list clustered by a segmenter. Then the cluster list compared with the historic filtered list. The background of the scene is apparently inconsistent enough that it is not mistaken for an object. The algorithm also contains a constant velocity model that allows objects to be estimated while occluded until they appear again.

Krüger *et al.* (1995) reformulated the obstacle detection problem into a state estimation problem. Using Kalman filters to track optical flow vectors over time. A *Chi Squared* test was used to determine if a given motion fits a stationary object. If the test failed the object was declared moving and ground plane and height constraints were used to interpret the optical flow.

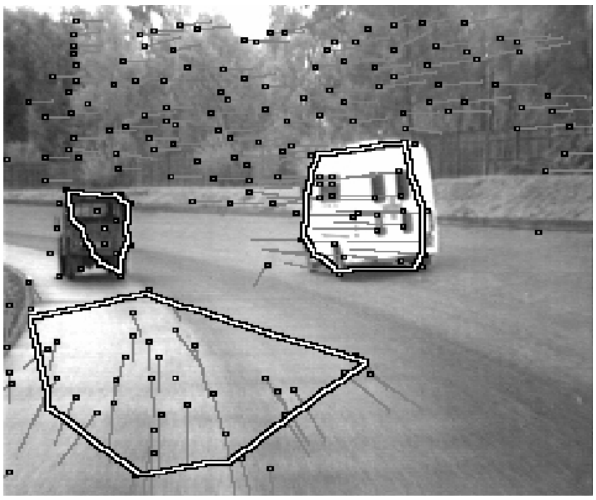


Figure 4.1: Asset2: Tracking Vehicles with feature based optical flow. Reproduced from Smith and Brady (1995)

Heisele and Ritter (1995) implemented colour blob tracking based on K-means to segmentation of the colour space. The method looks for blobs instead of a particular class of obstacle, yet they are still susceptible to the fickle nature of apparent colour.

Giachetti *et al.* (1998) used optical flow to detect vehicles. Model ego-motion and shocks. They demonstrate that shocks can be modelled as displacement of image by a constant. Pure translation modelled as simple linear equation. Turning motion modelled as 2nd order polynomial with shock constants.

(Krüger, 1999) used features on the road to estimate ego-motion relative to ground plane. A Kalman filter is used for state estimation which consists of changes in rotation, translation of camera relative to ground plane. Images are then warped based on the estimated ego-motion and differenced with consecutive images. Objects are detected by residual after differencing. Obstacle information is used to improve lane tracking.

Franke and Rabe (2005) achieved fast depth from motion estimation using banks of Kalman filters running with different initial conditions. The purpose of this approach is to support parallel range of hypothesis. This is another approach to temporary ambiguity tolerance. Our framework does this implicitly instead of managing a set of filters.

4.1.1 Range flow

(Franke and Heinrich, 2002) developed a system to detect people and objects such as basketballs at close range. Instead of modelling vehicle ego-motion they introduce a "flow/depth quotient". The quotient is linear for longitudinal motion.

Rotational ego-motion is also compensated for by matched filters for each axis of rotation. With rotation removed the the expected quotient forms a plane (in scene x,y, $z = quotientcoordinates$). Moving obstacles are detected by a large deviation from the expected quotient plane. Results show detection of child moving horizontally and basketball detection. Due to the optical flow limitation of ± 2 pixels per frame @ 25Hz, method is limited to a slow 25km/h. The group is now looking at using a multi-scale flow technique to overcome this problem (Franke and Heinrich, 2002). Like us this group has preferred to use generic obstacle detection techniques of stereo and optical flow, we too will need to use multi-scale flow. The use of a quotient between the stereo and optical flow is a nice approach but we will not use it in that it requires the stereo and the optical flow to be correct. In our experience these cues are incredibly noisy so best used were down stream algorithms don't require highly accurate values.

The range of optical flow values and disparities encountered in the road scene is large. Disparities and image motion in a single instance can easily range from 0 at the horizon to over 96 pixels in the near field. Franke and Heinrich (2002) limited the vehicle speed in their experiments so that the gradient based optical flow estimation constraint of image motion of less than 2 pixels per frame was honoured. They mention that future work could include a solution using Gaussian image pyramids to enhance the dynamic range possible in the flow estimation. We have adopted a image pyramid technique both in the optical flow and disparity map estimation. For the case of optical flow we implement a method similar to (Jähne and Haussecker, 2000). The optical flow is computed for the most coarse images then the result used to warp the next higher image resolution to maintain an acceptably small image motion at each level. The penalty for using a coarse to fine approach is that any errors occurring at any image resolution are propagated and amplified into the finer images.

4.1.2 Stereo algorithms

A ray in the 3D scene when projected onto a 2D plane will trace out a line on plane (see Figure 4.2). Given two images of a scene a point in one image actually represents a ray in the 3D scene that has been projected onto that image point. From the second camera the ray of projection traces out what is know as the epipolar line of the point. The epipolar line is in fact an intersection point of two planes, the image plane of the second camera with the epipolar plane which is a plane described by the two camera centres and the image point. For every point in either image there will be a complementary line in the alternate image. This relationship is known as the epipolar geometry between a pair of cameras. The projection of one camera centre onto the other image plane is known as the epipole. By the definition of the epipolar plane all epipolar lines must intersect the epipole. By knowing the epipolar geometry between a pair of cameras point correspondences between the two images becomes a 1D instead of a 2D search.

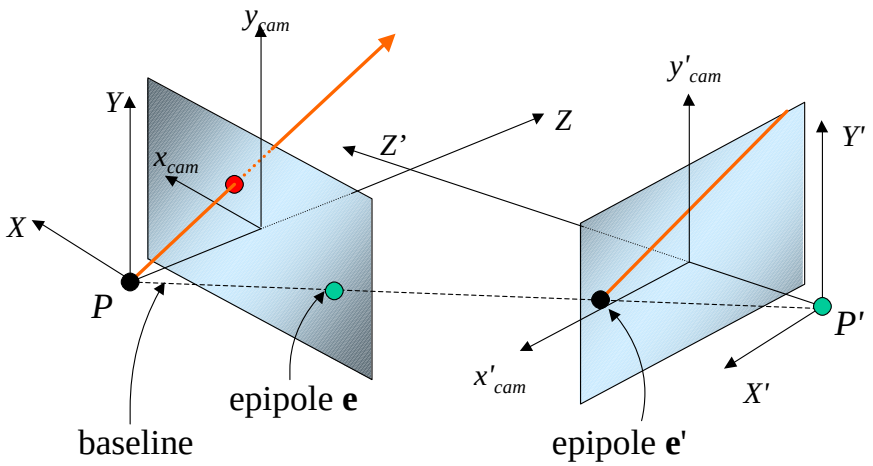


Figure 4.2: Epipolar geometry

A depth map is an image where the pixel intensity is proportional to the depth of the object projected onto that location on the image. A disparity map is often used as an interchangeable term but by definition is the pixel disparity between the reference image and a second image along the epipolar lines. The most common case for a disparity map is to have cameras with parallel optical axes, also referred to as placing the epipoles at infinity. Parallel optical axes introduce some advantages: the epipolar lines will are parallel, disparity becomes inversely proportional to depth and all disparities will be in the same direction (see Figure 4.3).

There have been numerous of approaches proposed for finding correspondences between stereo images (Fua, 1993a; Banks *et al.*, 1997; Scharstein *et al.*, 2001). Feature based techniques where edges or corners are first detected then matched have been successful in applications where sparse depth information is acceptable (Zhang *et al.*, 1997). However, area based techniques particularly correlation have been increasingly used as computational power and memory becomes cheaper (Aschwanden and Guggenbuhl, 1993). The common algorithms include denoting the template window as $\mathbf{I}_1$, the candidate window as $\mathbf{I}_2$, and summation over the window as $\sum_{(u,v)\in W}$, these are:

Table 4.1 shows the equations for the most common area based correlation metrics. Figure 4.4 demonstrates the difference of the correlation metrics. The sum of squared differences (SSD) is often favoured in the vision theory as it can be justified as minimising the squared differences a common practice in least squares or optimal filtering applications. Since this method requires a multiplication per pixel it is often simplified to the sum of the absolute differences (SAD). The minimum of a SAD match will be the same as the minimum for a SSD match. Zero mean normalised cross correlation (ZNCC) offers the best tolerance of image brightness changes between images. It performs the full two stages of normalisation (removing constant offset and normalising the contrast), given unlimited

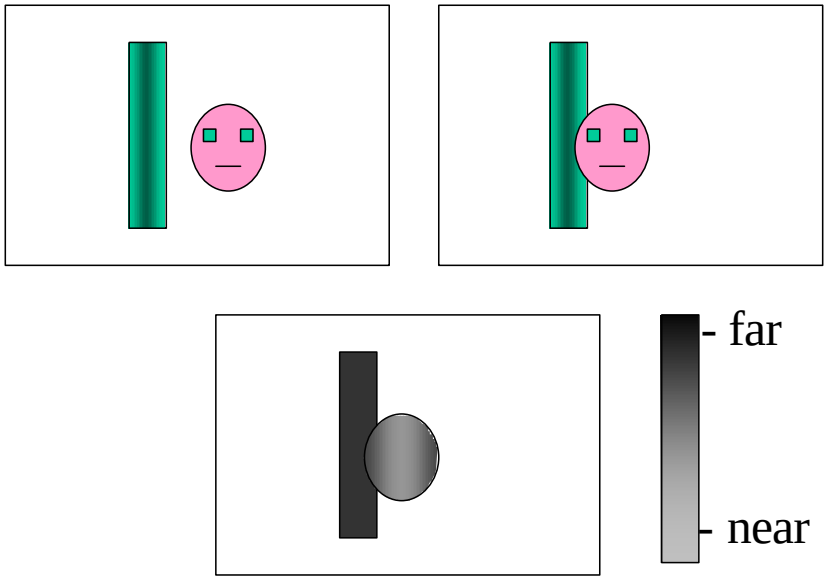
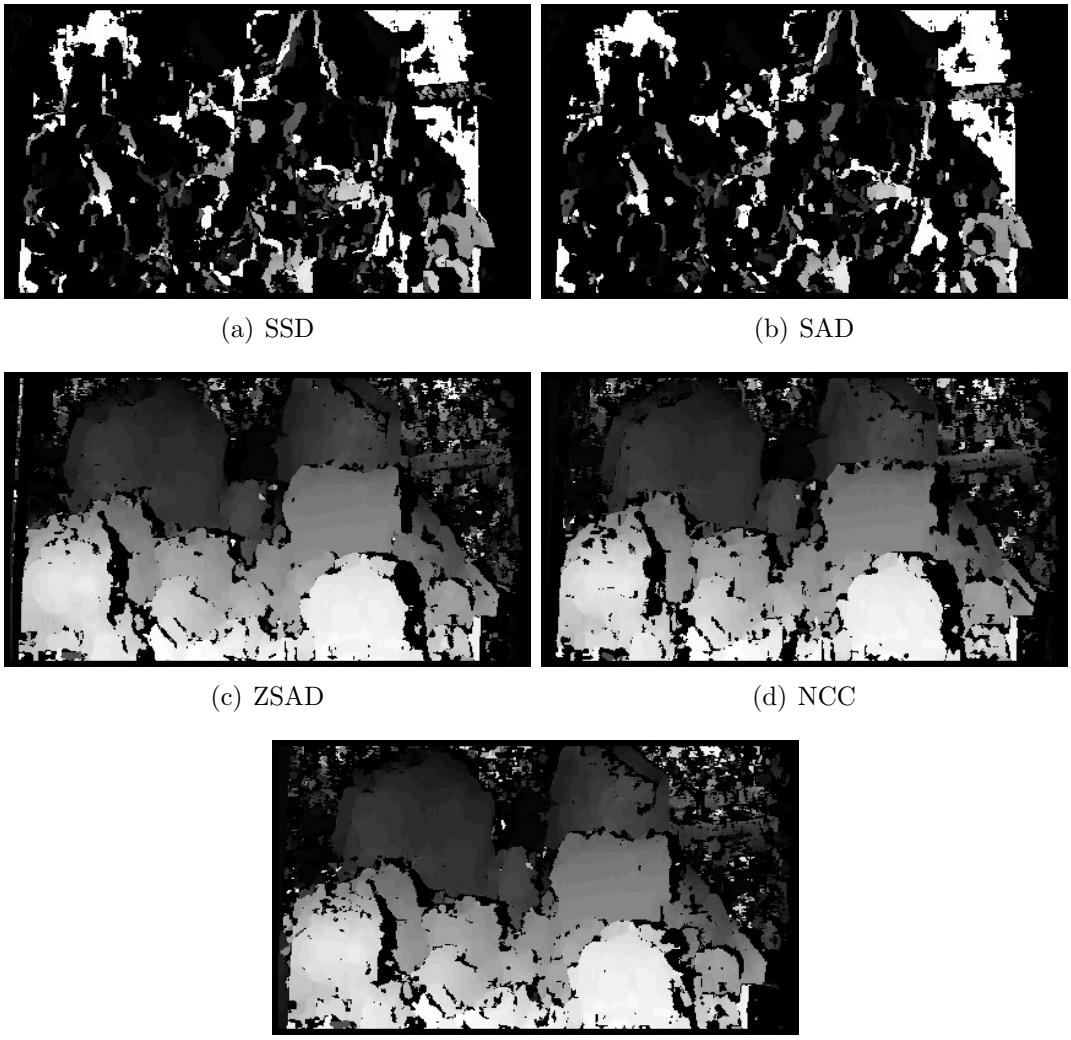


Figure 4.3: Disparity map construction

Method	**Abbrev.**	**Definition**
Sum of Squared Differences	SSD	$\sum_{(u,v)\in W}(\mathbf{I}_1(u,v)-\mathbf{I}_2(x+u,y+v))^2$
Sum of Absolute Differences	SAD	$\sum_{(u,v)\in W}\lvert\mathbf{I}_1(u,v)-\mathbf{I}_2(x+u,y+v)\rvert$
Normalised Cross Correlation	NCC	$\frac{\sum_{(u,v)\in W}\mathbf{I}_1(u,v)\cdot\mathbf{I}_2(x+u,y+v)}{\sqrt{\sum_{(u,v)\in W}\mathbf{I}_1(u,v)^2\cdot\sum_{(u,v)\in W}\mathbf{I}_2(x+u,y+v)^2}}$
Zero mean Normalised Cross Correlation	ZNCC	$\frac{\sum_{(u,v)\in W}(\mathbf{I}_1(u,v)-\overline{\mathbf{I}_1})\cdot(\mathbf{I}_2(x+u,y+v)-\overline{\mathbf{I}_2})}{\sqrt{\sum_{(u,v)\in W}(\mathbf{I}_1(u,v)-\overline{\mathbf{I}_1})^2\cdot\sum_{(u,v)\in W}(\mathbf{I}_2(x+u,y+v)-\overline{\mathbf{I}_2})^2}}$

Table 4.1: Image correlation measures.

computational resources it is often the best choice.
On many occasions however cpu resources are scarce, it can be safe to assume that brightness offsets or contrast doesn't vary considerably between cameras so cheaper methods like SAD are justifiable. Additionally there can be some reasons why the performance of SSD, and SAD measures are favourable. In real applications images can be blurred by vibrations or calibration can drift, SSD and SAD are more tolerant of these errors. SSD and SAD-based measures also generate



(a) SSD (b) SAD

(c) ZSAD (d) NCC

(e) ZNCC

Figure 4.4: Comparison of correlation functions. From (Banks *et al.*, 1997)

comparatively smooth minimums around the pixel disparity of best fit. These smooth minimums allow sub-pixel estimates to be made using weighted sum of the correlation values. NCC based measures however can vary abruptly even at pixel values adjacent to the correct disparity making similar sub-pixel estimation not useful. Also real image data can often contain regions with little contrast reflecting real-world objects of little contrast, image noise then dominates the contrast within a window. An example of this is the foreground of a road scene. The road is often featureless and the texturing in the image is solely due to image noise or specular reflection. By stretching the contrast NCC techniques often perform badly, generating erratic disparity matches. SAD, SSD results, while still not providing reliable results, tend to produce better "guesses" as the effects of noise are averaged out as opposed to emphasised.

A compromise between the two techniques is to use SSD or SAD metrics with

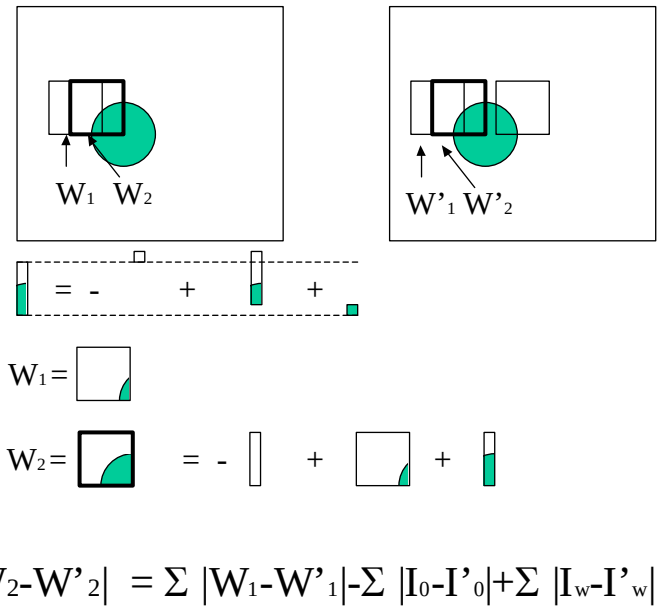
preprocessing. A Laplacian of Gaussian convolution or a rank transform can strengthen the assumption that normalisation between the image windows in not necessary by increasing the likelihood that the constant image offset in intensities is consistent and that the contrast is scaled similarly. While smooth minimum functions and better guesses in textureless regions still occur. Even after including the additional cost of image preprocessing the images, a large computational savings can be made. Banks *et al.* (1997) showed that preprocessing then SAD performs as well as ZNCC on natural images. The result of LOG filtering and SAD correlation will be similar to the ZSAD case shown in Figure 4.4 .

Scharstein *et al.* (2001) completed an excellent review of two-frame stereo correspondence algorithms. Comparisons were made on accuracy and speed. Real-time SAD technique generated a good result when weighted by fast execution time. Faugeras *et al.* (1993) pioneered the use of iterative box filtering methods to greatly optimise the computation of a correlation measure across the whole image. A naive implementation of a disparity map using area-based correlation would take $O(dw^2n^2)$ operations for an $n \times n$ is the image, $w \times w$ window and a d disparity range in pixels. Using an iterative implementation based on a box filtering approach, this is reduced to $O(dn^2)$. The number of operations is now independent of the correlation window size. Since typical window sizes range from four to 24 pixels this can amount to a 16 to 576 times speed-up compared with the equivalent non-recursive calculation. The approach is illustrated in Figure 4.5. Williamson and Thorpe (1999) used a modified SAD based recursive box filter for trinocular image matching for detection of road obstacles. Kagami *et al.* (2000) used a SAD and NCC algorithms for obstacle avoidance on a humanoid robot. The recursive algorithm tips the balance of interms of computational complexity, it permits a dense disparity map to be generated at frame-rate. We will use the SAD disparity map algorithm as part of our stereo perception system.

Fua (1993b) proposed a simple efficient and effective method for drastically reducing erroneous matches propagated from the disparity map is left-right, right-left consistency checking. With reference to Figure 4.6:

Given a pixel location T in the left image and its best match T' in the right image, to be consistent the best match for the point T' in the left image should be T. If this is not best match then there is an inconsistency and the pixel match is said to be inconsistent and discarded.

The assumption behind this consistency checking is that any point in the left image represents only one point in the real world scene and therefore should correspond to one point in the other image. In general this assumption is correct however if an image point represents an object that is partially transparent or occluded in the second image multiple points or no point may correspond. These cases are the exceptional enough that consistency checking provides more positives than negatives. Instead of computing the whole disparity map twice once looking for sub windows of the left image in the right, then looking for sub windows of the right image in the left, much of the required summations



$$\Sigma|W_2\text{-}W'_2| = \Sigma|W_1\text{-}W'_1|\text{-}\Sigma|I_0\text{-}I'_0|+\Sigma|I_w\text{-}I'_w|$$

Figure 4.5: Incremental additions and subtractions to the temporary variables in the iterative disparity map technique

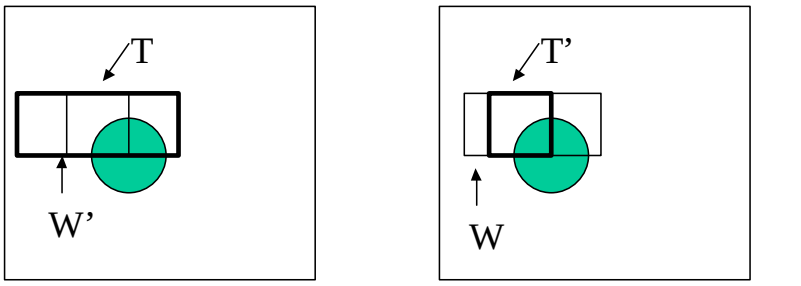


Figure 4.6: Consistency Check: Left $\Longleftrightarrow$ right matches checked for consistency

are performed to find the original disparity map. This is because the result of the correlation measure at disparity d_R at location (u_R, v_R) in the right image is equivalent to the correlation measure of disparity d_L where $d_L = -\delta$ at left image location (u_L, v_L) where $u_L = u_R + \delta$.

To create a Gaussian pyramid the original image is smoothed then over-sampled repeatedly until the image resolution is so coarse as to be of no value. A Laplacian pyramid differs from a Gaussian pyramid in that after the image has been smoothed the blurred image is subtracted from the original image at the current resolution. Figure 4.7 shows Gaussian and Laplacian images of a scene.

Gaussian pyramid algorithm:

Given the original image I_0.

1. $\mathbf{B}_n = \mathbf{I}_n * \mathbf{g}$ given Gaussian kernel g with $\sigma = 1$
2. $\mathbf{I}_{n+1} = \sum_{(i=0,j=0)}^{((M-1)/2,(N-1)/2)} \mathbf{B}_n(2i, 2j)$ (take every 2nd pixel in x and y)
3. repeat n times

Table 4.2: Gaussian pyramid algorithm.

Laplacian pyramid algorithm:
Given the original image $\mathbf{I}_0$.

1. $\mathbf{B}_n = \mathbf{I}_n * \mathbf{g}$ given Gaussian kernel $\mathbf{g}$ with $\sigma = 1$
2. $\mathbf{I}_n = \mathbf{I}_n - \mathbf{B}_n$
3. $\mathbf{I}_{n+1} = \sum_{(i=0,j=0)}^{((M-1)/2,(N-1)/2)} \mathbf{B}_n(2i, 2j)$ (take every 2nd pixel in x and y)
4. repeat n times

Table 4.3: Laplacian pyramid algorithm.

Given the conventional SAD correlation equation:

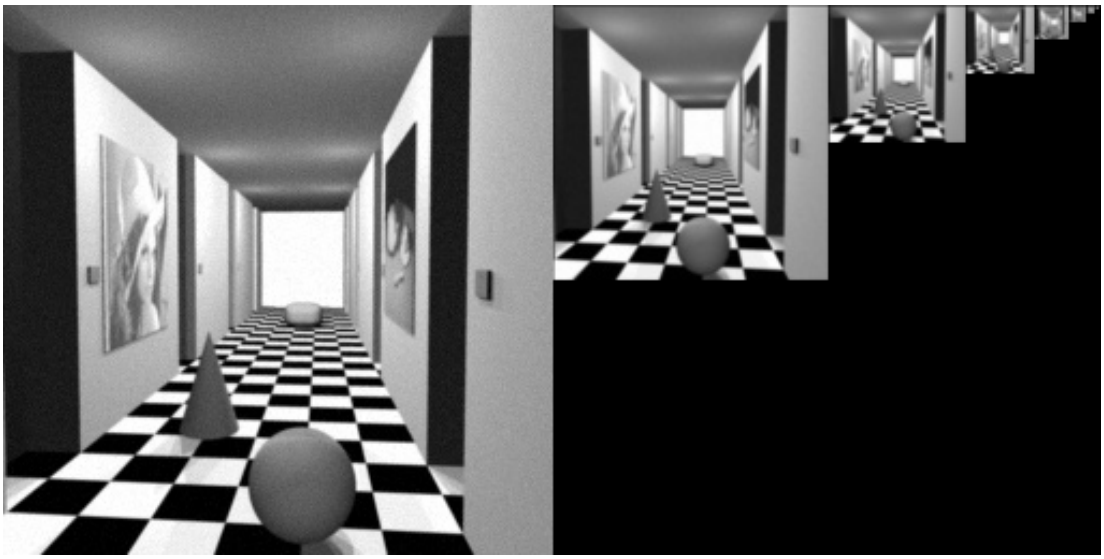
$$C(x, y, d) = \sum_{(u,v)\in W} |\mathbf{I}_1(x + u, y + v) - \mathbf{I}_2(x + d + u, y + v)| \tag{4.1}$$

$$O(x, y) = \min_d(C(x, y, d)) \tag{4.2}$$

Gaussian pyramid reconstruction algorithm: Starting at the coarsest level n.

1. $\mathbf{B}_{n+1} = \sum_{(i=0,j=0)}^{((2.M-1),(2.N-1))} \mathbf{I}_n(i/2, j/2)$ (where sub integer indices are interpolated)
2. $\mathbf{I}_{n+1} = \mathbf{B}_{n+1} * \mathbf{g}$ given Gaussian kernel $\mathbf{g}$ with $\sigma = 1$
3. repeat n times

Table 4.4: Image reconstruction from Gaussian pyramid algorithm.



(a)

(b)

Figure 4.7: (a): Five level Gaussian pyramid. (b): Five level Laplacian pyramid.

The iterative box filtering SAD correlation equations are:

$$P(x,y,d) = |\mathbf{I}_1(x,y) - \mathbf{I}_2(x+d,y)| \tag{4.3}$$

$$Q(x,0,d) = \sum_{j=0}^{W-1} P(x,j,d) \tag{4.4}$$

$$Q(x,y+1,d) = Q(x,y,d) + P(x,y+W,d) - P(x,y,d) \tag{4.5}$$

$$C(0,y,d) = \sum_{i=0}^{W-1} Q(i,y,d) \tag{4.6}$$

$$C(x+1,y,d) = C(x,y,d) + Q(x+W,y,d) - Q(x,y,d) \tag{4.7}$$

$$O(x,y) = \min_d C(x,y,d) \tag{4.8}$$

The iterative box filtering SAD algorithm we implement is inspired by the work of Kagami *et al.* (2000). It combines the iterative algorithm with the parallel, single instruction multiple data (SIMD) instruction set available in recent Intel Pentium processors. Using SIMD instructions available on the Pentium range of Intel processors a four to eight times speed-up is achieved. For each inner loop iteration 4 disparities are calculated.

Quadratic approximations are often applied to sub-pixel interpolation when using SAD correlation for example Kagami *et al.* (2000); Mühlmann *et al.* (2002). This is because the derivation of SAD correlation is usually a final step simplification of SSD correlation theory. When using sum of squared difference correlation it makes sense to use a quadratic approximation model since errors will grow quadratically. When using sum of absolute differences the error functions grow linearly, this justifies the use of a weighted linear sub-pixel approximation.
An investigation using some synthetic data confirmed this observation. We generated data in Matlab by taking a typical road scene and inserting a car at a known sub-pixel disparity, the car was placed at a whole pixel disparity on an enlarged version of the image and then resized using bicubic interpolation to the original size. As shown in 4.8 the quadratic approximation introduces an under estimation around the whole pixel disparity points and an over-estimation at the half-pixel disparity points. Linear interpolation tracks the ground truth disparity significantly better. The estimated disparity is the average across the scan line in the disparity image. There is a constant valued offset in disparity of the estimates and the ground truth due to underestimation due to boundary effects at the edge of the displaced vehicle.

Shimizu and Okutomi (2002) came to a similar conclusion, SAD and linear interpolation are better approximation for SSD with quadratic interpolation than SAD with quadratic interpolation.

Optical flow algorithms

Schrater *et al.* (2001) showed that changing object size without generating optical flow sufficient for humans to estimate time to contact, Therefore changes in car size in far field could be used to estimate vehicle motion without computing optical flow or using stereo vision.

Barron *et al.* (1992) undertook an assessment of optical flow implementations, and more recently Jähne and Haussecker (2000) compared popular methods of gradient, phase and image based optical flow. Most optical flow techniques are interpreted as solutions of the "brightness consistency constraint equation" (Barron *et al.*, 1992). The equation states that any change in image brightness (intensity) values must be due to motion in the image:

$$\Delta \mathbf{I}(\mathbf{x}, t) . \mathbf{v}(\mathbf{x}, t) + \mathbf{I}_t(\mathbf{x}, t) = 0 \tag{4.9}$$

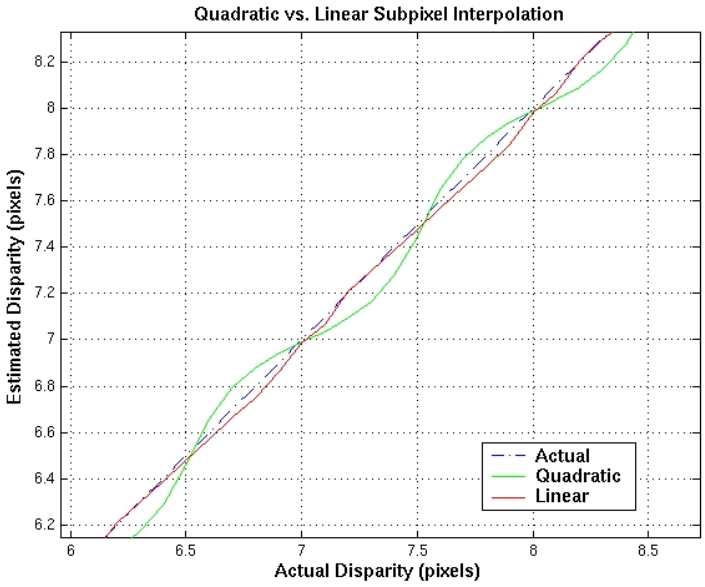


Figure 4.8: Quadratic vs. Linear sub-pixel interpolation

The equation has two unknowns and there by requires two constraints to solve. For the first constraint the common assumption is that the scene appearance does not change over time except for image motion:

$$\frac{d\mathbf{I}(\mathbf{x},t)}{dt} = 0 \tag{4.10}$$

There are three common choices to derive a second constraint:

1. Impose global smoothness constraints (regularisation) on the velocity field. Horn and Schunck (1981) proposed a technique used global smoothness constraint. The technique was simple to implement but required the minimisation of the following non-linear equation:

$$\int_D \left(\Delta\mathbf{I}.\mathbf{v} + \mathbf{I}_t\right)^2 + \lambda^2 \left(||\Delta u||_2^2 + ||\Delta v||_2^2\right) d\mathbf{x} \tag{4.11}$$

   Horn and Schunck (1981) used an iterative minimisation algorithm. The need to perform this large minimisation discourages the use of the technique for real-time applications. Also the global smoothness constraint has tendency to over smooth the resulting flow field, particularly fields containing multiple objects moving with different velocities (such as our application).

2. Use higher-order derivatives with an additional brightness conservation assumption. For example,

$$\frac{d^2\mathbf{I}(\mathbf{x},t)}{d^2t} = 0 \tag{4.12}$$

This equation implies rotation and dilation should not be present in the image sequence. Accurate second derivatives are significantly harder to obtain due to low signal to noise ratios than first derivatives. Aperture problems are worsen if a local neighbourhood is used to estimate the second derivative as it requires a larger neighbourhood than the first derivative.

3. Fit a constant or linear parametric model to the velocity field. Lucas and Kanade (1981) proposed the most common way to introduce this constraint. They integrated the image derivative s over a local window and then found the best least squares fit. The technique uses a weighted least squares fit of first order constraint equation with a constant velocity model in a small spatial neighbourhood (window). Minimizing:

$$\sum_{xin\Omega} w^2(x) \left[\Delta \mathbf{I}.\mathbf{v} + \mathbf{I}_t\right]^2 \tag{4.13}$$

where $w(x)$ is a weighted window function such as a Gaussian.

Since derivative based techniques have a limited bandwidth before aliasing will occur a Gaussian low pass spatio-temporal filter is applied to the image sequence to remove high frequency components likely to cause aliasing. In the original technique the temporal filter requires 15 frames temporal support for a 1.5 pixel standard deviation. Fleet and Langley (1995) used recursive infinite impulse response filters instead of the finite impulse response filter to reduce the frame storage to only three temporary images. Unreliable estimates can be seen as a large covariance of the least squares pseudo inverse or more simply as the size of the smallest eigenvalue (Simoncelli *et al.*, 1991). The source of unreliable optical flow estimates can be categorised into: a low signal to noise ratio due to poor contrast, a violation of the brightness consistency constraint assumption or an aperture problem.

Haussecker H. and B. (1998) took the standard least squares approach of Lucas and Kanade a step further by using a total least squares. Instead of finding the least squares fit by minimising the difference in image derivatives by varying the velocity x and y, the temporal dimension t is also varied in the minimisation. The problem was formulated as minimising the equation:

$$||e||_2^2 = \mathbf{r}^T \mathbf{J} \mathbf{r} \tag{4.14}$$

$$\mathbf{J} = \begin{bmatrix} < g_x, g_x > & < g_x, g_y > & < g_x, g_t > \\ < g_y, g_x > & < g_y, g_y > & < g_y, g_t > \\ < g_t, g_x > & < g_t, g_y > & < g_t, g_t > \end{bmatrix} \tag{4.15}$$

where: $\mathbf{r} = [r_1, r_2, r_3]^T$ and $< g_i, g_j >$ is inner product over a local region of the image derivatives g_i and g_j in the i and j directions respectively.
The image velocities $\mathbf{v} = [u, v]^T$ are recovered from the eigenvector $\mathbf{r}$ for the x

and y directions respectively by:

$$u = \frac{r_1}{r_3} \tag{4.16}$$

$$v = \frac{r_2}{r_3} \tag{4.17}$$

This simple extension has been shown to improve the overall flow estimates. Standard least squares has been shown to be biased toward smaller velocities, the total least squares approach is unbiased. A significant advantage of the total least squares approach is that the result provides a useful diagnostic of flow in the image using the rank of the J matrix or simple ratios of the eigenvalues to classify regions as:

- rank 0: constant brightness, no apparent motion
- rank 1: aperture problem area, only normal motion available
- rank 2: good spatial structure and constant motion
- rank 3: inconsistent motion (brightness constraint equation failed)

Finally, cases when the brightness constraint equation fails due to the assumption of constant image brightness has been investigated by a number of groups. Negahdaripour (1998) in particular redefined the optical flow constraint equation to incorporate more sophisticated brightness model. Haussecker and Fleet (2000) replaced the constant brightness assumption by linear and quadratic models with some success.

4.2 Our approach

The process of detecting and tracking obstacles is composed of three phases as shown in Figure 4.9. The phases are detection, distillation and tracking. All three phases run concurrently detecting new obstacles, monitoring and tracking existing objects. The most primitive phase "detection" uses a set of whole image techniques to search the image space for likely obstacle candidates. Stereo disparity and optical flow are principally used at this phase. Other visual cues can be used to shore up the detection during unfavourable circumstances, for instance possible obstacle candidates can also be derived looking for colour consistency differing from the homogeneous road surface.

The transition between the first phase ("Detection") and the second phase ("Distillation") occurs when sets of particles representing each obstacle candidate are injected into the particle filter state-space inside the Distillation algorithm. These potential objects are represented by distributing particles around the detected location in the state space. The obstacles are tracked in a state-space consisting of

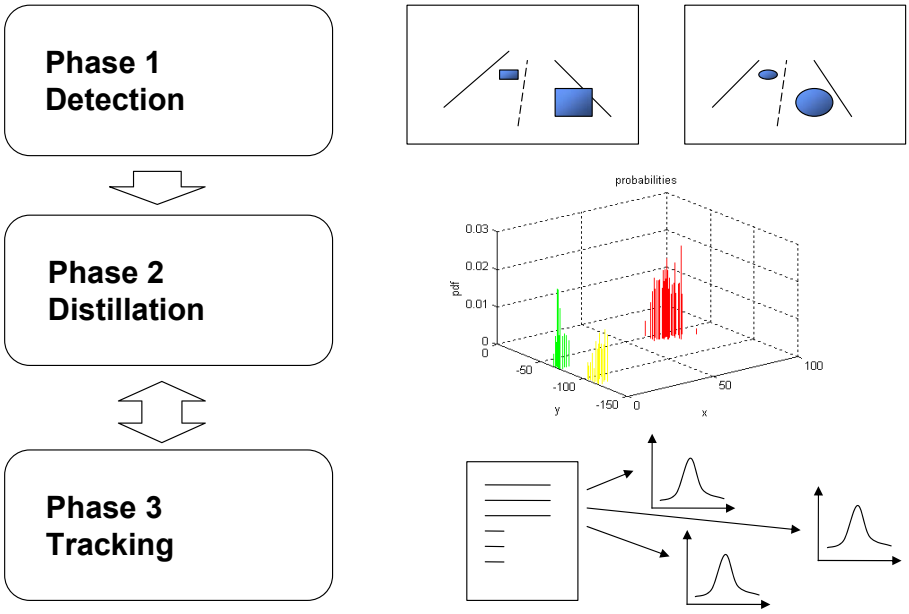


Figure 4.9: Three phases of the Obstacle detection and tracking engine

a the longitudinal and lateral position of the object with respect to the centre of mass of the research vehicle and one dimension representing the estimated obstacle size. Particles representing unsubstantiated obstacle candidates are eventually resampled to other obstacle candidates. The remaining potential obstacles are tracked within Distillation algorithm between frames.

Each cluster of particles that survive a minimum number of iterations is then checked for unimodality using the uncertainty deviation metric defined in Chapter 3. If the Gaussian distribution adequately describes the cluster a uniqueness operator is applied to the region of the obstacle in each if the stereo images and a set of correlation templates are taken at the most unique points of the obstacle. If the correlation templates are tracking reliably and the distribution is still sufficiently Gaussian a Kalman filter is spawned solely to track this obstacle, this is phase three "Tracking". The obstacles are then tracked using correlation templates alone. This list of objects tracked by Kalman filters is the primary output of the obstacle detection and tracking system.

If for any reason the Kalman filter starts to diverge the last object location is treated as a potential obstacle candidate again and particles are injected back into the particle filter and the Kalman filter is discarded. The object candidate is now back in the Distillation phase.

4.3 Detecting obstacles

The "bottom up" techniques used to segment potential obstacles are stereo disparity, optical flow and free space estimation. These visual cues have been shown to play a crucial role in human vision obstacle detection and supports our philosophy of a generic approach to obstacle detection as opposed to a implementation specific approach.

4.3.1 Stereo disparity

A Laplacian image pyramid coupled with sum of absolute difference correlation is used to generate a stereo depth map. Using image pyramids for disparity map estimation gives a several added benefits in addition to just an increased range of estimated disparities.

Finding a suitable neighbourhood or window size when using correlation techniques is a significant issue (Hirschmüller *et al.*, 2002; Lotti and Giraudon, 1994). A small window size is susceptible to image noise, large window sizes over smooth the feature map. A large window allows a more reliable match but causes overly smooth disparity maps. A small window size allows for finer features to be represented can introduce noise due to erroneous matches.

Fortunately, generally in road scenes, close objects induce large disparities and are apparently large in the image while distant objects, such as vehicles down the road, induce small disparities and are small in the image. Using an image pyramid and calculating the disparity for each image resolution with the same sized correlation window means that the correlation window is effectively halved for each image resolution going from coarse to fine. This property is exactly what is required to match large objects with significant disparities and smaller objects at minimal disparities. To reconstruct the disparity map from the layers of the pyramid we do not use image warping between the resolutions. This way we can avoid the propagation of errors from coarser pyramid layers. At higher resolutions we are interested in finding distant objects with small disparities, where large objects such as close vehicles are recovered at a coarse image resolutions. An issue arises that coarse resolution images can only resolve disparities to half the accuracy of the next higher resolution images. However, this works in opposition to the property of disparity estimates deteriorating as distances increase, the effect on the resultant disparity map is acceptable. We also save significant computation involved in warping the pyramid layers.

By using a Laplacian pyramid in preference to a Gaussian pyramid effectively apply Laplacian of Gaussian filtering to the source images before Sum of Absolute Difference correlation. This significantly reduces the intensity sensitivity of the SAD correlation. Using an image pyramid we only need to compute two disparities at each level for the disparity to overlap between the levels of the pyramid.

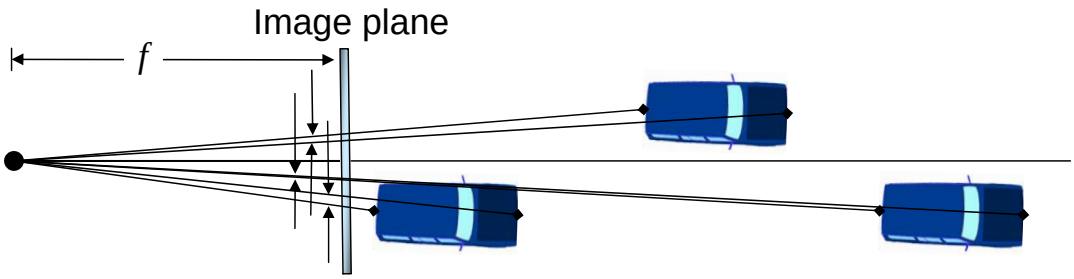


Figure 4.10: Reduced stereo depth resolution in the near field is countered by the larger disparities made by near field objects.

We elect not to warp the image between resolutions rather use a larger disparity range. The larger disparity range introduces redundancy between the levels that we use for noise rejection as we reconstruct the pyramid.

An n-level pyramid will allow the estimation of a maximum disparity of $d \times 2^n$. We need at least 96 disparities so we compute $d = 8$ disparities for each of a $n = 5$ level pyramid.

Right, left consistency checking is used to remove outliers from the correlation results. A second method we use to eliminate points of uncertain disparity. Correlation is to look for flat minimums in the correlation coefficients (see Figure 4.11). A textureless region will match almost as well with its neighbouring disparity as it does at the best matching disparity. Pixel locations where the correlation results are flat are discarded. This means that valid disparities of smoothly varying surfaces will erroneously be discarded, for example a valid disparity of around 5.50 will be discarded as it will be interpolated from the correlation results $C_{d=5}$ and $C_{d=6}$ which would be close to the same value, but this is preferable than propagating errors.

A test of the iterative box filtering SAD algorithm and the error correction techniques are shown on synthetic data and on real image data from a people tracking experiment (described in Chapter 3) in Figure 4.12 and Figure 4.13 respectively.

To build the image from the pyramid we start at the coarsest image resolution and double the image size and smooth the image as is standard for a Gaussian pyramid reconstruction. Since the intensity values actually represent disparities they are doubled then compared with the next higher level of the pyramid. If the difference between the two disparity estimates is a small tolerance the higher resolution value is kept otherwise the lower resolution value is used. This reconstruction technique favours the higher resolution image when the disparity seems reasonable, while propagating the coarse disparities when the higher resolution disparities are missing (due to error checking) or are too far off.

Figure 4.16 shows the resultant disparity map for a typical road scene. The large

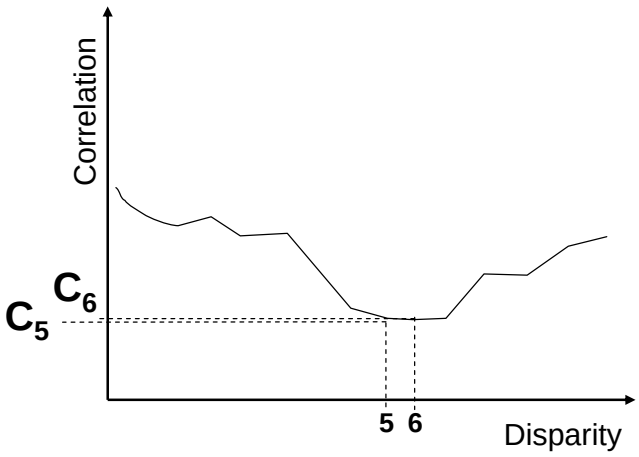


Figure 4.11: Texture Check: finding slope between minimum and neighbour, discarding points with near horizontal slope.

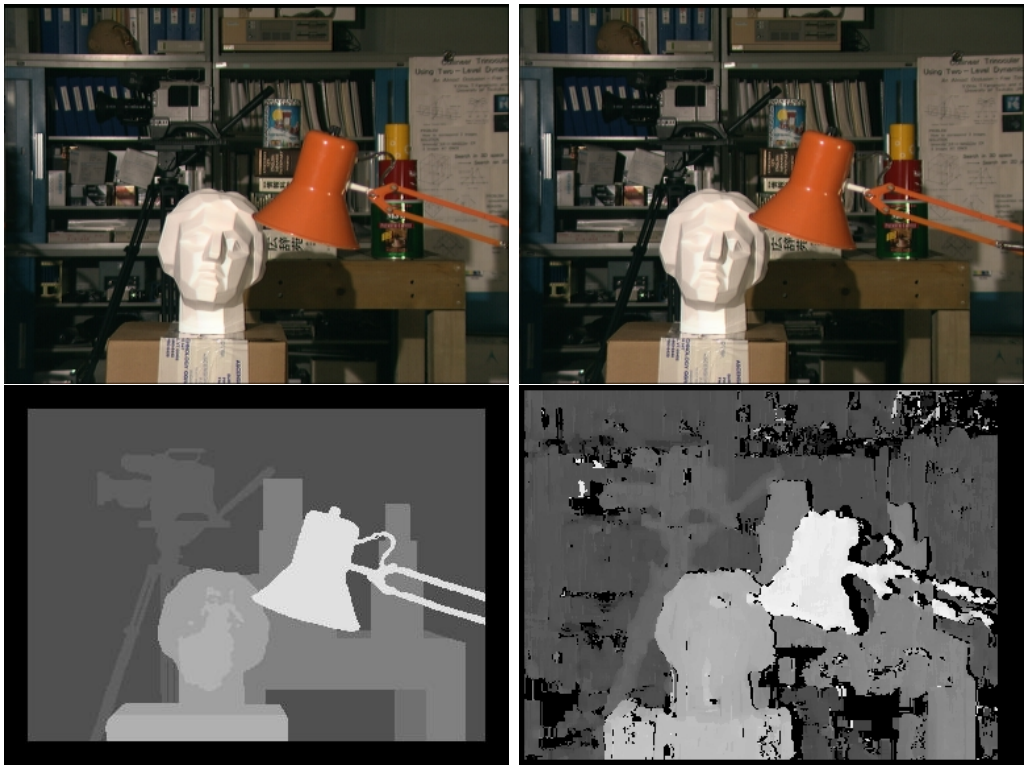


Figure 4.12: top: University of Tsukuba test images. bottom left: ground truth bottom right: estimated disparity

Figure 4.13: top: Typical stereo images from people tracking experiment bottom: estimated disparity

disparity of the close car to the left is shown in yellow, while the distant car is shown on the horizon in blue. Gross noise in the image is restricted to the sky which has no texture and no correct estimate in any level of the pyramid. The near field road is also underestimated in places due to insufficient texture.

The final stereo depth map algorithm is summarised in Table 4.5.

Segmentation

A V-disparity map is computed to remove the ground plane (though the ground need not be planar) from the image. The V-disparity map is an accumulation of the disparities per scan-line. The ground will show up on the V-disparity map as a curve. To estimate the ground plane we use a piecewise linear model and the RANSAC algorithm to obtain a robust line fit.

Results and discussion

Figure 4.14 shows the algorithm applied to synthetic corridor images distorted with additive Gaussian distributed noise. A corridor exhibits similar properties

Stereo disparity map algorithm:

1. Rectify gross image errors based on calibration data, particularly rotations about optical axis.
2. Use template tracking to ensure the central horizon has zero disparity between images.
3. Create an n-level Laplacian pyramids for the left and right image according to Table 4.3.
4. For each image resolution use a SAD iterative box filtering algorithm to find correspondences across d-disparities.
5. Perform left-right consistency checking.
6. Perform textureless region checking.
7. Perform sub-pixel interpolation.
8. Reconstruct the final disparity map from the levels of the pyramid using redundancy across the levels to verify the result.

Table 4.5: Stereo disparity map algorithm.

regarding scale as a road. Figure 4.15 shows the floor removed from the disparity image using the V-disparity map to estimate the floor, leaving three obstacles.

Figure 4.17 shows the algorithm applied to a typical road scene. In the case of the road scene the disparity map is further processed using the V-disparity map. A second constraint line is introduced representing the bound on the obstacle height in the scene (see Figure 4.18). This doesn't prevent taller objects from being detected but allows a substantial portion of the background to be culled, taller objects above three metres are truncated (they will still be represents, abet with the under estimated height)

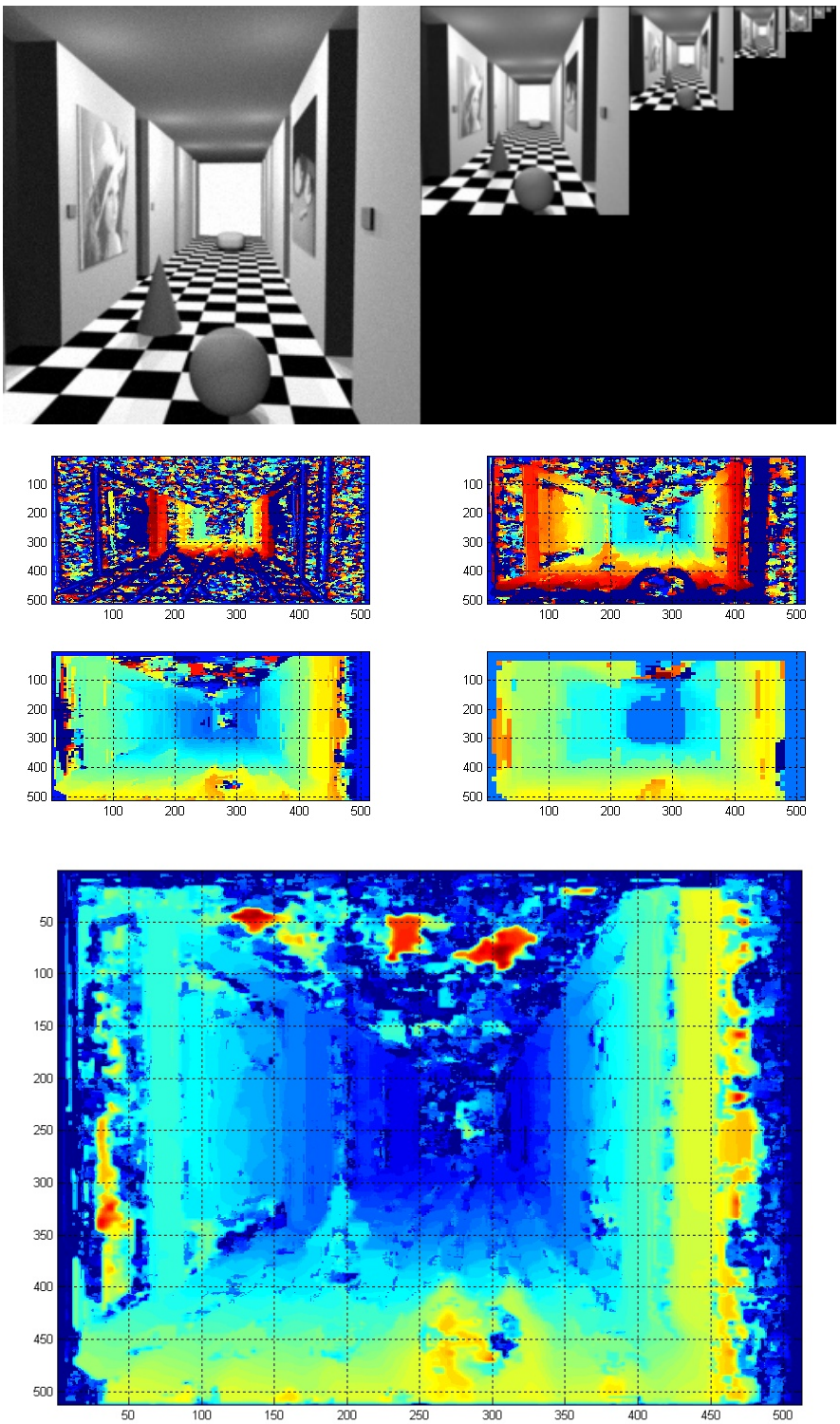


Figure 4.14: top: Grey-scale image, middle: disparity map at different pyramid levels bottom: estimated disparity from image pyramid (blue is far, yellow is close)

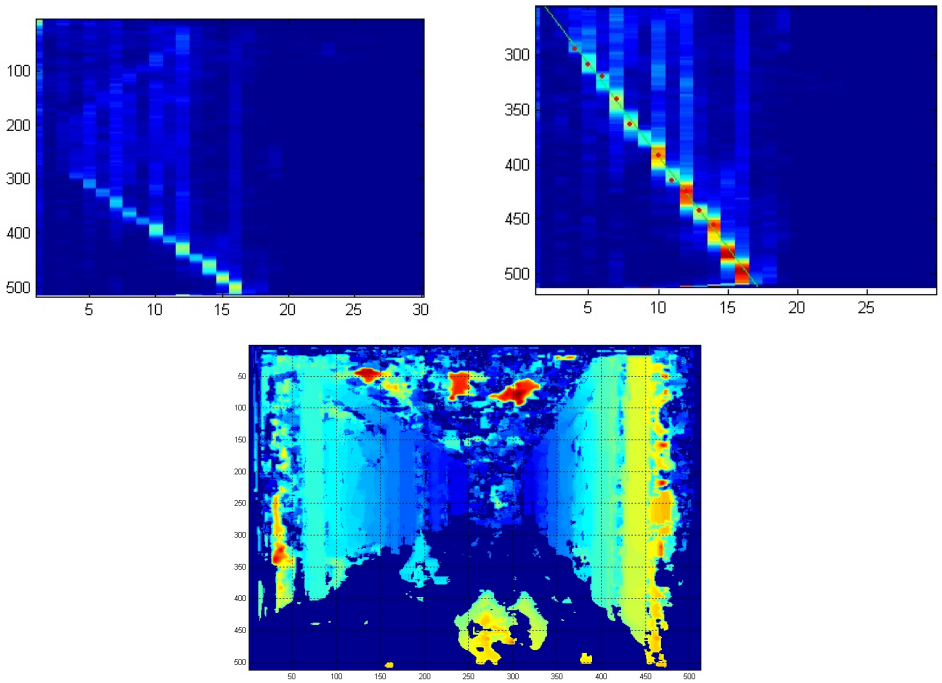


Figure 4.15: top: Grey-scale image, middle: V-disparity map showing ground curve, middle lower: V-disparity map with ground plane estimate overlaid, bottom: disparity map with ground removed (blue is far, yellow is close)

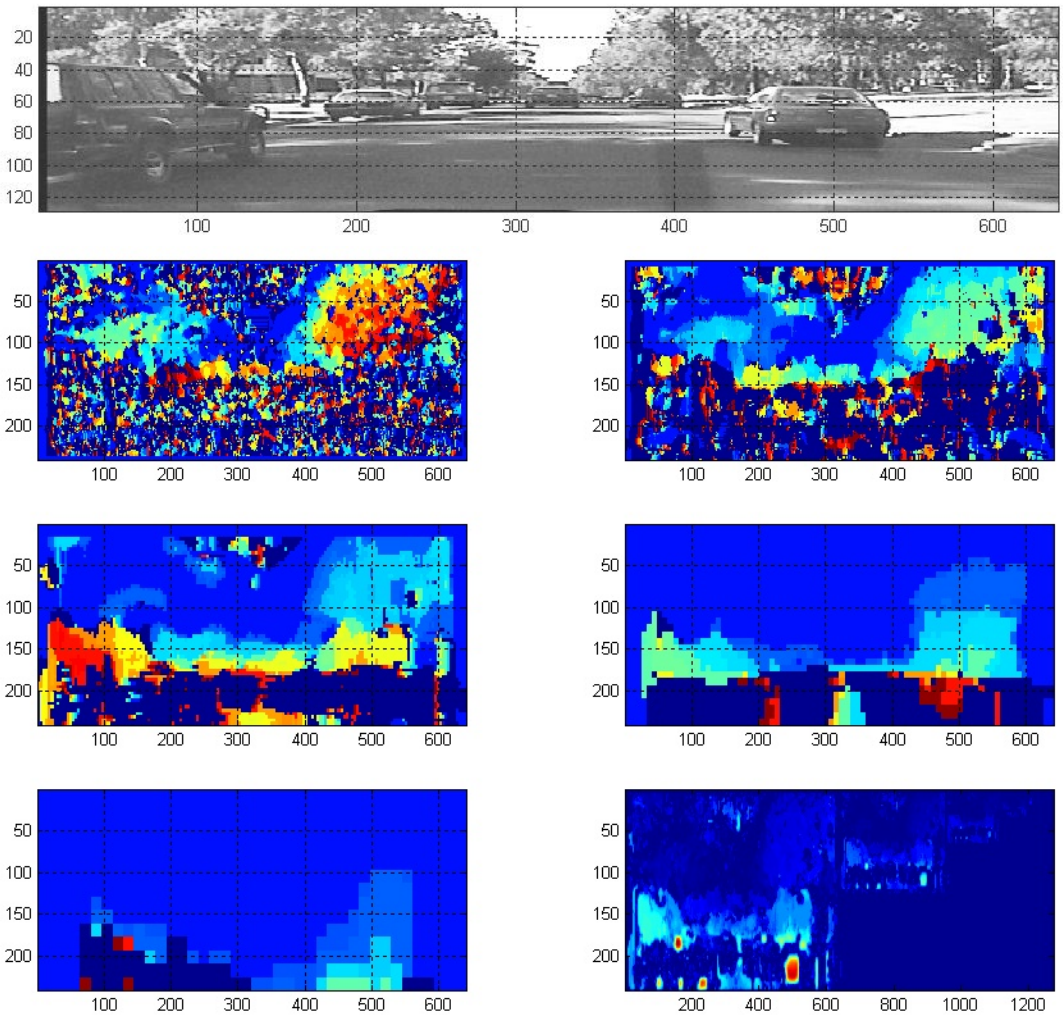


Figure 4.16: top: Grey-scale image, middle: from top left to bottom left (in a raster order) are the 5 resolution levels of the disparity map pyramid from fine to course (blue is far, red is close). bottom right: disparity map during reconstruction.

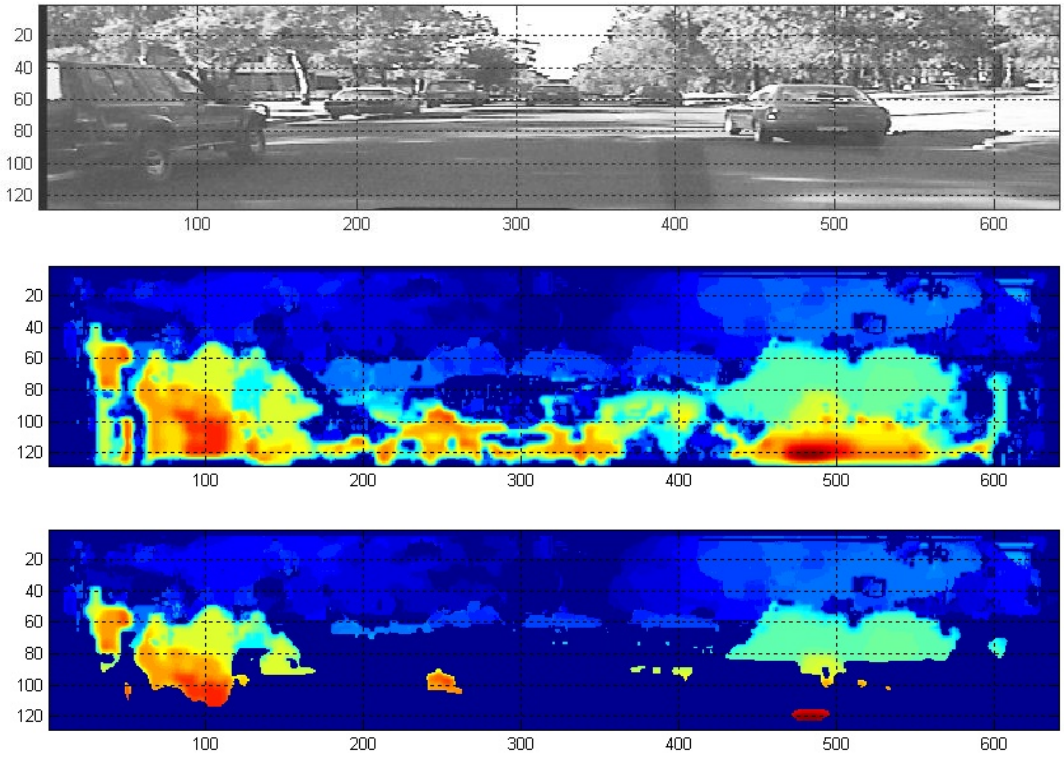


Figure 4.17: top: Grey-scale image, middle: disparity map from pyramid reconstruction bottom: disparity map with ground plane removed (blue is far, yellow is close).

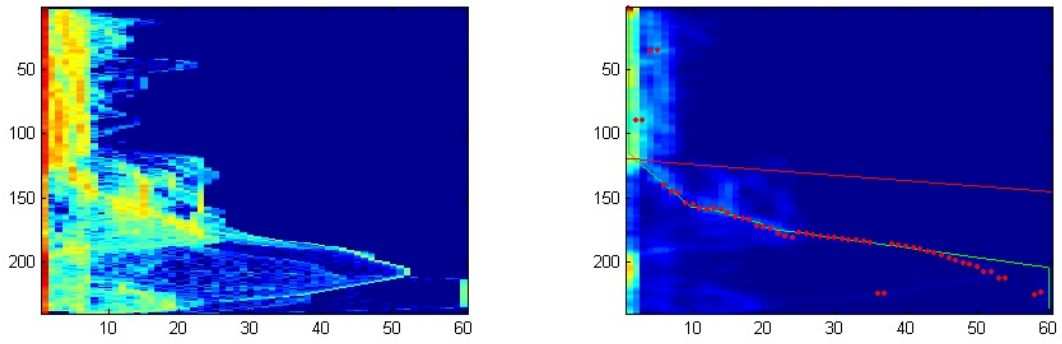


Figure 4.18: left: log(V-disparity map) for clarity, right: V-disparity map with ground plane and maximum obstacle height estimates overlaid.

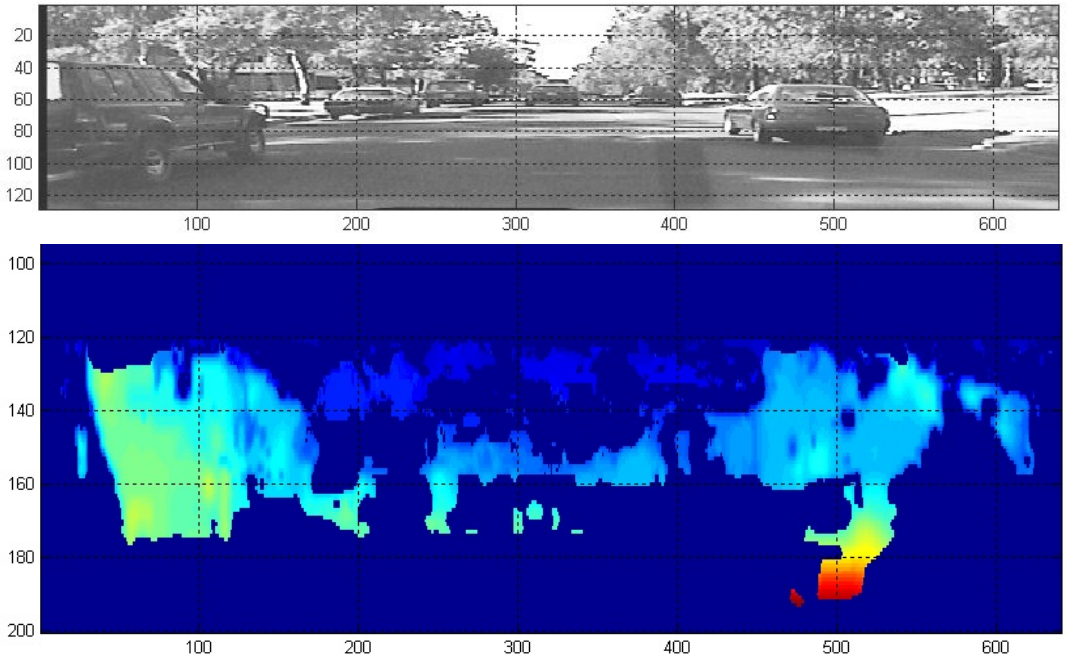


Figure 4.19: top: Grey-scale image, middle: disparity map with ground plane and height restrictions bottom: close up of obstacles remaining (blue is far, yellow is close). Car on left is across a disparity range from 15-23 pixels, car on right at 7 pixels, cars in distance are 4-6 pixels.

4.3.2 Optical flow

The perception of motion is fundamental to the human visual system. Research has shown smooth movement of the eye is not possible without a moving object to watch (Mallot, 2000). Vestibular reflexes are part of automatic ego-motion compensation systems. Experiments on primates have shown that there are intermediate level neurons in the visual pathways that not only can detect motion, but are tuned to detect particular flow directions and magnitudes (Marr, 1982). Other neurons detect overall flow properties like expansive verses contractive flow (Duffy and Wurtz, 1997). Primates have been shown to be predisposed toward a negative optical flow speed gradient. This is the most common gradient experienced when moving forward, the predisposition could represent an In-built ego-motion compensation mechanism.

In computer vision the estimation of motion projected into the camera image is termed optical flow (Horn and Schunck, 1981). Optical flow has been used in many applications in the past 20 years from detecting intruders to steering mobile robots (Okada *et al.*, 2001). Like stereo disparity optical flow can be computed in a dense field using pixel based techniques or a sparse set of points using feature based matching. Optical flow however, is well known for being computationally expensive and generating noisy results.

In order to use optical flow for road vehicle detection the we need to get an estimate of the flow speed ranges to expect. Using the assumptions listed in Table 4.6 we have estimated the expected flow for a several focal lengths and ego-motion down the camera axis (see Figure 4.20). The range of flow is large. Correlation based flow methods operate best with flow changes of the order of a few pixels between frames, while gradient based methods (Barron *et al.*, 1992) estimate flow of the order of a pixel or less.

Flow motion range Analysis:

Assuming:	Value	Unit
frame rate	30	hz
height h	1.6	metres
tilt angle	0	degrees
ccd_x	0.0048	metres
ccd_y	0.0036	metres
pix_w	640	pixels
pix_h	240	pixels
height displacement	3	metres
ground lateral displacement	3	metres

Table 4.6: Flow range settings

Longitudinal Distance	3	5	10	30	50	70	90
relative velocity	**60 km/h**		**16.6667 metres/s**		**0.55556 metres/frame**		
focal	**0.0054 metres**						
Delta theta u (degrees)	-5.82634	-3.05559	-0.92305	-0.10701	-0.03849	-0.01961	-0.01185
Delta theta v (degrees)	-4.78405	-1.84218	-0.46224	-0.05034	-0.01801	-0.00916	-0.00553
Delta pix u (pixels)	-73.4694	-38.4342	-11.6004	-1.34479	-0.48363	-0.2464	-0.1489
Delta pix v (pixels)	-30.1291	-11.5788	-2.90443	-0.31628	-0.11317	-0.05758	-0.03477
focal	**0.016 metres**						
Delta pix u (pixels)	-217.687	-113.879	-34.3716	-3.98456	-1.43299	-0.73008	-0.44119
Delta pix v (pixels)	-89.2715	-34.3074	-8.60572	-0.93712	-0.33531	-0.1706	-0.10303
focal	**0.05 metres**						
Delta pix u (pixels)	-680.272	-355.872	-107.411	-12.4517	-4.47808	-2.28149	-1.37872
Delta pix v (pixels)	-278.973	-107.211	-26.8929	-2.92851	-1.04786	-0.53312	-0.32198

Longitudinal Distance	3	5	10	30	50	70	90
relative velocity	**110 km/h**		**30.5556 metres/s**		**1.01852 metres/frame**		
focal	**0.0054 metres**						
Delta theta u (degrees)	-11.5555	-6.03384	-1.77114	-0.1993	-0.07122	-0.03619	-0.02184
Delta theta v (degrees)	-10.2259	-3.73085	-0.89015	-0.09376	-0.03333	-0.01691	-0.0102
Delta pix u (pixels)	-147.212	-76.1051	-22.2639	-2.50443	-0.89501	-0.45476	-0.27441
Delta pix v (pixels)	-64.9424	-23.4748	-5.5934	-0.58909	-0.20944	-0.10627	-0.06408
focal	**0.016 metres**						
Delta pix u (pixels)	-436.183	-225.496	-65.967	-7.42052	-2.65188	-1.34744	-0.81305
Delta pix v (pixels)	-192.422	-69.555	-16.573	-1.74544	-0.62055	-0.31486	-0.18988
focal	**0.05 metres**						
Delta pix u (pixels)	-1363.07	-704.676	-206.147	-23.1891	-8.28713	-4.21075	-2.54079
Delta pix v (pixels)	-601.318	-217.359	-51.7907	-5.45451	-1.93922	-0.98394	-0.59337

Figure 4.20: Flow range: pixel change between images for a range of focal lengths. The point of interest is displaced 3 m from the centre line and 3m above the road.

Results and discussion

We use an implementation of Fleet and Langley (1995)'s local least squares optical flow using infinite impulse response filters to find the temporal derivative with a small temporal delay. A three level Gaussian pyramid is used to cover the a wider flow range. The result of this technique is shown in figure 4.21.

We investigated modifying the implemented flow algorithm to compute tensor based total least squares instead as the Matlab results were promising. Using the tensor implementation the analysis of the eigenvalues enabled us to identify corner points used for template tracking in the tracing phase of the system without additional computation. The corner points detection algorithm generates better than points identified by a the Harris corner detector because the corner features have to withstand significant spatio-temporal smoothing, instead of being a strong feature in a single image. Figure 4.22 shows the result of the total least squares

Optical flow algorithm:

1. Rectify gross image errors based on calibration data, particularly rotations about optical axis.
2. Use template tracking to ensure the central horizon has zero disparity between images.
3. Create an n-level Gaussian pyramids for the right image according to Table 4.2.
4. Add each level of the pyramid to recursive optical flow algorithm of (Fleet and Langley, 1995).
5. Use the total least squares approach to find the reliable vectors.
6. Reconstruct the final flow image from the levels of the pyramid according to Table 4.4.
7. Compute and subtract the ego-motion from the flow vectors.
8. Segment sections of significant difference from ego-motion estimate.

Table 4.7: Stereo disparity map algorithm.

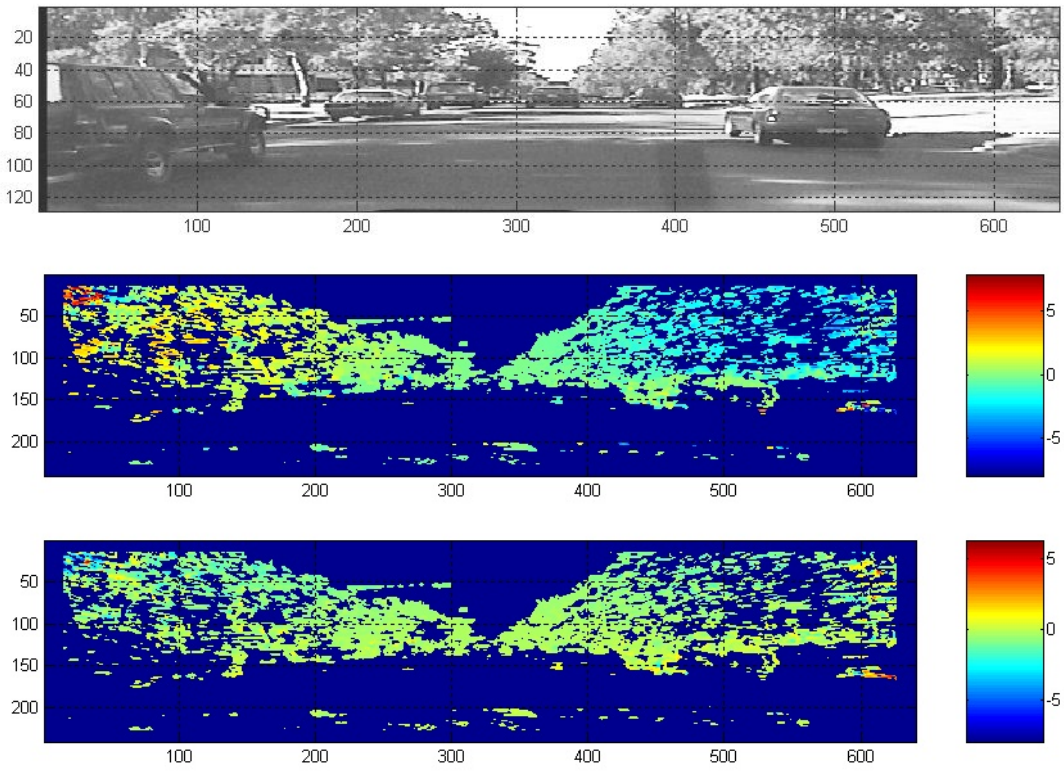


Figure 4.21: top: Greyscale image, middle: flow in x direction bottom: flow in y direction (up$>$ 0, left$>$ 0)

tensor method. The incoherent motion metric clearly shows the edges of the vehicles in the road scene as the vehicle motion contradicts the ego-motion.

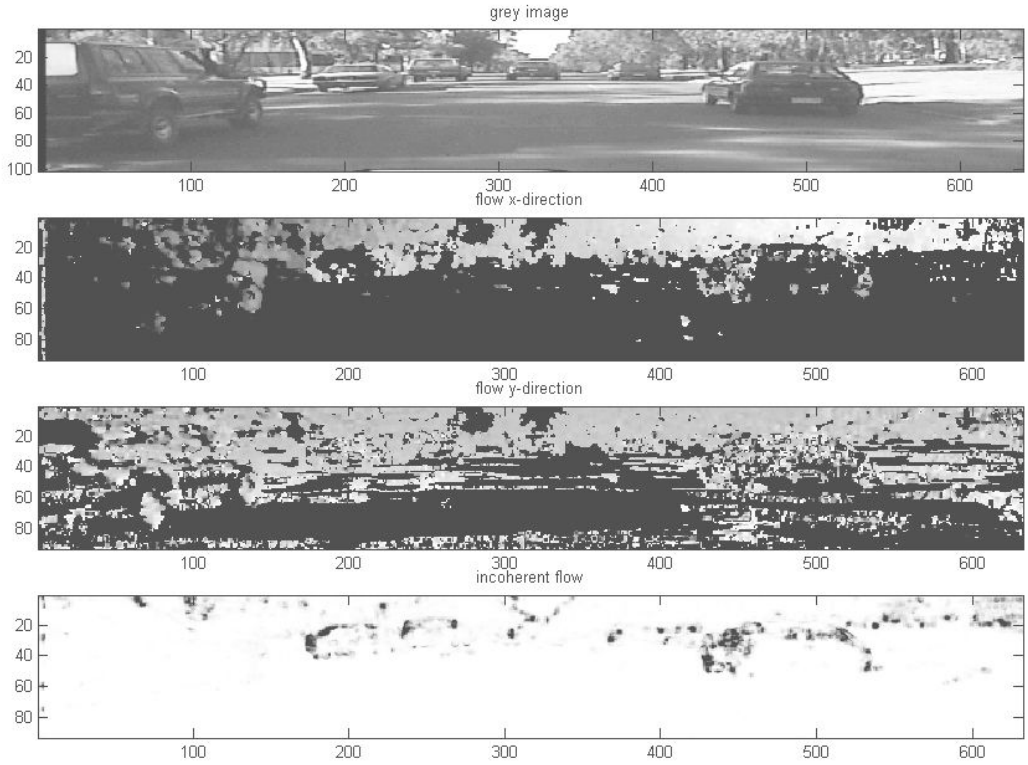


Figure 4.22: Total least squares tensor method. Top: Original grey image, Middle: flow in x then y directions, Bottom: incoherent motion shown as dark regions

Ego-motion compensation

Given a known scene geometry and a known camera motion it is possible to compute the expected optical flow. For an arbitrary camera motion the expected optical flow can be written as a function of the internal camera matrix $\mathbf{K}$, the extrinsic camera motion in terms of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$ as for some 3D point in the scene $\mathbf{X}$:

$$\begin{aligned} \mathbf{v} &= f(\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{X}) && (4.18) \\ &= diff(\mathbf{x}_t, \mathbf{x}_{t-1}) && (4.19) \\ &= diff(\mathbf{K}.[\mathbf{R}|\mathbf{t}].\mathbf{X}, \mathbf{K}.[\mathbf{I}|\mathbf{0}].X) && (4.20) \end{aligned}$$

We use $diff(,)$ to represent the subtraction of the 2D inhomogeneous points. Because the latter part of the 2nd equation assumes the use of homogeneous points, subtraction of the 2D homogeneous points is not equivalent.

We use image matching to compensate for rotation of the camera (which becomes a translation in the camera image). We ignore he effect of camera rotation between frames, for small rotations and obstacles deep in the scene the image matching removes the effect of rotation. For closer obstacles unmodelled rotations in nearer objects induces phantom detections.

We use the greatly simplified, still effective relation:

$$\mathbf{v} = \begin{bmatrix} \frac{a_i\,(X_z\,t_x - X_x\,t_z)}{(X_z + t_z)X_z} \\ \frac{a_j\,(X_z\,t_y - X_y\,t_z)}{(X_z + t_z)X_z} \end{bmatrix} \tag{4.21}$$

where: a_i and a_j are the focal lengths in pixels in the $i = x$ and $j = y$ directions, the 3D scene point is decomposed into: $\mathbf{X} = [X_x, X_y, X_z]^T$, the translation between frames is decomposed into: $\mathbf{t} = [t_x, t_y, t_z]^T$.

Note in this equation is that optical flow diminishes at the inverse square of distance.

In a vehicle with a forward looking camera since we assume the optical axis is close to parallel with the road we expect the translation along the Z axis will be proportional to the speed of the vehicle, translations in the X and Y directions will be minor in comparison to the Z axis. This simplifies the equation to:

$$\mathbf{v} = \begin{bmatrix} \frac{a_i\,(-X_x\,t_z)}{(X_z + t_z)X_z} \\ \frac{a_j\,(-X_y\,t_z)}{(X_z + t_z)X_z} \end{bmatrix} \tag{4.22}$$

If we substitute in the relation between a 3D scene point and it's 2D projection on the first camera ($x = \mathbf{K}.[\mathbf{R}|\mathbf{t}].\mathbf{X}$) we get:

$$\mathbf{v} = \begin{bmatrix} \frac{(i_0 - i)\,t_z}{(X_z + t_z)} \\ \frac{(j_0 - j)\,t_z}{(X_z + t_z)} \end{bmatrix} \tag{4.23}$$

where: i and j represent the screen coordinates, i_0 and j_0 represent the principle point.

Note that this equation is now independent of the focal length. However we have introduced the principle point.

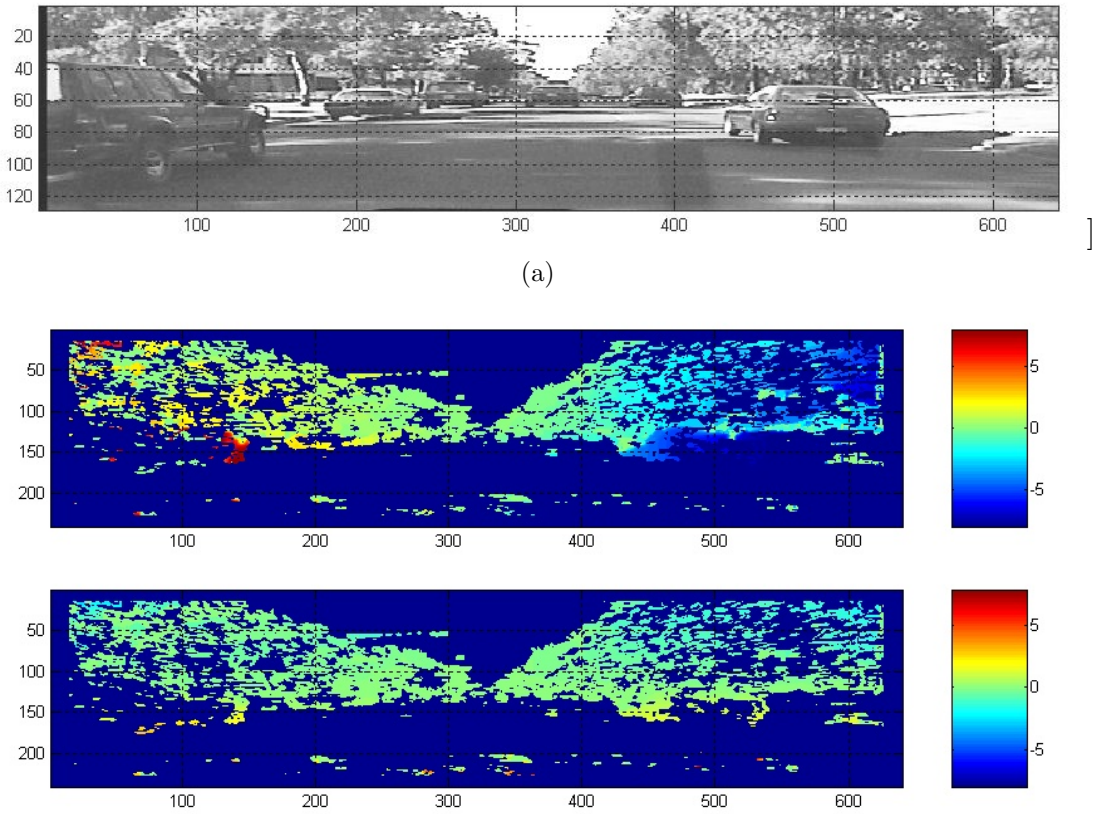
Finally, if we use the relationship between the depth X_z and the disparity d, namely: $X_z = \frac{a_i\,t_{baseline}}{d}$ where: $t_{baseline}$ is the baseline length of the stereo camera platform. We obtain an equation in terms of the pixel disparity, and distance travelled:

$$\mathbf{v} = \begin{bmatrix} \frac{(i_0 - i)\,t_z\,d}{(a_i\,t_{baseline} + t_z\,d)} \\ \frac{(j_0 - j)\,t_z\,d}{(a_i\,t_{baseline} + t_z\,d)} \end{bmatrix} \tag{4.24}$$

This simplified ego-motion estimation equation was found to work surprisingly well for straight vehicle motion and gentle turns

Results and discussion

Figure 4.23 shows the resultant ego-motion field based on the disparity map and velocity estimate from the tail shaft encoder. Figure 4.24 shows the log of the squared error between the estimated flow and ego motion estimate. The vehicles show a significantly greater error than the background. Note that the error on the trees at the sides of the image is due to an aperture problem.



(a)

Figure 4.23: top: Grey-scale image, middle: estimated ego flow in x direction bottom: estimated ego flow in y direction (up> 0, left> 0)

Segmentation

The resultant maps of the stereo cue and ego compensated flow cue are combined to produce an overall potential obstacle map. The map is then segmented using ranges in the pixel disparity. The disparity range reflect the inaccuracy of the

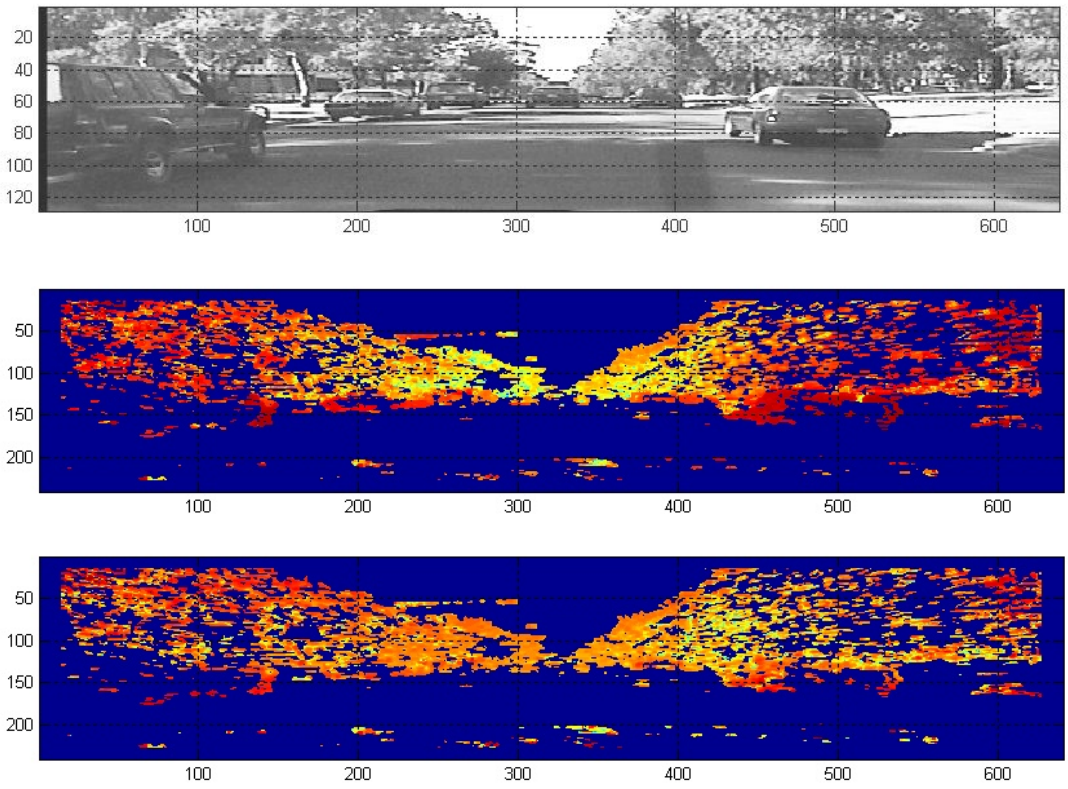


Figure 4.24: top: Grey-scale image, middle: $log((flow - ego)^2)$ in x direction bottom: $log(abs(flow - ego)^2)$ in y direction (red = large error, blue = small error)

disparity map at large distances and the fact that obstacles will span across several disparities in the near field (see figure 4.25). In the next phase, the Distillation algorithm consolidates duplicate detected objects. It is acceptable that the detection process generates multiple detections of the same obstacle rather than miss potential obstacles. These multiple detections or false positives are filtered out in the distillation phase.

4.3.3 Performance

Figure 4.26 shows the segmented disparity range and the potential obstacles identified. As mentioned previously the disparity ranges represent either the limit of the disparity error (± 1 pixel in the far field) or the approximate size of expected obstacles (disparity equivalent to ± 4 metre long object). The blobs are accepted only if they have a non-trivial size (an area greater than a metre) for their disparity (distance). Figures 4.27 and 4.28 show the bounding boxes of obstacles identified in the scene. Some sharp edges of shadows also cause false obstacles and obstacle boundaries can be detected, as mentioned above false detections are

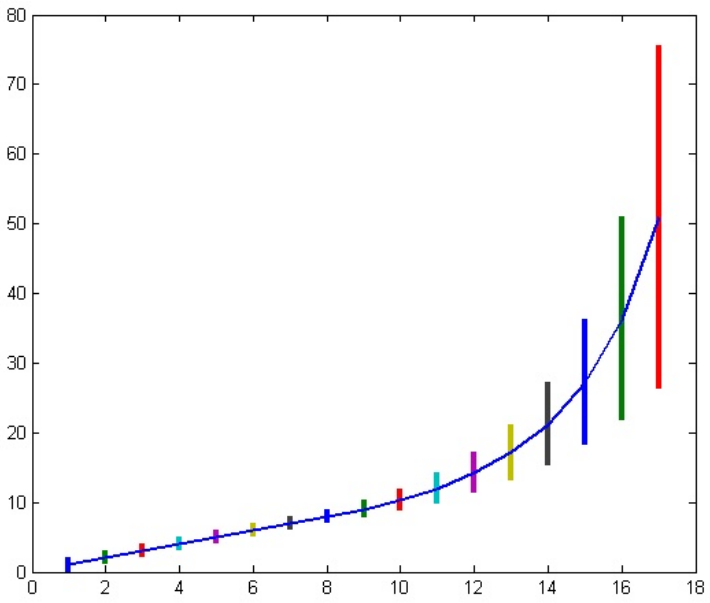


Figure 4.25: Ranges of disparity segmentation. The y-axis is disparities, the x-axis numbers the 17 different ranges.

preferable to false negatives. The distillation phase verifies that the tracking of obstacles is consistent over time.

The false negative rate per frame in our experiments was less than 8%, while the false positive rate per frame was 200 to 600%. The false negative rate across three consecutive frames was less than 2%.

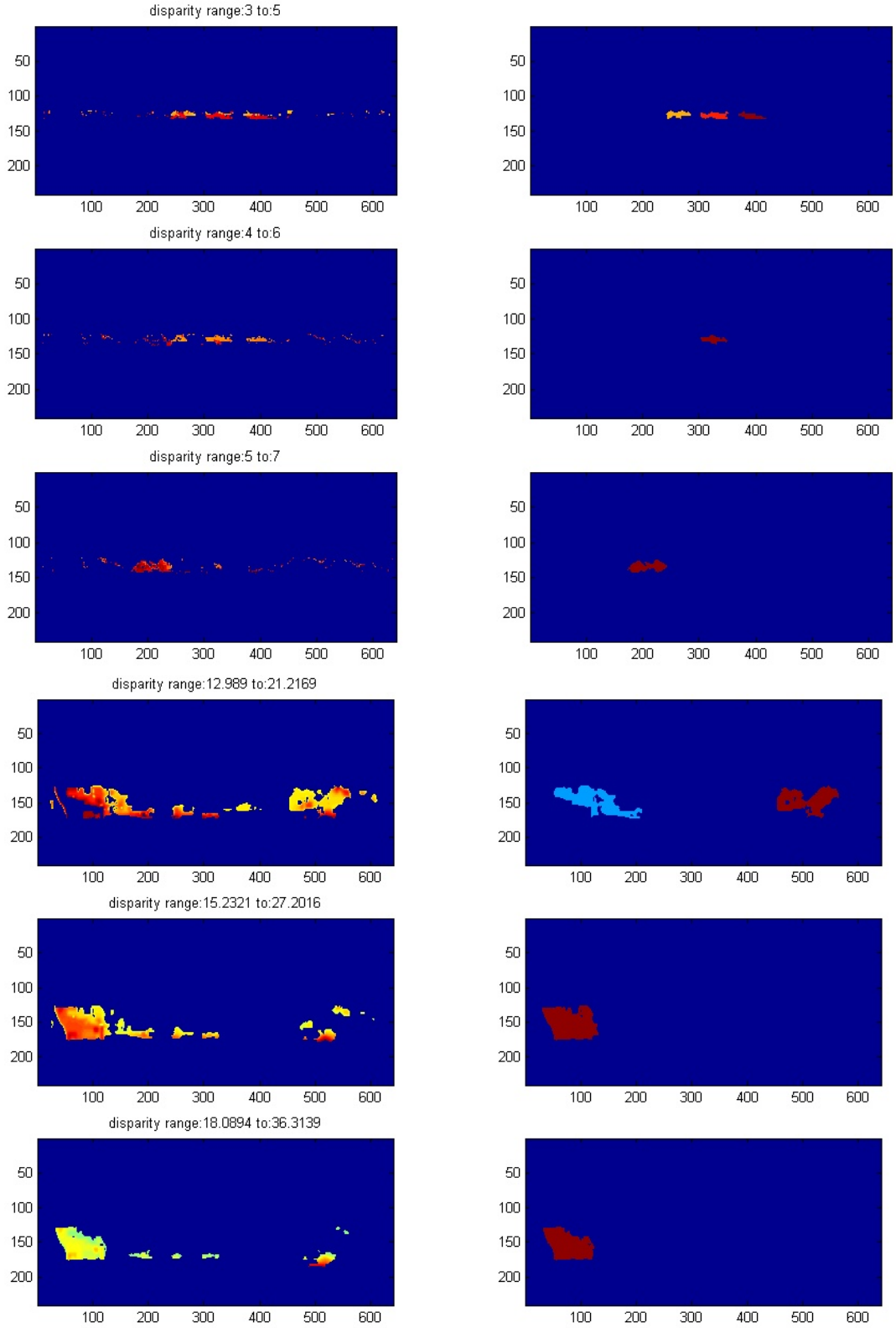


Figure 4.26: left:segmented disparity map. right: potential obstacles identified.

Figure 4.27: Detected obstacle candidates in multi-vehicle sequence. The video of this sequence is on the Appendix DVD-ROM (Page 257)

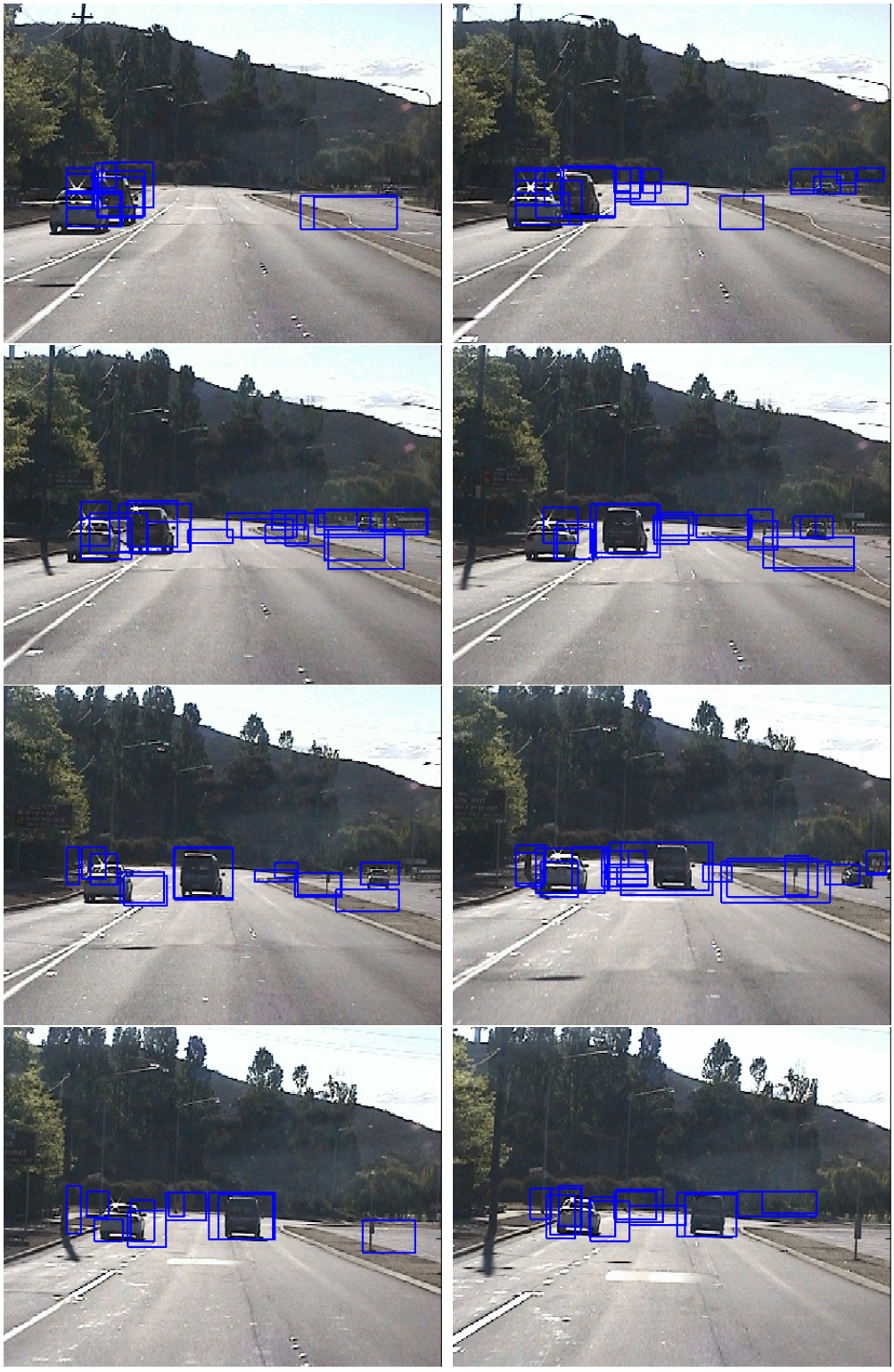


Figure 4.28: Detected obstacle candidates in two vehicle sequence. The video of this sequence is in the Appendix DVD-ROM (Page 257)

4.4 Distilling obstacles

The distilling of real obstacles from detected obstacles is done using the distillation algorithm. A full explanation of our multiple cue and multiple hypothesis computational framework was given in Chapter 3

To robustly track obstacles we need to deal with detected obstacles from the first phase and monitor them over time to refine the estimate of the obstacle's position and motion. In fact, as with human vision system (Mallot, 2000), rather than attempting to make the initial detection algorithm 100% accurate we tolerate false positives and check if potential obstacles stand the test of time. We expect many false detections from the noisy disparity and optical flow segmentation process, when tracked over a period of time we expect these false matches to behave inconsistently meaning that the tracking algorithm will eventually diverge from phantom objects. Real detected obstacles will behave in a consistent way allowing convergence in tracking. Once the tracking has found to have suitably converged the obstacle representation is moved into phase three.

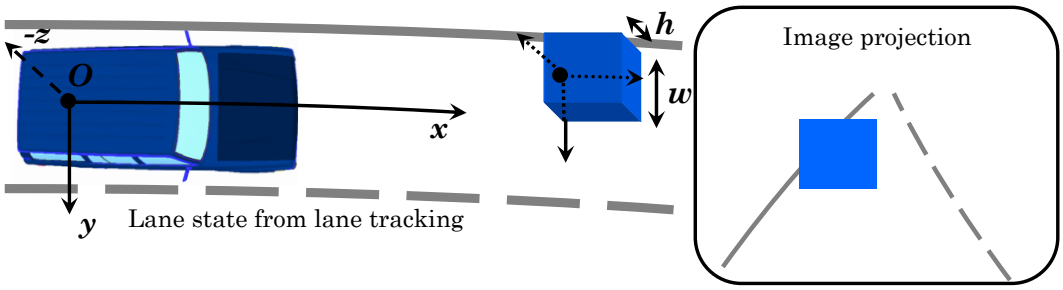


Figure 4.29: State model for obstacle detection.

Tracking obstacles the state space is defined in relation to the estimated lane position (see Figure 4.29). An obstacle candidate is represented by a 3D point with respect to the road at the vehicle centre of gravity. The height and width of the obstacle are estimated, length is modelled as the same as the width . The velocity or change in the lateral and longitudinal displacement is also estimated. This gives a seven dimensional state ($y(k), x(k), z(k), w(k), h(k), dy(k), dx(k)$). A constant velocity motion model is used to model obstacle motion:

$$x(k+1) = x(k) + dt.dx(k) \quad (4.25)$$

$$y(k+1) = y(k) + dt.dy(k) \quad (4.26)$$

Where dt is the time interval between filter interations k and $k+1$.

By default around 2500 particles were used to search the 7 dimensions. For each obstacle candidate from the detection phase fifty particles are injected around the state location of the candidate. The remaining particles were resampled according to the current overall probability distribution.

4.4.1 Preprocessing

The preprocessing involves the production of image pyramids, flow and disparity maps. There was some reuse of the image pyramids between the cues. Figure 4.30 illustrates the implemented resource sharing. The preprocessing was also shared between the detection and distillation phases.

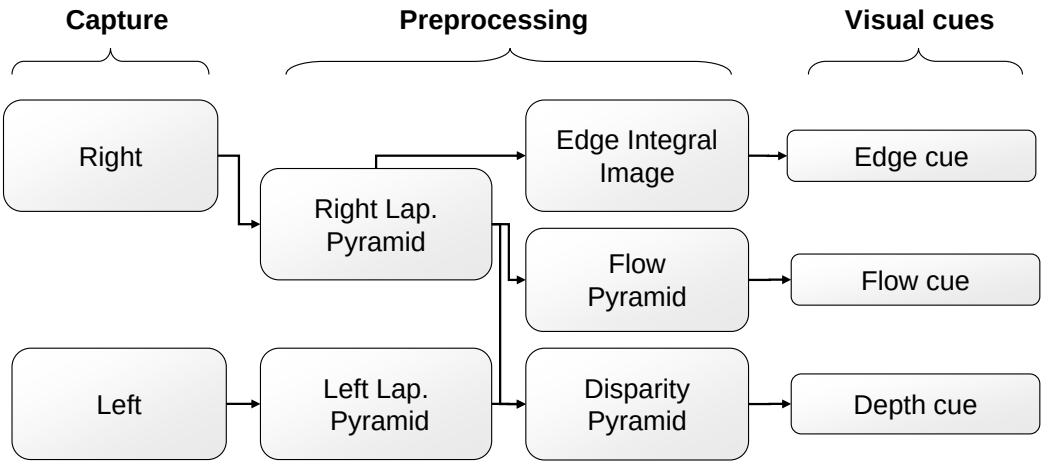


Figure 4.30: Preprocessing for obstacle distillation.

4.4.2 Visual cues

Similar to lane tracking, visual cues are used to evaluate the hypotheses represented by the particles compared with the image data.

Likewise Equation 4.27 is used to evaluate the sensor model.

$$P(e_t^{(j)}|s_t^i) = \frac{1}{\epsilon + N}\left(\epsilon + \sum_p^N I_t^{(j)}(s_p^{(i)})\right) \tag{4.27}$$

$s_p^{(i)}$ is the p^{th} pixel from the set of pixels S generated by particle i. $I_t^{(j)}(s)$ is the value of pixel s from the observation image I used of the j^{th} cue. N is the number of pixels in the region in image space. ϵ (set to 0.001) is used to support the possibility that the sensor is in error (as discussed in Chapter 3).

Stereo Disparity Cue: This cue measures the disparity across the projected bounding box of the obstacle candidate. The cue returns as probability inversely proportional to the deviation of the disparity from the sample depth.

Optical Flow Cue: This cue measures deviations from the expected ego-motion

flow in the projected bounding box. This cue only works for non-stationary objects, particularly moving in an opposite direction to the ego-motion.

A free space cue is used in addition to optical flow and stereo for the distillation phase.

Free space Edge Cue: This cue takes the highest resolution Laplacian image from the laplacian pyramid and computes an integral image. The area inside the projected bounding boxes from the particle filter hypothesis are then evaluated for sufficient texture for an obstacle.

4.4.3 Performance

Figure 4.31 shows the distillation algorithm tracking the leading vehicle on the road. This sequence comes from the Canberra-Geelong sequence were a handi-cam was used to collect the data so stereo vision is not available. Instead optical flow is used for the obstacle detection phase.

4.5 Tracking obstacles

Once the Distillation algorithm has some single mode clusters reliably following potential obstacles, an extended Kalman filter is spawned to track the vehicle more efficiently and free particles in the Distillation algorithm to track other emerging features. The modes were extracted by detecting the first minimum turning point in the histogram of the particle probabilities. The sample mean and variance computed, then these particles were excluded and the process repeated.

In this phase obstacles are tracked in the image space using normalised cross correlation (NCC). In this case because we are correlating a small number of templates compared with the disparity map generation, we can use the NCC more computationally expensive algorithm. Features are selected using the tensor flow corners identified in the previous section. The top n unique regions on the object. The features are tracked in both left and right images to provide a depth estimate, or for monocular footage the depth is estimated by the projected position in the road image.

If the template correlation is lost in consecutive frames the location of the obstacle is checked. If the obstacle is likely to have been passed then the spawned filter is destroyed. If tracking fails and the previous cases didn't appear to apply, particles are be injected into the particle filter at the last known location similar to when the vehicle was originally detected.

Figure 4.32 shows a the motion of an obstacle tracked using 15 points identified with a Harris point detector inside a given bounding box of the object. The object

Figure 4.31: Distillation algorithm tracking leading vehicle.

is then tracked using normalised cross correlation and a Kalman filter estimating the vehicle position. The vertical motion in the image is actually due to pitch in the experimental vehicle. Most research questions involving using an extended Kalman filter to track have been explored so little refinement was done on this phase. In fact

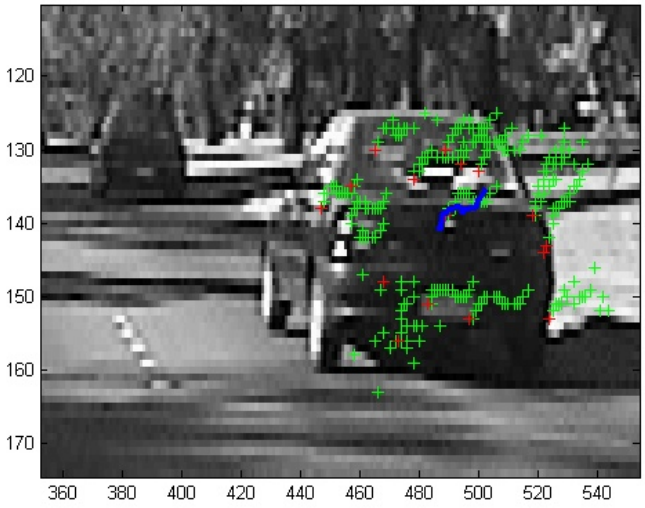


Figure 4.32: 15 points detected via Harris corner detector tracked using correlation and Kalman filter over 20 frames. Blue line represents object motion from right to left.

4.6 Performance

Figure 4.34 shows the obstacle tracking system following a vehicle. The video of this sequence is on the Appendix DVD-ROM (Page 257). The lane estimate is also shown as the road curvature is significant in this example illustrating how the obstacle detection system can use the lane estimate to maintain a physically grounded view of the road scene. The motion of the obstacle down the road is simply down the x axis of the obstacle detection state space, instead of the non-linear 3D motion that would be induced in a straight road, flat earth state space approach.

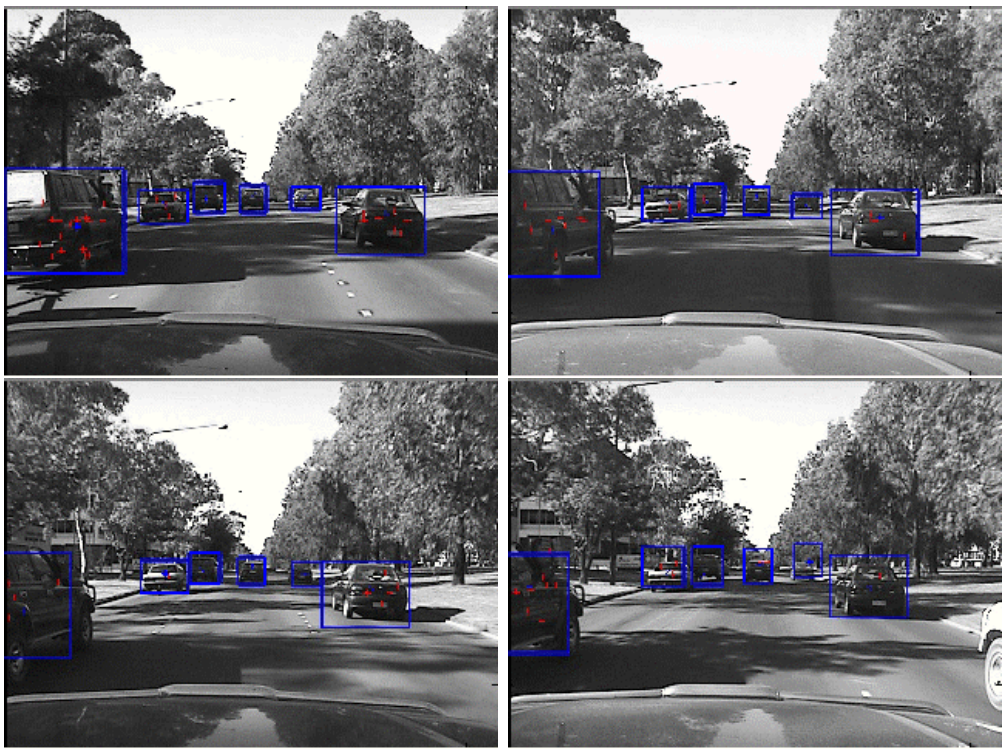


Figure 4.33: Tracking multiple obstacles using separate Kalman filters and correlation.

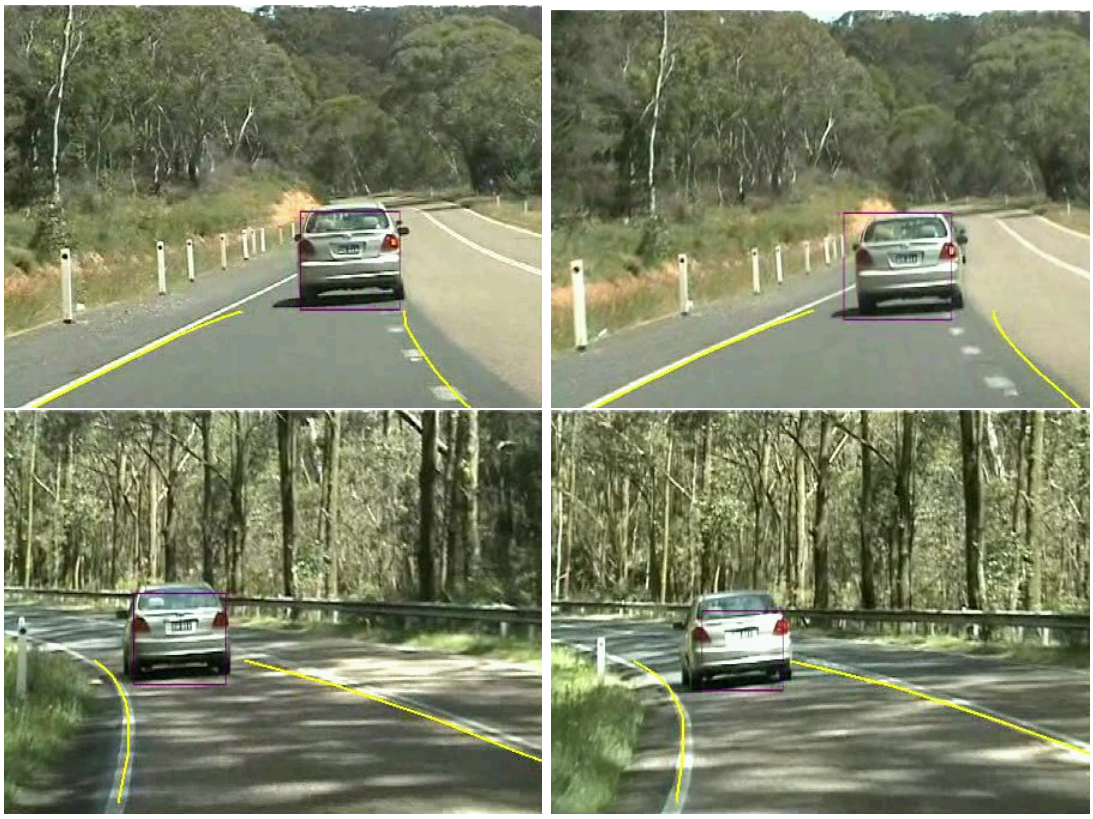


Figure 4.34: Obstacle detection and tracking. The video of this sequence is on the Appendix DVD-ROM (Page 257).

4.7 Summary

An obstacle detection and tracking system has been proposed that combines "top down" and "bottom up" approaches to detect and track obstacles. Similar in this way to the visual mechanisms thought to work in animals. Using a variety of visual cues stereo vision, optical flow and free space are obstacles detected and tracked in varying road environments. Cues which would often be too noisy to use for detection have been integrated in our framework in a three stage process of detection, distillation and tracking. Also the benefit of using the lane tracking state estimate as the foundation of the coordinate system for the obstacle detection system can be seen with curved non-flat pitch road, road obstacle detection.

One important object in the road scene requires not only detection but also recognition. Next will now examine road sign recognition.

Chapter 5

Road sign recognition

Road signs provide important information for interpreting the local road context. For autonomous systems road sign recognition is a crucial system component to maintain a comprehensive model of the road scene. Drivers may not notice a road sign due to distractions, occlusions or inattention due to another driving task. Road signs provide a classic example of how driver inattention, due to a missed road sign, can heighten crash risk a scenario that is addressed by our Automated Co-driver system.

In this chapter, after a review of road sign recognition in Section 5.1, we introduce a highly effective technique for detecting symbolic road signs (Section 5.2). The detection method is described and paired with a simple classifier to make a road sign recognition system in Section 5.3. The system is tested in Section 5.4. A significant issue when developing the sign recognition system is the small image size and low image quality of the road sign images to classify. We address these issues in Section 5.5 by developing an online image enhancement technique to create an enhanced image from a string of small, poor quality, road sign images.

Our work has concentrated on the detection of speed signs. Speed signs are arguably the most important road sign, as speeding is a leading contributing factor to road fatalities. In Chapter 2 we reported on research indicating that missed signs between speed zones causes higher speeds over the set speed limit than intentional speeding, substantially increasing the crash risk.

The approach we have developed is readily extensible to other symbolic road sign detection, such as stop, give way and roundabout signs as well as symbolic traffic signals such as traffic lights.

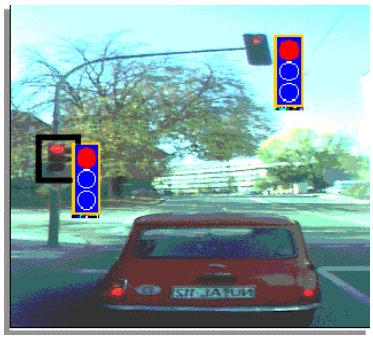
5.1 Review of road sign recognition techniques

We concentrate on symbolic road sign recognition as these are the signs which convey the most fundamental road scene information to the driver. Road signs included in this category are speed signs, stop signs, give way and roundabout signs. Road signs not in this category are informational signs that are used to navigate such as street names and highway route signs.

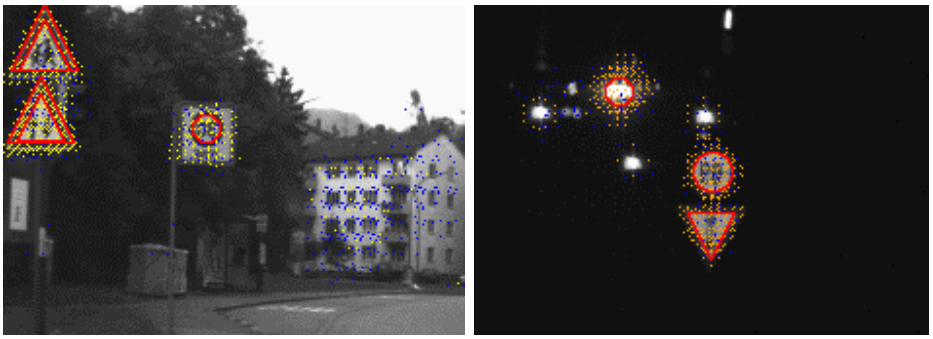
It is prudent to note that alternate methods for road sign information messaging have been proposed. Two alternate methods are GPS digital map tagging (Vaughn, 1996) and radio beacons (Tuttle, 2000). GPS map tagging works by inserting all speed zones into digital maps used in GPS navigational aids, as the vehicle approaches a position the speed is updated from the map. Radio beacon systems use active or passive (RFID tags) beacons positioned on each sign to broadcast the sign information to receivers fitted to each vehicle (Vaughn, 1996; Tuttle, 2000). The final solution for in-vehicle systems as acknowledged by vehicle technology commentators (Bishop, 2005) will almost certainly use a combination of approaches. Every technique has certain advantages. The key to our approach is that the Automated Co-driver has the same information as the driver, and that we do not need any additional road infrastructure to achieve this. The principal issue is then one of scene understanding - the driver and system must always have consistent information. Both systems read the same visual cue placed on the road way, as this way there is no possibility of an inconsistency between the road signs observed and any other parallel information source such as an out of date digital map or a missing RFID tag.

The most popular means of detecting road sign and signal features has been colour segmentation (Priese *et al.*, 1994; Piccioli *et al.*, 1996; Paclik *et al.*, 2000; Johansson, 2002; Fang *et al.*, 2003; Shaposhnikov *et al.*, 2002; Hsu and Huang, 2001). Most methods use normalised colour spaces to provide some regulation against varying illumination, while normalised colour spaces can help when the ambient light level varies in intensity (such as mild cloud cover). Most normalised colour spaces and colour segmentation techniques are not effective when the light source colour changes, which occurs at dawn or sunset, during heavy cloud cover, fluorescent, tungsten, sodium vapour lighting, or strong shadows (Austin and Barnes, 2003). Franke *et al.* (1999) used colour lookup table and neural network to detect traffic lights (see Figure 5.1). Like most groups they attained recognition rates in the order of 90%, with false positive rates as low as 2%, but due to the failure of the apparent colour assumption, improving the technique for a higher performance proved difficult. For the newer sign detection systems they have moved to a sign geometry approach (Gavrila, 1998).

One aspect of symbolic road signs that is highly regular is shape. Owing to the orthogonal alignment of signs with the road, the apparent shape of signs relevant to the approaching vehicle is constant. They do not change under different weather conditions or lighting, thereby forming a strong cue for detection. Gavrila (1998)



(a)



(b)

Figure 5.1: **(a):** Colour based traffic light detection achieves 90% detection but due to apparent colour failures is hard to make robust. From (Franke *et al.*, 1999). **(b):** The group Opted for a geometric technique for symbolic road sign recognition. From (Gavrila, 1998).

used a distance transform to find symbolic road signs by their symmetric shape. This technique appears quite effective, however, the distance transform is quite computationally expensive. This group used a series of template correlations to estimate the distance transform in real-time.

The key insight into our sign detection technique is the observation that symbolic road signs are positioned to face the driver (and the driver assistance system). Therefore unlike other objects in the road scene, from the driver's view point a circular sign, for example, will appear as a circle and not an ellipse due to 2D image projection. Using a strict shape detection algorithm and checking for circles consistent in time and space is sufficient to substantially reduce the volume of information that needs to be further processed into a manageable set of regions of interest.

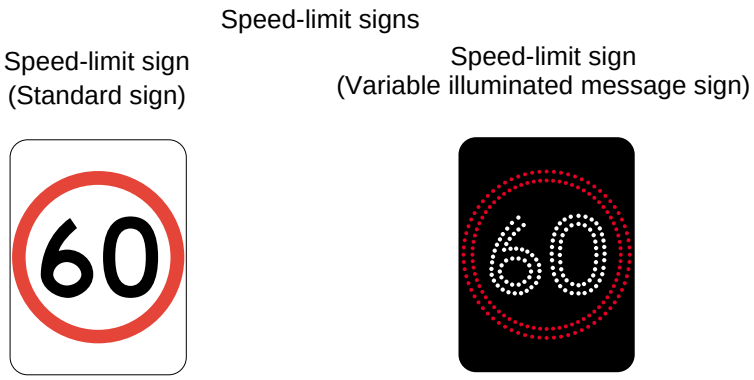


Figure 5.2: Australian speed sign geometry. For lower speeds such as '40','50' and '60', the speed signs are a minimum of $50x70cm$. Faster speeds and wider roads have larger signs. From (ANRTC, 1999).

5.2 Detecting road sign candidates

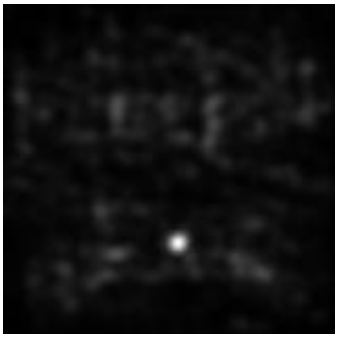
Road signs are detected by locating sign-like shapes in the input image stream. Australian speed signs are required to have a dark (typically red) circle enclosing the speed limit (see Figure 5.2)(ANRTC, 1999). The circles provide a strong visual feature for locating speed signs. We apply the fast radial symmetry transform(FRST)developed by (Loy and Zelinsky, 2003) to detect the circular features, thereby identifying potential speed sign candidates.

The Fast Radial Symmetry Transform (FRST) was originally developed by Loy and Zelinsky (2003) and used for detecting eyes in faces. It is a robust algorithm for detecting circles in images (see Figure 5.3(a)). The technique tallies votes from contributing edge pixels in the gradient image in a similar manner to the Hough transform (Hough, 1959). Given a radius r each gradient image pixel votes for a point r pixels away in the gradient direction. The centroid of a perfect circle will receive one vote from each pixel on the circumference of the circle, partial circles receive a proportion of this total. The voting space is scaled to emphasise complete or near complete circles. Circles show up as peaks in the voting space located at the centroid of the circles. The process is repeated for a number of different radii and then interpolated to find an estimate of detected radius.

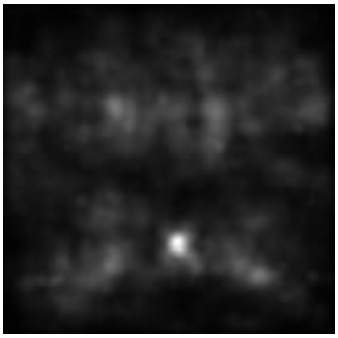
This method can also be extended to detect triangular, diamond (square) and octagonal signs (Loy and Barnes, 2004). In our research we focus on the speed sign case. For completeness a brief definition of FRST is given in Algorithm 5.1. Figure 5.3 shows the result of running the FRST on an image containing a speed sign. The speed sign centroid appears as the dominant maximum in both the response at radius 20pixels (the closest to the true target radius) and the full response (radii 15, 20 and 25 pixels).

(a)

(b)



(c)



(d)

Figure 5.3: (a) fast symmetry transform identifying eyes in faces. (b) Circles in speed signs, input image obtained from the internet. (c) the response at radius 20, and (d) the sum of responses for radii of 15, 20 and 25 pixels.

5.2.1 Application of radial symmetry

Using a shape-based approach to sign detection based on a vote tallying algorithm provides a strong tolerance to changing illumination. The algorithm detects shape based on edges and not on edge segmentation. The voting based scheme means a alternate location must win more votes before the result is effected. The result returns the centroid of candidate signs in addition to an estimate of the scale. Scale tolerance comes from computing a set of FRST images across a range of radii. The resulting image with the highest peak is regarded as the scale closest to the actual radius of the sign in the image.

The subsequent computation for classification is well targeted, and comparatively little further processing is required to assess each candidate. The fast radial symmetry transform was implemented and applied to the road scene image stream.

For speed sign detection we empirically determined a radius search range of 5 to 9 pixels performed for a 320x240 video stream. Signs smaller than a 5 pixel radius tended to lack significant image gradients due to coarse image sampling. Radii above 9 pixels could be used but larger radii indicate a closer sign on the road

Fast Radial Symmetry Transform algorithm:

For a given pixel, p, the gradient, g, is calculated using an edge operator that yields orientation, such as Sobel. If p lay on the arc of a circle, then its centre would be in the direction of the gradient, at distance of the circle radius. Robustness to lighting changes is achieved by applying the discrete form of the detector, and insignificant gradient elements (those less than a threshold) are ignored. The location of a pixel that will gain a vote as a potential shape centroid is defined as:

$$p_{+ve} = p + \text{round}\left(\frac{g(p)}{||g(p)||}n\right), \tag{5.1}$$

where $n \in N$ is the radius, and N is the set of possible radii. (A negative image is defined similarly, facilitating constraining the operator to find only light circles on dark backgrounds and vise-versa.) A histogram image O_n is defined by counting the number of votes awarded to each pixel, and truncated to form $\tilde{O}_n$ as follows

$$\tilde{O}_n(p) = \begin{cases} O_n(p), & \text{if } O_n(p) < k_n, \\ k_n, & \text{otherwise.} \end{cases} \tag{5.2}$$

where k_n is a scaling factor that subsequently normalises $\tilde{O}_n$ across different radii. The response for radius n is then determined as

$$S_n = G * \left(sgn(\tilde{O}_n(p))\left(\frac{|\tilde{O}_n(p)|}{k_n}\right)^{\alpha}\right), \tag{5.3}$$

where G is Gaussian, and α is the radial strictness parameter (typically 2). Each radii of N votes into a separate image to facilitate recovery of radius. The full transform is the mean of the contributions over all radii considered:

$$S = \frac{1}{|N|}\Sigma_{n \in N} S_n. \tag{5.4}$$

See Loy and Zelinsky (2003) for full details.

Table 5.1: Fast Radial Symmetry Transform(FRST) algorithm.

shoulder, often too close and blurring becomes significant due to camera motion. Figure 5.4 shows example images from sequences where speed signs have been detected. A video demonstrating the sign detection technique is included on the Appendix DVD-ROM (Page 257).

The algorithm proved remarkably effective at detecting signs in rural and urban street-scapes. The strict circle (over ellipses) assumption proved to substantially cull potential false positives. Interestingly tree foliage could momentarily be detected as a circular candidate, these detections didn't last more than individual frames and are readily rejected with spatio-temporal constraints or in classification. As expected man-made circles in the road scene particularly on buildings

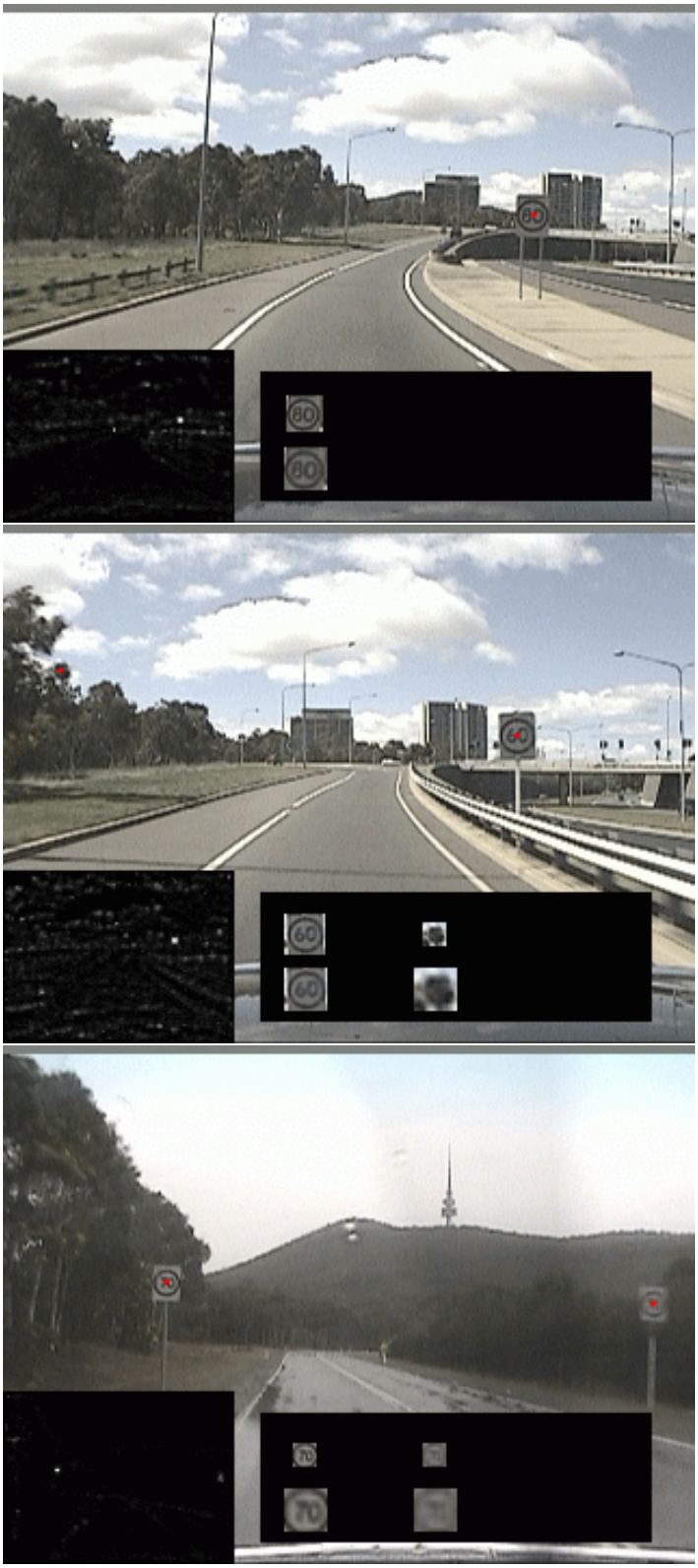
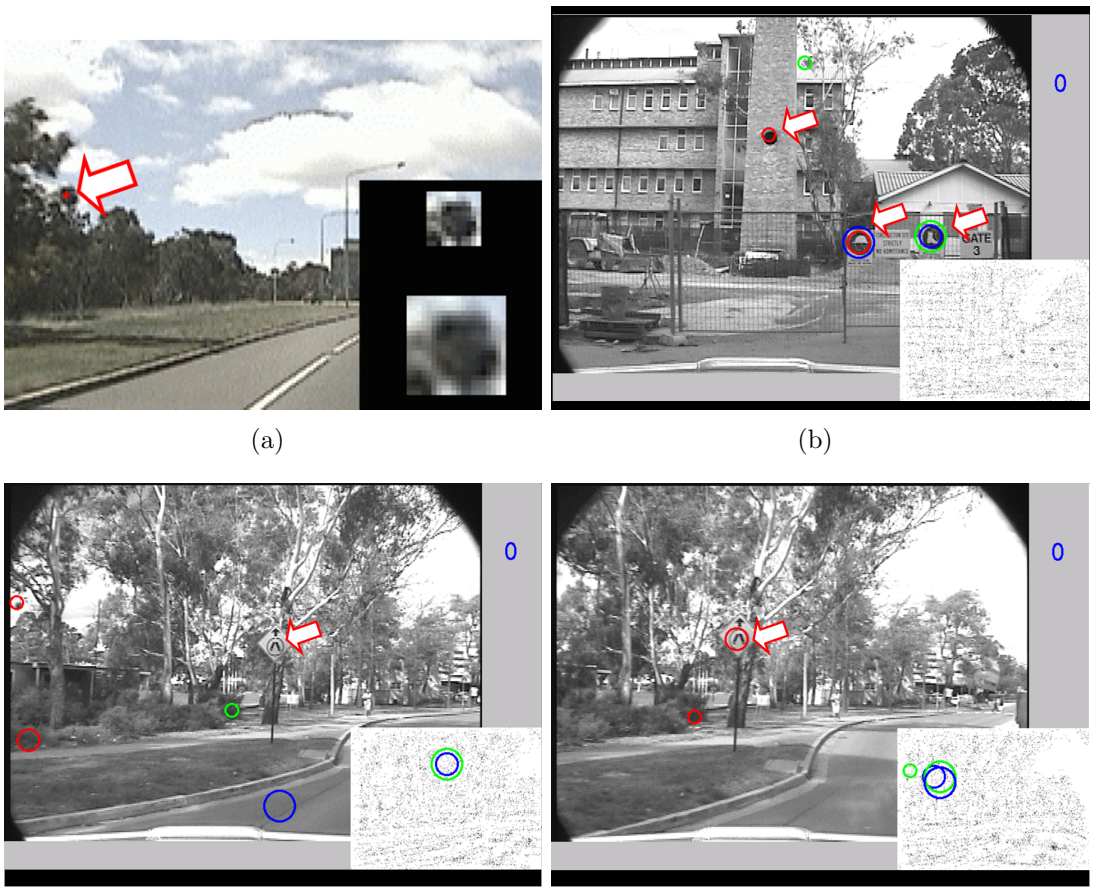


Figure 5.4: Speed sign detection using Fast Radial Symmetry Transform. Bottom left corner shows FRST image, The red crosses highlight detected signs.



(a) (b)

(c) (d)

Figure 5.5: False speed sign detections (not classifications), all cases were rejected by the classifier. (a) : Circular foliage. (b): Circular window and non-speed signs. (c): Circle on roundabout sign not highlighted because of oblique angle. (d): Circle highlighted as sign becomes more orthogonal to vehicle.

were highlighted, but only when parallel to the camera image plane. Examples are shown in Figure 5.5, the rejecting these cases is not a problem to the classifier.

Speed sign candidates are selected from the top N peaks in the FRST image. Each candidate is can be cropped based on which radius contributed the most to the overall transform. We describe how N is determined in the following section.

5.2.2 Efficacy of detection

To ascertain the best value for some of the parameters used in the detection algorithm we conducted a number of trials. First we varied the number of consecutive frames and the number of peaks required for a positive detection. Figure 5.6

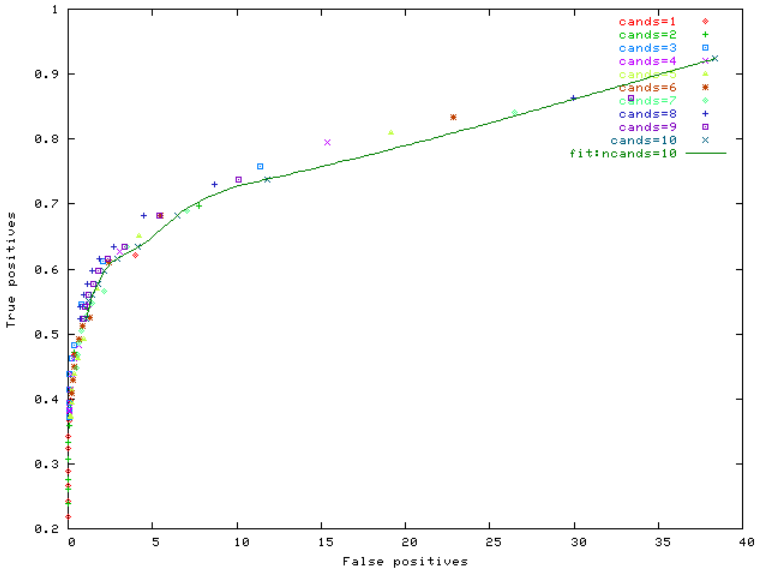


Figure 5.6: ROC curve of detected signs varying number of peaks and consecutive frames required for a detection. The number of consecutive frames required is varied from $0-9$ and the number of FRST image peaks was varied from $1-10$. The fit curve is for 10 FRST image peaks. From (Barnes, Fletcher and Zelinsky, 2006)

shows the ROC curve of true and false detections using the top one to ten peaks in the FRST image and one to ten consecutive frames. For the various numbers of FRST image peaks used, the spread of detection results is quite small (indicated by the small amount the plotted points drift from the curve fitted to the 10 peak candidate line). The number of consecutive frames required is a function of the positive detection rate required and number of false positives permissible by the downstream algorithm. Figure 5.7 shows the ROC curve of detected signs using top three peaks in the FRST image and varying the number of consecutive frames required for detection from zero to nine.

The positive detection rate acceptable is a function of how many opportunities the detector will have to detect the sign. Assuming that the sign is in a suitable region of the image for half a second (15 frames of a 30Hz camera). The detector using two consecutive frames would have 7 opportunities. For two consecutive frames the detector provides a detection performance of 65%. The number of false negatives per frame for the top three candidates is 0.1894 so the over all false detection rate is 0.1894^7 or nine per million.

Using the top three peaks in the FRST image per radius and two consecutive frames to detect signs would generate around 5 false positives per frame. These false positives are easily filtered by the classification phase.

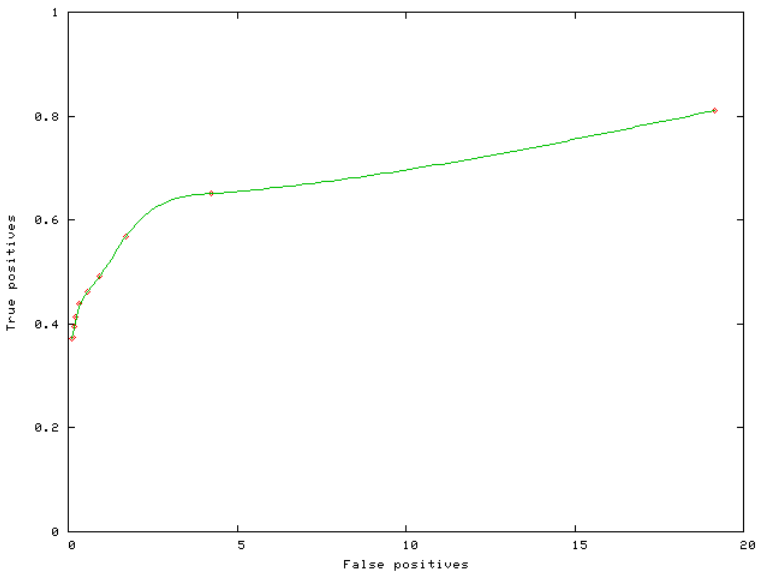


Figure 5.7: ROC curve of detected signs using top three peaks and varying the number of consecutive frames required for detection from $0-9$. From (Barnes, Fletcher and Zelinsky, 2006)

5.3 Classification of road signs

The detection phase of the sign recognition process is highly effective at culling potential sign candidates i.e., circular objects moving consistently over time, only a simple classification scheme is necessary. The classification needs to achieve two tasks. First to reject circular objects that are not signs and second to differentiate between a small set of potential symbols or speeds on a sign. Circular objects that are not signs are rejected as a consequence of symbol classification. It is extremely rare that a phantom feature will have a circular border, move consistently over time and have an appearance resembling a road sign symbol inside the circle.

Classifying the sign between a set of potential symbols is a more challenging problem. Symbol misclassifications far outweigh false detections. This topic is examined further in Section 5.5.1. As discussed above, the top three peaks in the FRST image that exhibit temporal and spatial consistency (i.e., small movements) over two frames begin to be classified. The classification is done with a normalised cross correlation (NCC) (see Chapter 4, Table 4.1)template correlation of the text against stored templates. The recognised speed of the sign was determined by the highest correlating template across the range of speeds that pass a nominal correlation threshold. That is the correlation threshold excludes false positives from the detection phase. The highest correlation across the template determines the speed recognisee. For these trials the speeds '40', '50', '60', '70' and '80' were classified. The sign recognition system signals a sign has been detected when consecutive frames are consistently classified with a high

Figure 5.8: Speed sign classification templates

correlation. Similar to the detection phase key parameters for the classification phase, number of consecutive frames and the correlation threshold are determined experimentally (in the following Section).

5.4 Results and Discussion

To determine suitable quantities for the number of consecutive frames for classification and the NCC correlation threshold, a set of trials was conducted. The road sign sequences were extracted from the pool of video sequences in the project. The sequences make a varied and difficult data set, including a significant number of cases where the sign is not readable to a human observer. Figure 5.9 shows the ROC curve for correct speed sign recognition across 50 road sign sequences. The number of consecutive frames of the same class was varied from $0-5$ frames and the NCC correlation threshold was varied from $0.5-0.7$. The spread of recognition rates was not large compared to the mean curve. Using two consecutive frames for classification and a correlation value of 0.6 produced the best values to use. Even though it is a simple classification strategy across a difficult data-set the true positive rate is 85% (Nine out of 10 signs are detected) and the false positive rate 20% (One sequence out of 5 produces a erroneous speed). As a baseline result this is acceptable to make a workable system for our experiments. There are a number of refinements that could be made to the classification but since our aim is not classifier research (in fact classifier research is an extensive field in computer vision) we concentrate on other issues.

Figure 5.10 shows typical results of the on-line speed sign recognition system. The system has been demonstrated on a laptop computer using web camera as well as in the experimental vehicle. By default the system uses 320x240 pixel images and comfortably runs at the frame-rate of the camera (30Hz) on a Dell Latitude D800 Pentium M 1.6GHz Laptop with 768MB RAM. 640x240 pixel resolution source image (using letter-boxing) executes at around 15Hz on the demonstration laptop. A video demonstrating the sign recognition system is included on the Appendix DVD-ROM (Page 257).

The most pressing issue for the sign recognition system was the poor image resolution of the number on the sign. Recognising the number often generated erroneous classifications early in the sequence, until the sign became sufficiently large enough to recognise. At this stage, however, the sign is often close to the vehicle and moving fast at the edge of the image, resulting in only a couple of frames to

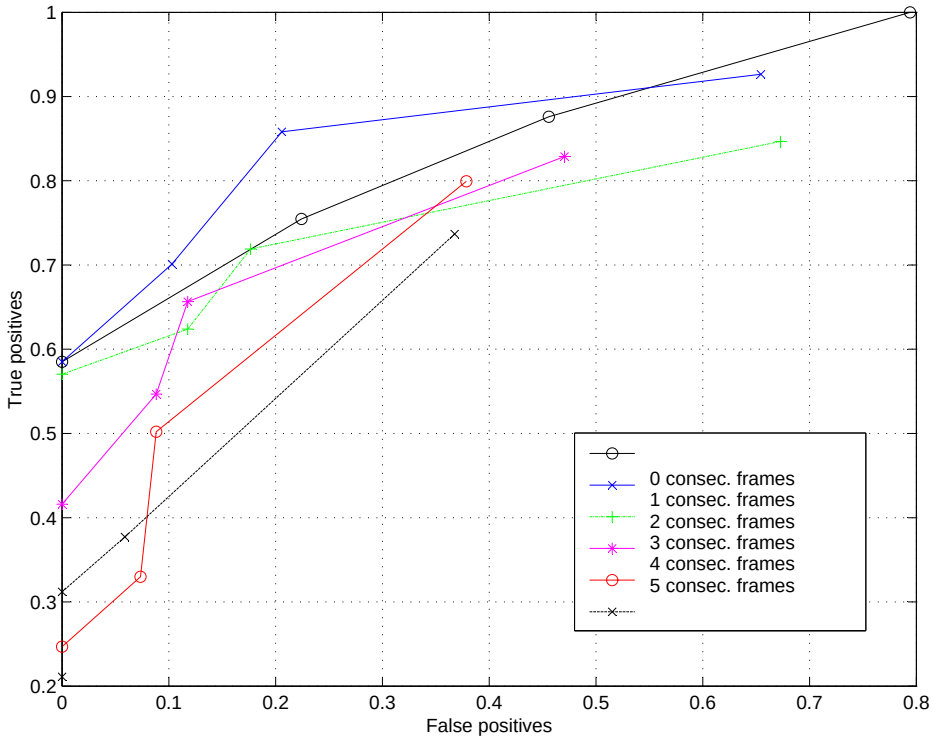


Figure 5.9: ROC curve for correlation based sign recognition. The number of consecutive frames was varied from $0-5$frames and the NCC correlation threshold was varied from $0.5, 0.6, 0.65, 0.7$

classify the sign. If at this late stage the sign becomes occluded recognition fails.

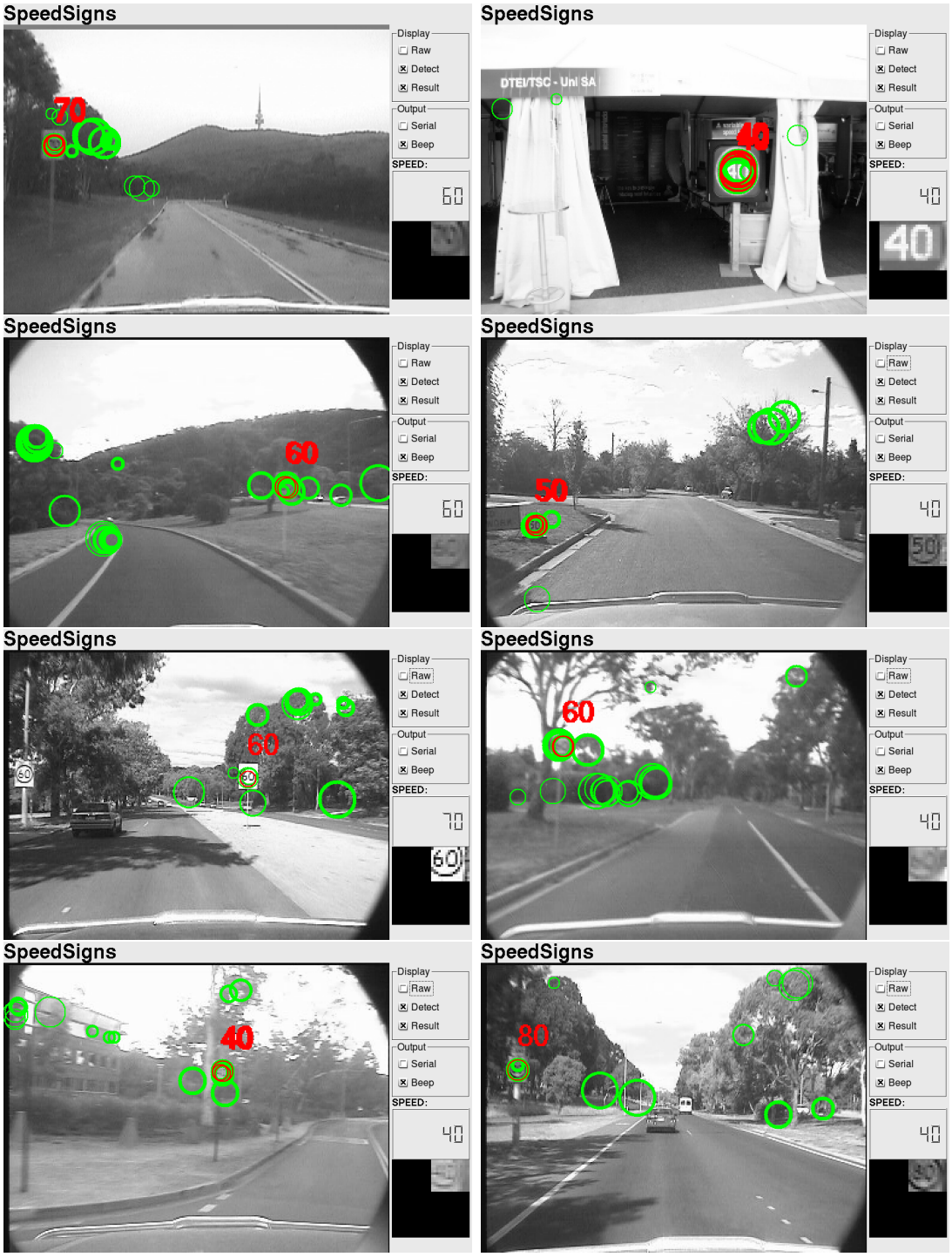


Figure 5.10: Speed sign recognition examples. Green circles represent peaks in the FRST image. Red circle and number represents detected sign and speed. Speed on the right of screen-shots represents the previous speed detected. A video of this system is included on the Appendix DVD-ROM (Page 257).

5.5 Improved road sign classification

We describe a method developed for the use of real-time image enhancement for improved road sign classification.

We are focused on improved speed sign recognition tracked using a circle detector in image streams to improve image quality for classification. We concentrate on a fast technique capable of running at frame-rate. Finally, we show that robust and reliable sign classification is possible with fewer image frames using enhanced images.

As with the other components of the Automated Co-driver system, for the system to be effective, sign detection and classification must be done in a reliable and timely manner. By enhancing the image, we aim to reliably classify the sign several frames sooner than with the raw image data.

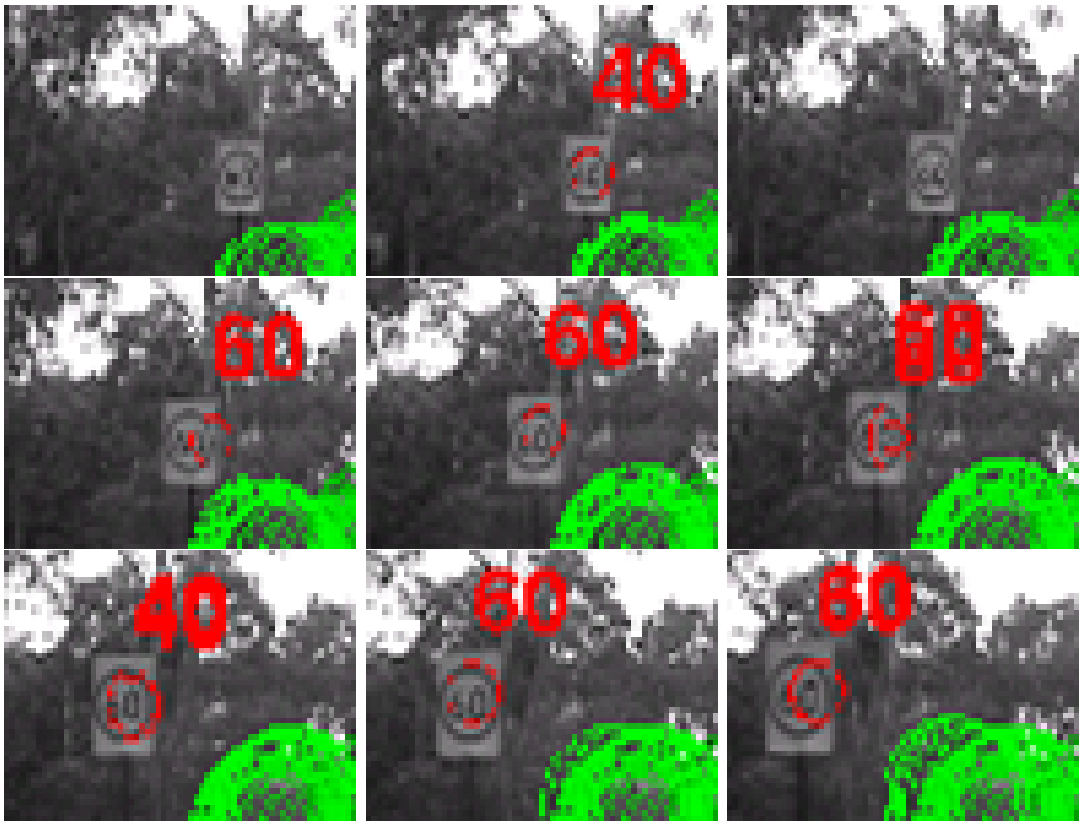


Figure 5.11: '60' Sign speed misclassified.

Video cameras used in robotics typically have low resolution. While digital still cameras on the market have 12 mega-pixels (4096x3072), video based computer vision research often works with a mere 300K (640x480) pixels.

The road sign detection strategy described in this chapter has proven to be effective at selecting sign locations in the image data. As mentioned earlier, the

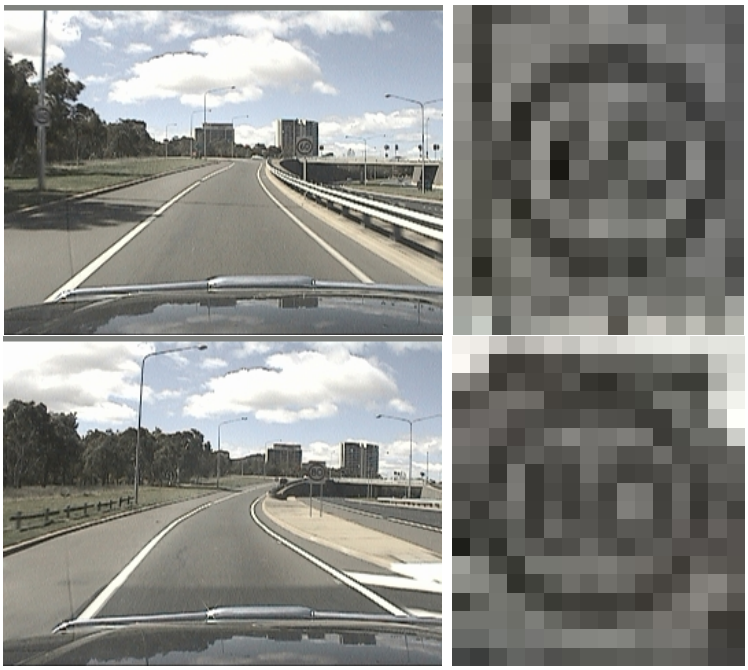


Figure 5.12: **left:** Still frames from a video camera, **right:** Close up of speed sign. Classifying the signs as 60 or 80 is not obvious.

detection strategy works so well that only a simple classification strategy has been required.

One problem that arises is that the road sign detection can work at much lower resolutions than the sign classification strategy. Figure 5.11 highlights the problem, the sign shape is detected and tracked well before the sign text can reliably be read. The misclassification is understandable, the second image was misclassified as a '40' instead of a '60'. The poor resolution is most noticeable when examining still frames of video. Figure 5.12 shows a frame from a video sequence, the right image shows the speed sign enlarged. Note that from a casual glance the speed sign seems well formed and readable in the original frame. However, upon closer examination we find substantial distortion of the text. The detection algorithm detects road signs from 5 to 11 pixels in radius. These radii correspond to areas of 78 pixels (or around 10×8 pixels) to 380 pixels (or around 22×16) respectively. Although these areas seem possible for number recognition, when several pixels are removed for the border circle and white space in between, the number of pixels remaining for two digit number recognition becomes too small for reliable detection. As shown in Figure 5.12 for similar numbers such as 60 and 80 the number of pixels to encode the crucial difference is in the order of $2 - 4$ pixels.

In addition to low resolution other factors confound the classification of the sign. The sign 'appears' in the distance with an apparent size of a few pixels and expands and accelerates in the image until it is lost from the field of view of the camera. Our problem is similar to the super-resolution problem. We examine

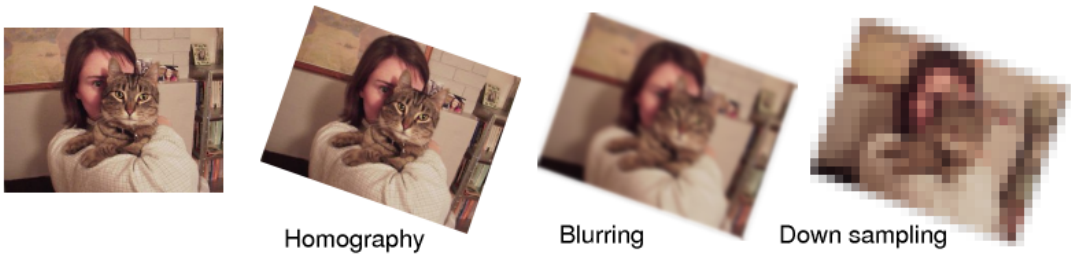


Figure 5.13: High resolution surface undergoes homographic transform, motion & optical blurring, then down-sampling.

super-resolution and online image enhancement methods. We select a method of image enhancement. Finally, the method is evaluated against the popular maximum a posteriori (MAP) super-resolution algorithm used byCapel and Zisserman (2000).

5.5.1 Review of image enhancement techniques

Super-resolution is the process of combining multiple low resolution images to form a higher resolution result (Tsai and Huang, 1984). The super-resolution problem is usually modelled as the reversal of a degradation process. A high resolution image $\mathbf{I}$ undergoes a homographic transformation followed by a motion & optical blurring, then finally, image space sub-sampling to generate the low resolution observation images $\mathbf{O}$ (as shown in Figure 5.13). The order of these operations and extensions such as illumination changes and colour space transformations has also been considered by various groups (Capel and Zisserman, 2003; Farsiu *et al.*, 2003).

$$\mathbf{O}_k = S\downarrow(b_k(H_k\mathbf{I})) + n_k \tag{5.5}$$

where H_k is the homography, $b_k()$ is the blurring function and $S\downarrow()$ is a down sampling operation for the kth observation $\mathbf{O}_k$. n_k is a noise term representing all other errors not modelled.

The solution of the problem amounts to the 'undoing' of the degradation in Equation 5.5. This consists of *image registration* , which recovers the alignment between the images, followed by *image reconstruction* where the images are combined to resolve an estimate of the original image (Baker and Kanade, 2000). Registration is usually done by matching feature points with one of the observation images used as a reference and computing the geometric or homographic transforms between each observation image (Capel and Zisserman, 2000).

Reconstruction requires combining the registered images accounting for the effects of $S\downarrow()$, $b_k()$ and n_k. Optical blurring is often modelled with a point spread

function (PSF) used to represent lens, CCD and discretization effects combined along with a separate motion blur. Schultz and Stevenson (1996) explicitly addressed the modelling of the motion blur. Cheeseman *et al.* (1996) developed a Bayesian method for the reconstruction.

Ideally using a Bayesian framework shown in Equation 5.6, a prior probability model $p(+\mathbf{I})$ can be included if something is known about the kind of image being resolved. Research into the best kinds of prior probability functions has been done for major classes of image, such as text recognition or people tracking. Prior probabilities for text, most useful for us, have been investigated by Capel and Zisserman (2000) and Dellaert *et al.* (1998b).

$$p(\mathbf{I}|\mathbf{O}_n...\mathbf{O}_1) = \frac{p(\mathbf{O}_n...\mathbf{O}_1|\mathbf{I})\dot{p}(\mathbf{I})}{p(\mathbf{O}_n...\mathbf{O}_1)} \tag{5.6}$$

where $p(\mathbf{I}|\mathbf{O}_n...\mathbf{O}_1)$ is the probability of the original texture given all the observations, $p(\mathbf{O}_n...\mathbf{O}_1|\mathbf{I})$ is the probability of the observations given the original texture, $p(\mathbf{I})$ is the prior probability of the original texture and $p(\mathbf{O}_n...\mathbf{O}_1)$ is the probability of the observations.

Since $p(\mathbf{O}_n...\mathbf{O}_1)$ does not depend on the original texture it can be ignored in the estimation process of $\mathbf{I}$. Most research has focused either on finding a maximum likelihood (ML) estimate or finding a maximum a posteriori (MAP) estimate of $\mathbf{I}$. The ML estimation maximises: $p(\mathbf{O}_n...\mathbf{O}_1|\mathbf{I})$). While a MAP estimate maximises: $p(\mathbf{O}_n...\mathbf{O}_1|\mathbf{I}).p(\mathbf{I})$) when something is known about the prior probability $p(\mathbf{I})$. Such knowledge could be that the expected image is of text or faces, properties of these kinds of images can then be represented in $p(\mathbf{I})$. Text, for example, has sharp contrasts between foreground and background regions which can be represented by higher probabilities for larger gradients in the image.
While these approaches have given impressive results most are not suitable for our particular application due to the processing time required. Good examples of this state of the art can be found in (Baker and Kanade, 2000; Capel and Zisserman, 2003; Farsiu *et al.*, 2003).

A novel approach was advocated by Dellaert *et al.* (1998b). This method tracked an object in an image sequence and used a Kalman filter to estimate the pose and augmented the state with the super-resolved image. With some optimising assumptions the group was able to perform online pose and image estimation. The effect of prior probabilities is incorporated quite neatly into this framework in the derivation of the Jacobian matrices.

Lee and Tang (2006) demonstrated an effective technique for motion blur compensation in license plate image enhancement. The technique applied a deconvolution filter to the images based on the detected motion between frames. Since our detection approach provides the region of interest in each frame this approach is readily applicable. Currently motion blur is not as dominant a problem as poor resolution. The signs are still sufficiently far away that motion blurring

doesn't appear substantial. In fact blurring more readily occurs from momentary jerks/vibrations (in vertical direction) than due to fast horizontal motion. We will not implement this technique, but it is an effective approach and may be an avenue for future work.

5.5.2 Our approach

For our application, objects approach from a minute size near the centre of the image and then expand and accelerate out of the left or right field of view. Image registration is primarily using the Fast Radial Symmetry Transform FRST).

Speed signs are detected as spatially and temporally consistent peaks in the FRST image sequence as described in Section 5.2. From the detection phase the dominant circle is cropped from the video frame and resized. The image is resized using bi-cubic interpolation to the size of the high resolution result image. Baker and Kanade (2000) found, as a good rule of thumb, eight times magnification is the upper limit for super-resolution. In our case, with the lower radius from the Fast Radial Symmetry Transform of 4 pixels (diameter of 8 pixels), the high resolution image is 64 x 64 pixels. The image is then correlated using normalised cross correlation to with the current integral (result) image to locate the latest image accurately. The latest image is shifted accordingly and combined with the integral image. For simplicity and speed the correlation and shift is only performed to an integer accuracy, since the correlation is done on the higher resolution images the shift is equivalent to a sub-pixel shift in the original video frames. The justification is that the correlation coefficients are flat in the correlation region due to the substantial increase in the image size. This indicates that sub-pixel interpolation is likely to just be driven by image noise. Also other sources of error such as uncorrected rotation about the optical axis and errors in the estimated radius from the FRST detector appear to dwarf the effect of this approximation.

Images where the correlation coefficient is below a certain threshold (usually less than 0.5) are discarded as these tend to be gross errors in radius estimate by the FRST or momentary competing circles near the tracked sign.

The next step is the reconstruction of the enhanced image. The aim is to get an immediate improvement in classification from each additional frame. The reconstruction is considered as a series of incremental updates of the resolved image from the observations as shown in equation 5.7. The update is a first order infinite impulse response (IIR) filter shown in equation 5.8 allowing a fast and efficient implementation. This could be considered a simplification of the Dellaert *et al.* (1998b) method.

$$\hat{\mathbf{I}}_k = \hat{\mathbf{I}}_{k-1} + \lambda c(S\uparrow(\mathbf{O}_k) - \hat{\mathbf{I}}_{k-1}) \tag{5.7}$$

Online image enhancement algorithm:

Once a sign is detected, for each frame in the image sequence:

1. Segment image from frame using location and radius of peak in FRST image.
2. Up-sample the image $\mathbf{O}_k$ using bicubic interpolation to $64x64$ pixels to give $S\uparrow(\mathbf{O}_k)$.
3. Use NCC correlation to align the up-sampled image $S\uparrow(\mathbf{O}_k)$ with the integrated image $\mathbf{I}_k$. For $k = 0$ use $S\uparrow(\mathbf{O}_k)$.
4. If the NCC correlation result c is too low reject the frame, otherwise:
5. Apply 4-connected homogeneous point operator to the up-sampled image $S\uparrow(\mathbf{O}_k)$.
6. Apply Erosion to the up-sampled image $S\uparrow(\mathbf{O}_k)$.
7. Combine images using Equation 5.8
8. Apply Classification as described in Section 5.3.

Table 5.2: Online image enhancement algorithm.

$$\hat{\mathbf{I}}_k = (1 - \lambda c)\hat{\mathbf{I}}_{k-1} + \lambda c S\uparrow(\mathbf{O}_k) \tag{5.8}$$

where $S\uparrow()$ is the up-sizing function for the kth observation $\mathbf{O}_k$ of the estimated enhanced image $\hat{\mathbf{I}}_k$. λ is a weighting constant and c is the above mentioned normalised cross correlation result. The constant λ is set so that when combined with the correlation coefficient the update weighting (λc) is around 0.15 to 0.25. The correlation result is a scalar between 0.0 and 1.0, correlations of contributing frames are around 0.6 to 0.9 so λ is set to 0.25. This weighting scheme allows better estimates, particularly later on in the sequence as the sign gets larger to have a greater impact on the result.

To recover text on a sign, we know the expected image will have a smooth background and lettering with sharp in contrast in-between. To incorporate the text prior into the real-time implementation we pre-emphasis the up-sampled images before they were integrated by equation 5.8. We perform a contrast enhancing homogeneous point operator and erosion on the grey images. This sharpens the discontinuity between the foreground and background and also reduces the spread of (skeletonises) the text. These steps sharpen the textual boundaries and help compensate for the over-smoothing of the filter.

Method	True positives	False Positives
Original images	0.65	0.094
Enhanced images	0.98	0.003

Table 5.3: Performance of enhanced image classification.

5.5.3 Image enhancement results and discussion

From the original sequences used in Section 5.4 we selected a set of sequences that had a significant number of frames where the sign was detected to attempt image enhancement. Not knowing how many frames would be needed we selected sequences with more than 30 frames. On these sequences we ran the enhancement algorithm described Table 5.3 shows the results of sign recognition using the online image enhancement compared with Section 5.4.

Figures 5.14,5.15 and 5.16 show the the generation of the enhanced images in several video sequences. Only every 10th image is shown from the sequence. The enhanced image is an improvement from the original up-sized image. The enhanced image appears resistant to fluctuations in the observations such as size errors (as the right 2nd row of Figure 5.16). The image does deteriorate toward the end of the Figure 5.15 sequence as the FRST has only been computed to a radius of 10 pixels and as the sign becomes larger there is a constant underestimate of the size of the sign in the last few frames. A video demonstrating the sign image enhancement is included on the Appendix DVD-ROM (Page 257).

The efficacy of the original and super-resolved image in speed classification was then tested by correlating a template image of '40', '60' and '80' signs with the images. The template images were taken from a highest clarity and resolution image available from different image sequences and are enlarged to match the resolution of the test images.

Figure 5.17 show the correlation results for a '40', '60' and '80' sign sequences. The drop outs such as in '60' sequence represent misses of the sign by the FRST detection phase. In all cases the enhanced image sequence showed a consistent improvement in reliability over the original image. Both the original and enhanced sequences show the expected upward trend in correlation value over time as the sign becomes bigger. The absolute correlation value did not show a significant improvement between the good original frames and the enhanced image. This may be due to the original lower resolution of the template image or over smoothing. The relative differences between the correlation coefficients for the templates and the consistency over time do justify the expectation of better classification.

(a) (b) (c)

Figure 5.14: '40' sign enhancement. **(a):** Every 10th frame from Forty sequence. **(b):** Resized cropped original image. **(c):** Enhanced sign image.

(a) (b) (c)

Figure 5.15: '60' sign enhancement. **(a):** Every 10th frame from Sixty sequence. **(b):** Resized cropped original image. **(c):** Enhanced sign image.

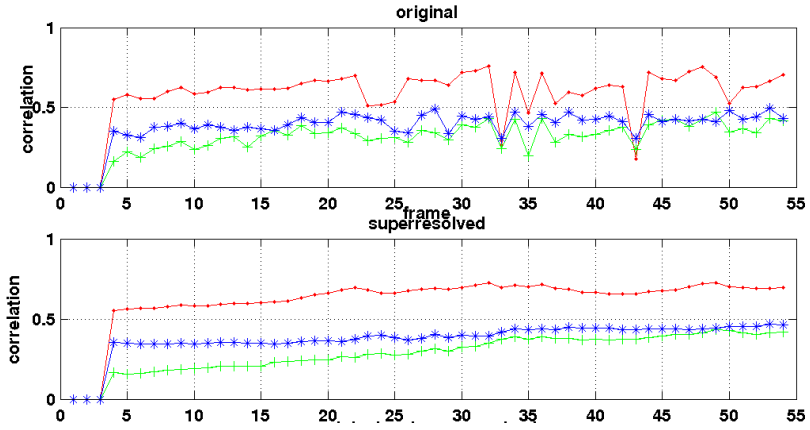
Verification

To verify our enhancement technique we implemented a recent super-resolution algorithm based on global optimisation and compared the results. The method used was the MAP algorithm used by Capel and Zisserman (2000). Since the algorithm involves solving a mid sized global optimisation problem, the algorithm is not suitable for an online implementation. Also, either a new optimisation would need to be done each iteration or the result would be unavailable until all the sign frames had been gathered. In this method a penalty function is used to influence the result based on the prior $p(\mathbf{I})$. A suitable text prior/penalty is

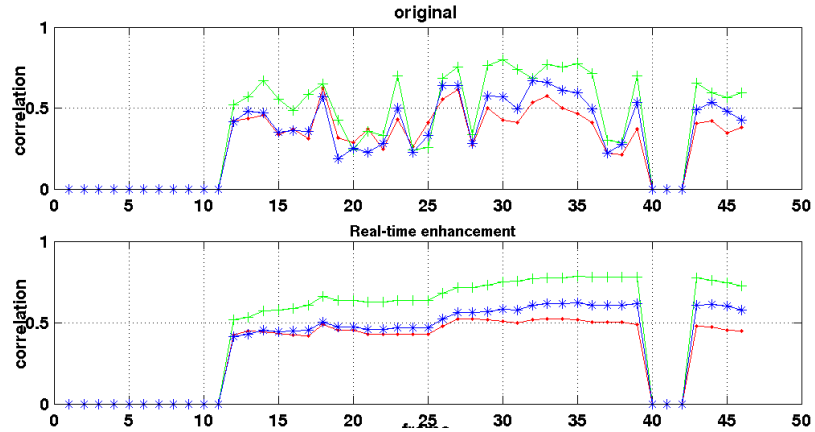
(a) (b) (c)

Figure 5.16: '80' sign enhancement. **(a):** Every 10th frame from Sixty sequence. **(b):** Resized cropped original image. **(c):** Enhanced sign image.

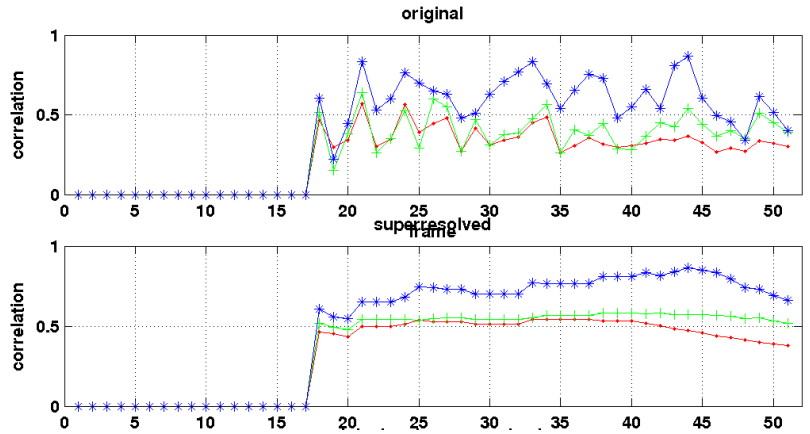
implemented as a function of the gradient magnitude of the image. For small gradient magnitudes the penalty is a quadratic ($f(I'^2)$); as the gradient magnitude increases and crosses the threshold α the penalty has linear 'tails' ($f(|I'|)$). Our implementation used the Mathworks Matlab^{TM}application global optimisation function *fmincon()* with a scalar error composed of the sum of the squared differences plus the weighting (λ) of the penalty contribution. Best results were obtained with $\lambda = 0.025$ and $\alpha = 40$. Please refer to Capel and Zisserman (2000) for a full description of the implementation. Figure 5.18 shows the result of the minimisation. The off-line image had more consistent intensity within the foreground and background regions but lost some contrast overall. While an im-



(a)



(b)



(c)

Figure 5.17: Correlation (NCC) coefficients for sign image sequences. **top graphs:** Original resized image sequences. **bottom graphs:** Enhanced image sequences. *'.':* '40' template. *'*':* '60' template. *'+':* '80' template.
. (a) '40', (b)'60' and (c) '80' sequence.

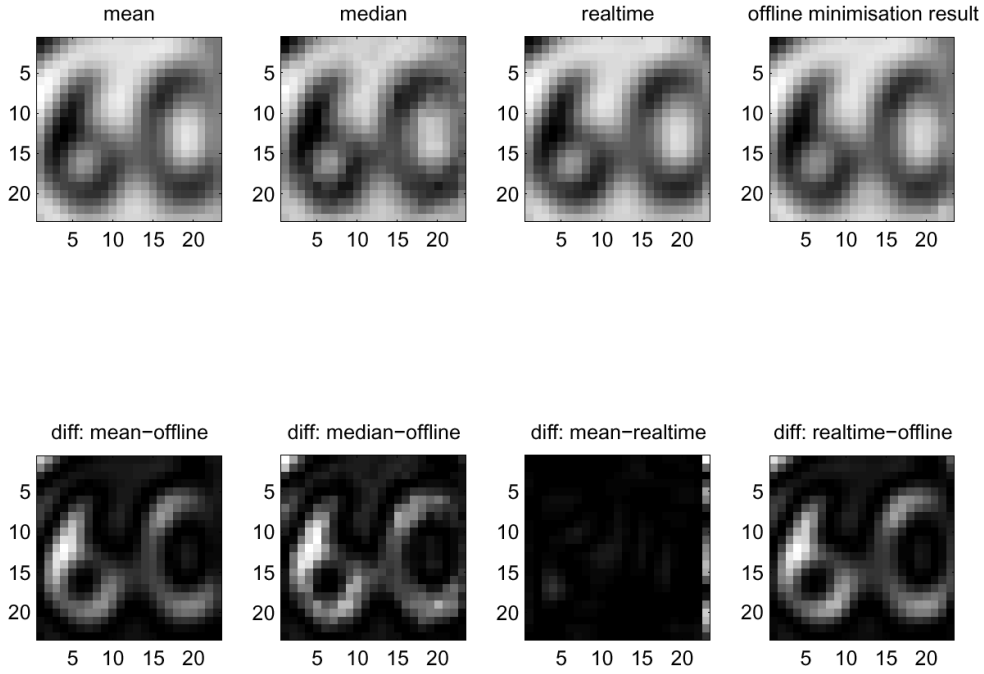


Figure 5.18: Comparison with off-line super-resolution.

provement on the temporal mean image is achieved, the tuning of λ and α that would provide a significant benefit across all the test image sequences proved difficult. It is likely that the hyper-plane for this minimisation resembles a large convex minimum around the temporal mean image with a small global minimum spike at optimal super-resolved image so finding robust λ and α parameters may not be possible. As could be expected, the temporal median image also provides a promising result, unfortunately it is not obvious how to achieve an efficient real-time temporal median image.

The real-time result was similar to the off-line technique although it exhibited more artifacts from the latter end of the image sequence, which is to be expected with the incremental update approach. We are satisfied that our approach is equivalent to the state of the art algorithm with the advantage that the technique runs in real-time.

5.6 Summary

We have presented a fast, effective, image-based symbolic sign detection technique. The technique uses strict shape-based image filtering to substantially reduce the data volume from the image stream for a simple classifier stage. The work is being continued and built on with a strong chance of industry and commercial outcomes by the National ICT Australia project partners. Even a simple classification strategy like image correlation teamed with the detector was able

to produce a usable system.

Online image enhancement is worthwhile and achievable even when the apparent size of the object changes substantially across the set of observed images. The super-resolution inspired image enhancement provides a significant benefit in the reliability of down stream sign classification. As with most work with super-resolution, over smoothing is a significant issue. Introducing a prior probability in a global optimisation could help but have tuning and computational complexities to overcome for the real-time case. Pre-emphasis of the observation images based on the prior, provides benefit along similar lines.

With a system able to read speed signs, we now are able to create an Automated Co-driver able to detect inattention due to missed road signs and speed adaption. These timely interventions to reduce speeding have the potential to reduce road fatalities.

Next, instead of explicitly detecting features of the road scene, we will assess the road scene complexity over time. Driver alertness suffers due to a monotonous road environment and distractions during busy intersections can cause accidents. By understanding the road scene complexity our Automated Co-driver, just like a human observer, will understand when the driver is bored (leading to fatigue) and when the driver is too busy to talk (or to receive secondary distractions).

Chapter 6

Road scene complexity assessment

A great irony of transportation systems research is that advances in road and vehicle safety may end up causing new threats to road users.
Drivers now enjoy improved vehicle design:

- suspension design to minimise skeletal repetitive strain injury.
- sound damping to reduce road and traffic noise.
- mirror placement and instrumentation design to minimise driver effort.
- climate control to maintain a constant temperature regardless of the weather.
- cruise control to reduce the strain of driving over long periods.

And improved road infrastructure:

- smooth low-curvature divided roads.
- multiple lanes or overtaking zones to reduce instances where drivers are stuck behind slower vehicles or need to overtake using the oncoming traffic lane.
- screening along highways to minimise traffic noise and deter pedestrian & animal hazards.

In effect, car manufacturers and infrastructure authorities have collaborated to attenuate stimulation from the off-driving tasks and ease the on-driving task. The unfortunate consequence is that drivers, now more than ever, are disengaged from the road environment other than the lane keeping task. If the task of lane keeping is under-stimulating, even for periods less than 20 minutes, the driver is susceptible to fatigue (Thiffault and Bergeron, 2003). The phenomon known as "Highway hypnosis" was coined by Williams (1963) to describe the ability

of drivers to drive long stretches on monotonous, periodic or uneventful roads with only minimal attention. In this reduced attention state the onset of fatigue is more likely and has even been reported regarding climate controlled mining dump trucks (Shor and Thackray, 1970; Thiffault, 2004). Consequently, sections of road that were once prone, for example, to head-on collisions, have become fatigue accident zones after divided multi-lane road redesigns. Ingwersen (1995) demonstrated that improved roads have led to more fatigue-related accidents.

In Section 6.1 we review fatigue management research and the problem of monotony. Section 6.2 describes our approach to the visual monotony detection. Again we define "visual monotony" as visual environments consistent with roads which are periodic, monotonous, or uneventful in accord with the well known "Highway hypnosis" phenomenon of Williams (1963), developed in relation to fatigue by Shor and Thackray (1970). In Section 6.3 we use lane tracking to handle cases where the road may appear featureless but is not likely to be monotonous. The approach is then extensively tested in Section 6.4. Finally, in Section 6.6 we investigate techniques for measuring road scene complexity. Road scene complexity is a useful determinant for driver workload management. Driver workload management, which is at the opposite extreme of the fatigue problem, involves assessing the difficulty of the current driving task to determine suitable scheduling of secondary tasks. We develop a heuristic for estimating the number of salient points relevant to the driver in a road scene.

Both visual monotony and road scene complexity metrics enable Automated Co-driver determinants to assess the driver's fitness to manage the driving task. Our Automated Co-driver will use these determinants to assess the monotony of the road scene, and therefore make judgements about the driver's ability to remain within safe limits on the road. Our system will have the potential to combine these determinants with direct driver monitoring. As stated in Chapter 1 conducting trials on fatigue, especially in vehicles (as opposed to in a simulator), is notoriously difficult and beyond the scope of this work. Instead in this chapter will examine the effectiveness of the metrics on long video data sets and then, in Chapter 7, simply demonstrate how the combined assistance system would operate.

6.1 Review of fatigue detection techniques

Fatigue presents insidious risks on the road. Unlike many other crash-contributing factors, fatigue has proved notoriously hard to police. Even experienced drivers can be oblivious to the deterioration in their condition (Torsvall and aAkerstedt, 1987).

Haworth *et al.* (1988) identified several indicators that which correlate well with the onset of fatigue:

- less controlled steering movements and larger variance in lateral position in lane
- driver blink patterns showing prolonged periods of eye closure
- slumping of driver head posture
- sleep-like brain activity (EEG readings)
- eye movement, particularly a change to a less efficient scanning pattern

Automated detection of these indicators are in active development. Driver head pose and gaze monitoring are particularly promising.

Indicators that have been found to be uncorrelated in detecting fatigue include:

- speed variation, acceleration and brake reaction time
- heart rate variance (ECG readings)
- analysis of bodily fluids (blood, urine, breath testing)

Continuous in-vehicle monitoring is the only mechanism capable of detecting inattention and fatigue. However, Haworth *et al.* (1991) also found that warning-only countermeasures (such as head tilt sensing or eye closure sensing glasses) did not allow the driver to drive for longer or prevent deterioration of driving performance due to fatigue.

To address this problem, research has begun into how to monitor and maintain driver vigilance.

Slower visual and auditory reaction time to tasks (such as tapping a button when a light flashes) are also a known indicator, although this is thought to contribute to fatigue as well as detect it. The above indicators show that in order to reliably detect fatigue, we need to bring together driver monitoring and the evaluation of driver behaviour as reflected through the vehicle.

There are a number of initiatives in place to combat driver fatigue. Driver log books are the primary tool used to manage fatigue in professional drivers, however the systematic under reporting of driving hours remains common. Braver *et al.* (1992) found two thirds of drivers regularly under reported their hours sponsoring calls for tamper-proof on-board loggers.

The Safe-T-cam system developed by Auty *et al.* (1995) has been effective. The system uses fixed cameras on overpasses to track heavy vehicle progress along arterial routes. However, its effectiveness is principally as a visible deterrent rather than its ability to secure convictions. Drivers can, for example, argue that they alternated with another driver between Safe-T-cam check points.

More broadly, Balkin *et al.* (2000) developed the 'sleep watch actigrap' for military and other personnel required to remain active at odd hours. The watch records the movement (such as the respiration rate) of the wearer to determine periods

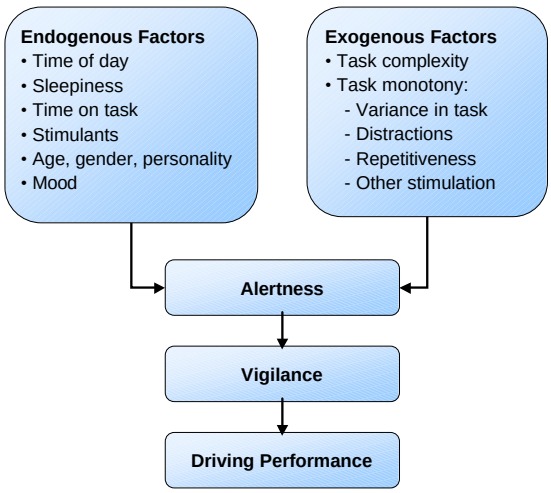


Figure 6.1: Endogenous and exogenous factors contributing to fatigue. Recreated from (Thiffault and Bergeron, 2003).

of sleep and wakefulness. The watch then uses a extensively developed algorithm based on the hours slept over recent days, time of day and circadian cycles to estimate the alertness of the wearer. This way the wearer can arrange rest to avoid critical levels of fatigue.

In-vehicle driver monitoring has shown much promise in the detection of inattention and fatigue. PERCLOS (Wierwille and Ellsworth, 1994) eye closure, percentage road centre eye gaze (Victor, 2005) and variance in steering wheel movement (SWM) are a few examples of metrics found to significantly benefit fatigue detection. However, one aspect which hampers these metrics is lack of context awareness. For example, many fatigue monitoring techniques struggle in urban and suburban driving scenarios; frequent blind-spot checking and intersection negotiation disrupt eye monitoring, and frequent manoeuvring, disrupts steering metrics. By detecting monotony in the road scene, fatigue monitors can be made context aware and thereby able to bias their metrics to minimise false positive warnings.

Also, automated detection of fatigue using the indicators above integrates measurements over minutes to robustly detect fatigue (Wierwille and Ellsworth, 1994; Victor, 2005; Thiffault and Bergeron, 2003). Fatigue related road fatalities can happen in a moment of inattention. An Automated Co-driver can address fatigue-related crashes by providing earlier warning of inattention-related road events.

Driver monitoring for fatigue detection is an important research area. The potential of emerging systems such as faceLAB^{TM} is considerable and there are many people working in this field. Therefore, as outlined in Chapter 2, the commercially available system is sufficient for our experiments. Our Automated Co-Driver is designed to support future fatigue-detection systems but in this work, we focus

on eye-gaze detection in a robust framework with road and vehicle monitoring. This allows us to concentrate on the specific, and largely unaddressed, problem of inattention.

The contributing factors of fatigue can be divided into endogenous (internal origin) and exogenous (external origin) sources. Lack of sleep can be considered an endogenous factor while lying in a darkened room would be an exogenous factor. Figure 6.1 shows the decomposition of contributing factors of fatigue. A recent insight in psychology literature has been to define monotony as an exogenous factor as opposed to a mental state (which would be endogenous, similar to boredom)(Thiffault and Bergeron, 2003). In this way monotony can be used as an attribute of a task in a particular context. That is, a task can be explicitly labelled as monotonous or non-monotonous (i.e., stimulating).

The key point is that the monotony of the task can be decoupled from the actual mental state of the person. So regardless of how a task affects a person, if there is infrequent (or periodic) stimulus, low cognitive demand and low variance of task, it can be termed monotonous. For example, the task of driving on a straight country road with little scenery on a clear day can be described as monotonous regardless of whether the driver is becoming fatigued or not. Whether a driver actually finds the trip fatiguing is dependent on the combined effect of the internal and external factors. A person driving home after being fired from his or her job is unlikely to become fatigued by the monotonous task.

Our system will combine monitoring of internal factors, through attention to driver gaze, and external factors, through monotony detection in the road scene, to address this combined effect. As outlined in Chapter 2, Victor (2005) has shown that driver eye-gaze direction is likely to be effective for estimating the cognitive state of the driver. Our approach will bring this estimation of the driver's cognitive state together with the detection of monotony in the external environment.

We develop an automatic method of assessing visual monotony and, by assessing the road scene complexity, identify stretches of road which are likely to induce fatigue. Our approach to gaze monitoring is to come in 7.

6.2 Visual monotony detection

As the primary sense used for driving, vision is also the primary sense to maintain alertness. We close our eyes and prefer a darkened room to sleep, but sounds, smells and touch can be slept through. The monotony of the driving task can be decomposed into a visual stimulus component and non-visual component. As mentioned earlier, the non-visual sensory stimuli have been attenuated by road and vehicle design, so we aim to measure the visual stimulus component.

Road Type	Scenery	Disruptions	Road Curvature	Monotony
Urban road	Cluttered	Frequent	High	**Low**
Country road	Moderate	Few	Varying	**Moderate**
Minor highway	Sparse	Varying	Moderate	**Moderate**
Major highway	Periodic	Varying	Low	**High**
Outback road	Sparse	Few	Low	**High**

Table 6.1: Different driving environments and likely monotony level.

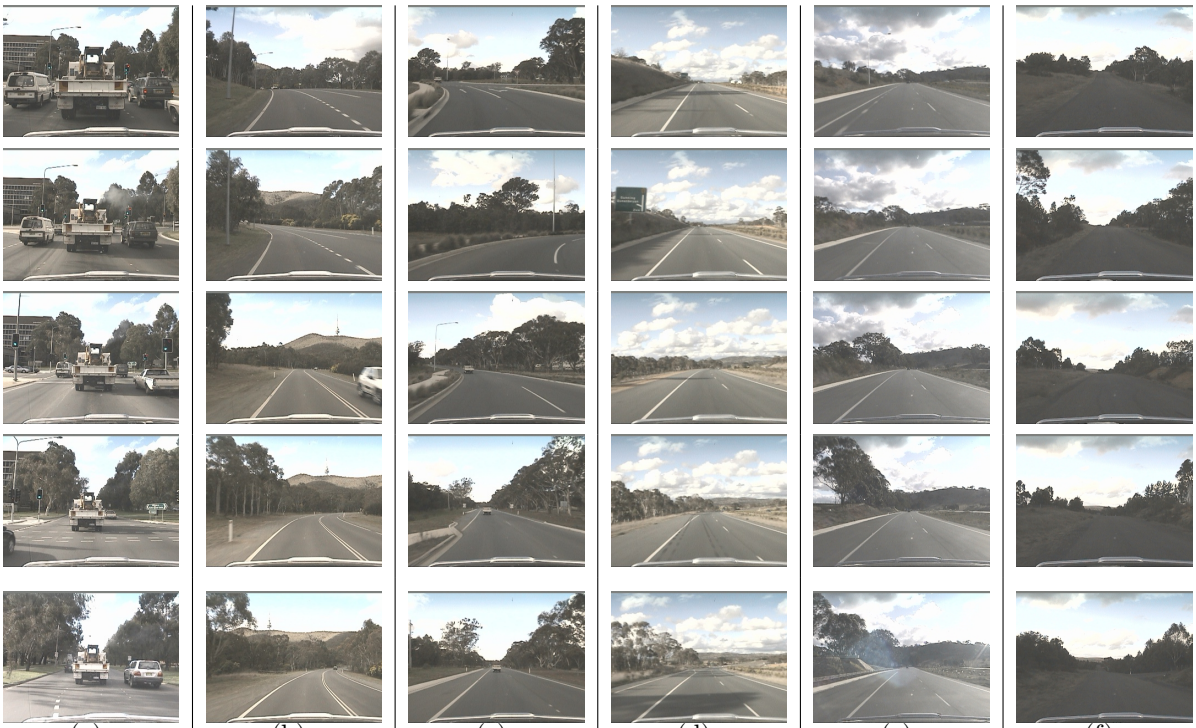


Figure 6.2: Different driving environments with approximately 3 seconds between frames. (a): city traffic, not monotonous. (b): curving single lane road, not monotonous. (c): roundabout on otherwise straight road, not monotonous. (d): dual-lane road with gentle curvature, potentially monotonous. (e): dual lane straight road, no traffic, potentially monotonous. (f): no lane marks, not monotonous.

Table 6.1 categorises a number of common road conditions into contributing factors for a monotonous environment. Figure 6.2 shows some sampled frames from a number of different road sequences. By observing the sampled frames an estimate of the monotony of the driving task, as judged by experienced drivers, can be made for each sequence.

To automatically measure the monotony in a road sequence we require a metric of

the variance of the video sequence over time. In essence, we need to estimate the information content of the sequence. The information content of a data signal is often defined by the stochastic complexity or Kolmogorov-Chaitin complexity. Kolmogorov-Chaitin complexity of a some original data is defined as the size of the smallest program (or encoding and decoder) which can reproduce the data (Chaitin, 1974). That is, the amount an information source can be compressed is a measure of the information content. Practically we need a measure that is robust against lighting variations common in outdoor scenes, periodic motion, and simple translational camera ego motion. Moving Picture Experts Group MPEG encoding fills these requirements. MPEG encoding can be thought of as an estimate of the Kolmogorov-Chaitin complexity because it changes based on the information content in the image sequence over time. However, as the compression is lossy it is important to note that it is not strictly a valid metric. Nevertheless for our purposes, the lossy nature of the compression and the effect on the metric is convenient, as we will examine.

6.2.1 MPEG Compression as a measure of visual monotony

MPEG encoding is a scheme for compressing a series of video or movie frames. MPEG exploits the property that in moving pictures only small regions of the image actually change substantially between frames. Most of the frame is static or translates in the image, varying marginally. Impressive compression ratios are achieved by coupling an effective lossy compression scheme with an update procedure that can efficiently represent small motions and appearance changes between frames. Briefly, the video sequence is encoded as a sequence of three kinds of frames. Key (I) frames are compressed, but are otherwise complete frames using compression similar to but not identical to JPEG compression. Prediction (P) frames consist of the set of changes required to modify the previous frame into the current one. The frame consists of a set of motion vectors detailing how each subregion of the image has moved between frames and a correction representing what needs to be done to the previous subregion to make it match the current frame. Bidirectional prediction (B) frames are similar to P-frames, except both previous and next frames are used to reconstruct the image. Benchmarking has shown that compression rates for each frame typically approach: 1-bit/pixel for I-frames, 0.1 bits/pixel for P-frames and 0.015 bits/pixel for B-frames (Motion Picture Experts Group, 2004).

The compression and correction is done using the discrete cosine transform (DCT), which is effectively a frequency domain transformation. Motion between frames is measured using block matching techniques common to computer vision such as Sum of Absolute Difference (SAD) correlation.

MPEG compressed sequences have been used in computer vision to:

- mosaicing recover camera ego motion using motion estimation vectors (Pilu,

1997).

- generate scene indexes of movies by the automatic detection of scene changes by large discontinuities between frames (Sethi and Patel, 1995).
- detect pedestrians using P-frame motion vectors for optical flow (Coimbra and Davies, 2003).
- segment and index objects using P-frame motion vectors (Mezaris *et al.*, 2004).

MPEG4 compression has the following attributes which make it especially suitable for monotony detection:

- JPEG-like compression of I-frames achieves better compression for "natural" images and worse compression for sharp edges, i.e. near-field cars compress badly.
- YUV colour-space means some tolerance to lighting changes, a shadow across a region of the image compresses as scalar multiple of the subregion.
- Sum of Absolute Difference or similar motion detection.
- Block matched motion down to half a pixel resolution.
- In intelligent encoders I-frames are included on an as-required basis based on error margin. So dramatic changes in the scene, like close moving vehicles will cause more I-frames to be added increasing the file size. One-off scene changes, like entering a tunnel, will cause a single I-frame to be introduced.
- MPEG compression chips makes embedded devices easily implementable.

MPEG compression seems to be ideally suited for monotony measurement. However several issues need to be discussed. First, since the MPEG algorithm was developed to compress video there is no guarantee that the motion detection used will actually capture any meaningful motion in the scene. The motion detection is not extensive, it is solely trying to reuse similar regions of the image to minimise the additional data required. We need to be wary as there is no embodied computer vision concept of features involved, let alone the concept of reasonable road scene motion. However, other groups have shown a high correlation between the optical flow of a scene and MPEG motion vectors (Coimbra and Davies, 2003).

A potentially more problematic issue is that certain categories of the road scene: rain, fog or otherwise featureless road scenes, will compress well even though this is not a monotonous situation. We will address this issue in Section 6.3.

6.2.2 Correlation between MPEG and monotony

To verify that MPEG encoding correlates with the monotony of a scene, we implemented a monotony detector using the open source libavcodec library (FFMPEG,

Sequence	MPEG File Size (Kb)	JPEG Seq. Size (Kb)	MPEG/JPEG Ratio
Seq. G. figure 6.2(a)	3664	5168	0.71
Seq B. figure 6.2(b)	2976	4672	0.63
Seq. C.	2684	4460	0.60
Seq. N.	2604	10080	0.26
Seq. H. figure 6.2(c)	2548	4460	0.57
Seq. A.	2504	4324	0.57
Seq. E.	2412	4364	0.55
Seq. L.	2248	9836	0.23
Seq. D.	2176	4216	0.52
Seq. J.	2108	9352	0.23
Seq. I. figure 6.2(d)	2024	4452	0.45
Seq. M. figure 6.2(e)	1972	9276	0.21
Seq. K. figure 6.2(f)	1784	8708	0.20

Table 6.2: Compression of video sequences.

2005). This library contains an MPEG4 encoder.

Every 60th image was selected from the forward-looking road scene camera for compression. This represents a one second gap between frames. A sliding window of 150 images was compressed, representing a time period of 2 minute 30 second window. The frames were filtered using reasonably strong temporally and spatially smoothing across adjacent frames before encoding. The filtering was to remove small differences in (Luma and Chroma) pixel values such as the fine texture on the road, low light graininess which do not represent the kind of "information content" we are interested in measuring, and also to remove JPEG artifacts if present. The frames were 320x240 colour images and compression took around one second on a Pentium IV 3.0GHz machine. Compression was performed every ten seconds. Most compression settings were left at the default values. SAD correlation was selected for motion vector estimation (from a limited set of methods supported in the library). A high maximum bit rate off 8 Mbit/s was selected, allowing the compressor to use as much data as needed to encode high frequency changes in the image. Lower maximum bit rates forsake high frequency changes in the sequence to minimise the bit rate, which causes the kinds of changes we are interested in to be lost.

Table 6.2 shows the file sizes for various MPEG encoded sequences similar to those shown in figure 6.2. The encoded files have a good spread of sizes with a factor of two difference between the smallest and largest files. The MPEG/JPEG ratio shows that there is no correlation between the size of a JPEG sequence (representing only scene complexity) and the MPEG sequence (representing the change in the image over time).

When compared to an experienced-driver judged monotony scale, the MPEG

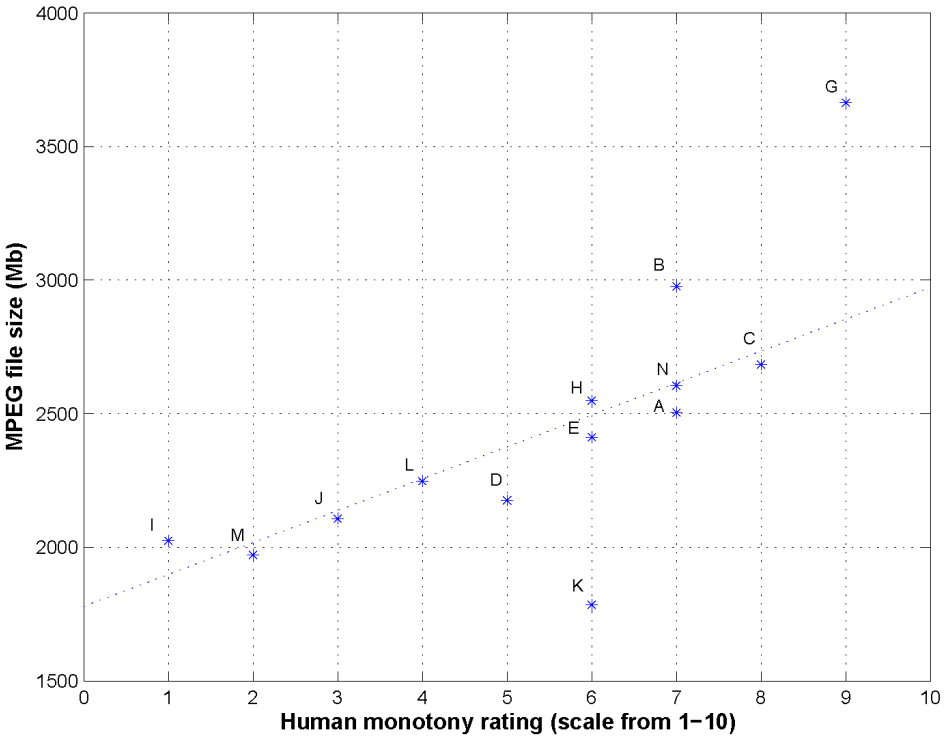


Figure 6.3: Various video file sizes versus an experienced-driver evaluated monotony scale. 1 = very monotonous, 10 = very stimulating. The sequences are included on the Appendix DVD-ROM (Page 257)

file size has a strong correlation (see figure 6.3). The sole outlier is the no lane markings sequence (figure 6.2(f)), which compresses very well but should not be considered monotonous. The lack of sharp lane boundaries seems to allow a gentle transition between the frames. The course of the road is quite straight throughout the sequence (similar to figure 6.2(d)). The lack of lane markings adds a degree of difficulty to the lane keeping task, thereby decreasing the monotony.

6.3 Augmenting MPEG with lane tracking

The primary failing of the MPEG monotony detector is in situations of poor visibility such as fog. The task is not monotonous yet the video will compress well. Detecting these cases would be possible as other groups have demonstrated systems capable of estimating the visibility of the road ahead. Hautire and Aubert (2003) implemented a system that decomposed the luminance of the road ahead to judge the visibility range. We will use the look-ahead distance in our previously developed lane-tracking system (Chapter 3) as a similar measure.

In our lane-tracking system, a confidence measure is used to vary the look-ahead

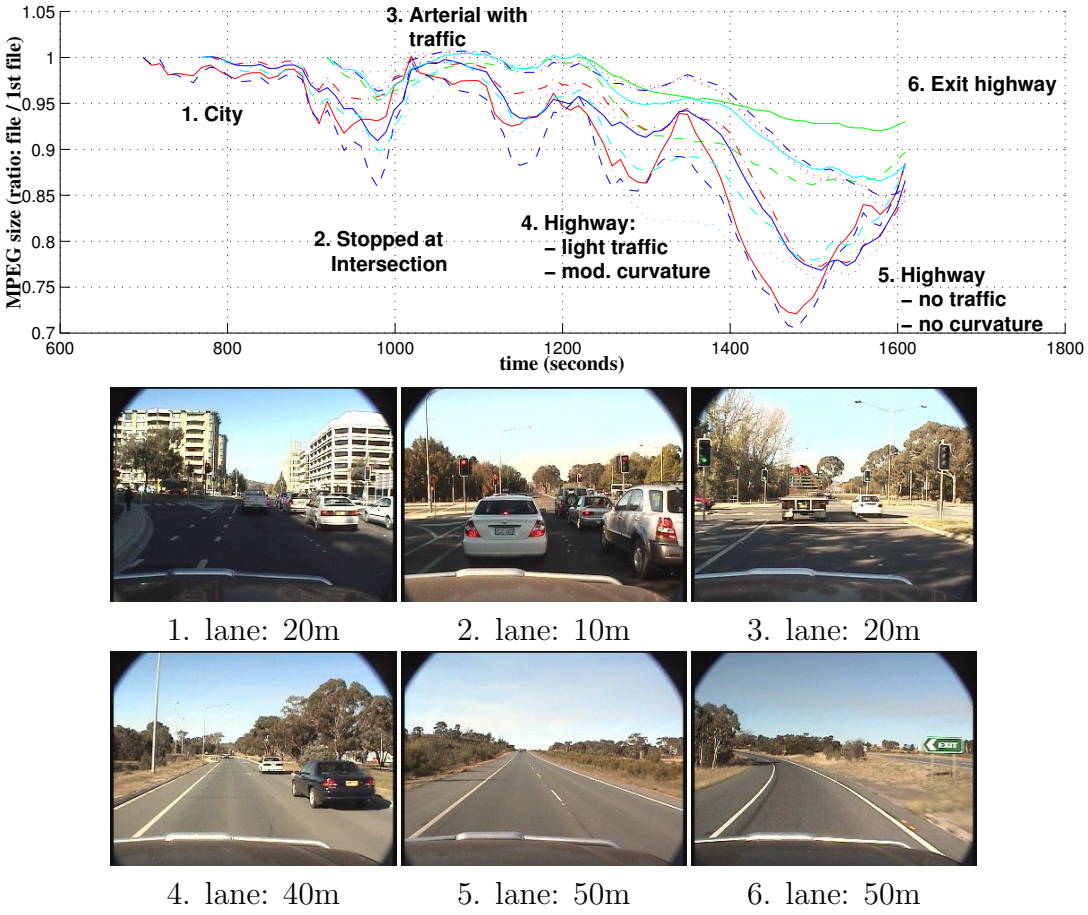


Figure 6.4: MPEG compression and lane tracking look-ahead during afternoon trial on city and arterial roads. Sample images from the camera are shown at the corresponding numbered points with the lane tracking look-ahead distance.

distance. When the variance of the primary road state variables (lateral offset, vehicle yaw and road width) increase beyond a small tolerance, the look-ahead distance is reduced to concentrate on robustly tracking the road directly in front of the vehicle at the expense of the far-field. As the near-field estimate converges, the look-ahead distance is increased once more. The lane tracking look-ahead distance has the additional benefit in the monotony detection system of detecting other subtle cases such as crests (which may not show up as significant changes in the compressed sequence) and the gravel road case shown in figure 6.2(f). On a gravel road, the lane tracker is still capable of tracking road using the colour difference between the road and the road shoulder and the weak edge information at the gravel boundary, but the increased uncertainty of the soft lane edges serves to keeps the look-ahead distance low, indicating non-monotonous conditions.

6.3.1 Initial road trials

We conducted trials during the day, dusk and at night. Each time the vehicle was driven from the city out on an arterial road, on to a highway then back on a country road to the city.

To investigate how best to use MPEG encoding to represent monotony, we encoded a set of movies every 20 seconds with various sampling rates and sequence lengths. We trialled sampling at frequencies of: 4Hz (15[/60 frames]), 3Hz (20), 2Hz (30), 1Hz (60), 0.5Hz (120) with total durations of 10 seconds to 5 minutes. Figures 6.4 and 6.5 show the results of a day trial driving out from the city and back on the country road, respectively. Figure 6.6 shows the result of a night trial along the same route. Overall the results were very promising. Both graphs show the largest trough in the MPEG file size when the car was stopped for a prolonged period at road works. Trends of smaller file size (or increased monotony) appear as the vehicle leaves the city for the highway and along the country road both during the day and at night. There is good consistency across all MPEG frequencies and durations, showing the monotony measure is well conditioned and not just an artifact of a particular sampling rate or total interval. As would be expected, the smaller duration sequences are more responsive to brief changes in conditions while the longer sequences reflect the longer term trends. The faster frame rates seem to vary more regardless of the total durations, indicating a better use of motion compensation. In the slower frame rates the vehicle motion between frames causes a significant difference in the image appearance. The change between images is too dramatic to capture using the motion compensation. The compression for these longer rates still represents a measure of the similarity of the scene over time, but it is more of a general scene appearance instead of sequential motion. The lane tracking look-ahead distance was effective in identifying sections of road with a higher monotony level than expected by the MPEG compression alone. In particular, cases such as moderate-curvature country roads, crests and sections with no lane marks were identified as less monotonous than the compressed MPEG file would suggest.

6.3.2 MPEG to a metric

An MPEG file is an amalgam of key frames which are equivalent to JPEG images and motion frames. However, the MPEG size does correlate significantly with the road scene variance. Using an absolute measure for the MPEG compression size seems arbitrary. Does it make sense to say that one size fits all, that a monotonous road is less than some X bytes for t seconds of video? It would be useful to have a metric that is not based upon an absolute number, but something that can be a rule of thumb across all video configurations. An attractive metric is the ratio of the MPEG size to the size of the set of JPEG images used to produce the sequence. The rationale is that the size of the JPEG images represents the scene complexity with no exploitation of image motion, whereas the MPEG size

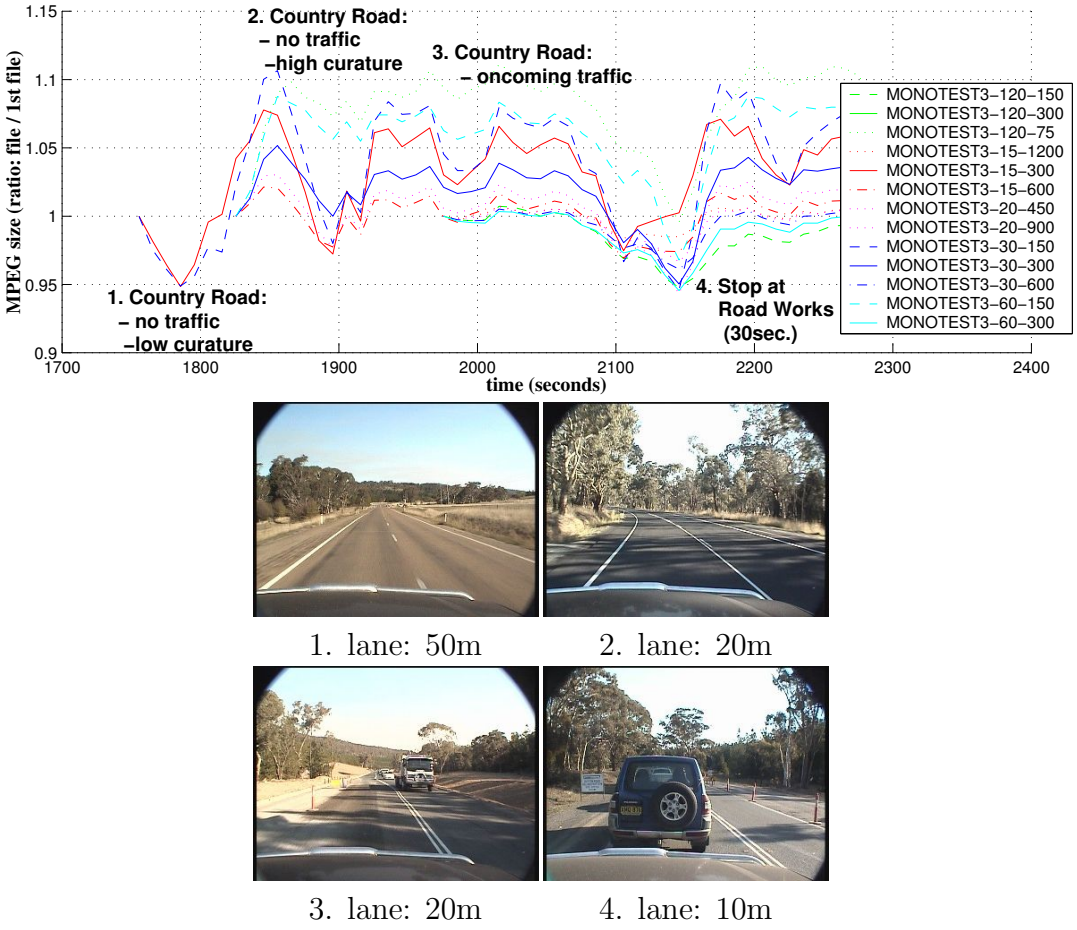
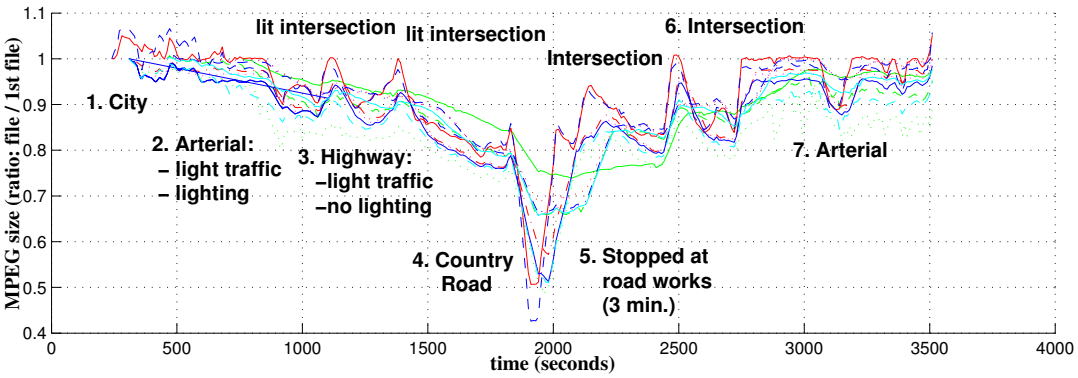


Figure 6.5: MPEG compression and lane tracking look-ahead during afternoon trial on a country road back into the city. Sample images from the camera are shown at the corresponding numbered points with the lane tracking look-ahead distance.

does exploit image motion. For example, two minutes of video waiting at an intersection in a city seen will have a large sum of jpegs (SJPEGS) size, while having a small MPEG movie size since little in the scene is changing.

6.4 Road trials

Two types of trials were conducted. First, a repeatability experiment to verify that the same road in a similar condition will return a similar response. Then, an extended trial to evaluate the metric over a comprehensive selection of road environments.



1.lane: 10m 2. lane: 40m 3. lane: 40m

4. lane: 40m 5. lane: 20m 6. lane: 0m

7. lane: 30m

Figure 6.6: MPEG compression and lane tracking look-ahead during a night trial on a city, arterial and country roads. Sample images from the camera are shown at the corresponding numbered points with the lane tracking look-ahead distance.

6.4.1 Repeatability verification

To validate that the developed monotony metric consistently evaluates the prevailing road conditions, we conducted a repeated trial on a likely monotonous stretch of highway. Three trials of the same stretch of road were conducted. Figure 6.7 shows the results of the trials. The have been folded on top of each other for comparison. MPEG files from these trials are included in the Appendix DVD-ROM (Page 257).

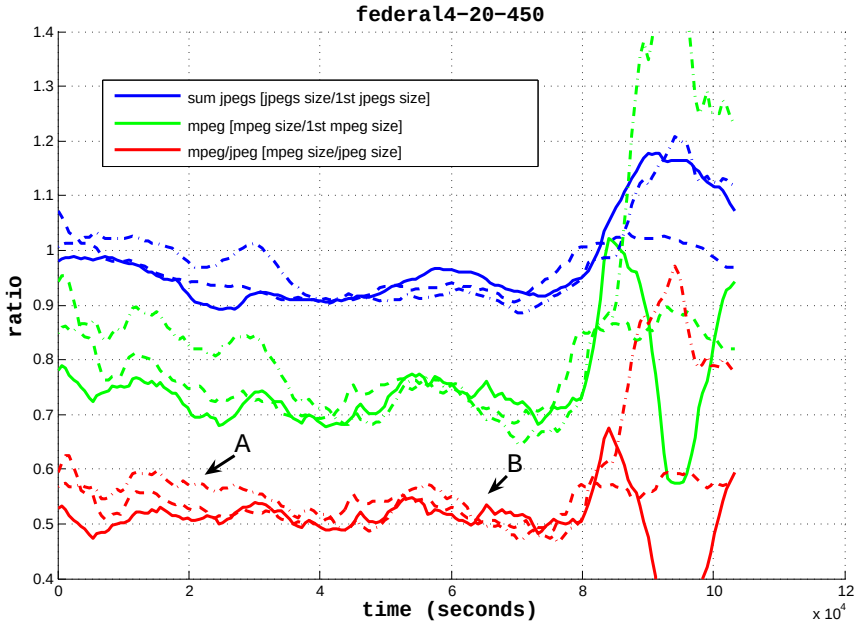


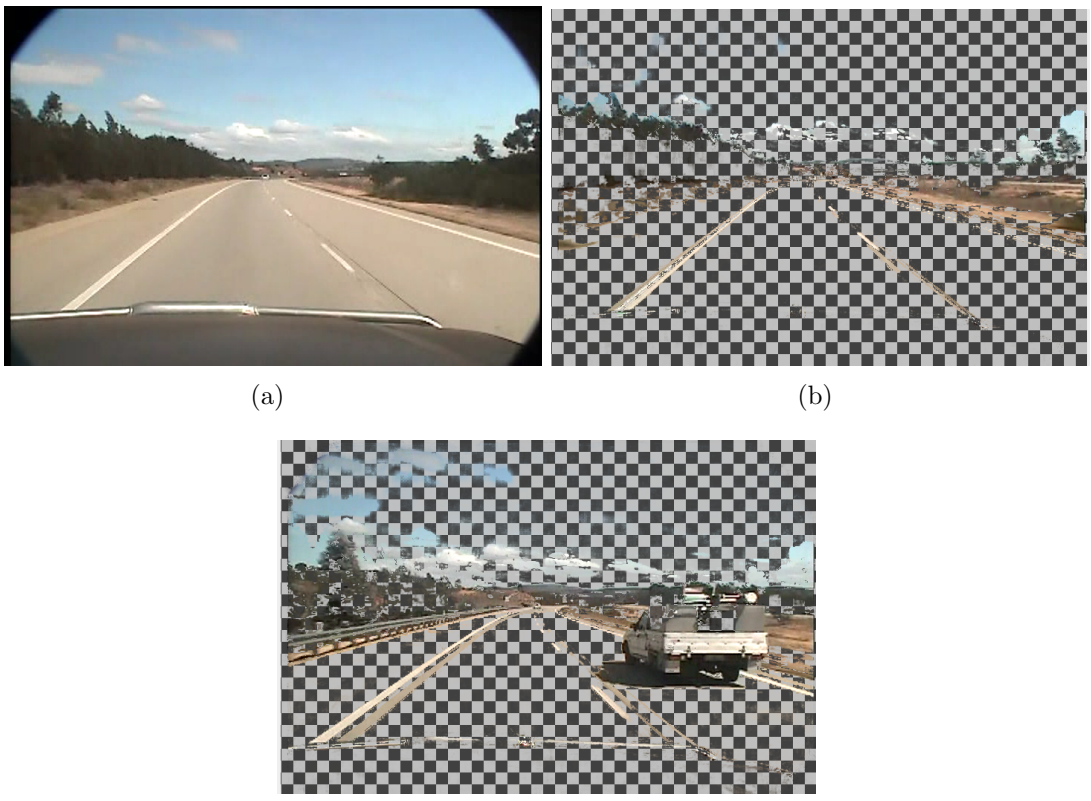
Figure 6.7: Repeated highway trial. Three trials of the same stretch of road were conducted. The trials have been overlaid. MPEG monotony metric (3Hz 2min. 30sec. sliding window). **blue:** Sum of contributing JPEG image files (ratio: current/first file, offset upward for clarity) . **green:** MPEG file size (ratio: current/first file, offset upward for clarity). **red:** ratio: MPEG file/Sum JPEG files. **A**: bored test driver decides to follow cars more closely on third trial. **B**: Traffic breaks monotony on first trial.

We examined the resultant MPEG files to verify that meaningful road scene motion made up the basis for the motion compensation. Figure 6.8 shows some sample incremental differences made to the road-scene video. The changed region of video illustrates that meaningful road scene change is being encoded in motion compensation. Figure 6.8(b) shows the encoded updates to the road scene (in Figure 6.8(a)). There are mainly low frequency (smooth) changes in the road appearance, some high frequency (sharp) changes at the top of tree line. In Figure 6.8(c) there are significant high frequency changes as a vehicle overtakes.

With confidence in the validity of the metric, we conducted a significantly more substantial trial.

6.4.2 Canberra to Geelong round trip trial

To test the metric over an extended duration and on a variety of road types, we logged data from a 1800 kilometre round trip. The route is shown in Figure 3.31. The first leg was a down a coastal highway, the return leg was on an inland motorway. Handi-cam data was logged on a coastal road trip from Canberra



(a) (b)

(c)

Figure 6.8: Changed region of video illustrating that meaningful road scene change is being encoded in motion compensation. (a) Original image (I-frame). (b) Accumulated updates showing road scene changes (several P-frames later). (c) Significant changes caused by overtaking vehicle (several P-frames later still).

to Batemans bay down to Geelong, following the Princes highway (Route 1). The return trip from Geelong to Canberra was direct via the Hume highway (Route M31). The dual-lane section of road between Albury and Melbourne has large diameter curves and little competition from road other users, making it the most likely section to qualify as monotonous. The stretch of road between Melbourne and Geelong is also multiple lane with large diameter curves. The road between Albury and Canberra, however, is often single-lane (in both directions) with periodic overtaking zones and more varying curvature. This road is likely to be tedious under heavy traffic, but not monotonous. The accident database referred to below reflects a pattern of likely fatigue-related accidents along the Albury-Melbourne and Geelong-Melbourne roads (see Figure 6.11).

On the coastal trip from Canberra to Batemans Bay, the road crosses the Great Dividing Range. This section of road is very winding and unlikely to be monotonous. The road from Batemans Bay to Geelong was long and a section in north-western Victoria, where the road cuts through bushland, was known among drivers to be quite monotonous.

Sixteen 90-minute videotapes log the passage of this trip. The journey down the coast road took twice as long as the inland return journey.

Figure 6.9 chronicles the trip from Canberra to Geelong via the coastal road. The monotony metric confirms the stimulation of the Great Dividing Range crossing before arriving at the coastal town of Batemans Bay. To a human driver, very little of the route is monotonous. From Batemans Bay to Geelong the road has a reasonable amount of scenery. Some stretches on arterial roads toward Melbourne felt monotonous to the driver.

Figure 6.10 chronicles the trip from Geelong to Canberra via the inland highway. The highway is an arterial route up the east coast of Australia.

The Geelong to Melbourne road, as well as some stretches north of Melbourne, were evaluated as monotonous by the metric. The Geelong to Melbourne road is known to be fatigue black spot.

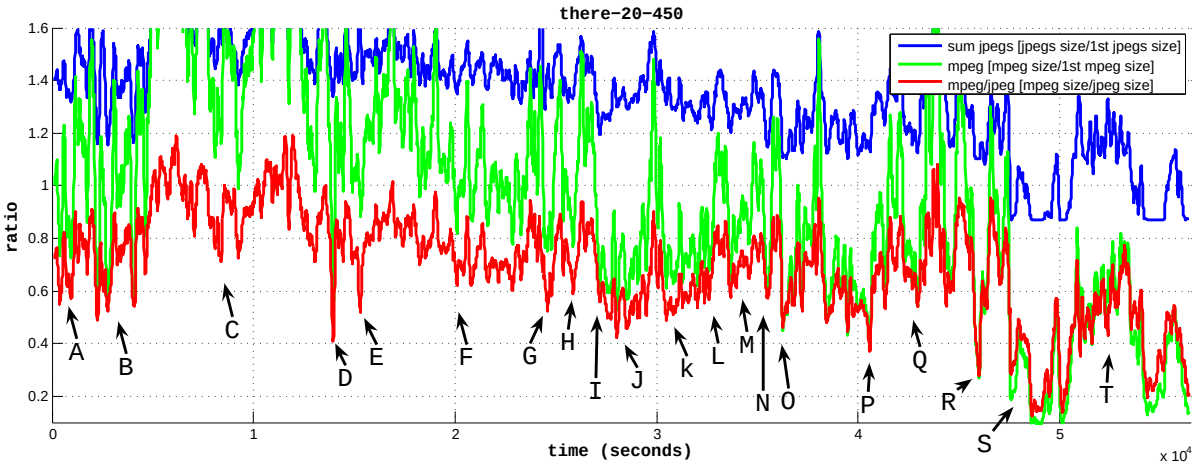
Figure 6.11 maps night-time highway crashes over the period of 1989 to 2005. This map was produced using the Victorian government (Vicroads) 'CrashStats' online application. Using the query interface, terms were selected to narrow the results down to likely fatigue cases. As with the ATSB (2006a) referred to in Chapter 2, this database has a limited number of facts about vehicle accidents that can be selected against. Again, this data does not include additional data from police reports such as the suspected contributing factors, so fatigue can not directly be selected. Instead the query is phrased to catch a representative sample of these crashes. In this case: night crashes causing serious injury or death were selected on highway roads where the accidents involved lane departure and collision with road side obstacles, trees, or barriers.

On the whole, crashes correlate to traffic volumes of the roads, however it is interesting to note regions on the Princes highway coming into Melbourne, in between Geelong and Melbourne and north of Melbourne on the Melbourne-Albury road feature in this mapping.

MPEG files from this road trip are included in the Appendix DVD-ROM (Page 257).

6.5 Discussion

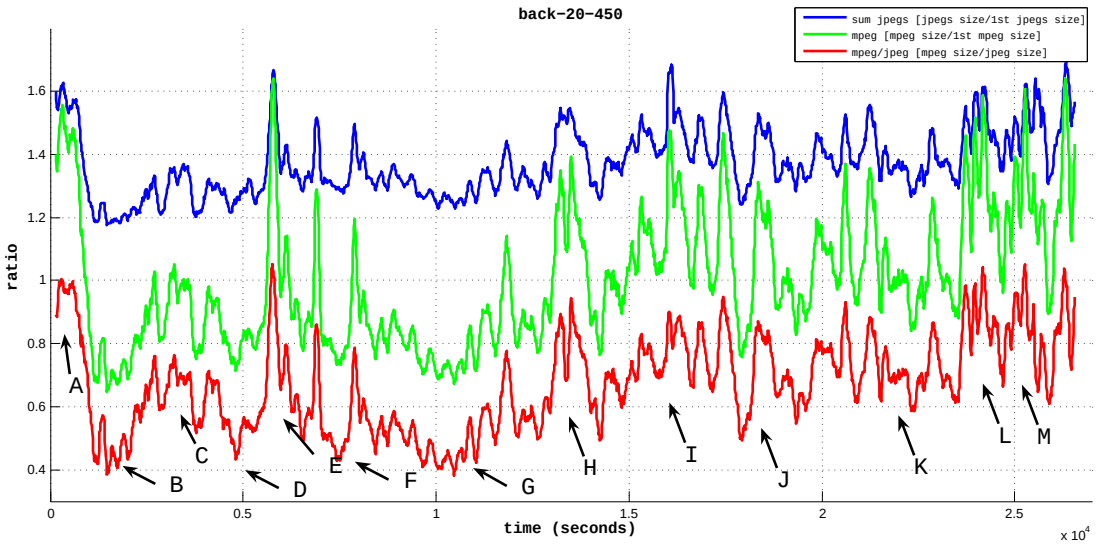
The MPEG monotony metric does provide a repeatable robust estimate of the road scene variability over time. Coupled with lane tracking look-ahead distance and the road speed, a useful warning system for potential danger points could be created. For a final system, longer window durations would be preferable to reduce false positives due to monotonous scenes moments after a dramatic road scene change. A higher frame-rate improves variance regardless of duration as more continuity of motion can be used between frames. As shown in Figures 6.4, 6.5 and 6.6 there is good consistency of the metric across MPEG



(a)

Key	Time (sec.)	Description
A	500	Arterial roads out of Canberra.
B	3,000	Winding but otherwise open roads on Kings Highway.
C	6,000	Tight winding roads and traffic through Great Dividing Range.
D	14,000	Waiting at road works.
E	15,300	Waiting at road works.
F	20,000	Winding roads with less traffic.
G	24,600	Stop to clean windshield.
H	25,900	Small section of relatively straight open road.
I	27,500	Heavy rain at the coast.
J	27,900	Sections of gravel road.
K	31,000	More sections of gravel road.
L	32,500	Country town.
M	35,000	Undulating country highway.
N	35,500	Open straight road.
O	36,300	Open straight road behind car.
P	41,000	Open straight arterial road into Melbourne at twilight.
Q	44,000	Melbourne outskirts in twilight.
R	46,000	Open straight arterial road in twilight.
S	48,000	Night. Unfortunately the camera hasn't been set to near infrared sensitive mode.
T	53,000	Night. Reasonable results where there are enough street lights.

Figure 6.9: Canberra to Geelong Coast road trial. **(a):** MPEG monotony metric (3Hz, 2min. 30sec. sliding window). **blue:** Sum of contributing JPEG image files (ratio: current/first file, offset upward for clarity) . **green:** MPEG file size (ratio: current/first file, offset upward for clarity). **red:** ratio: MPEG file/Sum JPEG files.



(a)

Key	Time (sec.)	Description
A	500	Geelong outskirts.
B	1,500	Geelong Road.
C	3,600	Melbourne Ring Road.
D	5,000	Hume highway, just north of Melbourne, light traffic.
E	5,200	Hume highway, north of Melbourne, heavy traffic.
F	7,500	Hume highway, north of Melbourne, open road.
G	10,000	Hume highway, mid. Victoria, open road.
H	14,000	Albury city (not bypassed).
I	16,000	North of Albury, open but winding road. Often single lane, behind traffic.
J	18,000	Small section of divided dual lane open road.
K	22,500	More but winding road. often single lane, behind traffic.
L	23,800	Winding narrow road near Murrumbateman near Canberra.
M	25,000	Arterial roads on edge of Canberra.

Figure 6.10: MPEG monotony metric - Geelong to Canberra Hume highway trial. **(a):** MPEG monotony metric (3Hz, 2min. 30sec. sliding window). **blue:** Sum of contributing JPEG image files (ratio: current/first file, offset upward for clarity) . **green:** MPEG file size (ratio: current/first file, offset upward for clarity). **red:** ratio: MPEG file/Sum JPEG files.

durations and frame frequencies so there is no one required choice. Around 5 minutes duration with a 4 Hz frame-rate would be a reasonable default.

As a rule of thumb, a monotony ratio (MPEGS / Sum JPEGS) of less than 0.5 is a potential point of concern for visual monotony. Checks also need to be made of significant lane tracking look-ahead distance and a non-trivial vehicle speed.

City scenes benefit significantly from motion compensation. However monotonous scenes are still more extreme, regardless of the underlying scene complexity.

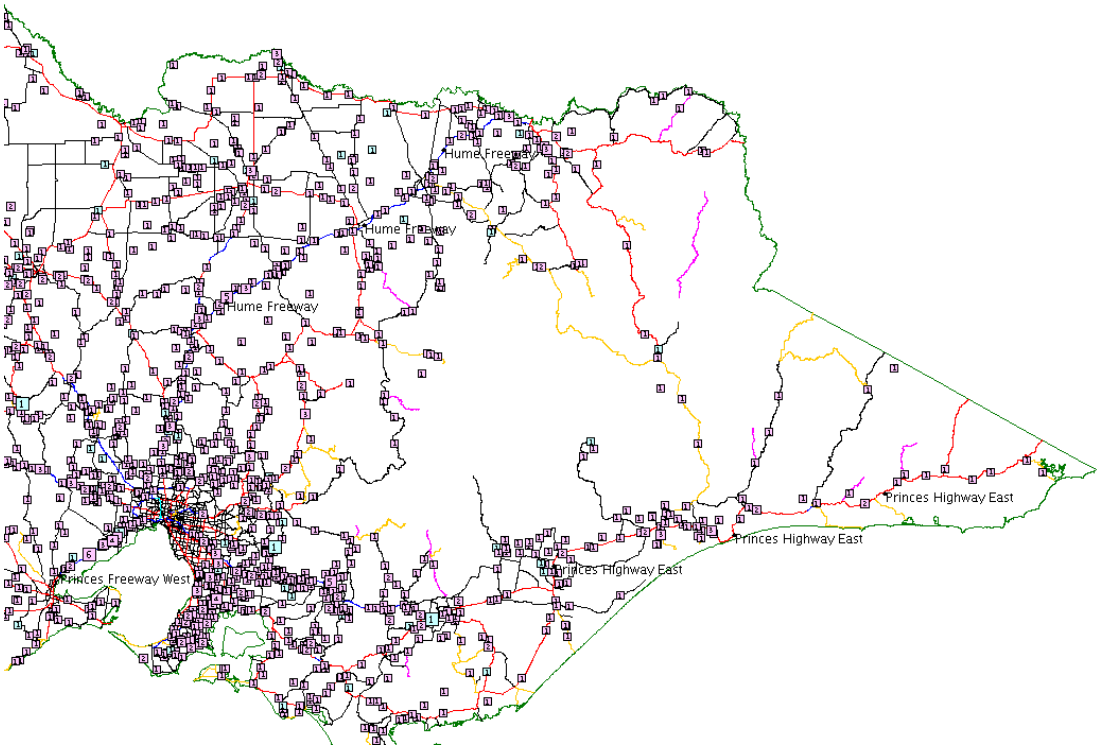


Figure 6.11: Map of Victorian night-time highway road departure crashes 1989-2005. Produced using VicRoads (2006).

The JPEG images vary substantially in size (a factor of 2) and may also contain information relevant to assess the road scene. The trend follows the city-to-highway scene complexity change.

6.6 Visual clutter

Mourant *et al.* (1969) found that the visual workload of the driver had a significant impact on driving performance. Current research by Edquist *et al.* (2005) speculates that road scene complexity is highly correlated with driver distraction.

Making an Advanced Driver Assistance System able to independently assess road scene complexity would be a significant advance in road safety. An online system able to judge when the vehicle is at a busy intersection, or on a quiet road, and make an on-the-spot assessment of the road-scene complexity would provide valuable input for a driver-workload manager.

In our Automated Co-driver, a visual-clutter system would be an important addition to monotony detection for driver workload management.

Although digital maps are able to provide some context for the road scene, such

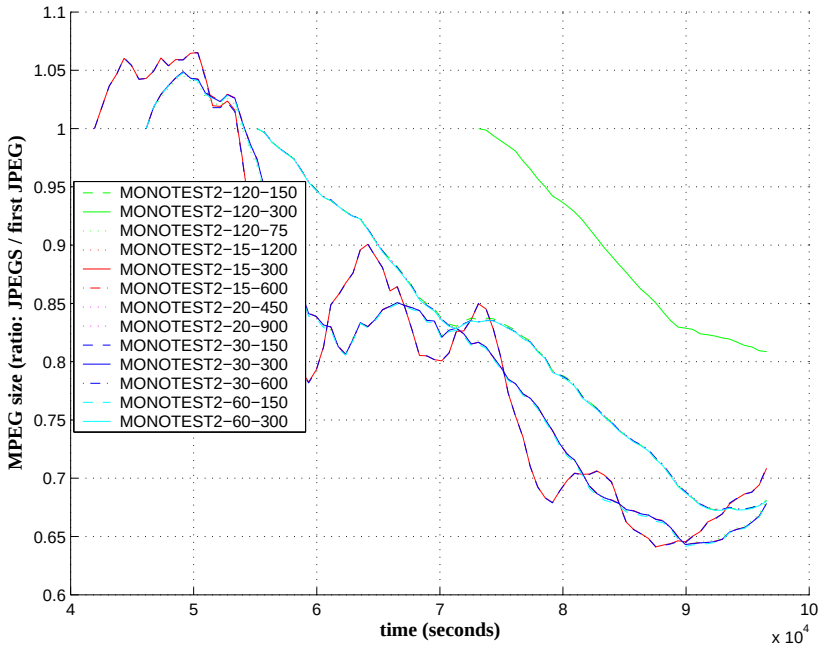


Figure 6.12: JPEG file size in the city to country trials. Note the sharp steps from city to arterial to highway scenes.

as tagging the road type, there are two outstanding issues. First, urban, highway, and country scenes are not homogeneous. In the city there are stretches of arterial road that are made substantially easier to navigate than city routes. Second, traffic volumes change dramatically across the day, affecting the concentration required to drive.

During the investigation into the use of MPEG compression to represent monotony, it became evident that there also was as strong correlation between scene complexity and the JPEG file size. This correlation could also have a use in assessing the road environment. Figure 6.12 shows the JPEG file size across the sliding window period used in the MPEG encoding above. Using the sum of the JPEG file sizes or using a short window MPEG encoding could deduce how busy the road scene is.

Care must be taken when examining JPEG files in this way. While the file size correlates with the scene complexity, there is no interpretation of the scene. So unlike even the previous MPEG approach, where the motion vectors did map to salient road scene content, JPEG image size is solely a measure of the image complexity (not the road scene complexity). There is much potential here but we must leave it to future work to use the JPEG size or small time window MPEG files to estimate driver workload in our system.

6.7 Summary

A monotony detector has been developed using MPEG compression to measure the change of information content of the road scene over time. The detector shows promise as a useful measure of the monotony of the driving task. Such a measure can be used to reduce false positives of fatigue-monitoring systems. Though the correlation between MPEG compressibility and monotonous sequences is high, there are some special cases such as low visibility weather and gravel roads that need to be handled explicitly. Using additional image analysis in the form of the lane tracking look-ahead, these cases can be managed.

A reasonably fast frame rate (minutes) seems to be the most promising for capturing monotony. Using a ratio of the MPEG file size over the sum of equivalent JPEG images (at 85% quality) generates a consistent monotony metric. A ratio of 0.5 or less indicates potentially monotonous road conditions. The metric was tested in a repeated verification and 1800 kilometres of road data.

This chapter concludes our exploration of road-scene vision processing. Next we will integrate our road-scene analysis with our driver gaze and vehicle monitoring systems to develop an Advanced Driver Assistance System modelled on the unique and powerful concept of an Automated Co-driver which could revolutionise future safety systems in road vehicles.

Chapter 7

Automated Co-driver experiments

Finally in this chapter we are ready to integrate the developed components into a prototype Automated Co-driver Advanced Driver Assistance System. As mentioned in Chapter 2, our proposition is that the majority of accidents occur due to driver inattention to the on-driving task crucial seconds before the incident. Many drivers have avoided accidents due to a warning from a vigilant passenger. We argue that if a real-time in-vehicle system could be invented to detect momentary driver inattention then many crashes may be avoided. Similar to a vigilant passenger many life critical vehicles have a co-pilot so we call this concept an Automated Co-driver. To accurately interpret driver behaviour, particularly driver inattention, requires simultaneous knowledge of the driver, the vehicle and the road context. To detect driver inattention, our approach is to model the driving task as a closed loop system involving the vehicle, driver and the road environment. The need to understand the driver's observations is crucial to detecting driver inattention. The systems implemented will not only be responsive to the road events or the driver's actions but also respond to the driver's observations. As illustrated in Figure 7.1, by reconciling the driver eye-gaze direction with detected road scene features we can estimate the driver's observations.

To investigate the feasibility of the technologies to estimate the driver's observations, we undertook a series of trials.

- **Road *centre* inattention detection Automated Co-driver** - A simple application using the driver observation monitoring and the vehicle speed alone. This system implements instantaneous driver inattention detection using an assumed road centre.
- **Road *event* inattention detection Automated Co-driver** - This system integrates a single road environment cue, namely speed sign recognition, with gaze monitoring to demonstrate the basic cases of driver observation monitoring.

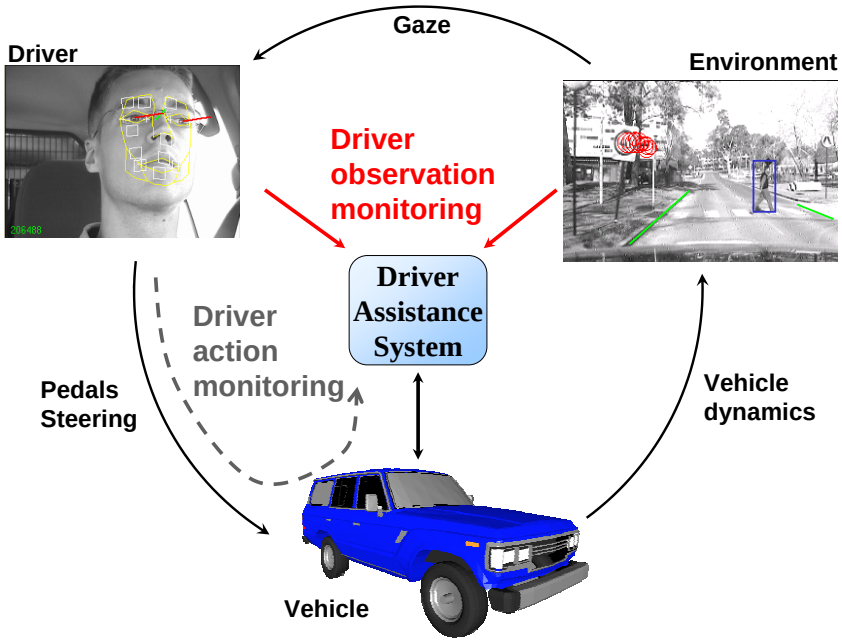


Figure 7.1: Implemented driver observation based Advanced Driver Assistance System.

- **Comprehensive inattention detection Automated Co-driver** - The final system integrates all the subsystems to demonstrate our best approximation to a functional Automated Co-driver.

In Section 7.1 we start by investigating and validating the use of direct driver eye-gaze monitoring to determine where the driver is looking. We quickly discover that the converse problem of where the driver is *not looking* is the correct formulation. Testing for this case is shown to be feasible. After a brief discussion about how our Automated Co-driver Advanced Driver Assistance Systems are actually constructed in Section 7.2 we detail our first Automated Co-driver in Section 7.3. This first system uses driver eye-gaze monitoring and the vehicle speed to implement an instantaneous inattention detection system. Section 7.4 extends the first Automated Co-driver by adding road scene event awareness in the form of speed sign recognition. This Automated Co-driver is shown to select interventions according to the behaviour matrix in Table 7.1. Finally, in Section 7.5 our final system demonstrates driver observation monitoring applied to other road scene events including lane departure, visual monotony and obstacle detection.

7.1 Correlating eye gaze with the road scene

The key mechanism required for driver observation monitoring is to reconcile objects in the road scene with the driver's eye-gaze direction.

7.1.1 Review of gaze monitoring in vehicles

Direct driver monitoring has been the subject of clinical trials for decades, however monitoring for use in Advanced Driver Assistance Systems is relatively new.

Head position and eye closure have been identified as strong indicators of fatigue (Haworth *et al.*, 1988). In addition to direct observation for fatigue detection, driver monitoring is useful for validating road scene activity. When augmented with information about the vehicle and traffic, additional inferences can be made.

Gordon A. D. (1966) conducted an in-depth analysis of perceptual cues used for driving. He described driving as a tracking problem. In on-road systems, Land and Lee (1994) investigated the correlation between eye gaze direction and road curvature, finding the driver tended to fixate on the tangent of the road ahead. Apostoloff and Zelinsky (2004) used gaze and lane tracking to verify this correlation on logged data, also observing that the driver frequently monitored oncoming traffic. Ishikawa *et al.* (2004) explored back projecting the driver's gaze direction onto the scene, but scene features weren't identified. Takemura *et al.* (2003) demonstrated a number of associations between head and gaze direction and driving tasks on logged data.

Salvucci and Liu (2002) showed, in simulator studies, that drivers shift their primary focus to the destination lane at the very start of a lane change maneuver. We can look for this change in our assistance systems to identify unintended lane departures.

The behaviour of the driver several seconds before and after an important detected road event is used by a vigilant passenger and can be used by us to decide whether to issue a warning. Recall that Neale *et al.* (2005) found 78% of accidents involved driver inattention 3 seconds or less before the incident. Driver monitoring is achieved via an eye-gaze tracking system and vehicle instrumentation. The developed road scene vision systems provide road scene context and detect events.

7.1.2 Proving a correlation

Scene camera and eye configuration is analogous to a two-camera system (see Figure 7.2). Gaze directions trace out epipolar lines in the scene camera. If we

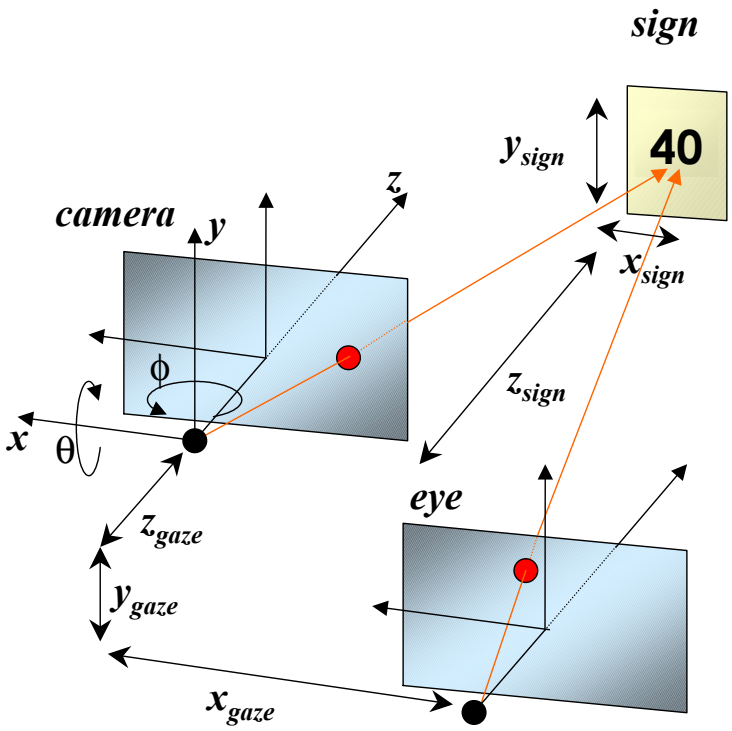


Figure 7.2: The scene camera and gaze direction is analogous to a two camera system.

had a depth estimate of the driver's gaze, we could project to a point in the scene camera. Similarly, if we had the object depth we could re-project on to the eye and estimate the required gaze. A depth estimate of the gaze is hard to obtain. A common assumption is that the gaze angle and angle in the scene camera are the same. In practice this assumption amounts to supposing that either the scene camera is sufficiently close to the driver's head (equivalent to a trivial baseline) or that the objects of interest are near infinity (Land and Lee, 1994; Takemura *et al.*, 2003; Ishikawa *et al.*, 2004). In these cases error bounds on the gaze direction (not fixation duration) are infrequently used and even less frequently justified.

The object depth could be estimated using a second scene camera running the same detection software or assumptions on object size and/or road layout. However, it is desirable to maintain flexibility of the implemented object detection systems which could use a single camera and has no strong assumptions on the object scale. If we assume the depth of the object is unknown, we can instead model the effect of the disparity error in our confidence estimate.

The effect of an unknown stereo disparity will be a displacement along the epipolar line defined by the gaze direction on to the scene camera. The disparity, as with any stereo configuration, will be most apparent for close objects and reduce by a $1/x$ relationship with distance from the baseline. The angular deviation reduces as the angle becomes more obtuse. To get an upper bound of

the likely disparity deviation we can compute the worst-case disparity for our camera configuration. With reference to Figure 7.2, and using the scene camera centre as a world reference frame, the scene camera and gaze angles for a sign at $(X_{sign}, Y_{sign}, Z_{sign})$ can easily be derived as the following equations;

$$\Delta\theta = (\theta_{cam} - \theta_{gaze}) = \arctan\frac{X_{sign}}{Z_{sign}} - \arctan\frac{X_{sign} + X_{gaze}}{Z_{sign} + Z_{gaze}}, \tag{7.1}$$

$$\Delta\phi = (\phi_{cam} - \phi_{gaze}) = \arctan\frac{Y_{sign}}{Z_{sign}} - \arctan\frac{Y_{sign} + Y_{gaze}}{Z_{sign} + Z_{gaze}}. \tag{7.2}$$

The worst-case disparity then translates to when the object is closest to the vehicle on the driver's side of the road, equivalent to an object on the right shoulder of a single-lane road (note that our system is in a right-hand drive vehicle). The field of view of the scene camera limits the closest point at which the object is visible. The closest visible object is at $(-3.0, -1.6, 8.0)$ for the 50° field of view of the camera. The worst-case height of the object relative to the scene camera, -1.6, would be when it is on the ground (this is worse than any actual case as the object would not be visible due to the bonnet). With pessimistic estimates of the driver (far) eye position relative to the scene camera manually measured to be: $(X_{gaze} = 0.22, Y_{gaze} = 0.1, Z_{gaze} = 0.2)$ the final errors become;

$$\Delta\theta = (\theta_{cam} - \theta_{gaze}) = (20.6° - 18.7°) = 1.9°, \tag{7.3}$$

$$\Delta\phi = (\phi_{cam} - \phi_{gaze}) = (11.3° - 10.4°) = 0.9°. \tag{7.4}$$

Therefore the worst expected deviation due to stereo disparity is ±1.9° horizontally and ±0.9° vertically, which is on par with other error sources in the system. The expected deviation for the majority of cases where the sign is further away is significantly less. The deviation is twice as large in the horizontal direction, implying that a suitable approximation of the tolerance region will be an ellipse with a horizontal major axis.

To determine the overall tolerance of the system, two further factors need to be accommodated. The gaze tracking system has an accuracy of ±3° and the field of view of the foveated region of the eye is estimated to be around ±2.6° (Wandell, 1995). The accumulated tolerance is the sum of these sources which for our experimental setup comes to ±7.5° horizontally and ±6.6° vertically. This allows us to claim that the driver was very unlikely to have seen the object if the object and gaze directions deviate by more than this tolerance.

7.1.3 System setup

To align the scene camera with the gaze direction, the default rotation and scaling between the gaze coordinate system and the scene camera must be determined.

While these parameters can be obtained through knowledge of the scene camera parameters and the relative position of the gaze tracking system, an online initialisation is effective and allows easy re-calibration (for zoom changes, gaze head model changes etc.). The driver is asked to look at several (≥ 4) features visible from the scene camera. The best features are points that approximate points at infinity such as along the horizon. The gaze direction is measured and the points are manually selected in the scene camera. The rotational offset and scaling can then be computed using least squares.

7.1.4 Verifying the foveated field of view

Figure 7.3: Screen-shot of PC test application to gather data on gaze object recognition field of view. A test subject is requested to fix gaze on the cross. The Speed sign periodically varies in position and speed (10,20,30,...,90). A road video sequence was played in the background to add context for the trials. The subject entered the 'guessed' sign via numeric keypad ('1'-'9').

To get a sense for the field of object discrimination ability of the driver, a simple trial was constructed using a desktop PC. Wandell (1995) states the foveated region of the eye to be around $\pm 2.6°$, though as illustrated in Figure 2.13(b) in Chapter 2, a substantial cone density exists up to around twenty degrees. A simple program was made to present speed signs featuring varying speeds to the subject at different positions on the screen. Highway road footage was shown in the background to add a small amount of realism to the scene. The test subject was asked to fix their gaze on a stationary cross on the screen and to press the corresponding number on the keypad to match the presented sign. If the subject broke their gaze or otherwise invalidated the trial, they pressed no key or '0'

and the result was discarded. Figure 7.3 presents a typical screen-shot from the test program. A set of five subjects with normal (or corrected) eye sight were each presented with 200 signs (100 fixating to the left, 100 fixating toward the right). The result (see Figure 7.4) shows that reliable discrimination ability drops off at over 4 degrees from the fixation point. The test was conducted on three subjects with normal vision or corrected vision with glasses normally worn for driving. This trial is by no means intended to be the last word on the field of discrimination of the eye. The use of the trial was to provide a rule of thumb for the discrimination ability of the driver in the context of the road objects. The estimated field of discrimination will be used to judge whether the driver would be able to see a particular road object.

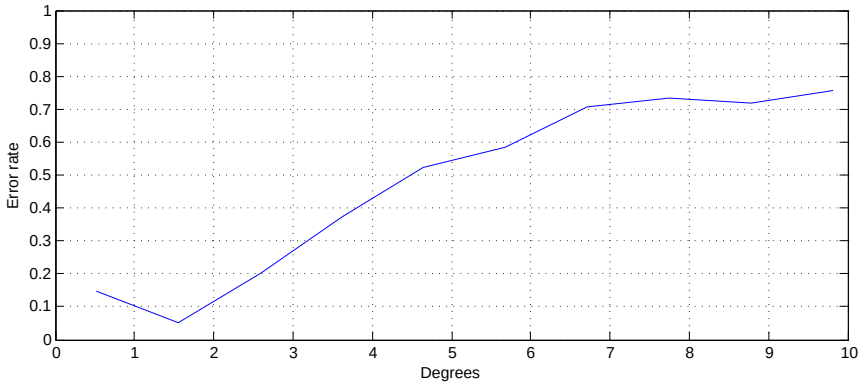


Figure 7.4: Gaze PC test error rate. Above a 4.5° angle from the fixation point the test subjects were recognising one in two signs correctly.

7.1.5 In-vehicle verification

We conducted a verification experiment to test that the Automated Co-driver was indeed able to detect whether the driver missed an object. The driver was asked to fix his gaze on an object in the scene. A sign was then placed at a certain distance from the fixation point. The driver was then asked to identify the object. The object was one of eight possibilities. The proportion of correct classifications was logged along with the driver-gaze angle and apparent sign position in the scene camera. 30 metre, 20 metre and 10 metre depths were tested against four different displacements between the object and fixation point. The object size was 0.45 metres in diameter. For each combination of depth and displacement, ten trials were done.

Figure 7.5 shows the driver's object classification error rate versus the angle between gaze and object position. Expected recognition rates fall as the object becomes more peripheral in the driver's field of view. The results of this trial verify our expectation that while it is hard to prove the driver saw an object, it is possible to estimate, with reasonable confidence, when the driver was unlikely

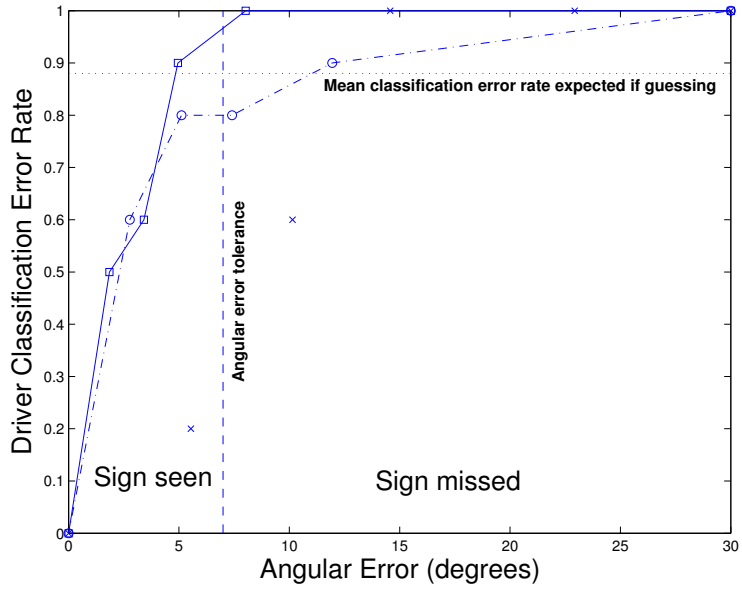


Figure 7.5: Driver recognition rate of objects in peripheral vision for various depths. Dotted horizontal line shows expected value due to chance. Vertical dashed line represents ±7.5° derived tolerance. it squares: 30 metre points. it Circles: 20 metre points. it crosses: 10 metre points.

to have seen the object. A curious effect was noticed (represented by a cross in the middle of the graph) when the driver was very close to the object. The large apparent size of the object in the driver's field of view seemed to aid the recognition rate. However, this only occurred when objects were close to the vehicle, which is not when drivers typically see road objects. The driver reported not consciously being able to see the object in this case.

This verification demonstrates the expected strength of the system: the ability to detect when the driver has missed an object. It is impossible to determine whether the driver saw the object, as, even with perfect measurement of a perfect gaze direction match, the driver's attention and depth of focus cannot be determined. If the driver is looking in the direction of the object, it is an ambiguous case whether the driver noticed the object, thus no warning is issued.

7.2 Automated Co-driver design

The ability to estimate what the driver is looking at is a key mechanism for implementing an intuitive and unobtrusive system. By monitoring where the driver is looking many unnecessary warnings can be avoided. Unnecessary warnings can be suppressed and necessary warnings can be made more relevant. As long as a road event, such as an overtaking car, or wandering pedestrian, is noted by the

driver no action needs to be taken.

The Automated Co-drivers detailed in this chapter can be thought of (and are implemented) as instances of the ADAS logic engine. The ADAS logic engine is the software module in control of the macro behaviour of the system. Figure 7.6 illustrates how the components discussed in previous chapters are related in the driver assistance application. This chapter outlines the development of an Automated Co-driver to combat inattention, then a co-pilot to aid with sign recognition and finally, brings all the systems detailed in this thesis together to demonstrate our Automated Co-driver concept.

The principal difference between the different Advanced Driver Assistance Systems implemented is the behaviour of the ADAS logic engine (as discussed in Appendix A).

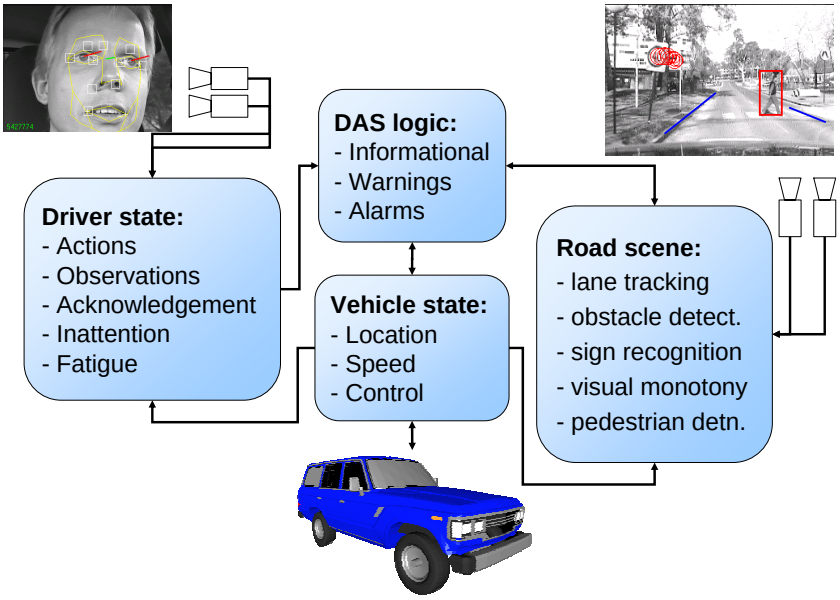


Figure 7.6: Implemented ADAS software component architecture. The "DAS Logic" module dictates the actual purpose of the driver assistance system. All the other modules represent sets of data servers.

It is important to note the the limits of the proposed trials. The trials aim to demonstrate the efficacy of the technologies to create such a system. Enough trials will be conducted to demonstrate that the results are deterministic, not chance. The outcome of the trials will hopefully form the investigative ground work for a clinical trial of an Automated Co-driver. However the trials will not have the statistical significance to prove a safety improvement due to an Automated Co-driver. Such a trial would involve the long term deployment of the technology in a significant number of vehicles. Instead we are limited to a single vehicle and a pool of test drivers restricted to students or staff members in the Department

authorised to drive the vehicle. As such each system was trialled on a minimum of three drivers from the Department but not working directly on the Research. The drivers were coached through the eye-gaze tracking system calibration process and were aware their gaze was being tracked while driving, however they were not aware of the details of how the gaze information was being used and the details of the experiments. The drivers were instructed to drive normally through and around the University campus on a route directed by the Experimenter.

7.3 Road *centre* inattention detection Automated Co-driver

The purpose of our Advanced Driver Assistance System is to demonstrate the immediacy of direct driver gaze monitoring. Previous systems used in simulation or road trials have used metrics such as the variance in steering wheel movements to gauge the fatigue level of the driver. Because these systems compute statistics over time, there is a significant lag between driver inattention and action by the system. Online real-time driver gaze monitoring can easily be used to detect short periods of driver distraction.

Similar to the percentage road centre metric, driver gaze can be analysed to detect even shorter periods of driver distraction. The faceLAB^{TM} system readily allows the implementation of an online distraction detector. The gaze direction is used to reset a counter. When the driver looks forward at the road scene, the counter is reset. As the driver's gaze diverges, the counter begins. When the gaze has been diverted for more than a specific time period, a warning is given. The time period of permitted distraction is a function of the speed of the vehicle. As the speed increases, the permitted time period could drop off either as the inverse (reflecting time to impact) or the inverse squared (reflecting the stopping distance). We use the inverse square (see Figure 7.7). The warning can be auditory, tactile or visual but should be capable of degrees in intensity, raised to the extent which the diversion is over time. Once the driver is observed to have had a stable gaze at the road ahead, the counter and the warning is reset until the next diversion. Since the vehicle speed is considered, normal driving does not raise the alarm. As more dramatic movements such as over the shoulder head checks occur at slow speeds, the tolerance is longer. Situations, such as waiting to merge, when the vehicle is not moving permit the driver to look away from the road ahead indefinitely without raising the alarm.

Using the faceLAB^{TM} system, we can monitor driver head pose and eye gaze direction via the driver state engine. The watchdog timer inside the driver state engine will be used to verify whether the driver has not looked at the road for a significant period of time. An auditory warning is given if the driver is seen to be inattentive. The auditory warning is included on the Appendix DVD-ROM (Page 257).

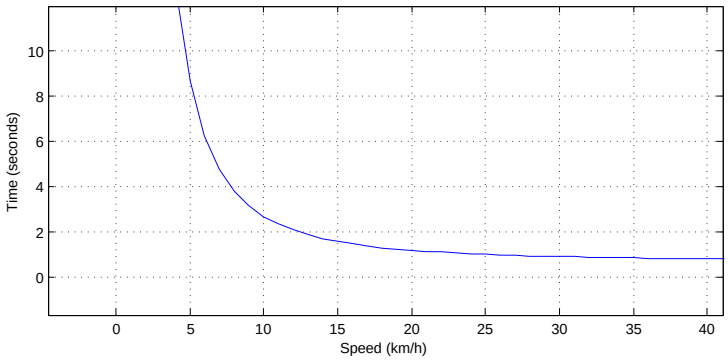


Figure 7.7: Duration of permitted inattention for a given speed.

7.3.1 On-road trials

Figure 7.9 is a sequence of screen-shots showing the typical response with a distracted driver. Once the distraction threshold is crossed for the given speed, audio warnings increase until the driver looks forward again. Then the warnings are reset until next time. One conclusion from the trials was that a road position estimate would help the system. In some cases when the road had significant curvature, the system was triggered because the gaze direction was substantially away from straight ahead.

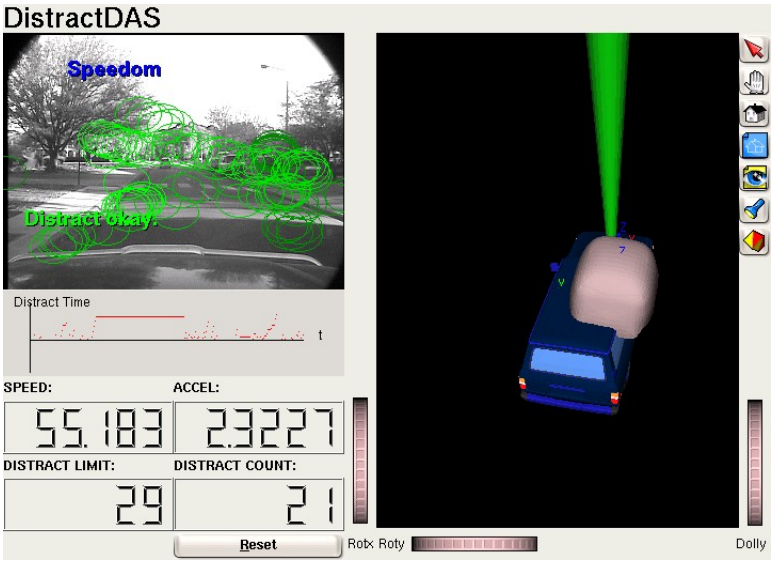
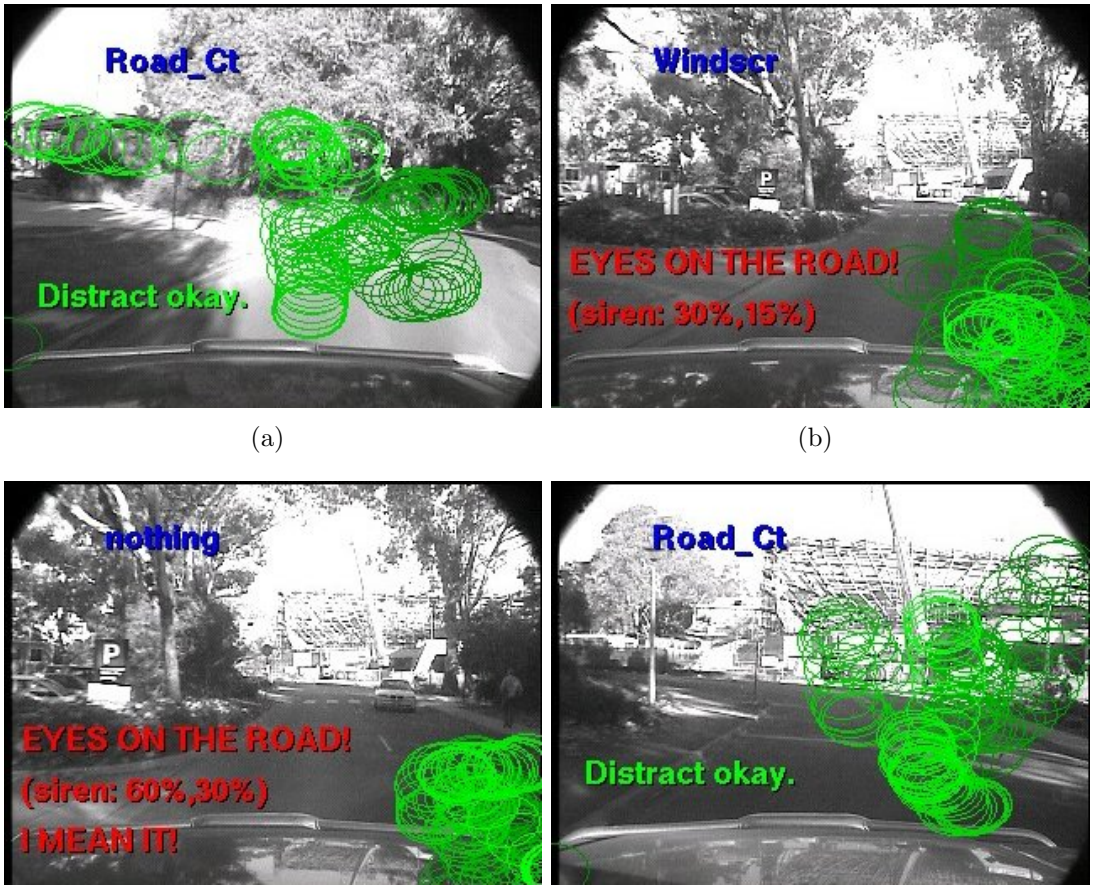


Figure 7.8: Road scene inattention detection Automated Co-driver screen-shot.



(a) (b)

(c) (d)

Figure 7.9: Road scene inattention detection Automated Co-driver screen-shot sequence. Circles represent driver gaze. (a): driver looking forward. (b): driver looking at right mirror, alarm sounding. (c): driver still distracted, alarm louder. (d): driver gaze has returned to the road ahead, alarm silenced.

7.4 Road *event* inattention detection Automated Co-driver

This section presents a Road event inattention detection Automated Co-driver. An autonomous detection system recognises a road event, in this case speed signs. At the same time, a driver monitoring system verifies whether the driver has looked in the direction of the road sign. If it appears the driver is aware of the road sign, the information can be made available passively to the driver. If it appears the driver is unaware of the information, an auditory warning is generated.

The macro behaviour of the assistance system can be described according to the matrix in Table 7.1.

Driver Behaviour	**"Seen"**	**Missed**	**Acknowledge**
Consistent with road event	OK	INFO	OK
Inconsistent with road event	INFO	WARN	INFO

Table 7.1: Driver behaviour matrix used by Automated Co-driver to determine the appropriate. *OK*: No action required. *INFO*: Provide visual reminder. *WARN*: Provide auditory and visual reminder.

For this case, if the driver appears to have "seen"[1] a speed sign, the current speed limit can be simply recorded on the dashboard adjacent to the speedometer. However, if it appears the driver has not looked at the road sign, and over time, a speed adjustment is expected and has not occurred, a more prominent warning would be given. This still leaves the driver in control of the critical decision, but supports him or her in a way that aims not to be overly intrusive. Warnings are only given when the driver is not aware of the change of conditions. Finally, the warning can also be cancelled by observing the driver: a glance at the speedometer confirms that the driver is aware of his or her speed and the new detected limit.

Using the reasoning in Section 7.1.2, the worst expected deviation due to stereo disparity is $\pm 1.9°$ horizontally and $\pm 0.9°$ vertically which is on par with other error sources in the system. The expected deviation for the majority of cases where the sign is further away is significantly less. The deviation is twice as large in the horizontal direction, implying that a suitable approximation of the tolerance region will be an ellipse with a horizontal major axis.

To determine the overall tolerance of the system, two further factors need to be accommodated. The gaze tracking system has an accuracy of $\pm 3°$ and the field of

[1] The word "seen" for the rest of this chapter we simply mean that the driver's eye-gaze was close to the road feature. We can't say that the driver saw the sign because that requires cognition.

Figure 7.10: Screen-shot showing '60' sign detected and seen by driver. **top left:** Live video showing eye gaze (*large circles*) and status (*overlaid text*). **bottom left:** Last detected sign (*small circles*) and eye gaze (*large circles*). **top right:** 3D model of current vehicle position, eye gaze (*oversize head*) and sign location. **bottom right:** Current detected speed limit, vehicle speed, acceleration and count down for speeding grace period in frames.

view of the foveated region of the eye is estimated to be around $\pm 2.6°$ (Wandell, 1995). The accumulated tolerance is the sum of these sources, which for our experimental setup comes to $\pm 7.5°$ horizontally and $\pm 6.6°$ vertically. This allows us to claim that the driver was very unlikely to have seen the sign if the sign and gaze directions deviate by more than this tolerance.

To correlate the eye-gaze with the sign position, the histories of the two information sources are examined. The sign detection sub-system provides a history of the sign location since detected. This includes all frames from when the sign was first detected before the sign was able to be verified or classified. Similarly, the faceLAB^{TM} data provides the historical head pose and gaze direction. When a sign has been classified, the sign angles and gaze directions are checked back in time to when the sign was first detected. If the angles from any previous frame fall within the tolerance, the sign is reported as being seen by the driver. If the angles never coincide, the sign is reported as missed. The system provides a four second tolerance for the driver to achieve the speed limit. The timer is instigated when the measured speed exceeds the limit and the measured acceleration is not significantly decreasing.

(a) '40' missed. (b) '80' seen.

Figure 7.11: Primary scenarios for Road Signs Automated Co-driver when the vehicle is speeding. **left:** Live video feed showing current view, eye gaze (*dots / large circles*) and current status (*overlaid text*) during screen-shot. **right:** Last detected sign (*small circles*) and eye gaze (*dots / large circles*).

7.4.1 On-road trials

A video of the system is included on the Appendix DVD-ROM (Page 257). The system was able to detect speed signs around the University and evaluate the implications for the driver. Figure 7.10 shows a screen-shot of the system demonstrating a typical case. Figures 7.11 and 7.11 illustrate the primary scenarios encountered. In Figure 7.11(a) the driver was watching a pedestrian and failed to notice a '40' sign. The Automated Co-driver has detected that the driver did not see the sign and has issued a red *sign: missed!* warning. Figure 7.11(b) shows an instance where an '80' sign was detected; the driver saw the sign and the vehicle was not speeding so no red warning was issued. Similarly, in Figure 7.12(a) a '40' sign was detected. The driver saw the sign, the system assumed the driver was intentionally speeding so a warning was displayed but no sound alert generated. In Figure 7.12(b) the driver has missed the last sign and is speeding for more than a predefined grace period without decelerating. The *SLOW DOWN!*

(a) '40' seen. Car speeding.

(b) '40' missed. Car speeding.

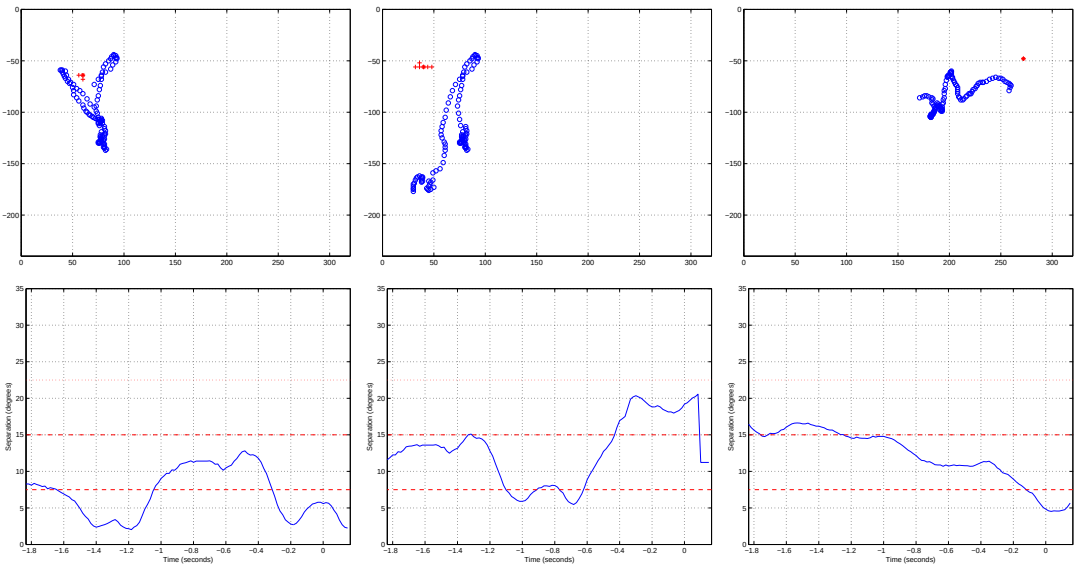
Figure 7.12: Primary scenarios for Road Sign Automated Co-driver when the vehicle not speeding. **left:** Live video feed showing current view, eye gaze (*dots / large circles*) and current status (*overlaid text*) during screen-shot. **right:** Last detected sign (*small circles*) and eye gaze (*dots / large circles*).

warning is shown and an alert sound issued.

Figure 7.13 shows the sign and the gaze direction separation angle for typical signs classified as "seen" by the system. Note the troughs in the separation angle graphs reflecting times when the driver looked toward the sign.

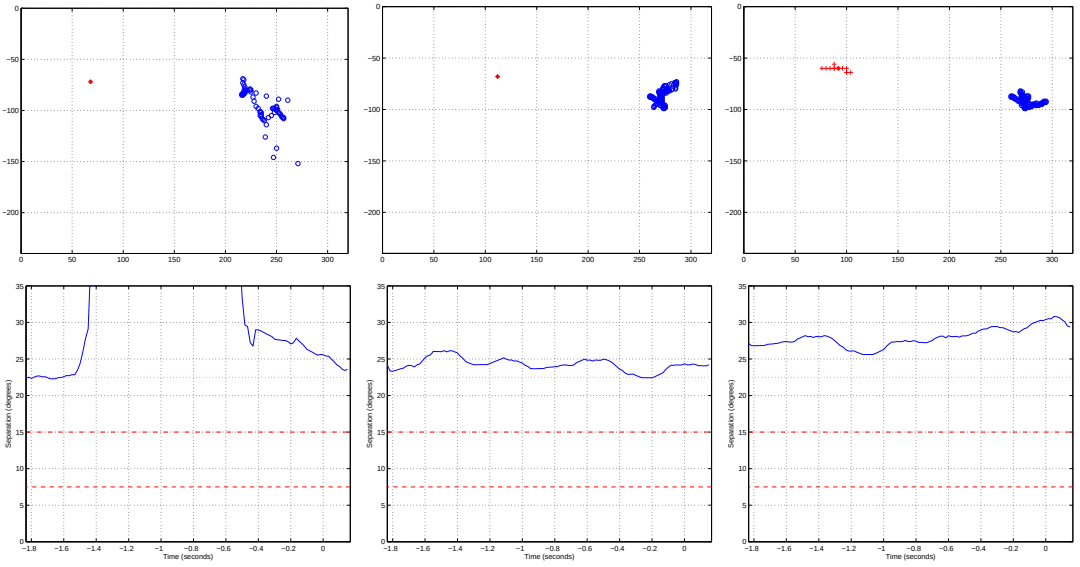
At one location in the test area there was a speed sign just past an entry road to the University. Because the of the intersection geometry and a bend shortly along the entry road the road sign while in plain view on the road sholder was not prominent. After passing the sign the drivers were asked whether they saw the sign, none of the drivers noticed the sign. Figure 7.14 shows the sign and the gaze direction separation angle for three test drivers. The system classifies these signs as missed. Notice the lack of directed gaze toward the sign, instead gaze is steady on the road and merging traffic to the right.

In Figure 7.15 are some border line sign gaze angle separation cases. There



(a) (b) (c)

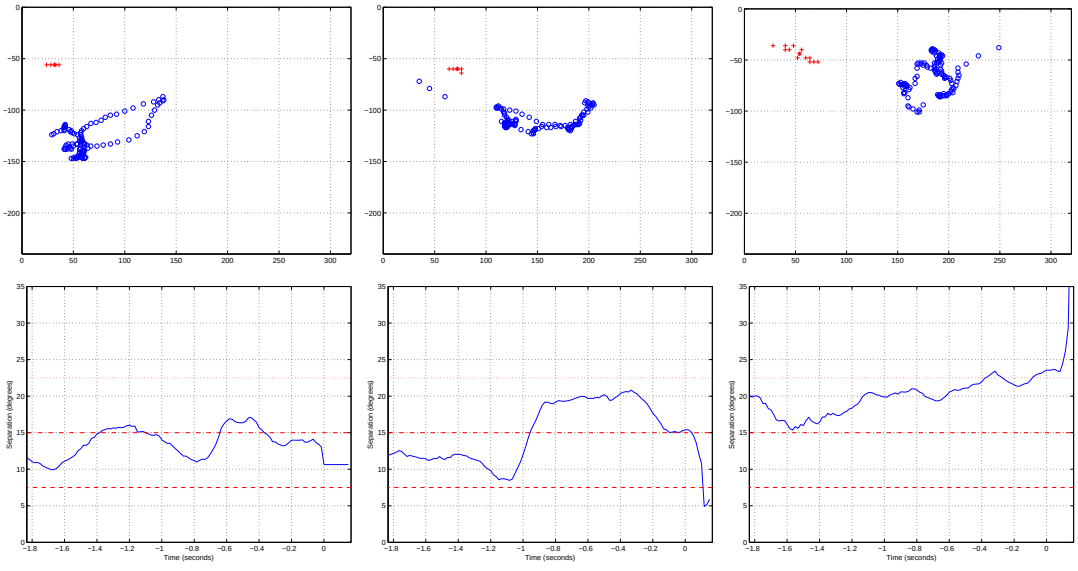
Figure 7.13: "Seen" sign, eye-gaze direction separation angle. Origin on the time axis represents the final detection of the sign. *(top)*: *(Blue 'o')*: Eye-gaze. *(Red '+')*: Sign position. *(bottom)*: Sign - gaze separation angle. *(Red dashed lines)*: 1x 2x 3x 7.5^o error tolerance.



(a) (b) (c)

Figure 7.14: "Missed" sign, eye-gaze direction separation angle. *(top)*: *(Blue 'o')*: Eye-gaze. *(Red '+')*: Sign position. *(bottom)*: Sign - gaze separation angle. *(Red dashed lines)*: 1x 2x 3x 7.5^o error tolerance.

appears to be some directed eye movement however the angle is greater than the angle error tolerance. Watching the gaze and sign detection it becomes obvious that the sign may be read well before the sign is detectable in the image. The driver eye with much better acuity than the video camera can recognise the sign further away.

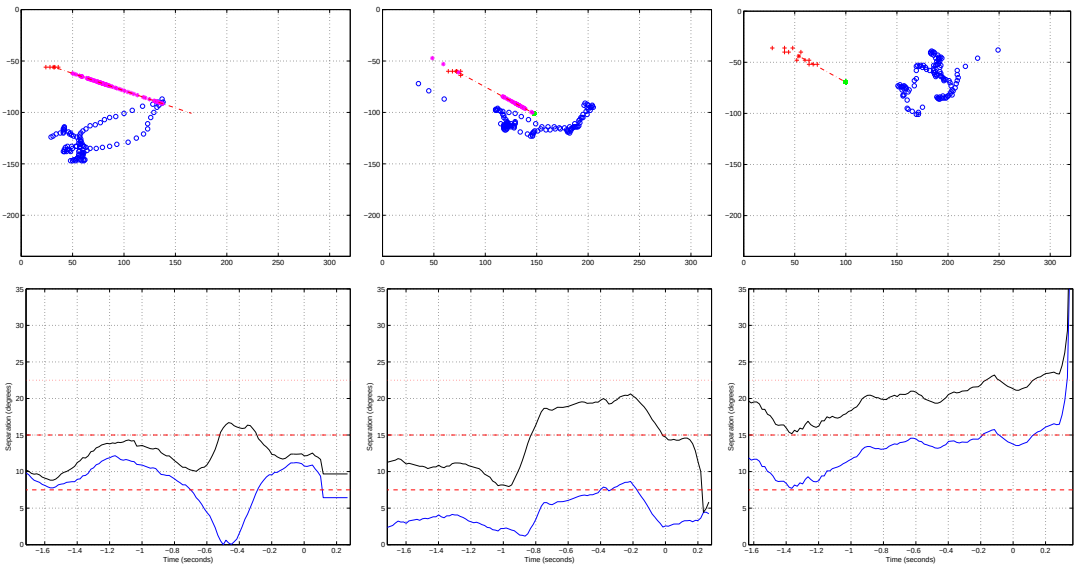


(a) (b) (c)

Figure 7.15: "Borderline" sign, eye-gaze direction separation angle. *(top)*: *(Blue 'o')*: Eye-gaze. *(Red '+')*: Sign position. *(bottom)*: Sign - gaze separation angle. *(Red dashed lines)*: 1x 2x 3x 7.5^o error tolerance.

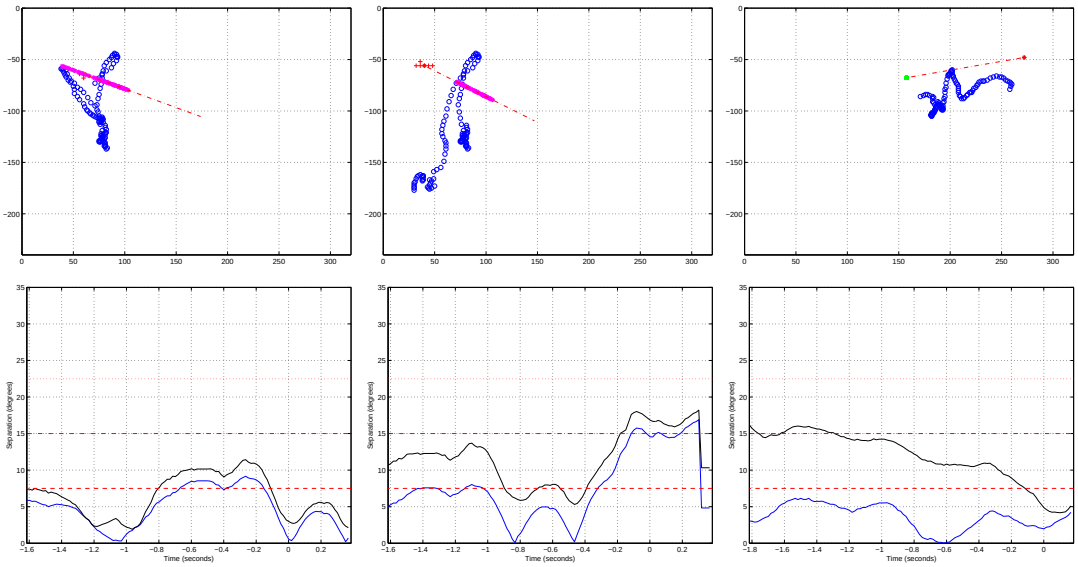
To address this case we projected the sign position back according to the recent vehicle egomotion. The sign - gaze separation angle is then taken as the minimum distance to this path. Figure 7.16 shows the projected paths and the revised sign - gaze angle separation. Now we see that Figures 7.16(a) and (b) are classified as seen and (c) remains classified as missed. Similar to Figure 7.14 in Figure 7.16(c) the sign is on a bend in the road so the sign location remains on the left of the camera image (see the "40" sign sequence 2/3rds of the way through video of the system on the Appendix DVD-ROM (Page 257).

For completeness we include Figures 7.17 and 7.18 showing the impact of the back projection of the "seen" and "missed" signs. The classifications remain unchanged.



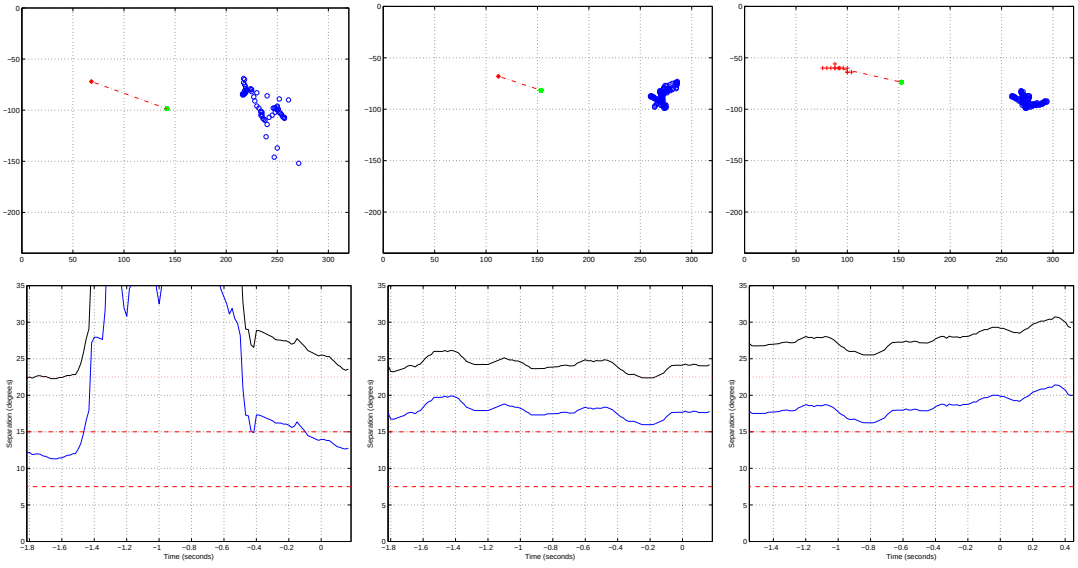
(a) (b) (c)

Figure 7.16: Sign position projected back in time to estimate sign, eye-gaze direction separation angle before the sign was large enough to track. *(top)*: *(Blue 'o')*: Eye-gaze. *(Red '+')*: Sign position. *(Magenta and green '*')*: Projected gaze point on sign path. *(bottom)*: Sign - gaze separation angle. *(Black plot)*: original separation. *(Blue plot)*: back projected separation. *(Red dashed lines)*: 1x 2x 3x 7.5^o error tolerance.



(a) (b) (c)

Figure 7.17: "Seen" sign, eye-gaze direction separation angle. Origin on the time axis represents the final detection of the sign. *(top)*: *(Blue 'o')*: Eye-gaze. *(Red '+')*: Sign position. *(bottom)*: Sign - gaze separation angle. *(Black plot)*: original separation. *(Blue plot)*: back projected separation. *(Red dashed lines)*: 1x 2x 3x 7.5^o error tolerance.



(a) (b) (c)

Figure 7.18: "Missed" sign, eye-gaze direction separation angle. *(top)*: *(Blue 'o')*: Eye-gaze. *(Red '+')*: Sign position. *(bottom)*: Sign - gaze separation angle. *(Black plot)*: original separation. *(Blue plot)*: back projected separation. *(Red dashed lines)*: 1x 2x 3x 7.5^o error tolerance.

7.5 Comprehensive inattention detection Automated Co-driver

The final experiment is an integration of the subsystems outlined above. That is: visual monotony estimation, sign reading, lane tracking, vehicle detection and pedestrian detection with driver gaze monitoring. This demonstrates our Automated Co-driver.

7.5.1 Implementation

A pedestrian detection system from elsewhere in the Smart Cars project was integrated with the developed subcomponents to create the Automated Co-driver system.

Pedestrian Detection

Gandhi and Trivedi (2006) presents a concise summary of recent vision based pedestrian detectors. We will use a system developed within the research group. A pedestrian detection system was developed by Grubb and Zelinsky (2004) (see Figure 7.19). The system used stereo cameras, V-disparity and support vector machines to achieve a high false-negative rate. This system and the work of Viola *et al.* (2005) has sponsored the development of a second system by our project collaborators, the National ICT Australia group. Viola *et al.* (2005) implemented a method of detecting pedestrians using Haar features and Ada-boost[1]. The module is used to detect and track pedestrians for our assistance system.

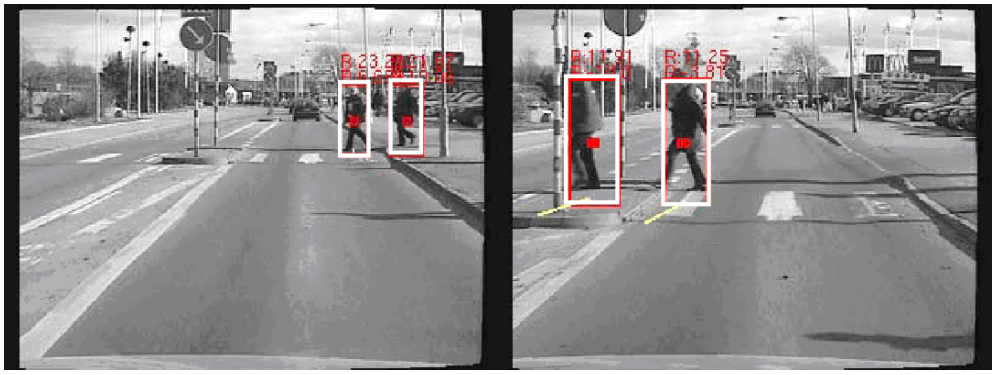


Figure 7.19: Online pedestrian detection. The basis for the system used in the Automated Co-driver.

[1]footnote: Ada-boost is a method of taking a collection of weak classifiers and "boosting" them so that together a combination of classifiers is derived that can strongly classify the data set

Road departure calculation

To determine whether the vehicle will depart the lane we use the intersection of the estimated lane geometry with the Ackermann motion model of the vehicle (see Figure 7.20). The intersection point provides a time till departure estimate assuming the driver maintains the current steering angle. To find this point we solve the lane model and Ackermann equations numerically to find the arc length to departure and, for a known speed, the time till departure.

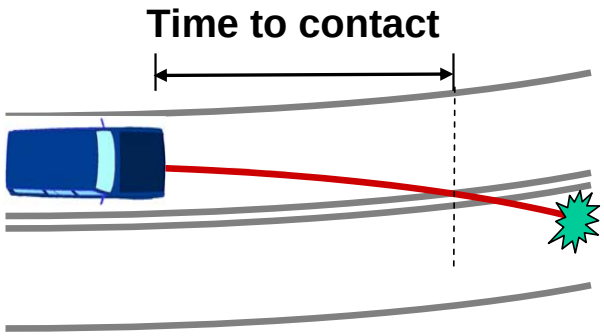


Figure 7.20: The current lane departure point is found at the intersection of the estimated lane geometry and the vehicle motion model.

To determine whether a lane departure is intentional we use the departure point, the turning indicators and the driver eye-gaze. If the turning indicators are operating and the lane departure point is in the same direction the warning is suppressed. The warning is also suppressed if the driver gaze has been sufficiently close to the departure point. If neither of these criteria are met an auditory warning is given.

Decision logic

Since multiple tasks are required the final Automated Co-driver The Co-driver uses a set of simple rules to determine when warnings would be given. Table 7.2 paraphrases the implemented decision logic used by the co-driver system. In this case we hard-coded the logic in the application for simplicity. A suitable refinement would be to use a logic engine such as so that the logic would be configurable.

7.5.2 On-line road trials

Figure 7.21 and Figure 7.22 shows some typical screen-shots of the Automated Co-driver system. Like all of our systems the interface is for the experimenter

Sample decision-tree rules used by Co-driver:

Pedestrians:
y_{ped} - lateral displacement of pedestrian.
dy_{ped} - lateral velocity of pedestrian.
y_L - lateral position of left lane boundary.
y_R - lateral position of right lane boundary.
w_{lane} - lane width.
g_{ped} - Gaze - pedestrian intersection.

1. Pedestrian on road:
   IF $(y_{ped} > y_L)$ AND $(y_{ped} < y_R)$ THEN ...
   IF (NOT g_{ped}) THEN alert:"PEDESTRIAN!" ELSE warn:"Pedestrian."

2. Pedestrian moving into danger:
   IF $(((y_L - w_{lane}) > y_{ped} > y_L)$ AND $(dy_{ped} > 0))$ OR $((y_R > y_{ped} > (y_R + w_{lane}))$ AND $(dy_{ped} < 0))$ THEN ... (ELSE info:"Ped. Moving out of danger.")
   IF (NOT g_{ped}) THEN alert:"PED. Moving into danger!" ELSE warn:"Ped. Moving into danger."

3. Pedestrian far from road:
   IF $(y_{ped} < (y_L - w_{lane}))$ AND $(y_{ped} > (y_R + w_{lane}))$ THEN info:"Ped. (no danger)"

Inattention:
t_{inatt} - Duration of inattention.
T_{inatt} - Inattention tolerance in seconds.

1. Compute T_{inatt} based on vehicle speed. $T_{inatt} \propto \frac{1}{s_{veh}^2}$

2. IF $(t_{inatt} > T_{inatt})$ THEN warn:"Inattention!"

Monotony:
m_{mono} - MPEG monotony ratio.
M_{mono} - MPEG monotony ratio threshold.
c_{lanes} - Lane tracking confidence.
C_{mono} - Monotony lane tracking confidence threshold.

1. IF $(m_{mono} < M_{mono})$ AND $(c_{lanes} < C_{mono})$ THEN warn:"Monotony!"

Table 7.2: Co-driver decision logic.

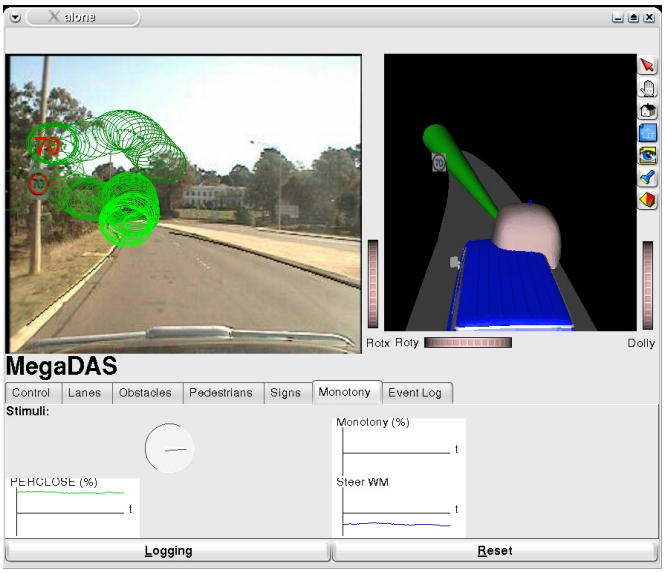


Figure 7.21: Screen-shot of the experimenter interface of the Automated Co-driver assistance system. Overlaid blue circles represent gaze direction.

not the driver, audio warnings and gaze cancellation provide the interface to the driver. A video of the system is included on the Appendix DVD-ROM (Page 257).

Figure 7.22: Screen-shot of the experimenter interface of the Automated Co-driver assistance system. Overlaid blue circles represent gaze direction.

Now follows some illustrative cases of the Automated Co-driver. Like all of our systems the interface is for the experimenter not the driver, audio warnings and gaze cancellation provide the interface to the driver. The application cycles through a decision tree to determine possible alerts. First potential pedestrian collisions, then obstacles then lane departures are verified. Then sign events, inattention, vehicle status and finally monotony events are checked. Alerts are

given with unique sounds. Approaching obstacles and pedestrians observed by the driver are not warned, nor are lane departures provided the driver is indicating or has gazed in the departure direction.

Figures 7.23, 7.24 show cases where direct driver observation enabled the system to verify that the pedestrian threat had been observed, so an alert could be suppressed.

(a)

(b)

Figure 7.23: Automated Co-driver screen-shot sequence. Circles represent driver gaze. **(a):** Approaching pedestrian detected. **(b):** Pedestrian determined to be no threat. Arrow added manually afterward.

Figure 7.25 shows the Automated Co-driver detecting a speed sign. The vehicle was not speeding so no warning was given. The system detects an acknowledgement when the driver looks at the speedo anyway.

Figure 7.26 shows inattention detected by gaze monitoring. A glance back at the

(a)

(b)

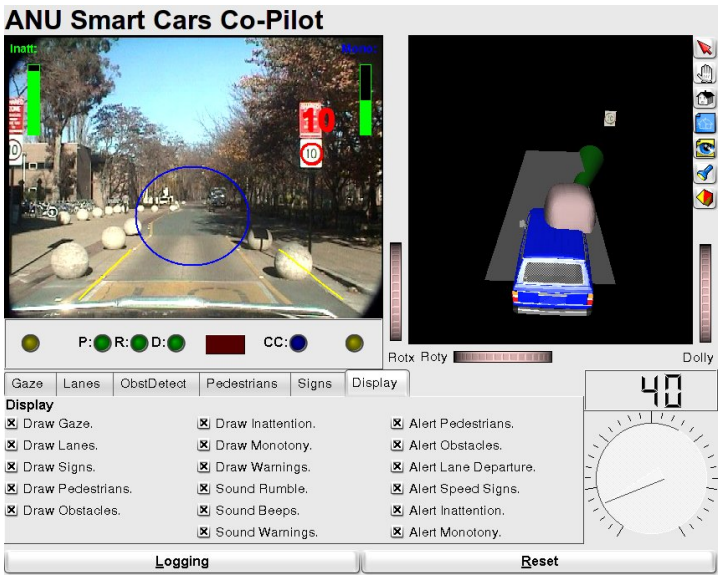
Figure 7.24: **(a):** Approaching pedestrian detected. **(b):** Pedestrian seen by driver. Arrow added manually afterward.

road resets the inattention alarm.

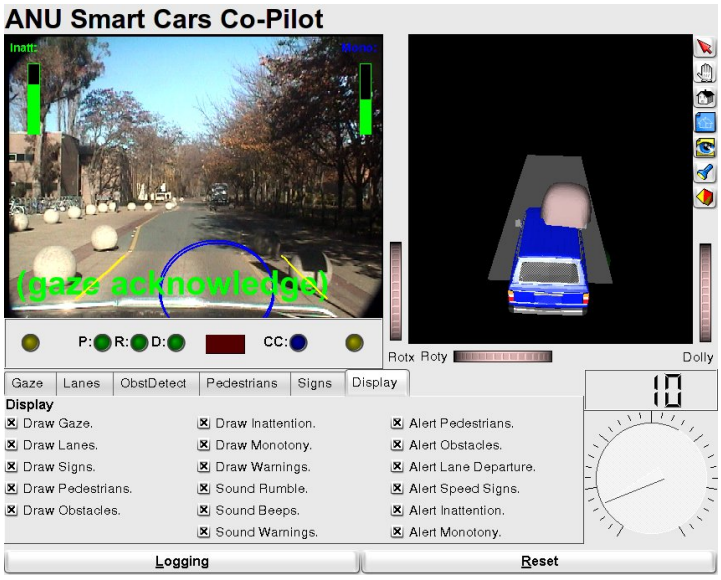
In Figure 7.27 the inattention alarm has been reset though now the driver is speeding. Direct driver observation enabled the system to verify that the speed sign had been observed so a visual warning is given, the auditory alarm is suppressed.

Figures 7.28 and 7.29 demonstrate cases of detected lane departures. Direct driver observation enables the Automated Co-driver to permit the driver to depart the lane in the sequence in Figures 7.28

Figure 7.29 shows several cases of intentional lane departure. The final case shows a lane departure without eye-gaze verification. The driver is turning left but has



(a)



(b)

Figure 7.25: Co-driver DAS screen-shot. Circles represent driver gaze. (a): sign detected. (b): Driver glanced at speedo.

not indicated or looked in that direction. Direct driver observation enables us to detect this final case without having to warn the driver during the first 3 intentional cases.

Figure 7.30 shows the lane estimate with gaze direction projected onto the ground plane. In these cases the driver eye-gaze shifts focus to the destination lane before the lane change. These cases are detected as intended lane changes.

In contrast Figure 7.31 shows an unintended lane change. The vehicle moves into

Figure 7.26: Screen-shots of the Automated Co-driver. Prolonged inattention detected. *Large circles:* driver gaze.

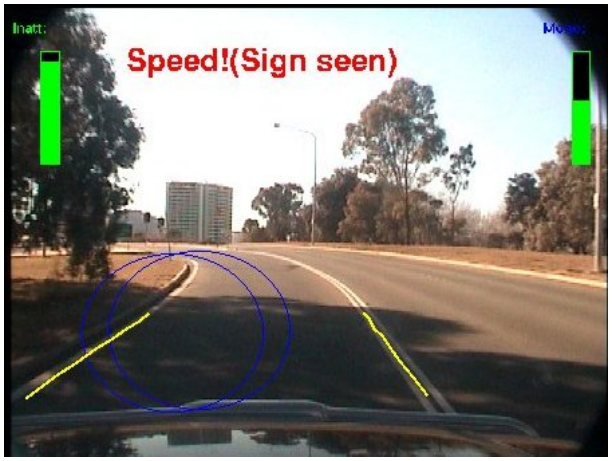


Figure 7.27: Screen-shots of the Automated Co-driver. Vehicle speeding, though last speed-sign was observed by the driver. *Large circles:* driver gaze.

the adjacent lane without an eye-gaze transition. The system reports this case as an unintended lane departure. The driver then corrects lane position bringing the vehicle back into the original lane.

Finally, Figure 7.32 demonstrates a case of a visually monotonous stretch of highway. Visual monotony is detected after several minutes of monotonous road conditions. A visual alert is given indicating the heightened fatigue risk in this scenario. The monotony is broken as the driver approaches a slower vehicle.

Lane tracking during the early automated codriver trials was poor due to a software bug introduced expanding the Distillation Algorithm source to support obstacle detection. Strong direct sunlight caused some uncertainty in the lane tracking and, at times, even disrupted the gaze tracking. When gaze tracking

Figure 7.28: A sequence of Screen-shots of the Automated Co-driver. A lane departure that was seen by the driver is shown. *Large circles:* driver gaze.

was strong and the driver was attentive, uncertainty in the road scene vision did not warrant a warning. When the gaze tracking was failing or when prolonged inattention was detected while the road scene uncertainty was high a low volume tone was sounded.

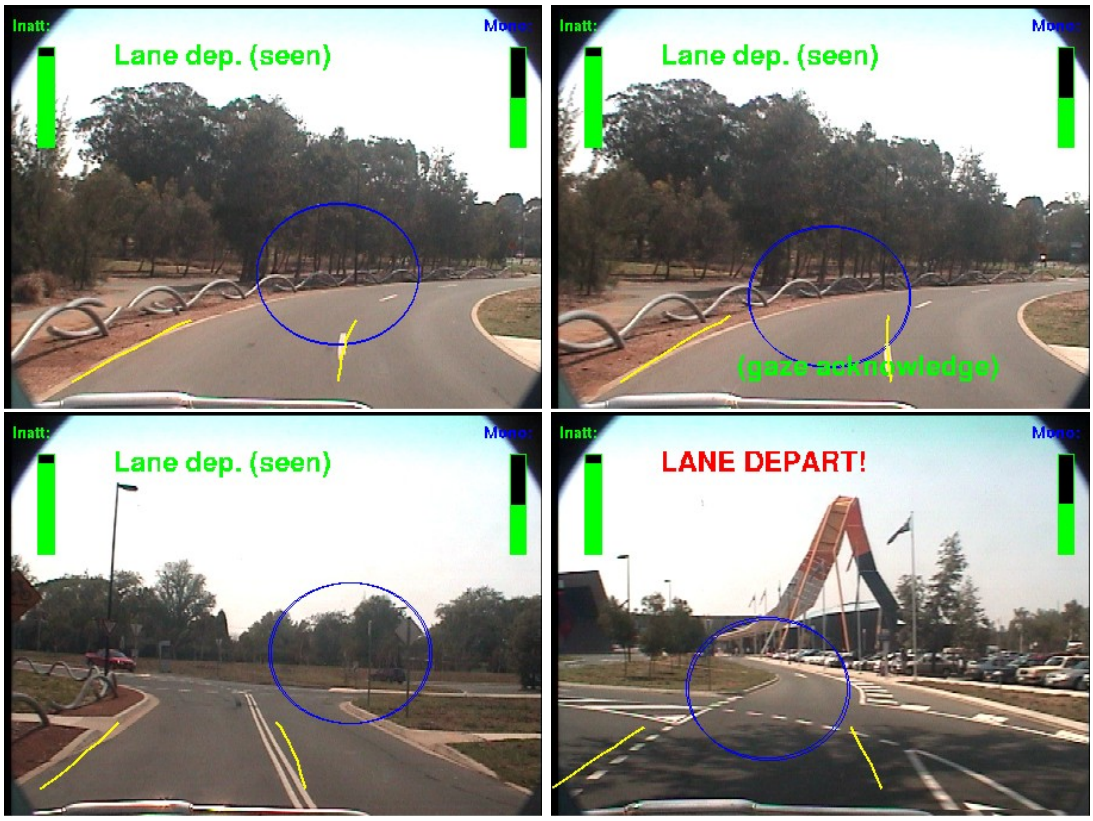
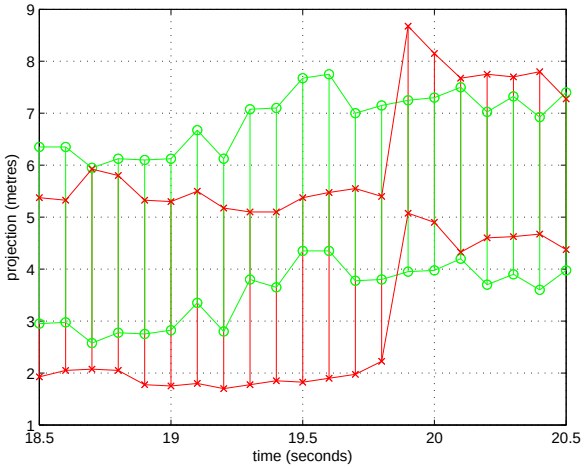
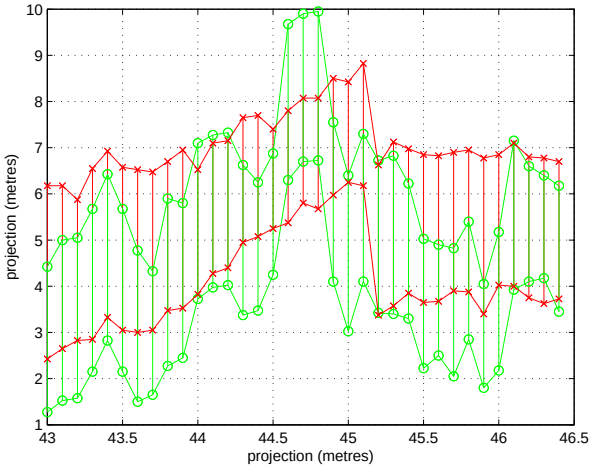


Figure 7.29: Screen-shots of the Automated Co-driver. Lane departure warnings. The final case shows when the driver was turning a corner without indicating or looking in the departure direction. *Large circles:* driver gaze.



(a)



(b)

Figure 7.30: (a): Gaze change at 19.3 seconds then intended lane change at 19.9 seconds. (b): Gaze change at 44.9 seconds then intended lane change at 45.3 seconds. *(Green "o" and lines)*: Sampled gaze error extents projected onto ground plane. *(Red "x" and lines)*: Sampled lane position and width.

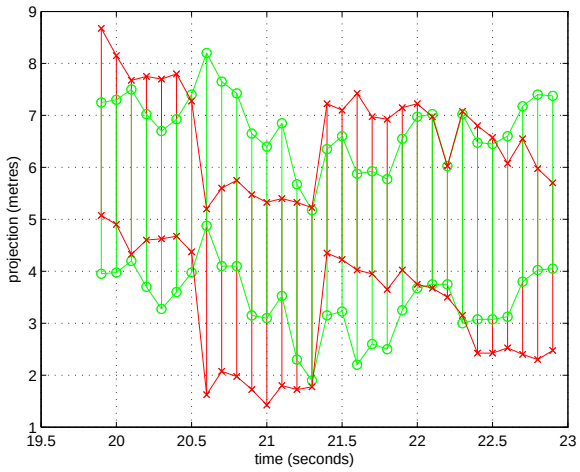


Figure 7.31: Unintended lane change. No descernable gaze change to correlate with lane change at 20.6 seconds. Driver reverts to the original lane at 21.4 seconds. *(Green "o" and lines)*: Sampled gaze error extents projected onto ground plane. *(Red "x" and lines)*: Sampled lane position and width.

Figure 7.32: Sequence of screen-shots of the Automated Co-driver showing a Monotony warning occurring. *Large circles:* driver gaze.

7.6 Summary

This chapter brought together components detailed in the previous chapters to produce on-line real-time context sensitive Advanced Driver Assistance Systems, demonstrating our Automated Co-driver concept. The use of driver gaze as a cue to detecting driver inattention was developed and the feasibility of the approach verified. By not only monitoring the driver's actions, but also the driver's observations, we were able to infer whether the driver was likely to have seen a particular road event. Due to the "look but not see case", we show that road events missed by the driver were particularly well suited for detection by the system. Prototype systems were presented with an examination of unintentional lane departure and unintentional speeding due to a missed road sign.

Chapter 8

Conclusion

This work has assessed the feasibility of an Automated Co-driver as the next significant step in road safety.

We have argued that the key problems remaining in road safety are driver-centred and require a driver-centred intervention. The road-safety hard cases of fatigue and distraction, and their common outcome, driver inattention, can be addressed with a closed-loop system which brings together the strengths of humans and intelligent systems.

Autonomous vehicle research has acheived many impressive engineering feats yet still lacks in-roads into consumer vehicles. Meanwhile much research in driver modeling for road safety had lamented the need for road scene information to supplement driver action monitoring and eye-gaze monitoring to achieve a coherent understanding of the driver behaviour. This work has combined autonomous vehicle technologies with driver eye-gaze monitoring to attempt "driver observation monitoring" to detect potential periods of driver inattention.

8.1 Summary

We began in Chapter 1 by introducing the concept of a second set of eyes for the driver. We proposed that an Automated Co-driver may be able to significantly reduce the risk of road fatalities by integrating the tireless vigilance of machines with the flexibility of the human driver.

In Chapter 2 we stated the case for a Automated Co-driver to be the next significant step in road safety. We examined the problem of road safety and current interventions. We investigated how drivers drive and what has been achieved in vehicle automation. The chapter concluded with set of requirements for a potential Automated Co-driver.

A key issue in the use of computational vision is that although many techniques exist to find and track objects in video footage, these techniques do not operate across the variance encountered in many practical applications. To use these algorithms practically, an adaptable framework is needed which is capable of using the technique when it can and using other methods when a particular approach fails. The "Distillation algorithm" was derived and developed in Chapter 3 as an extension of tracking using "Condensation Isard and Blake (1998)". The Distillation algorithm is an tracking algorithm designed to facilitate the tracking of uncertain objects in a noisy environment using multiple visual cues and a visual-ambiguity tolerant framework. The developed algorithm is demonstrated first in the laboratory for people tracking, then applied to lane tracking. Robust lane tracking was implemented incorporating multiple visual cues and curvature estimation, extending the flexibility of such systems.

Distillation was then used in Chapter 4 for obstacle detection. Combining "top down" and "bottom up" approaches, obstacles were detected and tracked. The Distillation framework allowed a number of issues facing object detection, segmentation and tracking to be handled implicitly instead of explicitly.

Reading signs is an integral part of road-scene understanding and hence understanding the driver. Chapter 5 describes our symbolic sign reading system. Under the common constraint of low resolution imagery, an on-line image enhancement was developed to significantly improve the sign classification reliability.

By Chapter 6 we had developed vision systems for the principal components of the road scene. One aspect, however, needed consideration. A key role of the Automated Co-driver would be to assess the condition of the driver. To assess the condition of the driver, a human co-pilot uses an implicit model of the driver's behaviour. Crucial to this model is an assessment of the environmental conditions: day or night, hot or cold, busy or quiet. To complete the capabilities of an Automated Co-driver, we developed a useful measure of visual variance in the road scene, estimating the monotony of the driving task.

In Chapter 7 we integrated the developed components into a demonstration of an effective Automated Co-driver. The systems integrate driver eye-gaze monitoring with road scene analysis to achieve driver observation monitoring, a crucial skill of any Automated Co-driver. A speed sign assistance system combining vehicle monitoring, speed sign detection and driver gaze monitoring was demonstrated. Then finally, an integration of vehicle monitoring, road object detection, lane tracking, sign recognition, pedestrian detection and visual monotony detection along with driver monitoring are used to demonstrate an Automated Co-driver. Potential threats likely to have been already known to the driver made available to the driver passively. Rising threats likely to have been missed by the driver generated a warning. At any time, concerns were acknowledged with a glance (at the console or the detected threat if still present).

8.2 Achievements

The invention of the first Advanced Driver Assistance System based on the concept of an Automated Co-driver. This work combined automated: road object detection, lane tracking, sign recognition, pedestrian detection and visual monotony detection with driver eye-gaze monitoring to attempt "driver observation monitoring" to detect potential periods of driver inattention. While the "Look but not see" phenonomen precludes the definite assertion that the driver has seen a particular road event. We have shown that it is possible to use gaze direction to determine when the driver is highly likely to have missed a road event. Exploiting this case we have demonstrated some assistance systems able to alert the driver when it is reasonably likely that the driver has missed the event. The backward projection in time using past vehicle egomotion and road geometry can determine if road feature - eye gaze intersection was likely to have occurred prior to the feature's detection.

An original use of driver eye-gaze as an acknowledgement was also demonstrated. Warnings were cancelled by a glance at the dashboard or the threat if still present.

A novel visual cue fusion tracking "Distillation algorithm" was developed, particularly capable of:

- tolerating visual ambiguities by supporting multiple tracking hypotheses.
- combining visual information arriving different rates;
- exploiting synergies between different visual cue processing;
- executing a simple cost benefit analysis between visual cue performance and execution time to dynamically control visual cue execution, and
- incorporating 'late' arriving sensor data.

This work was carried out in conjunction with Gareth Loy and Nicholas Apostoloff while at the Australian National University. The Distillation algorithm was first demonstrated on a people tracking problem.
In my research this algorithm was then extended to robust lane tracking including multiple cameras, road curvature and pitch. The lane tracker was validated on 1800 kilometres of road video consisting of a highway route and a country road route. Tracking was achieved for 96.2% of the trip.
Finally, the Distillation algorithm was applied to a real-time obstacle detection and tracking system again based on visual cue fusion.

A fast and effective real-time road sign reading system was realised. The system based on shape finding and a voting space was found to perform well in a variety of challenging scenarios including partial shadows, varying illumination and faded signs. An online road sign image enhancement algorithm capable of reliably

recognising the sign at a greater distance from the same low resolution image sequence data was presented.

A real-time metric of visual monotony was developed and validated on 1800 kilometres of highway and country road footage. This metric could be used as part of an in-vehicle fatigue monitor but also by road makers as a quantitative measure to assess the fatigue risk due to monotony on a given stretch of road. The augmentation of lane tracking enabled discrimination of cases such as fog and traffic queuing from visually monotonous roads. A similar metric was investigated for road scene complexity measurement.

An extensive hardware and software framework for Driver Assistance Systems was designed, developed and tested. The developed software framework is a cornerstone of the intelligent-vehicle research conducted by the ANU Smart cars project as well as partners National ICT Australia and the CSIRO.

8.3 Further work

The obvious next step for this work would be to conduct clinical trials of an Automated Co-driver ADAS. Due to the experimental nature of our vehicle, test subjects in our experiments had to be students or staff from the Department.

An interesting extension would be a trial where an avatar presents itself as the inbuilt co-driver. The trial could reveal interesting information as to how a driver could come to relate to the assistance. The phenomenon of superimposing a personality on devices such as cars and computers is very common. Once the trial subject has the concept that the car contains an intelligence that is collaborating in the driving task, he or she would hopefully drive accordingly - taking the system interaction in this vein, as opposed to feeling judged or threatened by the system.

The principle failure modes of the vision systems were due to either limited field of view or specularities from direct sunlight. Limited field of view prevented some signs on the far sides of roads to be detected and also prevented the lane tracking from working with very tight curvature encountered on mountainous roads.

The cues used for obstacle detection work well in the near field but still lack sufficient discrimination ability to detect distant oncoming vehicles. Texture based cues such as Haar feature cascades would be a good complementary cue for this case.

The area of road scene complexity assessment is a definite direction for future work. The use of our visual monotony detector by road builders to assess the fatigue risk of roads is an avenue for future exploration. The use of our developed visual-monotony metric with shorter sliding time windows may be sufficient to

Figure 8.1: Prominent bilateral symmetry of signs.

work as a driver-workload handler.

Another approach we hoped to investigate was to exploit a property common to many road scene objects, yet not restricted to one class of object. One such metric is bilateral symmetry. As shown in Figure 8.1, bilateral symmetry is in road signs, road obstacles and pedestrians. An application that estimates the number of bilateral symmetrical objects in the road scene could be effective in judging road-scene complexity and managing driver workload.

There is more to be done to realise the full potential of automated visual clutter detection. The scene variance and bilateral symmetry approaches show great promise. A measure of the complexity of the road environment has a ready application in contributing to future generations of cognitive load management systems, especially in the wake of the mobile technology flood currently afflicting road users.

One intriguing extension to this work that we are interested in following up is the possibility of road-user gaze communication. Driving can often involve making eye contact with other road users. Other drivers, pedestrians and cyclists all use eye contact to ensure a mutual consensus on the order through an intersection or validation of presence. A fascinating extension to this work would be to attempt, first through eye-gaze tracking between drivers and then through high-resolution tracking of pedestrians and cyclists, to track the eye-gaze and there by the intentions of all road users. Simply detecting whether a pedestrian or other driver is looking your way is of enormous benefit to drivers and sounds feasible given sufficient image resolution.

Digital mapping and ad-hoc communications also open up a wealth of possibilities for automoated co-driver systems, not only in the improved performance of tasks we have mentioned, but also new capabilities. Perhaps automated vehicles will

be able to use inter-vehicle communication to replicate the communication link available to human road users through eye-contact.

Appendix A

Test-bed development

The author was the principle developer of the experimental vehicle software and electrical infrastructure. The vehicle mechanical infrastructure was developed in collaboration with the excellent staff (Jason Chen), and students (Luke Cole, Harley Truong and Anthony Oh) of the RSISE Robotics Systems Lab.

An experimental test-bed was developed to fulfil the requirements of the Automated Co-driver that was defined in Chapter 2. The vehicle is a shared resource across the diverse research ANU Smart cars project so some components on board will not be required for our work. Our experiments require road scene, vehicle and driver monitoring. This chapter provides a detailed description of the experimental vehicle and all associated systems in Section A.1. An extensive software infrastructure is also required to support the research work. Section A.2 describes the distributed modular software infrastructure that was developed. The software provides the flexibility to combine system components to create the systems of systems framework that is required for an Automated Co-driver.

A.1 The test vehicle

First we provide a review summary of intelligent vehicles. This is followed by a detailed discussion of our experimental vehicle.

A.1.1 Review of intelligent vehicles

Our design decisions for the test vehicle were shaped by the experience of previous intelligent vehicle projects.

The first generation of intelligent vehicles were by necessity large. The Univer-

(a) 1985: UBM's VaMoRs

(b) 1986: CMU's Navlab1

(c) 1992: CMU's Nablab2

(d) 1993: UBM's VAMP

(e) 1995: CMU's Navlab5

(f) 1995: UParma's ARGO

(g) 2003: INRIA's Cybercab

(h) 2004: CMU's Navlab11

(i) 2005: Stanford's Stanley

Figure A.1: 20 years of intelligent vehicles.

sität der Bundeswehr Münich (UBM) group developed the "VaMoRs" (translated as:"Experimental Vehicle for Autonomous Mobility and Computer Vision"), a five-ton Mercedes D508 van (Figure A.1(a)) equipped with banks of transputers, the only cost effective processors able at that time to support real-time computer vision (Dickmanns and Graefe, 1988b,a). Project partner Daimler-Benz developed a similar vehicle known as VITA (VIsion Technology Application). The University of Parma (UParma) developed the MOB-LAB (mobile laboratory)(Bertozzi and Broggi, 1998). The Fraunhofer- Institut für Informationsund Datenverarbeitung (IITB) system in collaboration with the Robert-Bosch auto part manufacturer developed a van (Fhg-Cop Van). In Carnegie Mellon University(CMU)'s Navlab the first few experimental vehicles were vans (Thorpe, 1990a). In 1995 CMU's "Navlab 5", which crossed America, was a Pontiac transport (Figure A.1(e))(Pomerleau, 1995). "Navlab 11" is a robot Jeep Wrangler (Figure A.1(e)) equipped with a wide variety of sensors for short-range and mid range obstacle detection (Sun *et al.*, 2006). The goal of autonomy loomed large. The vehicles were fitted with actuators to control steering. Some, like those in

the UBM group, also had throttle and brake control.

With keen interest from car makers and advances in computing technology, many groups then developed complementary "light weight" versions of their vehicles. UBM produced the "VAMP" (VaMoRs Mercedes Passenger vehicle) was a Mercedes S-class sedan (Figure A.1(d)) mirrored by the VITA II by Daimler-Benz. IITB and Bosch produced the Fhg-Cop car (sedan). UParma ported their systems to the ARGO a Lancia Thema passenger car (Figure A.1(f)) with actuated steering (Bertozzi and Broggi, 1998).

All groups used video cameras as their primary sense. The primary goal of these systems was lane tracking and many groups fitted cameras behind the windshield around the position of the central review mirror. UBM developed the "MarV-Eye" Multi-focal, active/reactive Vehicle Eye system which consisted of an active camera platform to stabilise the video images. The camera configuration is illustrated in Figure A.2. Multiple focal length cameras with overlapping fields of view were used to get acceptable resolutions in the far field as well as directly in front of the vehicle (Dickmanns, 1999b). Both UParma ARGO and CMU-Navlab used fixed wide baseline cameras to achieve high accuracy multi-view reconstruction for obstacle detection (Williamson and Thorpe, 1999). With high resolution cameras prohibitively expensive, the "VaMoRs" and "Fhg-Cop" vehicles used several diverged cameras for a large near field of view (Dickmanns, 1999b; Enkelmann, 2001). The Daimler-Chrysler VITA-II was fitted with 18 CCD cameras for omni-directional monitoring.

A different approach was taken by INRIA's CyberCars project. This project concentrated on autonomy in smaller vehicles (Figure A.1(g)). The focus of this group was areas such as city centres where road traffic shares the roads with pedestrian traffic. This group argues that clogged cities need to move away from conventional vehicles to fleets of public electric vehicles. The vehicles make use of the emerging autonomous technologies to build in an underlying framework of safety systems (Parent, 2004).

Given the success of Lidar systems in mobile robotics in the early 1990s, lidar systems quickly found their way on to vehicles as near-field obstacle detectors. The working range of the sensor of less than 80 metres - but often in practice around 25 metres - limited their applications (Thrun *et al.*, 2006). With longer ranges, microwave and millimetre wave radar have been used for obstacle detection and car following on various vehicles. Until recently these systems have either been only capable of generating range measurements in a single direction or were very expensive. Millimetre wave radar has huge potential in the automotive field as it has been shown to be tolerant of heavy snow, fog and rain, conditions which can blind visual systems (Brooker and Durrant-White, 2001). As both lidar and radar systems are active sensors (i.e. they project then measure the reflection of a beam) they need some mechanism to prevent cross talk from adjacent and oncoming vehicles (Roberts and Corke, 2000; Brooker and Durrant-White, 2001).

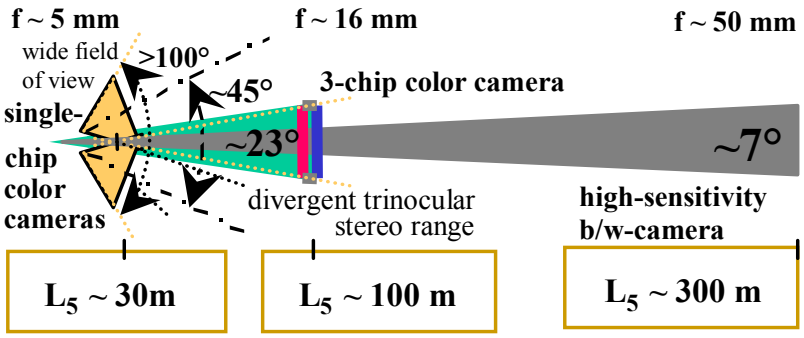


Figure A.2: Universität der Bundeswehr Münich (UBM) group camera arrangement (Dickmanns, 1999b).

In recent years the DARPA Grand Challenge has spawned a new generation of intelligent vehicles. These vehicles have much in common with the first generation of vehicles. As the end goal is the automation of defence vehicles, a requirement is that the vehicles are over two tonne. Most of the entrants were 4-wheel drive vehicles (DARPA, 2005). Actuators are fitted to steering, throttle and brake mechanisms. Additional functions are automated since these vehicles are configured to work as robots. Tasks include working from a standing start, reverse manoeuvring and moving to an arbitrary location (i.e. functionality such as gear selection and starting and stopping the engine are intelligent as well). As described in Chapter 2, since new technology is embedded in production vehicles today, most teams interfaced directly to the vehicle CAN bus to obtain speed and instrumentation data from existing devices in the vehicle. Due to the competitive need to have the best possible sensing, instead of an economic imperative, the vehicles were packed with redundant sensing devices. Many vehicles contain several lidar scanners, multiple GPSs as well as multiple video cameras. The DARPA 2005 Grand Challenge winner Stanford Univerity's "Stanley" (Figure A.1(i)) was a Volkswagen Touareg R5 fitted with five lidar scanners, a colour video camera and two millimetre wave radars (Thrun *et al.*, 2006).

Early in real-time image processing research, since image processing was too computationally intense for digital computers, some research teams implemented operations such as image derivatives using analog filters. Out of these approaches in the late 1980s and early 1990s, several custom image processing hardware solutions were developed using transputers and custom combinations of digital signal processors (DSPs) to implement real-time image edge detection. During the mid-1990s generic processors such as the Intel 386 began to outperform these custom built systems. Because of their generic nature, these software solutions allowed much greater variations in the implemented algorithm. Several years later a second wave of image processing hardware arrived in the form of image correlation cards, and more custom devices like "Datacubes" which started a swing back

to dedicated hardware solutions. At the close of the 1990s generic processors struck back with pipelined and small scale parallel single instruction multiple data (SIMD) instructions (eg. MMX SSE 3DNow instruction sets). Again the "software solution" offered more flexibility in algorithms. CMU's "Stanley" featured six Pentium M computers (Thrun *et al.*, 2006). Currently there is a swing toward dedicated hardware in the form of field programmable gate arrays (FPGAs) and the use of graphical processing units (GPUs). Graphical processing units are the core of video display devices used in personal computers. They are devices designed to achieve the massively parallel and fast limited accuracy computations needed for 3D graphics. The demand for ever greater visual effects has driven these devices to incredible capabilities (Yang *et al.*, 2004). These technologies are, however, likely to find their way into generic processors in future years. FPGAs are a new breed in the hardware vs software divide. Field programmable gate arrays are reprogrammable hardware devices. Their potential lies between a fixed hardware and a marginally parallel software solution. FPGAs offer an attractive path for embedded vision hardware in vehicles. Although we will develop our computer vision and advanced driver assistance systems algorithms with future FPGA embedded solutions in mind, we will not use this technology in the research phase of this project.

A.1.2 TREV:Transport Research Experimental Vehicle

In the scope of the survey discussed above, the vehicle used in our research is in generation X: after the first wave of intelligent vehicles and before the second wave (DARPA Challenge vehicles).

The vehicle development commenced in late 1999 to be a generic testbed for research on-road or off-road and for autonomous driving or for Advanced Driver Assistance Systems. At this time technologies such as CAN bus interfaces were unavailable. Also, unlike many of the vehicles developed above, no technical companies were offering services to implement functionality such as steering control or sensor interfacing. As a result, all work on the vehicle was done within the project with the exception of some wiring which was done by a local automotive electrician.

The objective in developing a experimental platform for on/off-road vehicle research was to have a robust, adaptable solution. It needed to be relatively easy to use and also flexible enough to cater for the unknown needs of current and future researchers. The methodology used was to develop a platform with a true system of systems approach. Wherever possible commercially available subsystems were used to leverage off the robustness and reliability of the product. Commonly used standards were used (even, at times, at the expense of efficiency) to support ease of use and reconfigurability. Such standards include ethernet based communications between devices, mains (240Volt) based power for computers and standard desktop computers for processing. Though these standards, such as the provi-

Figure A.3: TREV: Transport Research Experimental Vehicle.

sion of a reliable mains power supply, took significant effort, the rewards are now reaped in that experimental PCs can be replaced easily.

The Transport Research Experimental Vehicle (TREV) is a 1999 Toyota Land Cruiser 4WD (Figure A.3). A four wheel drive vehicle was chosen for the provision of:

- A strong and robust platform capable of surviving the rigours of experimentation.
- A large amount of interior and exterior space for installing sensors/computers.
- Support a large scope of research topics including off-road autonomous driving.

The base vehicle has an auxiliary battery and 1100 Watt inverter to provide a reliable mains supply. The vehicle has been augmented with sensing equipment, vehicle control devices and processing hardware.

A.1.3 Sensing

There are three sensor application domains considered, namely: road scene sensing, vehicle-state monitoring and driver monitoring.

Road scene monitoring involves analysing the environment surrounding the vehicle, the location and state of lanes as well as potential obstacles. The vehicle is fitted with a SICK LMS-221 laser range finder (lidar) mounted on the front of the vehicle. This sensor measures a horizontal 180 degree slice of depths to obstacles in front of the vehicle. As noted earlier, due to back scatter degrading the signal

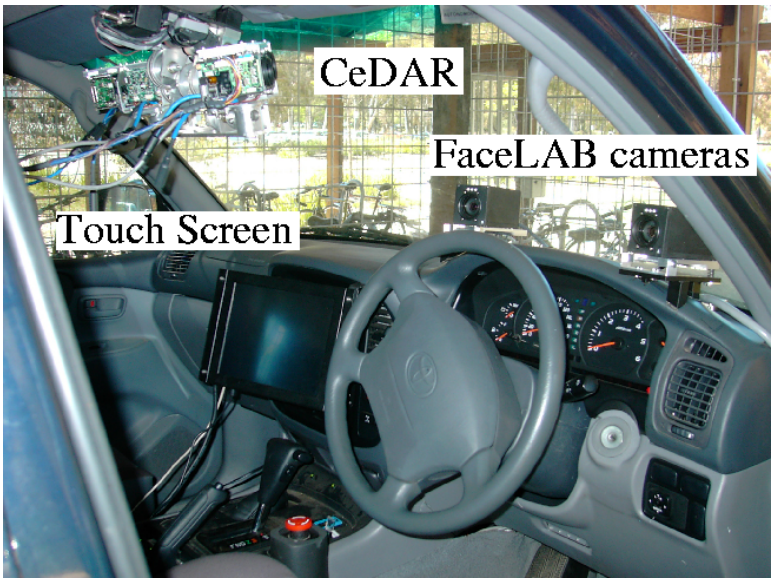


Figure A.4: The vision platforms in the vehicle. The CeDAR active vision head in place of a central rear vision mirror and faceLAB^{TM} passive stereo cameras on the dashboard facing the driver.

to noise ratio, the effective range of the lidar is around 25 metres (SICK AG, 2003). A millimetre wave radar is fitted to the roof of the vehicle and measures a point range in front of the vehicle (Mihajlovski, 2002). Neither of these sensors are used in our experiments as we are interested in using senses similar to the driver's sense of the road environment.

The vision system is a CeDAR camera platform (see Figure A.4) and is mounted at the top centre of the windscreen. The CeDAR camera platform provides a tilt and vergence mechanism for two pairs of cameras providing a level of responsiveness in the order of the human vision system (Sutherland *et al.*, 2000; Dankers and Zelinksy, 2003). The camera platform provides a field of view and vantage point similar to that of the driver. A more detailed description of the camera platform and how it is used is included in Section A.1.4.

In addition, a panoramic camera is available for near field blind spot monitoring, though not used in our experiments.

The installed sensors consist of:

- A 4 pulse per revolution tail shaft encoder providing an estimate of the vehicle speed.
- A steering angle potentiometer providing an absolute estimate of the front wheel angle.
- An encoder on the steering actuator which gives a relative estimate of the steering angle.

- A six degree of freedom DMF-FOG inertial navigation sensor (INS) mounted near the centre of gravity of the vehicle, which provides an estimate of the linear and angular acceleration of the vehicle.
- A Trimble AG132 GPS with differential correction from the Australian Maritime Safety Authority coastal beacon.

The above sensors combined with an Ackermann steering model and recursive estimator track the vehicle state between video frames. The Ackermann steering model, illustrated in Figure A.5, models the steering geometry of a road vehicle. The model is an approximation of road vehicle motion over small distances and low curvatures. The model assumes a constant steering angle, all four wheels trace out circular paths around the instantaneous centre of curvature (ICC).

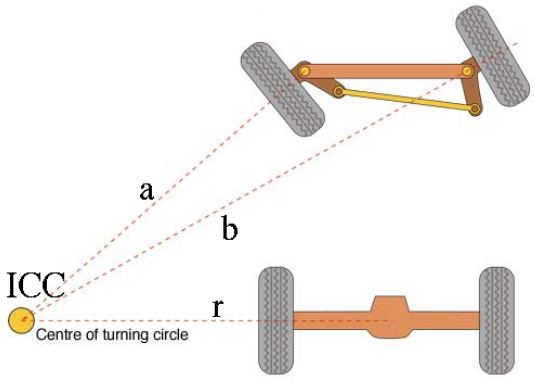


Figure A.5: Ackermann steering model.

The driver state is estimated from:

- A brake light monitor.
- The gear selection switch.
- Turning indicator switches.
- Horn and emergency stop switch.
- Strain gauges on steering shaft to measure torque applied by the driver.
- Touch screen interface.
- Driver head pose and eye gaze tracking system.

The sensors provide information on the controlling actions of the driver in addition to monitoring the driver's behaviour.

A.1.4 CeDAR: the CablE Drive Active vision Robot

CeDAR is a 3 DOF active camera platform arranged in a Helmholtz configuration (cameras share a common tilt axis). Figure A.6 shows the design of the system.

Figure A.6: CeDAR: Cable Drive Active vision Robot in vehicle. Four cameras are arranged in two stereo pairs. Cameras are fitted with polarising filters and far field cameras use glare hoods.

In the experimental vehicle CeDAR was mounted upside down in place of the centre rear vision mirror. The active-camera platform can be used for intersection spotting or vehicle-pitch compensation, but in our experiments the cameras are fixed. Pitch compensation is achieved in software. Instead of a single camera, two cameras are located on the left and right sides of the platform. The cameras were configured as two stereo pairs with two different focal lengths, providing sufficient resolution in the near and far fields. The near field cameras have a focal length of 5.7mm and field of view of 48°. The far field cameras have a focal length of 11.3mm and a field of view of 29°. All cameras are Sony FCB-EX470L analog NTSC, single 1/4 inch CCD video cameras. The cameras are designed as OEM solutions for handi-cams and feature good automatic outdoor exposure correction, electronic zoom, automatic focus and zero lux response (night vision). The cameras are fitted with polarising filters to cut reflections from the windshield and glare from the road.

Calibration

The coordinate system of the active camera platform used is shown in Figure A.7(a). Each camera was calibrated to find the internal camera parameters using Zhang's algorithm (Zhang, 1999) with the Matlab^{TM}camera calibration toolbox (Bouget, 2002). There are no degrees of freedom between the cameras on the same side of the camera platform. The external camera parameters are calibrated between these cameras, using point correspondences and the predefined internal camera matrices. The external relationship between the left and right camera pairs is a function of the left and right vergence angles on the head. Similarly, the relationship between the cameras and the base of the platform is a function of the tilt angle. All these relationships can be written as transformations in terms of these angles.

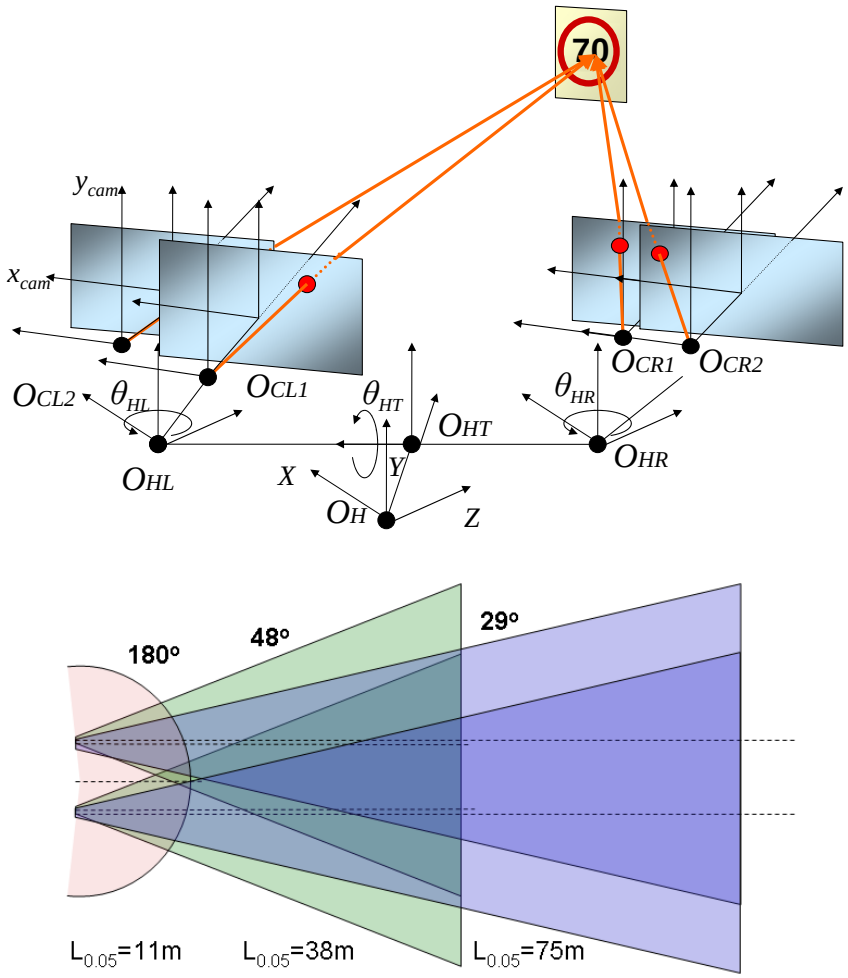


Figure A.7: Road scene camera configuration for 4 camera active camera platform. (a) Camera coordinate system for the: O_h-head, O_{ht}-head tilt, $O_{cl1}O_{cl2}$-camera left and $O_{cr1}O_{cr2}$-camera right. (b) Camera field of view.

Image rectification is performed on each image to remove lens distortion and camera mounting deviations from the idealised geometry. Figure A.8 shows the result of such a rectification. Rectification is done using bilinear interpolation using a lookup table computed off-line. Again the Matlab^{TM}camera calibration toolbox (Bouget, 2002) is used with some modification to generate the lookup tables for each camera.

Below are the intrinsic parameters for each of the wide field cameras and the extrinsic parameters between the cameras. These are transformed into the CeDAR head tilt coordinate frame.

```
Calibration results for left camera (with uncertainties):

Focal Length:          fc = [ 739.14758   363.22382 ] + [ 9.66077   4.69778 ]
Principal point:       cc = [ 372.20516   125.81629 ] + [ 5.34851   3.65983 ]
```

Figure A.8: Left wide field camera before and after image rectification.

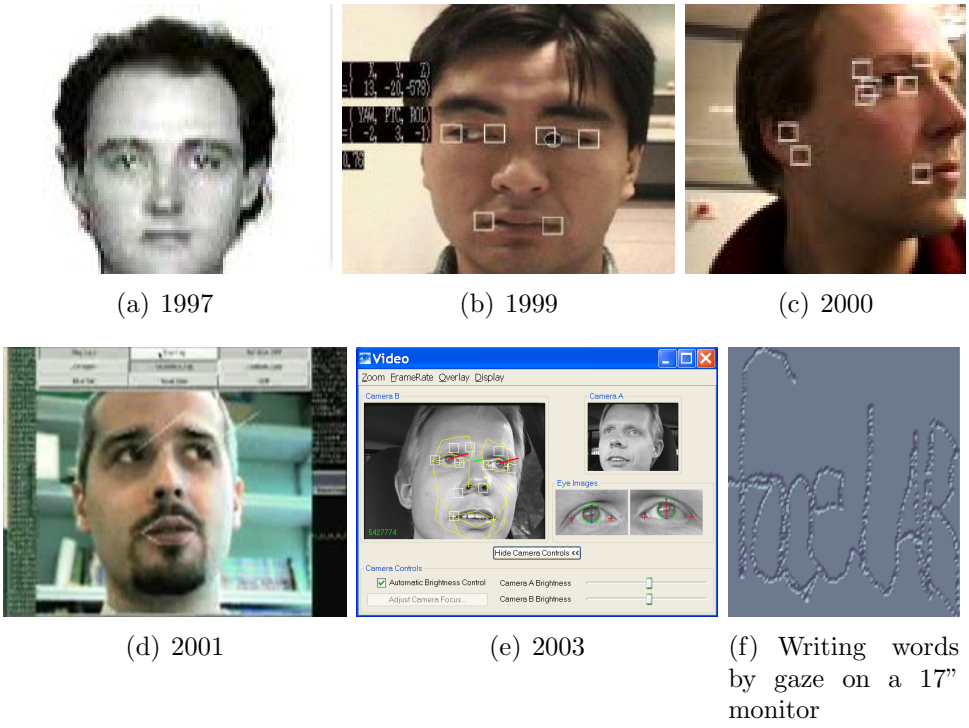
```
Skew:             alpha_c = [ 0.00000 ] + [ 0.00000  ]   => angle of pixel axes = 90.00000 + 0.00000 degrees
Distortion:            kc = [ 0.00000   0.00000   0.00000   0.00000  0.00000 ] + [ 0.00000   0.00000   0.00000
Pixel error:          err = [ 0.35863   0.28399 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).


Calibration results for right camera (with uncertainties):

Focal Length:          fc = [ 707.99615   349.56201 ] + [ 11.07373   5.35484 ]
Principal point:       cc = [ 319.50000   119.50000 ] + [ 0.00000   0.00000 ]
Skew:             alpha_c = [ 0.00000 ] + [ 0.00000  ]   => angle of pixel axes = 90.00000 + 0.00000 degrees
Distortion:            kc = [ -0.40457   0.00000   0.00454   -0.01697  0.00000 ] + [ 0.06579   0.00000   0.00411
Pixel error:          err = [ 0.35645   0.25338 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).

Extrinsic parameters (position of right camera wrt left camera):

Rotation vector:             om = [ -0.00324   0.15064  0.01598 ]
Translation vector:           T = [ -314.49544   2.63615  48.71482 ]
```

A.1.5 Driver face and eye-gaze tracking system

To localise and track the driver's head pose and gaze we use a computer vision system that was developed by our laboratory at the Australian National University and is now commercialised by Seeing Machines (2001). Figure A.9 chronicles the evolution of the ANU head and gaze tracking research. SeeingMachines' faceLAB^{TM} system uses a passive stereo pair of cameras mounted on the dashboard to capture video images of the driver's head (see Figure A.4). The video images are processed by a standard PC in real-time using image correlation and edge detection techniques to determine the 3D pose of the person's face ($\pm$1mm, $\pm1^{\circ}$) as well as the eye gaze direction ($\pm1^{\circ}$), blink rates and eye closure. The system uses a 3D head model and a physically realistic geometric model for the eye to achieve an impressive level of precision. The system can either log the data to a file or stream the data to a UDP socket.

A.1.6 Actuation

The vehicle is fitted with three actuators to control the vehicle: steering, braking, and throttle. For completeness we outline these systems, though in our experi-



(a) 1997 (b) 1999 (c) 2000

(d) 2001 (e) 2003 (f) Writing words by gaze on a 17" monitor

Figure A.9: Evolution of ANU gaze tracking research.

ments we do not use active actuators. As discussed in Chapter 2 the University is unwilling to insure the actuated control of the vehicle on public roads. The roads where all of our experiments are conducted. We are instead interested in vehicle control through the manipulation of the driver.

Throttle control was implemented by interfacing with an after-market cruise control fitted to the vehicle. The steering sub-system is based around a Raytheon rotary drive motor/clutch unit. The unit was designed for use in yachts to steer the rudder according to a given bearing. The actuator was installed in the engine bay alongside the steering shaft. The Raytheon drive unit was fitted with an encoder for improved closed loop control. A three-spur gear linkage was manufactured to connect the Raytheon actuator to the steering shaft: the first gear is attached to the steering shaft, the second to the motor shaft, and the third, an idler gear, sits between the first two. A key feature in the design is that the idler gear can be engaged and disengaged from the drive-train assembly. For "manual" driving the idler gear can be disengaged, providing the safeguard that the autonomous steering assembly cannot impede normal steering. The steering sub-system is shown in Figure A.10.

The braking actuator is based around a linear drive unit (produced by *Animatics*), and an electromagnet. The linear drive is a "smart" actuator controlled via a RS-232 serial interface. The linear drive is connected to one end of a braided steel cable via the electromagnet and base plate. The cable passes through a

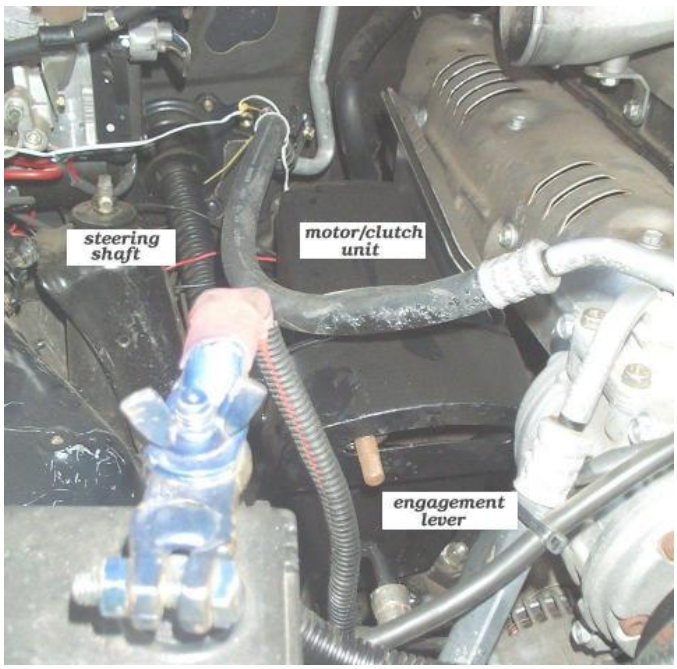


Figure A.10: Steering mechanism of vehicle containing drive motor, gears and clutch. The engagement lever moves an idler gear into position between a gear fitted to the steering column and the motor clutch.

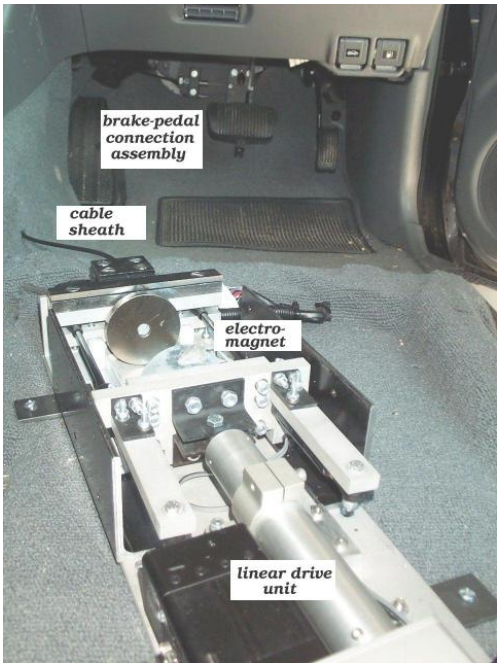


Figure A.11: Braking device showing linear actuator, electromagnet & cable mechanism.

guiding sheath to reach the brake pedal. The electromagnet is a safety feature of the mechanism. To apply the brakes, the electromagnet must be energised to pull on the cable. In an emergency situation, power is cut to the electromagnet and compressed gas struts ensure the cable goes slack immediately, returning all braking control back to the safety driver. Our implementation is designed with the assumption that there will always be a safety driver ready to take over control if necessary. The braking system is shown in Figure A.11. In the foreground the figure shows the linear drive and electromagnet, while in the background the brake pedal and its connection with the cable are shown.

A.1.7 Safety

The driver's seat is fitted with a 4-point harness in addition to the standard seat belt (see Figure A.12). The purpose of this harness is to restrict the driver's body so that the driver's head is unable to collide with the CeDAR head or faceLAB^{TM} cameras (shown in Figure A.4) in the event of an accident.

Figure A.12: Driver wearing 4-point safety harness. Rear seat belts fully extended and crossed for use as top harness straps.

A.1.8 Data processing hardware

Data processing is done using a standard PC architecture (see Figure A.13). Such hardware is readily available, easily replaced and inexpensive. Figure A.14 is a diagram showing how the computers in the test vehicle are configured. One PC is fitted with a Servo2go interface card (ServoToGo inc., 2000) and runs the servo loop to control the actuators and read analog and digital IO. A second PC supports the faceLAB^{TM} system. A further two PCs contain video grabbers and perform the necessary road scene image processing. All PCs with the exception of the faceLAB^{TM} system run Linux. The computers are networked with multiple ethernet adapters (to increase bandwidth) and a gigabit ethernet switch. One

Figure A.13: Back of vehicle featuring desktop computers, networking, relay and breakout racks.

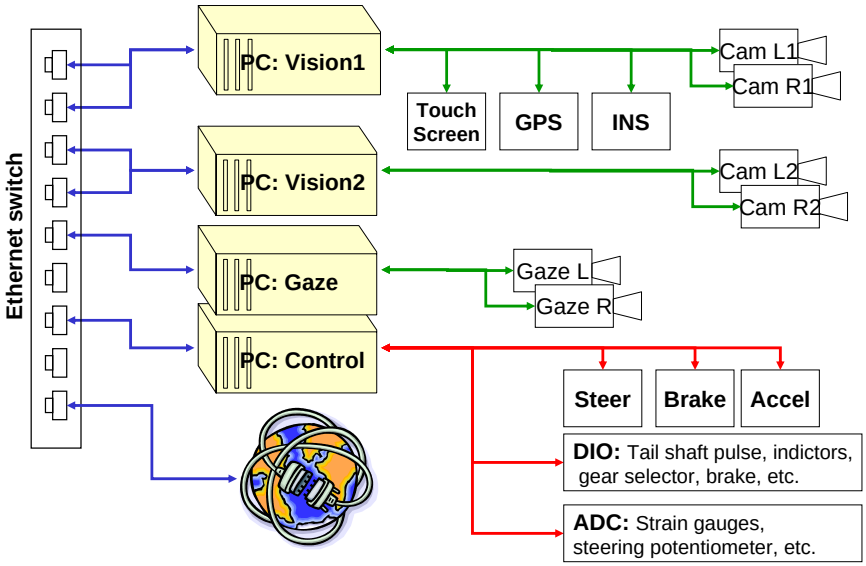


Figure A.14: Computer configuration in the test vehicle. Including a link to the outside world.)

PC contains a RAID 0 disk array (for high thoughput) with the home filesystem which is exported to the other pcs.

A.2 Software framework

In a vehicle fitted with multiple sensors, PCs and actuators, it is the software framework that effects the transformation into an integrated intelligent vehicle. The requirements of the software framework are:

- C/C++ implementation
- distributed execution
- reconfigurability
- reliability
- scalability
- modularity
- availability

We begin with a brief review of relevant architectures followed by a description of the developed software framework.

A.2.1 Review of software architectures

The Player/Stage robot interface and simulation system (Gerkey *et al.*, 2001) has achieved huge popularity due to its user-friendly design and relatively mature source base. The less mature Dave's Robotic Operating System (DROS)(Austin *et al.*, 2001) is free software for robot control. The software is light weight and fast, but not yet complete.

The main issue with these architectures is that the modelling does not quite fit with our application, i.e., mobile robot architectures have been developed for autonomous robots particularly focused on map building, navigation and localisation. We need an architecture for advanced driver assistance and for real-time computer vision and applications across machines and processors.

Recently, with the launch of the European Union AIDE project (European Union, 2004), efforts to understand, optimise and standardise human machine interface (HMI), for cars have been undertaken (Amditis *et al.*, 2006). Kun *et al.* (2004) outlined an architecture for driver support using speech interaction.

Figure A.15 shows the architecture used for UBM's VaMORs vehicle. The architecture is a combination of transputers (PP) and standard processors. Fast camera stabilisation was done with an analogue feed back loop from an INS sensor (Dickmanns and Graefe, 1988a).

Figure A.16 shows the architecture developed by the Stanford DARPA Grand Challenge team. The architecture is based on the three-layered approach of (Gat,

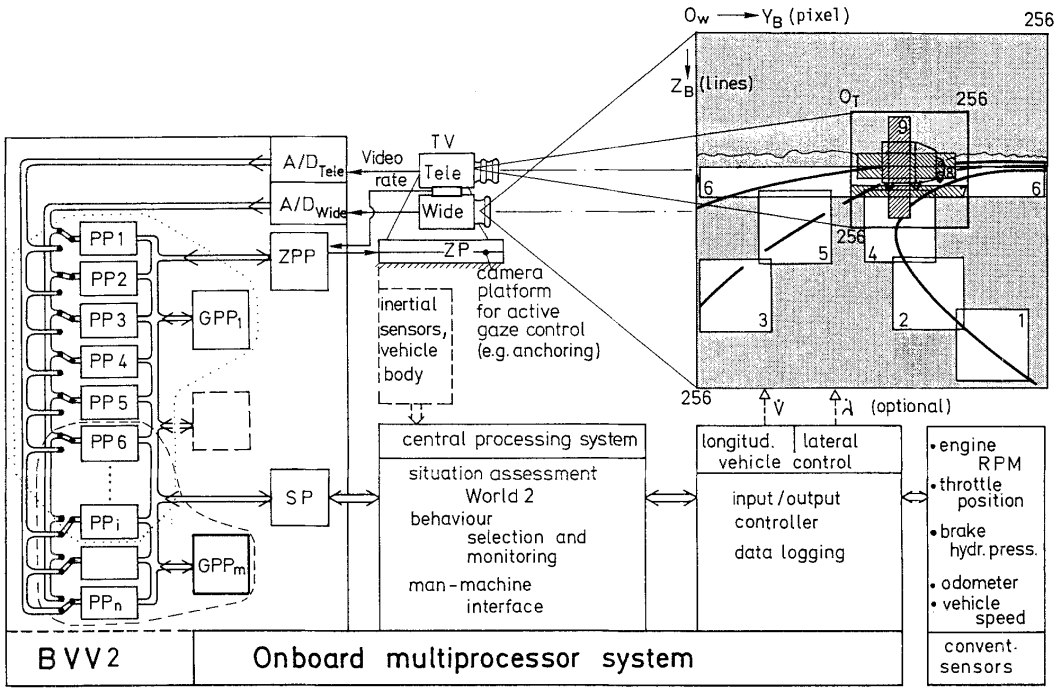


Figure A.15: Software architecture model for UBM's "VaMORs" vehicle. From (Dickmanns and Graefe, 1988a)

1998) and uses CMU's inter process communication (IPC) toolkit to implement a client server architecture. The group developed an architecture loosely grouped into Sensor, Perception, Planning and Control, and Action components with a few "Global services" components providing supporting functions (Montemerlo *et al.*, 2006). The use of a client server IPC and modular construction is parallel to our requirements. For our research not one of these architectures was available to build upon. When tallying the benefits of using these architectures against the list of modifications required to match our paradigm, together with the constraint of not being independent of the architecture, there appears to be no good choice.

Instead of using an existing architecture, we decided to use light weight class libraries to implement core functionalities. We then wrote libraries and application software using these basic classes. In a thorough investigation into the design of software frameworks for complex tasks, Petersson (2002) concluded that a layered approach to complex-system design offered the best outcome. The use of levels of abstraction from the raw device interfaces to the ultimate task provides a division that often mirrors the intuitive modularity of the complex system. The layered structure and explicit interfaces lead to a number of efficiencies. First, the approach leads to a natural development and testing schedule. Second, the validation of algorithms even in higher layers can be done in isolation from the complexities of the rest of the system. Even across the whole system, debugging can quickly be isolated to a particular layer and module. Finally, the layering

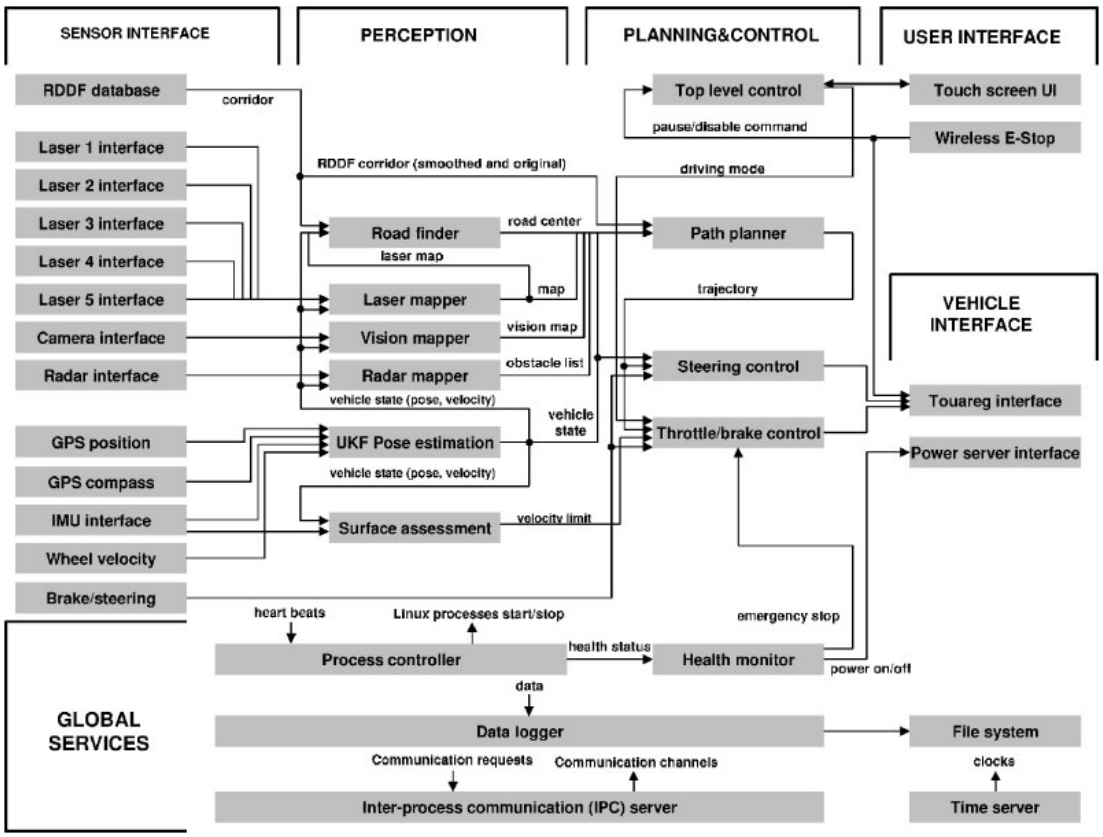


Figure A.16: Software architecture model for 2005 DARPA Grand Challenge winner Stanley. From (Thrun *et al.*, 2006)

segments algorithms of different levels of abstraction. This means that the decomposition and reuse of modules between systems is by design effective. Figure A.17 shows the user model approach to a complex system architecture defines layers of abstraction. In our case the "End user" is the driver. The "Application programmer" is the Experimenter with a particular Advanced Driver Assistance System implementation. The "Module programmer" develops the underlying base class libraries based on generalised concepts relevant to the application. The "Interface programmer" develops class interfaces based on abstracted models of the devices. The "Hardware designer" develops the capability to interact with the physical system.

A.2.2 Software implementation

An Advanced Driver Assistance System can be thought of as a system of systems. Figure A.18 shows the principal components of our Advanced Driver Assistance System.

As our experimental vehicle is used for a variety of driver assistance systems,

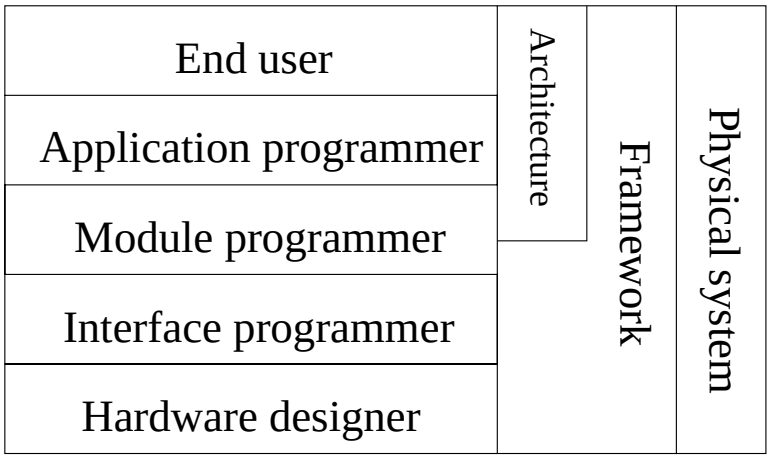


Figure A.17: User model driven approach to complex system architecture. From (Petersson, 2002)

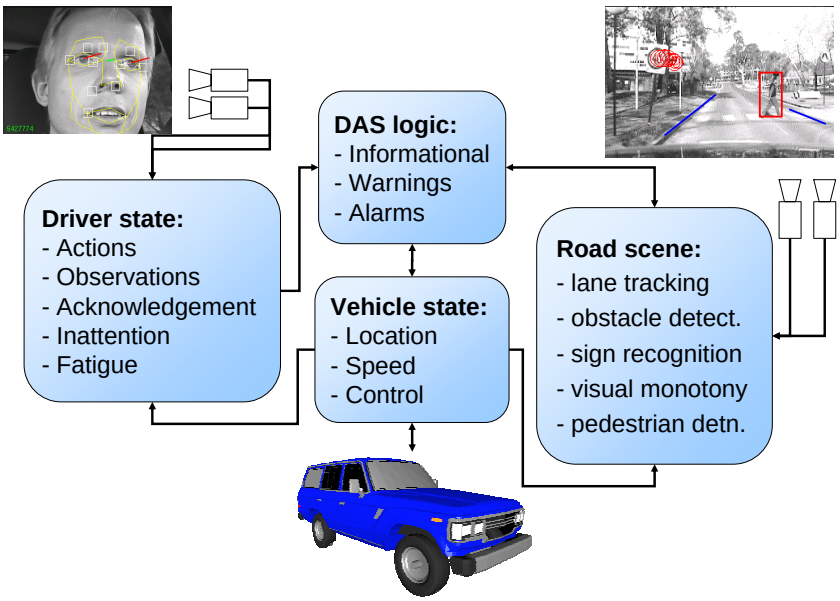


Figure A.18: Functional components of the Driver Assistive System.

a great deal of flexibility is desired in terms of which devices are to be used and how. To support this flexibility and to make the complexity of the system tractable, we opted for a common object request broker architecture (CORBA) based inter-process communication (IPC) system (OMG, 2002). This provides an implicit client-server style modularity across the system. For example the steering actuator can be controlled by the vision PC by accessing a remote steering object provided by the primary PC. The fact that the steering object is really implemented as a process on a remote host does not affect the application. Ap-

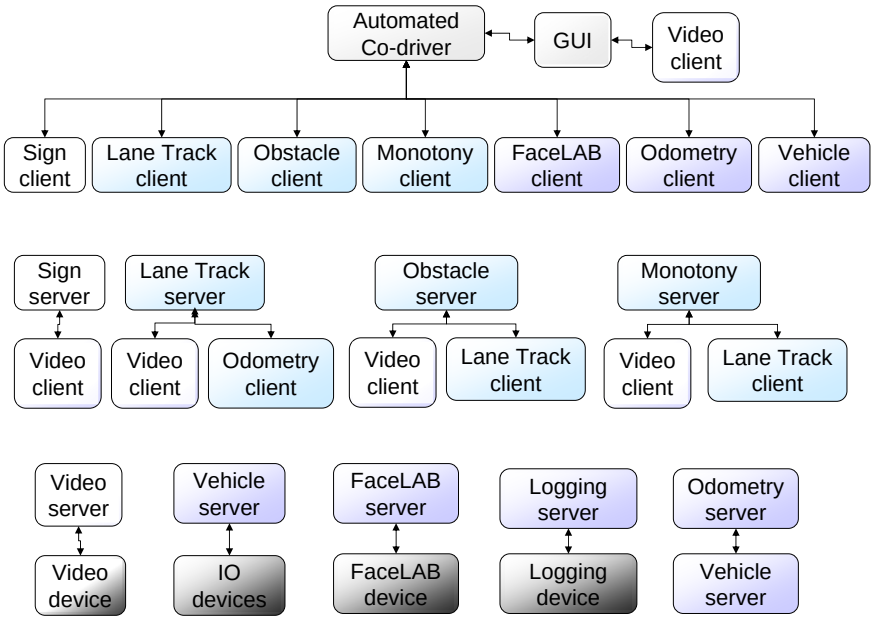


Figure A.19: Typical client-server software configuration for the co-pilot application. Each connected set of objects represent an executable program. Clients and servers can run on different machines.

plications can also provide their own CORBA remote object interfaces allowing a clear separation between the algorithms, application and the GUI. CORBA has the benefit over traditional IPC techniques of being inherently object oriented, platform independent and implementation independent. Hence several operating systems and programming languages can be used together in concert. Through the use of an interface definition language (IDL), CORBA can ensure the underlying clients and servers interpret data correctly without the programmer having to consider how communication is achieved. CORBA also incorporates name service so objects can be referred to by meaningful names instead of, for example, host and port numbers. Although Wulf *et al.* (2003) dismisses CORBA and several robot architectures due to their lack of the hard real-time guarantees, with sensible implementations (such as no CORBA objects in control loops) and sensible low-level watchdog functions, the flexibility of these frameworks can be used safely and reliably.

Figure A.19 illustrates a selection of CORBA objects in the software used in the co-pilot experiments. For any server there may be zero or more clients running on the local or remote PCs. Another feature of the software is that CORBA servers and clients can be stopped and restarted without affecting the other half. For example the video server can be stopped from grabbing live video images and restarted from a saved video-stream. Depending on the needs of the client, the client will either wait for the video server to restart or report back to the application that the video server was unavailable.

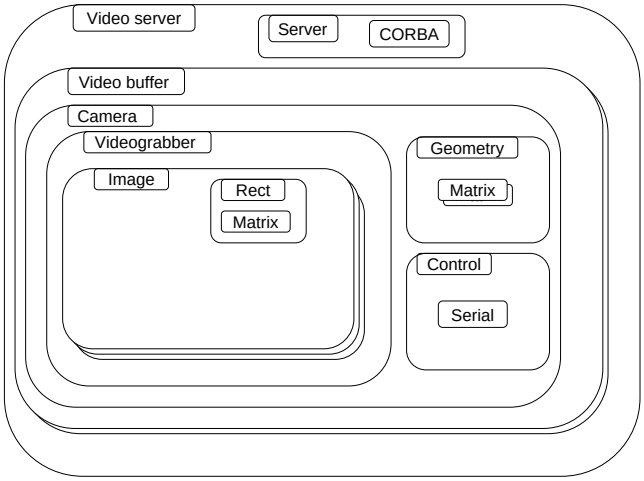


Figure A.20: Class composition of the video server component.

Shell scripts are used to start and stop servers as required for different experiments. Creating different driver assistance applications simply requires a different "DAS logic" module.

The design of the underlying software framework for the vehicle has purposely been kept flexible, fast and modular. The software has withstood the test of time as hardware components and research agendas have changed. The software's modularity and flexibility has led to uses never envisaged, such as distributed training of learning algorithms.

The source code for the basic shared class libraries used is included on the Appendix DVD-ROM (Page 257). Source for video-server shown above is included. All header files are present some of the source files have been deleted to avoid potential copyright/IP infringement of previous students.

A.2.3 The technicalities

A simple image class and matrix class are the basis of the software libraries. The image class is an evolution of previous computer vision programming. The basis of the matrix class is the Newmat library (Davies, 1994), an open source matrix class library which incorporates some conveniences like matrix singular value decomposition while still efficient for small matrices. Another class provides a multi-threading, semaphore and mutex support. Figure A.20 illustrates some core classes for the software. The algorithms and applications described in the rest of this document all use these shared classes and CORBA architecture.

The Trolltech QT library (Trolltech, 2006) and Xwindows XVideo extension is used for graphical user interfaces. XVideo particularly allows full rate video to be displayed with no computational load.

3D modelling is implemented using the open source version of SGI's Open Inventor library Coin3D(Coin3D, 2006). The Open Inventor library translates the specified scene into an openGL scene graph which executes on the graphics card GPU again causing no significant processing load.

Some computationally intense functions (such as stereo disparity maps, image correlation and optical flow) have been optimised in functions using Intel SIMD instructions (MMX, SSE and SSE2).

A.3 Summary

A significant effort has been invested in the development of a suitable test vehicle. Both hardware and software systems were developed and evolved to enable a flexible modular framework for intelligent vehicle research. Off the shelf components were used where possible, but necessity dictated a few custom solutions.

The use of a modular software architecture has proved its worth time and again when assembling different applications, debugging and even pursuing new machine learning (training based) research directions. Based on the CORBA framework, the software provides a natural structure for distributed client server applications.

In the next chapter we introduce our computer vision tracking algorithm. The algorithm was developed to design robustness into computer vision object tracking. The algorithm is developed and demonstrated on lane tracking problems before being put to use in obstacle detection in Chapter 4.

Appendix B

DVD-ROM contents

Accompanying this thesis is a DVD-ROM containing:

Introduction

- **tacad.avi:** Victorian Transport Accident Commission (TAC) 1994 television advertisement: "Nightshift (Fatigue)" discussed in Chapter 1.

Lane_tracking

- **people_track1.avi** and **people_track2.avi**: People tracking using Distillation as discussed in Chapter 3.
- **lanetracking_straight.avi** Original straight lane tracking using Distillation as discussed in Chapter 3.
- **lanetracking_curve_lookahead.avi** Curvature and look-ahead lane tracking using Distillation as discussed in Chapter 3.
- **circles.avi** Circular lane curvature verification as discussed in Chapter 3.
- **longtrip/** Long lane tracking using Distillation as discussed in Chapter 3.

Obstacle_detection

- **obst_detect.avi**: Obstacle detection using Distillation as discussed in Chapter 4.
- **obst_distillation.avi**: Obstacle distillation using Distillation as discussed in Chapter 4.

- **obst_track.avi**: Obstacle tracking using Distillation as discussed in Chapter 4.
- **obst_carfollowing.avi**: Obstacle detection and tracking as discussed in Chapter 4.

Sign_Recognition

- **sign_detect.avi** Symbolic road sign detection as discussed in Chapter 5.
- **sign_recognition.avi** Symbolic road sign detection as discussed in Chapter 5.
- **enhance_online.avi** Road sign image enhancement as discussed in Chapter 5.

Road_Scene_Complexity_Assessment

- **samples/**: Sample MPEG files used to assess judge monotony Chapter 6.
- **monotony/**: Sample MPEG files used to by the metric in road trials discussed in Chapter 6.

Automated_Co-driver_experiments

- **weowit.wav** and **rumble.wav**: Sounds of auditory warnings.
- **signs_codriver.avi**: Context sensitive speed sign Automated Co-driver discussed in Chapter 7.
- **automated_codriver.avi**: Comprehensive context sensitive Automated Co-driver discussed in Chapter 7.

Appendix: Testbed

- **trev/**: Most of the TREV source tree. Source files for code containing IP sensitive code has been removed, header files remain for these files.

Bibliography

Amditis, A., Kubmann, H., Polychronopoulos, A., Engstrom, J., and Andreone, L. (2006). System Architecture for Integrated Adaptive HMI Solutions. In *IEEE Intelligent Vehicles Symposium*.

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal filtering*. Prentice-Hall Information and System Sciences Series, Englewood Cliffs: Prentice-Hall.

ANRTC (1999). *Australian National Road Transport Commission, Australian Road Rules*.

Apostoloff, N. (2005). *Vision based lane tracking using multiple cues and particle filtering*. Master's thesis, Australian National University.

Apostoloff, N. and Zelinsky, A. (2004). Vision in and out of vehicles: integrated driver and road scene monitoring. *International Journal of Robotics Research*, **23**(4-5), 513–538.

Aschwanden, P. and Guggenbuhl, W. (1993). Experimental results from a comparative study on correlation-type registration algorithms. In Forstner and Ruweidel, editors, *Robust Computer Vision*, pages 268–289. Wichmann.

Assistware (2006). SafeTrac - Lane departure warning system. http://www.assistware.com/.

ATSB (2004a). Australian Transport Safety Bureau, Road deaths australia 2004 statistical summary. Technical report, Australian Government.

ATSB (2004b). Australian Transport Safety Bureau, Serious injury due to road crashes: road safety statistics report. Technical report, Australian Government.

ATSB (2006a). Australian Transport Safety Bureau, Fatal Road Crash Database. *http:www.atsb.gov.auroadfatal_road_crash_database.aspx*.

ATSB (2006b). Australian Transport Safety Bureau, Novice driver safety. *http://www.atsb.gov.au/road/novice_driver_safety/novice_driver_safety.aspx*.

Austin, D. and Barnes, N. (2003). Red is the new black - or is it? In *Australasian Conference on Robotics and Automation*, pages Brisbane, Australia.

Austin, D., Fletcher, L., and Zelinsky, A. (2001). Mobile robotics in the long term-exploring the fourth dimension. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 613–618, Maui, Hi, USA.

Australian Government (2000). Beyond the midnight oil: Managing fatigue in transport. Technical report, House of Representatives Standing Committee on Communications, Transport and the Arts.

Auty, G., Corke, P. I., Dunn, P., Jensen, M., Macintyre, I. B., Mills, D. C., Nguyen, H., and Simons, B. (1995). Image acquisition system for traffic monitoring applications. *Proceedings of SPIE - Cameras and Systems for Electronic Photography and Scientific Imaging*, **2416**, 119–133.

Baker, S. and Kanade, T. (2000). Limits on super-resolution and how to break them. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 372–379.

Baker, S., Matthews, I., Xiao, J., Gross, R., Kanade, T., and Ishikawa, T. (2004). Real-time Non-rigid Driver Head Tracking For Driver Mental State Estimation. Technical Report CMU-RI-TR-04-10,, Robotics Institute, Carnegie Mellon University.

Balkin, T. (2004). The heart of the matter: Management of performance in the operational environment. CARRS-Q 'Workshop on Monitoring Hypovigilance in monotonous conditions' presentation.

Balkin, T., Thorne, D., Sing, H., Thomas, M., Redmond, D., Wesensten, N., J. Williams, S. H., and Belenky, G. (2000). Effects of sleep schedules on commercial motor vehicle driver performance. Technical Report DOT-MC-00-133, U.S. Department of Transportation, Federal Motor Carrier Safety Administration, Washington DC.

Baluja, S. (1996). Evolution of an artificial neural network based autonomous land vehicle controller. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, **26**(3), 450–463.

Banks, J., Bennamoun, M., and Corke, P. (1997). Fast and robust stereo matching algorithms for mining automation. In *Proc. International Workshop on Image Analysis and Information Fusion*, pages 139–149.

Barron, J. L., Fleet, D., Beauchemin, S., and Burkitt, T. (1992). Performance of optical flow techniques. *Computer Vision and Pattern Recognition*, **92**, 236–242.

Batavia, P. H., Pomerleau, D. A., and Thorpe, C. E. (1998). Predicting lane position for roadway departure prevention. In *Proc. IEEE Intelligent Vehicles Symposium*.

Behringer, R. and Müller, N. (1998). Autonomous road vehicle guidance from autobahnen to narrow curves. *IEEE Trans. Robot. Automat.*, **14**(5), 810–815.

Bertgozzi, M. and Broggi, A. (1998). GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. In *IEEE Transactions on Image Processing*, volume 7, pages 62–81.

Bertozzi, M. and Broggi, A. (1998). Gold: a parallel real-time stereo vision system for generic obstacle and lane detection. In *IEEE Transactions on Image Processing*, volume 7, pages 62–81.

Bertozzi, M., Broggi, A., and Fascioli, A. (2000). Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, **32**, 1–16.

Bertozzi, M., Broggi, A., Cellario, M., Fascioli, A., Lombardi, P., and Porta, M. (2002). Artificial vision in road vehicles. *Proceedings of the IEEE*, **90**(7), 1258–1270.

Bishop, R. (2005). *Intelligent Vehicle Technology and Trends*. Artech House, first edition edition.

Borgefors, G. (1986). Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, **34**(3), 344 – 371.

Bouget, J. Y. (2002). Matlab camera calibration toolbox.

Braver, E., Preusser, C., DF.Preusser, HM.Baum, Beilock, R., and Ulmer, R. (1992). Long hours and fatigue: a survey of tractor-trailer drivers. *Public Health Policy*, **13**(3), 341–366.

Broadhurst, A., Baker, S., and Kanade, T. (2005). Monte Carlo Road Safety Reasoning. In *IEEE Intelligent Vehicles Symposium*.

Broggi, A., Bertozzi, M., and Fascioli, A. (2001). Self-calibration of a stereo vision system for automotive applications. In *IEEE*, pages 3698–3702, Seoul, Korea.

Brooker, G. and Durrant-White, H. (2001). Millimetrewave radar. In *Australian Conference on Robotics and Automation*, Melbourne, Australia.

Brown, R. G. and Hwang, P. Y. C. (1997). *Introduction to random signals and applied Kalman filtering*. John Wiley & Sons, Inc., 3rd edition.

Cai, J. and Goshtasby, A. (1999). Detecting human faces in color images. *Image and Vision Computing*, **18**, 63–75.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **8**(6), 679–698.

Capel, D. and Zisserman, A. (2000). Super-Resolution enhancement of text image sequences. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 600–605.

Capel, D. and Zisserman, A. (2003). Computer vision applied to super resolution.

Carsten, O. and Tate, F. (2001). Intelligent speed adaptation: The best collision avoidance system? In *17th International Technical Conference on the Enhanced Safety of Vehicles*, Amsterdam, The Netherlands.

Chaitin, G. J. (1974). Information-theoretic limitations of formal systems. *J. ACM*, **21**(3), 403–424.

Cheeseman, P., Kanefsky, B., Kraft, R., Stutz, J., and Hanson, R. (1996). Super-resolved surface reconstruction from multiple images. In G. R. Heidbreder, editor, *Maximum Entropy and Bayesian Methods*, pages 293–308. Kluwer Academic Publishers, Dordrecht, the Netherlands.

Chesher, G. (1995). Cannabis and road safety: an outline of research studies to examine the effects of cannabis on driving skills and actual driving performance. In *Parliament of Victoria, Road Safety Committee, Inquiry into the effects of drugs (other than alcohol) on road safety.*, pages 67–96.

Coimbra, M. and Davies, M. (2003). A new pedestrian detection system using mpeg-2 compressed domain information. In *Proc. of IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2003)*, pages 598–602, Madrid, Spain.

Coin3D (2006). Coin 3D high-level 3D graphics toolkit. http://www.coin3d.org.

Crisman, J. and Thorpe, C. (1993a). SCARF: A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automation*, **9**(1), 49 – 58.

Crisman, J. D. and Thorpe, C. E. (1993b). SCARF: A color vision system that tracks roads and intersections. *IEEE Trans. on Robotics and Automation*, **9**(1).

Crowley, J. L. and Berard, F. (1997). Multi-modal tracking of faces for video communications. In *Computer Vision and Pattern Recognition, CVPR '97.*

Dankers, A. and Zelinksy, A. (2003). A Real-World Vision System: Mechanism, control and vision processing. In *Third International Conference on Computer Vision Systems*, Graz, Austria.

DARPA (2004). Defense Advanced Research Projects Agency, Grand Challenge 2004. http://www.darpa.mil/grandchallenge04/index.htm.

DARPA (2005). Defense Advanced Research Projects Agency, Grand Challenge 2005. http://www.darpa.mil/grandchallenge05/index.htm.

DARPA (2007). Defense Advanced Research Projects Agency, Grand Challenge 2007. http://www.darpa.mil/grandchallenge.

Darrell, T., Gordon, G., Harville, M., and Woodfill, J. (2000). Integrated person tracking using stereo, color, and pattern recognition. *International Journal of Computer Vision*, **37**(2), 175–185.

Davies, R. (1994). Writing a matrix package in C++. In *The Second Annual Object Oriented Numerics Conference*, pages 207–213, Rogue Wave Software, Corvallis.

Dellaert, F., Pomerleau, D., and Thorpe, C. (1998a). Model-based car tracking integrated with a road-follower. In *IEEE*.

Dellaert, F., Thorpe, C., and Thrun, S. (1998b). Super-resolved texture tracking of planar surface patches. In *Proc. IEEE/RSJ International Conference on Intelligent Robotic Systems*, pages 197–203.

Desmond, P. and Matthews, G. (1997). Implications of task-induced fatigue effects for in-vehicle countermeasures to driver fatigue. *Accident Analysis and Prevention*, **29**(4), 515–523.

Dickmanns, E. D. (1999a). *Dynamic Vision for Intelligent Vehicles*. Unpublished.

Dickmanns, E. D. (1999b). An expectation-based, multi-focal, saccadic (ems) vision system for vehicle guidance. In *Proc. International Symposium on Robotics and Research*, Salt Lake City, Utah.

Dickmanns, E. D. and Graefe, V. (1988a). Applications of dynamic monocular machine vision. *Machine Vision and Applications*, **1**(4), 241–261.

Dickmanns, E. D. and Graefe, V. (1988b). Dynamic monocular machine vision. *Machine Vision and Applications*, **1**(4), 223–240.

Dickmanns, E. D. and Mysliwetz, B. D. (1992). Recursive 3-d road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**(2), 199–213.

Dixon, J. C. (1991). *Tyres, suspension and handling*. Cambridge University Press, Cambridge.

Draskczy, M. and Mocsri, T. (1997). *Present Speeds and Speed Management Methods in Europe*. Master's thesis, University of Lund, Sweden. KTI, Hungary.

Du, Y. and Papanikolopoulos, N. P. (1997). Real-time vehicle following through a novel symmetry-based approach. In *IEEE Int. Conf. on Robotics and Automation*, pages 3160–5.

Duffy, C. J. and Wurtz, R. H. (1997). Medial superior temporal area neurons respond to speed patterns in optical flow. *The Journal of Neuroscience*, **17**(8), 2839–2851.

Edquist, J., Horberry, T., Regan, M., and Johnston, I. (2005). 'Visual Clutter' and external-to-vehicle driver distraction. In *The Australasian College of Road Safety (acrs), StaySafe Committee NSW Parliament International Conference on Driver Distraction*.

Enkelmann, W. (2001). Video-based driver assistance - from basic functions to applications. *International Journal of Computer Vision*, **45**(3), 201–221.

European Union (2004). AIDE Adaptive Integrated Driver-vehicle InterfacE. http://www.aide-eu.org/index.html.

Fang, C. Y., Fuh, C. S., Chen, S. W., and Yen, P. S. (2003). A road sign recognition system based on dynamic visual model. In *Proc IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 750–755.

Farsiu, S., Robinson, D., Elad, M., and Milanfar, P. (2003). Fast and robust multi-frame super-resolution.

Faugeras, O., Hotz, B., Mathieu, H., Viéville, T., Zhang, Z., Fua, P., Théron, E., Moll, L., Berry, G., Vuillemin, J., Bertin, P., and Proy, C. (1993). Real time correlation-based stereo: algorithm, implementations and applications. Technical report, Institut National de Recherche en Informatique et Automatique, France.

FFMPEG (2005). Ffmpeg is an audio/video conversion tool. it includes libavcodec. http://sourceforge.net/projects/ffmpeg/.

Fleet, D. J. and Langley, K. (1995). Recursive filters for optical flow. *IEEE Trans. Pattern Anal. Machine Intell.*, **17**(1), 61–67.

Forsyth, D. A. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall.

Franke, U. and Heinrich, S. (2002). Fast obstacle detection for urban traffic situations. *IEEE Trans. Intell. Transport. Syst.*, **3**(3), 173–181.

Franke, U. and Rabe, C. (2005). Kalman Filter based Depth from Motion with Fast Convergence. In *IEEE Intelligent Vehicles Symposium*, pages 181–186.

Franke, U., Gavrila, D., Görzig, S., Lindner, F., Paetzold, F., and Wöhler, C. (1999). Autonomous Driving approaches Downtown. *IEEE Intelligent Systems*, **9**(6).

Fua, P. (1993a). A parallel stereo algorithm that produces dense depth maps and preserves image features.

Fua, P. (1993b). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, **6**(1), 35–49.

Gandhi, T. and Trivedi, M. M. (2006). Pedestrian collision avoidance systems: A survey of computer vision based recent studies. In *Proceedings of the Intelligent Transportation Systems Conference*, pages 976–981.

Gat, E. (1998). *Artificial Intelligence and Mobile Robots*, chapter Three-layer architectures. AIII Press/MIT Press, Menlo Park, California, USA.

Gavrila, D. M. (1998). A road sign recognition system based on dynamic visual model. In *Proc 14th Int. Conf. on Pattern Recognition*, volume 1, pages 16–20.

Gerdes, J. C. and Rossetter, E. J. (2001). A unified approach to driver assistance systems based on artificial potential fields. *Journal of Dynamic Systems, Measurement and Control*, **123**(3), 431–438.

Gerkey, B. P., Vaughan, R. T., Støy, K., Howard, A., Sukhatme, G. S., and Mataric, M. J. (2001). Most Valuable Player: A Robot Device Server for Distributed Control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1226–1231, Wailea, Hawaii.

Giachetti, A., Campani, M., and Torre, V. (1998). The Use of Optical Flow for Road Navigation. *IEEE Transactions on Robotics and Automation*, **14**(1), 34–48.

Goodrich, M. and Boer, E. (2000). Designing human-centered automation: trade-offs in collision avoidance systems design. *IEEE Transactions on Intelligent Transport SYstems*.

Gordon, N., Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *Proc. IEEE Conference on Radar Signal Processing*, volume 140, pages 107–113.

Gordon A. D. (1966). Perceptual basis of vehicular guidance. *Public Roads*, **34**(3), 53–68.

Green, P. (2000). Crashes Induced by Driver Information Systems and What Can Be Done to Reduce Them. *Society of Automotive Engineers*.

Gregor, R., Lutzeler, M., Pellkofer, M., Siedersberger, K.-H., and Dickmanns, E. (2002). EMS-Vision: a perceptual system for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, **3**(1), 48–59.

Grubb, G. and Zelinsky, A. (2004). 3D Vision Sensing for Improved Pedestrian Safety. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Parma, Italy.

Hartley, R. and Zissermann, A. (2000). *Multiple View Geometry in computer vision*. Cambridge University Press, Cambridge, UK.

Haussecker, H. W. and Fleet, D. J. (2000). Computing optical flow with physical models of brightness variation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 760–767, Hilton Head.

Haussecker H., S. H. and B., J. (1998). Tensor-based image sequence processing techniques for the study of dynamical processes. In *Proc. International Symposium on Real-time Imaging and Dynamic Analysis*, volume 32, pages 704–711, Hakodate, Japan. International Society of Photogrammetry and Remote Sensing, ISPRS.

Hautire, N. and Aubert, D. (2003). Driving assistance : Automatic fog detection and measure of the visibility distance. In *IEEE Intelligent Transport Systems*, Madrid, Spain.

Haworth, N., Vulcan, P., Schulze, M., and Foddy, B. (1991). Testing of commercially available fatigue monitors. Technical report, Monash University Accident Research Centre.

Haworth, N. L., Triggs, T. J., and Grey, E. M. (1988). Driver fatigue: Concepts, measurement and crash countermeasures. Technical report, Federal Office of Road Safety Contract Report 72 by Human Factors Group, Department of Psychology, Monash University.

Heisele, B. and Ritter, W. (1995). Obstacle Detection Based on Color Blob Flow. In *Proc. IEEE Intelligent Vehicle Symposium*, pages 282–286.

Hirschmüller, H., Innocent, P. R., and Garibaldi, J. (2002). Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, **47**, 229–246.

Hoffmann, C. (2006). Fusing multiple 2D visual features for vehicle detection. In *Intelligent Vehicles Symposium*, pages 406–411, Tokyo, Japan.

Horn, B. and Schunck, B. (1981). Determining optical flow. *Artificial Intelligence*, **17**, 185–204.

Hough, P. V. C. (1959). Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*. CERN.

Howarth, N. L., Heffernan, C. J., and Horne, E. J. (1989). Fatigue in truck accidents. Technical Report 3, Monash University Accident Research Centre.

Howat, P., Sleet, D., and Smith, I. (1991). Alcohol and driving: is the 0.05% blood alcohol concentration limit justified? *Drug and Alcohol Review*, **10**(1), 151–166.

Hsu, S.-H. and Huang, C.-L. (2001). Road sign detection and recognition using matching pursuit method. *Image and Vision Computing*, **19**, 119–129.

Ibeo AS (2007). Ibeo lux automotive laser sensor. http://www.ibeo-as.com.

Ingwersen, P. (1995). Tracing the problem of driver fatigue. *Driver Impairment, Driver Fatigue and Driving Simulation, Hartley, L. (Ed.), Taylor & Francis, London*, pages 76–86.

Isard, M. and Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. In *Proc. of European Conf. on Computer Vision*, volume 1, pages 343–356.

Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, **29**(1), 5–28.

Ishikawa, T., Baker, S., and Kanade, T. (2004). Passive driver gaze tracking with active appearance models. In *11th World Congress on Intelligent Transportation Systems*.

Iteris (2002). Welcome to iteris. Internet http://www.iteris.com/.

Jähne, B. and Haussecker, H., editors (2000). *Computer Vision and Applications, A Guide for Students and Practitioners*. Academic Press, San Diego, CA.

Jochem, T., Pomerleau, D., and Thorpe, C. (1993). MANIAC: A next generation neurally based autonomous road follower. In *Proceedings of the International Conference on Intelligent Autonomous Systems*. Also appears in the Proceedings of the Image Understanding Workshop, April 1993, Washington D.C., USA.

Johansson, B. (2002). *Road sign recognition from a moving vehicle*. Master's thesis, Centre for Image Analysis, Swedish University of Agricultural Sciences.

Kagami, S., Okada, K., Inaba, M., and Inoue, H. (2000). Design and implementation of onbody real-time depthmap generation system. In *Proc. IEEE Int. Conf. on Robotics and Automation*, California, USA. IEEE Computer Press.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, pages 35–45.

Kenue, S. K. (1989). Lanelok: detection of lane boundaries and vehicle tracking using image-processing techniques — parts i and ii. In *Proceedings, SPIE Mobile Robots IV, 1989*, pages 221–244.

Kim, S.-H. and Kim, H.-G. (2000). Face detection using multimodal information. In *Proc. of IEEE International Conference on Face and Gesture Recognition*, pages 14–19.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5**(1), 1–25.

Kittler, J., Hatef, M., Duin, R., and Matas, J. (1998). On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, **20**(3), 226–239.

Kluge, K. and Thorpe, C. (1992). Representation and recovery of road geometry in YARF. In *Proceedings of the Intelligent Vehicles '92 Symposium*, pages 114 –119.

Krüger, W. (1999). Robust real-time ground plane motion compensation from a moving vehicle. *Machine Vision and Applications*, **11**, 203–212.

Krüger, W., Enkelmann, W., and Rossle, S. (1995). Real-Time Estimation and Tracking of Optical Flow Vectors for Obstacle Detection. In *Proc. IEEE Intelligent Vehicles Symposium*, pages 304–309.

Kuge, N., Yamamura, T., Shimoyama, O., and Liu, A. (1998). A driver behavior recognition method based on a driver model framework. *Transactions of the Society of Automotive Engineers*.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, **22**(1), 79–86.

Kun, A. L., Miller III, W. T., Pelhe, A., and Lynch, R. L. (2004). A Software Architecture Supporting In-Car Speech Interaction. In *IEEE Intelligent Vehicles Symposium*, pages 471–476.

Labayrade, R., Aubert, D., and Tarel, J.-P. (2002). Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In *Proc. IEEE Intelligent Vehicle Symposium*, France.

Land, M. and Lee, D. (1994). Where we look when we steer. *Nature*, **369**(6483), 742–744.

Lee, K. F. and Tang, B. (2006). Image Processing for In-vehicle Smart Cameras. In *Intelligent Vehicles Symposium*, pages Tokyo, Japan.

Lee, S. and Kwon, W. (2001). Robust lane keeping from novel sensor fusion. In *IEEE*, pages 3740–3745, Seoul, Korea.

Leonard, J., How, J., Teller, S., Berger, M., Campbell, S., Fiore, G., Fletcher, L., Frazzoli, E., Huang, A., Karaman, S., Koch, O., Kuwata, Y., Moore, D., Olson, E., Peters, S., Teo, J., Truax, R., Walter, M., Barrett, D., Epstein, A., Mahelona, K., Moyer, K., Jones, T., Buckley, R., Attone, M., Galejs, R., Krishnamurthy, S., and Williams, J. (2008). A perception driven autonomous urban robot. submitted to *International Journal of Field Robotics*, (-), –.

Lotti, J.-L. and Giraudon, G. (1994). Correlation algorithm with adaptive window for aerial image in stereo vision. In *Proc. International Conference on Pattern Recognition*, volume A, pages 2310–15, Rome, Italy. Institut National de Recherche en Informatique et Automatique.

Loy, G. and Barnes, N. (2004). Fast shape-based road sign detection for a driver assistance system. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems.IROS2004*. in press.

Loy, G. and Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. *IEEE Trans Pattern Analysis and Machine Intelligence*, **25**(8), 959–973.

Loy, G., Fletcher, L., Apostoloff, N., and Zelinsky, A. (2002). An adaptive fusion architecture for target tracking. In *Proc. The 5th International Conference on Automatic Face and Gesture Recognition*, Washington DC.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679.

Mallot, H. A. (2000). *Computational Vision: Information Processing in Perception and Visual Behaviour*. A Bradford Book, MIT press, Massachusetts, USA.

Maltz, M. and Shinar, D. (2004). Imperfect In-Vehicle Collision Avoidance Warning Systems Can Aid Drivers. *Human Factors*, **46**(2).

Marr, D. (1982). *Vision: A computational investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Company, New York, USA.

Matsumoto, Y., Heinzmann, J., and Zelinsky, A. (1999). The essential components of human-friendly robots. In *International Conference on Field and Service Robotics FSR'99*.

McCall, J. C. and Trivedi, M. M. (2006a). Human Behaviour Based Predictive Brake Assistance. In *IEEE Intelligent Vehicles Symposium*, pages 8–12.

McCall, J. C. and Trivedi, M. M. (2006b). Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transport Systems*, **7**(1), 20– 37.

McCall, J. C., Achler, O., and Trivedi, M. M. (2004). A collaborative approach for human-centered driver assistance systems. In *IEEE lntelligent Transportation Systems Conference*, Washington, D.C., USA.

McEwen, B. and Schmeck, H. (1994). *The Hostage Brain*. The Rockefeller University Press, New York.

Mezaris, V., Kompatsiaris, I., Boulgouris, N. V., and Strintzis, M. G. (2004). Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval. *IEEE Transations on Circuits and Systems for Video Technology*, **14**(5), 606–621.

Mihajlovski, A. (2002). Autonomous road vehicle: Radar sensor characterization and adaptive cruise control. Australian National University Engineering Honours Thesis.

MobilEye Vision Technologies Ltd (2002). Mobileeye homepage. Internet http://www.mobileye.com/home.html.

Montemerlo, M., Thrun, S., Dahlkamp, H., Stavens, D., and Strohband, S. (2006). Winning the DARPA Grand Challenge with an AI Robot. In *American Association of Artificial Intelligence 2006 (AAAI06).*, Boston, USA.

Motion Picture Experts Group (2004). Mpeg information & frequently asked question page. *http:www.mpeg.orgMPEGstarting-points.html#faqs*.

Mourant, R. R., Rockwell, T. H., and Rackoff, N. J. (1969). Drivers' eye movements and visual workload. *Highway Research Record*, **299**, 1–10.

Mühlmann, K., Maier, D., Hesser, J., and Männer, R. (2002). Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, **47**, 79–88.

Nageswara, S. R. (2001). On fusers that perform better than best sensor. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, **23**(8).

Neale, V. L., Dingus, T. A., Klauer, S., Sudweeks, J., and Goodman, M. (2005). An Overview of the 100-Car Naturalistic study and findings. In *Proceedings of the International Conference on Enhanced Safety of Vehicles*, Washington, USA.

Necker, L. A. (1832). Observations on some remarkable optical phaenomena seen in Switzerland; and on an optical phaenomenon which occurs on viewing a figure of a crystal or geometrical solid. *The London and Edinburgh Philosophical Magazine and Journal of Science*, **1**, 329–337.

Negahdaripour, S. (1998). Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, **20**(9), 961–979.

NHTSA (1999). National Highway Traffic Safety Administration, The autonav/dot project: baseline measurement system for evaluation of roadway departure warning systems. Technical report, U.S. Department of Transport.

Nilsson, T., Nelson, T. M., and Carlson, D. (1997). Development of fatigue symptoms during simulated driving. *Accident analysis and prevention*, **29**(4), 479–488.

NSWStaySafe (2005). Parliament of New South Wales Joint Standing Committee on Road Safety (Staysafe): Inquiry into Driver Distraction. http://www.parliament.nsw.gov.au/prod/parlment/committee.nsf/V3Home.

OECD (2006). Organisation for Economic Co-operation and Development, Factbook 2006 - Economic, Environment and Social Statistics: Quality of life. http://www.sourcecd.org.

OECD/ECMT (2006). Organisation for Economic Co-operation and Development, Transport Research Centre, Working group on Achieving Ambitious Road Safety Targets Country reports on road safety performance. In *OECD European Conference of Ministers of Transport (ECMT)*, http://www.cemt.org/JTRC/WorkingGroups/RoadSafety.

Oesch, S. L. (2005). Automated Red Light and Speed Camera Enforcement. In *District of Columbia Public Roundtable on Automated Enforcement*. Insurance Institute for Highway Safety.

Okada, K., Inaba, S. K. M., and Inoue, H. (2001). Walking human avoidance and detection from a mobile robot using 3d depth flow. In *International Conference on Robotics and Automation (ICRA)*, pages 2307–2312, Seoul, Korea.

Oliver, N. and Pentland, A. (2000). Graphical models for driver behavior recognition in a SmartCar. In *Intelligent Vehicles Symposium*.

OMG (2002). Object Management Group, CORBA/IIOP 2.4 specification. internet: http://www.omg.org/.

Paclik, P., Novovicova, J., Somol, P., and Pudil, P. (2000). Road sign classification using the laplace kernel classifier. *Pattern Recognition Letters*, **21**, 1165–1173.

Parent, M. (2004). From drivers assistance to full automation for improved efficiency and better safety. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 5, pages 2931 – 2934.

Petersson, L. (2002). *A Framework for Integration of Processes in Autonomous Systems*. Ph.D. thesis, Computational Vision and Active Perception Laboratory, Royal Institute of Technology (KTH), Stockholm, Sweden.

Photon Focus (2006). LinLog Technology. http://www.photonfocus.com/html/eng/cmos/linlog.php.

Piccioli, G., Micheli, E. D., Parodi, P., and Campani, M. (1996). Robust method for road sign detection and recognition. *Image and Vision Computing*, **14**(3), 209–223.

Pilu, M. (1997). On Using Raw MPEG Motion Vectors To Determine Global Camera Motion. Technical Report HPL-97-102, Digital Media Department, HP Laboratories Bristol, UK.

Pomerleau, D. (1995). Ralph: Rapidly adapting lateral position handler. In *Proc. IEEE Symposium on Intelligent Vehicles*, pages 506 – 511.

Pomerleau, D. and Jochem T. (1996). Rapidly Adapting Machine Vision for Automated Vehicle Steering. *IEEE Expert: Special Issue on Intelligent System and their Applications*, **11**(2), 19–27.

Priese, L., Klieber, J., Lakmann, R., Rehrmann, V., and Schian, R. (1994). New results on traffic sign recognition. In *Proceedings of the Intelligent Vehicles Symposium*, pages 249–254, Paris. IEEE Press.

Redelmeiser, D. and Tibshirani, R. (1997). Association between cellular telephone calls and motor vehicle collisions. *The New England Journal of Medicine*, **336**(7), 453–458.

Regan, M. A. (2005). Keynote address. In *The Australasian College of Road Safety (acrs), NSW Joint parliamentary standing committee (Staysafe) International Conference on Driver Distraction.*, Sydney, Australia.

Roberts, J. and Corke, P. (2000). Obstacle detection for a mining vehicle using a 2d laser. In *Proc. Australian Conference on Robotics and Automation (ACRA)*, pages 185–190, Melbourne, Australia.

Salvucci, D. D. and Liu, A. (2002). The time course of a lane change: Driver control and eye-movement behavior. *Transportation Research Part F*, **5**, 123–132.

Scharstein, D., Szeliski, R., and Zabih, R. (2001). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *In Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision*, Kauai, HI.

Schrater, P. R., Knill, D. C., and Simoncelli, E. P. (2001). Perceiving visual expansion without optic flow. *Nature*, **410**(6830), 816–819.

Schultz, R. R. and Stevenson, R. L. (1996). Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, pages 996–1011.

Schweiger, R., Neumann, H., and Ritter, W. (2005). Multiple-cue data fusion with particle filters for vehicle detection in night view automotive applications. In *IEEE Intelligent Vehicles Symposium*.

Seeing Machines (2001). FaceLAB: A face and eye tracking system. http:www.seeingmachines.com.

ServoToGo inc. (2000). Servo To Go Motion Controller Card. http://www.servotogo.com/, 8117 Groton Lane Indianapolis, IN, USA.

Sethi, I. K. and Patel, N. V. (1995). Statistical approach to scene change detection. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 329–338.

Shaposhnikov, D. G., Podladchikova, L. N., Golovan, A. V., and Shevtsova, N. A. (2002). A road sign recognition system based on dynamic visual model. In *Proc 15th Int Conf on Vision Interface*, Calgary, Canada.

Shimizu, M. and Okutomi, M. (2002). An analysis of sub-pixel estimation error on area-based image matching. In *International Conference on Digital Signal Processing*, pages 1239– 1242.

Shor, R. E. and Thackray, R. I. (1970). A program of research in highway hypnosis: a preliminary report. *Accident Analysis and Prevention*, **2**, 103109.

SICK AG (2003). *SICK LMS manual*. SICK AG, Germany, www.sick.com.

Simoncelli, E. P., Adelson, E. H., and J., H. D. (1991). Probability distribution of optical flow. In *International conference on Computer Vision and Pattern Recognition*, pages 310–315.

Smith, S. M. and Brady, J. M. (1995). Asset-2: Real-time motion segmentation and shape tracking. *IEEE Trans. Pattern Anal. Machine Intell.*, **17**(8), 814–820.

Soto, A. and Khosla, P. (2001). Probabilistic adaptive agent based system for dynamic state estimation using multiple visual cues. In *Proc. of Int. Sym. Robotics Research (ISRR)*.

Southall, B. and Taylor, C. J. (2001). Stochastic road shape estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, Vancouver, Canada.

Stutts, J., Reinfurt, D., Staplin, L., and Rodgman, E. (2001). The role of driver distraction in traffic crashes. Technical report, Foundation for Traffic Safety, USA.

Summala, H. and Nicminen, T. (1996). Maintaining lane position with peripheral vision during in-vehiclc tasks. *Human Factors*, **38**, 442–451.

Sun, Z., Bebis, G., and Miller, R. (2006). On-Road Vehicle Detection: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(5), 694–711.

Sutherland, O., Truong, H., Rougeaux, S., and Zelinsky, A. (2000). Advancing active vision systems by improved design and control. In *Proceedings of International Symposium on Experimental Robotics (ISER2000)*.

TAC (1994). Victorian Transport Accident Commission, Advertisement: Nightshift (fatigue). Television adverstisement, http://www.tac.vic.gov.au.

Takemura, K., Ido, J., Matsumoto, Y., and Ogasawara, T. (2003). Driver monitoring system based on non-contact measurement system of driver's focus of visual attention. In *Proc. IEEE Symposium on Intelligent Vehicles*, pages 581–586, Columbus, Ohio.

Thiffault, P. (2004). Environmental and personality precursors of driver fatigue and hypovigilance. CARRS-Q 'Workshop on Monitoring Hypovigilance in monotonous conditions' presentation.

Thiffault, P. and Bergeron, J. (2003). Monotony of road environment and driver fatigue: a simulator study. *Accident Analysis and Prevention*, **35**, 381–391.

Thorpe, C., editor (1990a). *Vision and Navigation: The Carnegie Mellon NavLab*. Kluwer Academic Publishers.

Thorpe, C. (1990b). *Vision and Navigation: the Carnegie Mellon Navlab*. Kluwer Academic Publishers.

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (2000). Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, **19**(11), 972–99.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, **128**(1-2), 99–141.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The Robot that Won the DARPA Grand Challenge. *Field Robotics*, **23**(9), 661–692.

Torsvall, L. and aAkerstedt, T. (1987). Sleepiness on the job: continuously measured EEG changes in train drivers. *Electroencephalography and Clinical Neurophysiology*, **66**(6), 502–511.

Treat, J., Tumbas, N., McDonald, S., Shinar, D., Hume, R., Mayer, R., Stansifer, R., and Castellan, N. (1979). Tri-Level study of the causes of traffic accidents: Final report - Executive summary. Technical Report DOT-HS-034-3-535-79-TAC(S), Institute for Research in Public Safety, University of Indiana, Bloomington, IN, USA.

Triesch, J. and von der Malsburg, C. (2000). Self-organized integration of adaptive visual cues for face tracking. In *Proc. of IEEE International Conference on Face and Gesture Recognition*, pages 102–107.

Trivedi, M. M., Gandhi, T., and McCall, J. (2005). Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. In *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety*.

Trolltech (2006). QT cross platform application development library. http://www.trolltech.com.

Tsai, R. Y. and Huang, T. S. (1984). Multiframe image restoration and registration. *Advances in Computer Vision and Registration*, **1**, 317–339.

Tuttle, J. R. (2000). RFID system in communication with vehicle on-board computer. US Patent 6112152, Micron Technology, Inc.

Underwood, R. T. (1991). *The Geometric Design of Roads*. The Macmillan company of Australia pty. ltd., 1st edition. IBSN: 0 7329 0585 0.

Vaughn, D. (1996). Vehicle speed control based on GPS/MAP matching of posted speeds. US patent 5485161, Trimble Navigation Limited.

VicRoads (2006). Crash Stats online accident database. *http://www.vicroads.vic.gov.au*.

VicRoadSafety (2006). Parliament of Victoria Road Safety Committee: Inquiry into Driver Distraction. http://www.parliament.nsw.gov.au/prod/parlment/committee.nsf/V3Home.

Victor, T. (2005). *Keeping Eye and Mind on the Road*. Ph.D. thesis, Uppsala University, Sweden.

Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, **63**(2), 153–161.

Wandell, B. A. (1995). *Foundations of vision*. Sinauer Associates, Sunderland, Mass. USA.

WHO (2001). World health report. Technical report, World Health Organisation, http:www.who.intwhr20012001mainenindex.htm.

Wierwille, W. W. and Ellsworth, L. A. (1994). Evaluation of driver drowsiness by trained raters. *Accident analysis and Prevention*, **26**(5), 571–581.

Williams, G. W. (1963). Highway hypnosis: An hypothesis. *International Journal of Clinical Experimental Hypnosis*, **11**, 143–151.

Williamson, T. and Thorpe, C. (1999). A trinocular stereo system for highway obstacle detection. In *Proc. International Conference on Robotics and Automation (ICRA99)*.

Wulf, O., Kiszka, J., and Wagner, B. (2003). A Compact Software Framework for Distributed Real-Time Computing. In *Fifth Real-Time Linux Workshop*, Valencia, Spain.

Yang, R., Pollefeys, M., and Li, S. (2004). Improved Real-Time Stereo on Commodity Graphics Hardware. In *IEEE Conference on Computer Vision and Pattern Recognition - Workshop*.

Young, K., Regan, M., and Hammer, M. (2003). Driver distraction: A review of the literature. Technical Report 206, Monash University Accident Research Centre (MUARC), Melbourne, Australia.

Yowell, R. O. (2005). The Evolution and Devolution of Speed Limit Law and the Effect on Fatality Rates. *Review of Policy Research*, **22**(4), 501.

Zhang, Z. (1999). Flexible camera calibration by viewing a plane from unknown orientations. In *In International Conference on Computer Vision (ICCV'99)*, pages 666–673, Corfu, Greece.

Zhang, Z., Weiss, R., and Hanson, A. R. (1997). Obstacle detection based on qualitative and qantitative 3d reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **19**(1), 15–26.

Zheng, P. and McDonald, M. (2005). Manual vs. adaptive cruise control - Can driver'ss expectations be matched? *Transportation Reseach Part C*, **13**, 421–431.