School of Computer Science and Software Engineering
Monash University

Bachelor of Digital Systems (Honours), Clayton Campus

Literature Review - 2002

A smarter computer controlled model car

November 4, 2002

Supervisors Dr. Lloyd Allison, Dr. Ronald Pose

Abstract

This Literature Review presents the theoretical background to autonomous navigation, in particular computer controlled cars, outlining the work done in the fields applicable to the project, the algorithms and hardware available and how suited each are to the tasks involved in implementing a computer controlled car. Other research projects based in the same field are also investigated, their strengths and weaknesses analysed and relationships to this project are outlined.

Contents

1 Introduction

The implementation of a Computer Controlled Car (CCC) poses a multi-faceted problem that can be solved using techniques from a wide variety of study areas in the fields of computer science and physics. Goldberg (1995) noted that robotics in general involves the application of technology in disciplines including kinematics, dynamics, control and programming. This project, the Monash CCC, presents a robotics challenge, as both a hardware and software problem. It is based upon research in the above fields, specifically image processing, motion control, programming and hardware design. The sections below outline the theoretical and practical backgrounds to autonomous navigation and specifically, the major components involved in the implementation of an autonomous vehicle. Firstly, an outline of work done by researchers on autonomous vehicles is presented, including previous work on the Monash CCC, showing the achievements and relevance of each. This is followed by sections highlighting more specific aspects of autonomous vehicles relevant to the Monash CCC.

2 A brief selected history of the field

Robotics is a field of application that has existed for many years. Since the 1960's research has generally been focused on robotic manipulators, as dictated by industry (Cox and Wilfong 1990). The 1980's saw a boom in the use of robotics in applied tasks, as the technology matured and evolved further. With this boom came a new diversity in the applications of robotic technology. Scientists from Europe and America both began work on autonomous vehicles as an area of application of robotics.

There have been many autonomous and semi-autonomous car projects, aimed at a wide variety of applications. These projects have been created from passenger sized vehicles, as well as small model vehicles as is the case with the Monash CCC. It is usual for such projects to be created as experimental platforms for technology that could be incorporated in the automobile or construction industries. RJ (Taylor 1995), ALVINN (Thorpe and Herbert 1997), VaMoRs (Graefe and Kuhnert 1991), the development of AICC projects (Tribe 1996, Franke and Gavrilla 1999), CyCab, Urbie, the urban robot, NOMAD (Apostolopoulos, Wagner and Whittaker 1999) and, most importantly to this project, the Monash CCC, are some examples.

Early autonomous vehicles did not necessarily use computer vision to achieve perception, as the Monash CCC does. Early vision systems for robotics were expensive but, with recent dramatic drops in the price of components, artificial vision systems are becoming more commonplace in robotics (Adams 2000), with many vision based autonomous vehicles being developed in the last decade.

3 Autonomous Car Projects

This section covers work done on many well known autonomous vehicles, finishing with previous work done on the Monash CCC by past honours students. Also covered are projects that do not constitute autonomous vehicles in themselves but, from the perspective of this project, have made a relevant contribution to research in the field.

3.1 Fully Autonomous Vehicles

Fully autonomous vehicles, those not requiring human or external input to function, are not commonplace on roads today. In fact, the 100% fully autonomous commercial road vehicle is still some way off in terms of research. These vehicles are ideally able to take care of all their own path planning, low-level navigation, collision avoidance etc., leaving the human in a supervisory role only (for safety purposes). All interaction would be ideally only through very high-level commands to the vehicle. While this is not yet seen in the commercial vehicle industry, it is emerging in scientific research in the laboratory, or uninhabited test fields. Such vehicles have even made it out to road-testing stages for several extended tours of freeways around the globe.

3.1.1 Carnegie Mellon University

Carnegie Mellon University's robotics institute[1], lead by Charles Thorpe, has been among the forefront of robotics advancement in the last decade. They have developed and documented a vast number of complete autonomous platforms as testbeds for many autonomous systems developed at Carnegie Mellon. Researchers at the Robotics Institute have been responsible for the Aibo, Millibot (Navarro-Serment, Grabowski, Paredis and Kholsa 1999), Pluto (Martin 1998) and many other useful projects. Of interest to this project is the NAVLAB division, who build robot cars, trucks and busses capable of autonomous driving and also driver assistance.

In particular the "No Hands Across America" tour, which employed the RALPH[2] computer program (Pomerleau 1995), is of interest to this project. During the tour, two scientists traveled across America in a passenger vehicle fitted out with equipment which controlled the lateral position, allowing them to drive with only their feet to control the speed. RALPH is a program that uses video images of the road surface in front of the vehicle to determine its own position relative to the road and then calculate the appropriate steering direction, keeping the vehicle on the road. RALPH is able to adapt quickly to changes in the appearance of lanes on the road, and the types of

---

[1] http://www.ri.cmu.edu/

[2] Rapidly Adapting Lateral Position Handler

lane markings, with little or no a priori model the road surface and features. Pomerleau (1995) showed that RALPH is able to identify and adapt quickly to changes in the road surface type with a high degree of robustness.

In this project, there is a higher degree of control available over the "road" (racetrack) surface than that available in the RALPH trials and therefore a stronger a priori model of the track can be built and exploited to reduce processor overhead. It is on this point that RALPH differs from this project and previous implementations on the Monash CCC (see later sections), as RALPH is built for generality whereas the Monash CCC is aimed at specific conditions (see Tung (2000)). Overall, RALPH is a very useful project that addresses the problem of autonomous vehicle flexibility, but uses methods far too general and costly to be implemented in this project. The only changes in lane features can be expected to be found during the course of a test on the Monash CCC by our car only under critical shadow conditions. It is expected that with investigation these conditions can be adapted too, or even ignored depending on the robustness of the final algorithm. It is possible that the lane detection method by hypothesis fitting used by RALPH may be implemented in the Monash CCC later in this project.

Another approach which has implications in autonomous navigation is found in the ALVINN (Pomerleau 1992) and MANIAC (Jochem, Pomerleau and Thorpe 1993) systems, both of which form part of a project to use neural networks, trained by observing real drivers, to develop a model for human-like driving. NAVLAB is also currently applying research from autonomous vehicles to driver assistance and collision warning systems, which may be integrated with existing automotive hardware without a great deal of modification (Thorpe, Duggins, Gowdy, MacLachlan, Mertz, Siegel, Suppe, Wang, and Yata 2002, Wang and Thorpe 2002, Duggins, McNeil, Mertz, Thorpe and Yata 2001). Based on current PC hardware, a neural network could slow the frame rate for the car below acceptable levels and thus will not be explored further by this project.

3.1.2 Universität der Bundeswehr – VaMoRs

In the mid 1990's, the VaMoRs vehicle (Graefe and Kuhnert 1991, Dickmanns, Behringer, Hildebrandt, Maurer, Thomanek and Schiehlen 1994) was the fastest autonomous vehicle in the world, able to drive at speeds limited only by the vehicle itself. Under strict conditions this 5 ton van has been clocked at up to 130km/h. The restrictions to this vehicle are that it works well on freeways but suffers a massive performance drop on unpainted roads and does not have the ability to drive on roads with other vehicles or non-static obstacles present.

The VaMoRs vehicle uses a single monochrome camera to capture images of the road surface, focusing only on regions of interest (Graefe and Kuhnert

1991, Broggi 1995b). This allows extremely fast processing of the images forming a robust basis for road detection. However this approach was found to be unsuccessful both in critical shadow conditions and when imperfections in the road surface were detected (Kluge and Thorpe 1990), making it ideal for freeway driving but of less use in urban or outback environments.

Since VaMoRs, both the Carnegie Mellon Robotics Institute (Thorpe and Herbert 1997) and other commercial researchers (Franke and Gavrilla 1999, Matthies 2000) have built systems capable of detecting obstacles and other vehicles (and in one case traffic signals – see 3.2.2). The system that will be developed for this project can afford to be much less general – refer also to Bruton (1999) for a discussion on this, and therefore forego much of the processing needed on road detection in favour of doing other important tasks.

3.1.3 Universitá di Parma – The ARGO

Under the framework of a European project called Prometheus, several vehicles and systems (see also 3.1.4 and 3.2.2) were developed from the late 1980's to the mid 1990's with the goal of researching autonomous vehicle design and hardware implementation. Out of this came the GOLD system (Bertozzi and Broggi 1998) and the ARGO vehicle (Broggi, Bertozzi, Fascioli and Conti 1999b, Broggi, Bertozzi and Fascioli 1999a). The ARGO, using the GOLD system for parallel stereo vision, was taken on a 2000 mile tour of Italy in June of 1998, with excellent results. The *"MileMiglia in Automatica"* [3] tour saw the ARGO vehicle drive itself on freeways around Italy from Parma to Firenze, through Milano and Bologna at an average speed of 88km/h and a top speed of 123km/h, driving itself over 90% of the time. This tour highlights to us the possibility of autonomous driving (or at least "supervised driving") in the near future.

The limitation of the ARGO is that the vehicle only drove on flat (of slowly changing) freeways during it's Italian tour. The system in use specifies that flat roads are an assumption for correct operation of the vehicle. This is similar to the VaMoRs vehicle, in that it makes assumptions of road quality specifications. The concept of specifying a minimum road quality is one that may be utilised in this project. Technologies from Prometheus such as ARGO, GOLD and MOB-LAB have shown proven robustness and accuracy in the field implemented on real working projects. Some of the algorithms are directly applicable to the Monash CCC, and could be implemented given the right host architecture.

---

[3] http://www.argo.ce.unipr.it/ARGO/english/

3.1.4 MOB-LAB

Another vehicle developed under the Prometheus umbrella was the MOB-LAB vehicle (Broggi 1995a, Broggi 1996). Developed by some of the same people as the ARGO, MOB-LAB was an early attempt at real-time autonomous drive processing. Several interesting algorithms, such as the transform techniques employed to remove perspective from images, came from MOB-LAB and were later seen in work done for the ARGO. Such algorithms might be useful in enhancing the accuracy of the Monash CCC's vision systems, as they have been shown to be robust to shadow problems that other vehicles, such as VaMoRs faced. It may be however, that the low cost massively parallel PAPRICA architecture used by MOB-LAB is not comparable at all to PC hardware in terms of performance on these applications, making it impossible to eventually use these algorithms on the current Intel-based host PCs.

3.1.5 Jet Power Laboratory – Planetary Exploration

It has been shown that it is possible to build a robot that can navigate and survive on hostile terrain with very limited human input. NASA's Jet Power Laboratory (JPL) were the primary body responsible for the FIDO[4] rover (Tunstel, Huntsberger, Aghazarian, Backes, Baumgartner, Cheng, Garrett, Kennedy, Leger, Magnone, Norris, Powell, Trebi-Ollennu and Schenker 2002) and have also worked on a number of other exploration related projects, such as Urbie (Matthies 2000). Among these they have developed many other projects intended for the exploration of Mars. Such projects require that the vehicle is able to navigate autonomously without human supervision for several minutes at a time, as the communication link between earth and an interplanetary robot is very limited and extremely low bandwidth compared to what can be achieved on Earth.

The design of the FIDO rover is such that a stream of commands are beamed to the rover, perhaps on another planet, telling the rover where to go next. It is up to the rover to figure out the best way to get there. Field trials have shown (Tunstel et al. 2002) that the rover can be "intelligent" enough to navigate for many miles with no human control. The rover is designed to cover hundreds of miles on its next mission.

3.2 Semi-autonomous Vehicles

Semi autonomous vehicles, including supervised semi-autonomous vehicles, individual components and vehicle technology, make up an area most widely used in the commercial automobile industry. Increasing focus in recent years on so-called "intelligent systems" for autonomous vehicles has seen a boom in

---

[4] Field Integrated Design and Operations

the research of these systems by vehicle manufacturers. Several key research papers and technologies in the field of autonomous vehicle control have come out of this. Jaguar have developed a radar based system for AICC (Tribe 1996)(see below) and worked with manufacturers such as Porsche, Mercedes and major Japanese car manufacturers on the Prometheus group of projects.

3.2.1 Passenger Vehicles and AICC

Autonomous Intelligent Cruise Control (AICC) is a system that is being developed by many organisations across the world for use in cars in the next decade (Richardson, Ward, Fairclough and Graham 1996). What the system entails is basically an evolutionary step from regular vehicle cruise control. It involves transferring control of the throttle and brake systems of the vehicle to a computer on board the vehicle, just as in cruise control. Where the AICC based systems differ is that the computer has also the ability to set the optimum speed of the cruise control, rather than having this set by the driver. This means that the vehicle will travel as per usual cruise control conditions until it encounters other traffic, when the AICC matches the vehicle speed to that of the surrounding traffic. This is a much more modest and realistic task than full autonomy and has been achieved quite well by several companies (Tribe 1996).

3.2.2 Mercedes-Benz – The VAMP

One step beyond AICC (see above) is the ability of the car to fully control the throttle and brakes as well as steering, rather than simply add another software layer to the cruise control system. AICC involves software that sits between the user and the existing functions of most automobiles, doing what the user normally does for themselves, usually with the aid of a radar.

The idea of integrating full computer control and urban environment analysis with existing passenger car technology was taken on by some researchers for Daimler-Chrysler (Franke and Gavrilla 1999). Not only did the system developed have autonomous "Intelligent Stop & Go" capabilities but could also recognise and classify a range of traffic signals and other vehicles. This system is by far the most advanced autonomous driving project to date with many promising features. The modified Mercedes-Benz saloon car (named VAMP) has been shown to react to pedestrians and act to avoid collision (as featured on *"Tomorrow's World"* (C) BBC, Autumn 1998). Many of the algorithms are computationally expensive but the system has been tuned well to work in real time on three 200MHz Power PCs.

The Monash CCC currently operates on a single-host setup where all the software is run on a single processor on one machine. Although parts of the code are optimised with multi threading (see Tung (2000)) the current hardware configuration limits the PC to single processor mode. This means that the parallelism required to run software of the magnitude of the Daimler research is not possible while keeping up the necessary frame rate. The VAMP is however the most fully-featured autonomous vehicle currently available and promises to bring the technology into commercial use very soon.

3.3 Previous work on the Monash CCC

This project should be seen as an extension to the work that has been done for previous theses (Bruton 1999, Tung 2000). The following sections re-visit that old work.

3.3.1 Computer Controlled Model Car – 1999

Bruton (1999) first implemented the hardware and software for the Monash CCC. This was a model which was sufficient to allow the host computer to steer the car around a track slowly, using the modified RF remote supplied with the car. Although that project fully achieved the goal of autonomous driving, some problems remained in the software (Tung 2000). The 1999 project included all of the features of an autonomous vehicle, with a top speed of $0.4m/s$. The configuration used was of course limited to running on an Indy due to the extensive use of Silicon Graphics' VL library and built in proprietary hardware on the Silicon Graphics Indy machine used.

3.3.2 Computer Controlled Car – 2000

The main achievement by Tung (2000) was in porting the existing configuration from a Silicon Graphics Indy computer to a PC. The justification given for this was the rapidly increasing power to cost ratio of PC's. The reimplementation on PC brought an increase in speed with the new found processing power, as well as better use of the video architecture to enhance frame capture speed (see below). Altogether the project was successful with regards to the aims.

Also completed in 2000 was a complete re-working of the control code for the car. Daniel wrote his own code in C++, without re-using any of Tim's code, as he considered the previous code to be inefficient, due to the use of forks rather than threads. Instead of reusing code he chose to implement a multi-threaded model in order to overcome the inefficiencies in switching between multiple forked processes (Muys 2000, Balakrishna 2000), and to enhance usability on a multiprocessor machine.

One of the problems remaining after Tung's project, as identified on his website[5] is the track identification algorithm. In some cases noted on the website the algorithm fails to correctly find the track edges.

---

[5] www.csse.monash.edu.au/hons/projects/2000/Daniel.Tung/

4 Vision

Vision is the most important single area of work related to vision-based autonomous systems and has been shown to be a vital part of systems based on multiple sensing methods, such as the JPL Urbie, for example (Matthies 2000). The most important computations performed by the host PC on signals from the Monash CCC are on image processing, for if the car cannot "see" (the chosen method of detecting the environment in this case) then it cannot be expected to successfully negotiate its environment. The major vision tasks of the CCC are firstly to get a visual signal from the car's camera to the host PC and then to detect the track present in that signal (and more generally – "objects" in the environment) and determine useful heading information and appropriate motion control.

The transmission aspect of vision has been implemented using an analog video camera and TV transmitter, allowing the host to receive signals. Originally this signal was decoded by a VCR which sent the signal to a Silicon Graphics Indy. Tung (2000) streamlined this process by eliminating the need for a VCR when the car is used on a PC with a video capture card.

4.1 Object Detection Algorithms

There are many useful object detection and feature extraction algorithms developed for different purposes (Sun 1992, Leu 1992). In the context of the Monash CCC, object detection primarily means track detection and extraction and secondly other car or obstacle detection as well as collision avoidance. McDonald, Franz and Shorten (2001) present a preliminary computational model for road detection and following based on the Hough transform (Hough 1962). As of the end of 2000, the only form of object detection implemented is detecting of the race track for navigation purposes. The CCC does not yet have the ability to detect other objects on or off the track and respond to them. The following sections show selected work that implements object detection, re-visiting the some previously mentioned projects, as well as specific methods used previously on the Monash CCC.

4.1.1 RALPH

The RALPH[6] computer program (Pomerleau 1995) (see also section 3.2.1) uses a simple and effective track detection method to determine the offset of the track relative to the car. By transforming a trapezoidal capture of the terrain in front of the vehicle to a square and then summing vertically all pixel intensities, RALPH builds up a graph of column intensity in the image. By hypothesising the actual track curvature in an attempt to "undo" the corners, RALPH can find the curve that produces the sharpest transitions

---

[6] Rapidly Adapting Lateral Position Handler

in the graph and therefore the straightest lines. Once a good straight line is found it is assumed that that model is appropriate for the track. RALPH has been shown to work extremely well for a full-sized car and may be of use in this project, if the processing time needed is not found to be excessive.

4.1.2 GOLD

The GOLD system (Bertozzi and Broggi 1998) was developed for lane detection using two cameras to capture a stereo image pair. As it was first implemented on a massively parallel SIMD system it was most computationally efficient to remove the perspective effect from the images in order that the same processing could then be performed on each image. This lead to the development of the inverse perspective mapping (IPM) algorithm, which translates $W = \{(x, y, z)\}$, a 3D world space, into $I = \{(u, v)\}$, a 2D screen image representation with no perspective.

Lane markings are assumed to be in the $z = 0$ plane of W space, which is represented in the remapped image as a pseudo-vertical line of bright pixels in contrast to darker ones. This allows for not only easy scan-line lane detection, but also simple distance information extraction from the remapped image, as distances are remapped uniformly. Due to this remapping, the system is inherently more robust than other algorithms to changed lighting and shadow conditions. Shadows are more likely to fall in a non-parallel fashion to the road markings and are also not as likely to continue for as great a distance. So by simple probability, a vertical summation will extract a greater magnitude of road surface as compared to noise.

Experimental results on GOLD (see section 4.2 of Bertozzi and Broggi (1998)) show that the system is capable of following road markings and hypothesising road structure even where some parts of the markings are obscured by traffic, given a clear enough image. Extensive testing has been carried out with the system proving both stable and accurate. A proposed extension to GOLD (Bertozzi, Broggi and Fascioli 1998) to handle roads of varying curvature allows the system to be further generalised for situations where the road surface is not in a single plane $z = 0$ as in the first paper. This gives GOLD a feature that many other systems do not yet have in terms of condition handling.

4.1.3 Intelligent stop and go on a Mercedes-Benz

Similar to GOLD (see above), Daimler-Chrysler have developed a system for "intelligent stop and go" (Franke and Gavrilla 1999) (see also section 3.2.2). Rather than generalising an algorithm for freeway-bound traffic, this system is targeted to urban environments where the scene is less defined and constantly variable.

Unlike GOLD (Bertozzi and Broggi 1998) where perspective is removed for parallel processing, intelligent stop and go runs on the raw image over three 200MHz PowerPC 604 computers. The need for a fast correlation analysis technique led to the development of a feature-based algorithm that classifies pixels according to the grey values of its four nearest neighbors. The system has 3^4 classes for encoding edges and corners at various orientations. Searching for corresponding pixels is reduced in the algorithm to classification of two pixels at a time based on computing whether they belong to the same feature.

Using estimates of camera parameters (height, pitch angle etc.) the algorithm creates a distance map which is then used to classify and track obstacles in succeeding frames. Detected edges and polygons are compared to a database of objects at various orientations in order to classify features in the image. This is coupled with intelligent control of vehicle movement allowing the vehicle to respond to identified features in the scene, which is built up from a stereo image pair of the surrounding environment, not just of the road surface as in GOLD.

The systems is highly flexible and can identify lane markings and road arrows, but also traffic lights, pedestrians and stop signs. Color and distance analysis of these features, coupled with the telemetry information form the car allow the system to stop the car at a red light or stop sign. From the standpoint of a car user this system provides the most features of interest to real driving, allowing the user to have cruise control enabled through even busy city streets.

4.1.4 Computer Controlled Model Car – 1999

Originally, the 1999 CCC (Bruton 1999) analysed 188 pixels of every video frame in order to achieve track detection in 2.8ms (Tung 2000). The speed of the vehicle using this method was approximately $0.4m/s$. This was achieved in conjunction with a path planning algorithm and parent / child architecture.

The 1999 project also implemented an optical flow algorithm (Quenot, Pakleza and Kowalewski 1997) for velocity measurement based on the movement of objects from one frame to another. This algorithm only slowed the track identification by a negligible amount, however the resulting resolution was shown to be too low to be useful for the CCC. This algorithm will be replaced with a hardware device later in this project.

4.1.5 Computer Controlled Car – 2000

A multi-threaded approach in C++ implemented by Tung (2000) yielded an improvement in track detection speed, using a system dubbed “autoline.” The system was robust enough to allow the car to travel at faster speeds. However, this speed increase was not applicable to cornering and in the worst case the algorithm degenerated to the equivalent of the original algorithm used by Bruton (1999).

Autoline was the final result of the research in 2000 into track detection. The algorithm was built on an intermediate system of what were known as “autonodes.” These autonodes are markers generated by the algorithm and placed automatically on the video image where the algorithm detects a pair of edges which might be the racetrack. Autoline then uses these nodes to calculate certain navigational information about the hypothetical track which has been located.

5 Motion and Control Systems

This section discusses the hardware systems and software algorithms used by robots and autonomous vehicles for motion control. This is important to the current project as motion control will have to be examined if a new hardware platform is to be developed.

5.1 Basic Movement

Basic movement is the ability of an entity to create motion. In this case the entity is a specific autonomous vehicle or robot. Basic movement covers only the fact of the motion, not any planning or design that may be working behind it. It is usual for these two things to be separated in the design of hardware and software, as in the Monash CCC.

5.1.1 Hardware Requirements

The hardware required for basic movement by a general autonomous vehicle is relatively simple and does not usually differ greatly from an ordinary vehicle. The Jet Power Laboratory showed that and autonomous vehicle need not necessarily be a car or van (Matthies 2000). ” Urbie,” the vehicle created for urban exploration had very limited applications however, especially in terms of the racetrack. Most high speed autonomous vehicles are in fact adaptations of existing human controlled vehicles (Graefe and Kuhnert 1991, Broggi et al. 1999b, Franke and Gavrilla 1999). The same is true of the Monash CCC, which is an adaptation of a successful toy car.

Model cars of interest to this project can be classified in one of two categories – proportional and non-proportional, referring to the capability of the model to steer and drive in a manner that approximates analog control. The current car is a non-proportional vehicle that can only turn its wheels 100% left or right and only has three speeds, reverse, forward and forward fast.

5.1.2 Low-level Control Algorithms

Low level control algorithms consist of the subroutines commonly found in robotics to take commands like “move 5 feet” and translate them into a series of hardware control operations to execute the command. In the case of the Monash CCC, as with most robots, these routines are specific to the robot’s architecture and must be implemented individually for each model of robot.

Bruton (1999) designed the car control algorithms and realised a major speed increase by implementing them as a separate child process, running the control system in conjunction with the track identification system, rather than in a single serial process. Tung (2000) modified this idea by implementing the separate processes as many individual threads, gaining a further speed increase. Researchers at Carnegie Mellon (Thorpe 1990) and JPL (Matthies 2000) have gone one step further by implementing some of the control algorithms in hardware and only requiring a periodic communication with the vehicles.

5.2 Motion Planning

Motion planning entails the forming of a possible (or feasible) path $\vec{P}$ from the current position in space to another position (the destination). Often the approach to planning this path recursively breaks down a sequence of one or more from A to B until all moves in the path are possible. There are several approaches to motion planning, ranging from learning methods (Overmars and Švestka 1995), probabilistic approaches (Overmars and Švestka 1996), mathematical methods (Boissonnat, Devillers and Lazard 1995) and one off methods (Taylor 1995), which require no prior knowledge of the exact environment.

5.2.1 Learning Methods

Learning methods involve in some way mapping or recording features of the terrain for later use in making decisions about the execution of high-level, global instructions. These methods are quite at home in the field of general robotics, where a robot exists in an environment and has a series of goals, either hard-coded or dynamically generated, that must be achieved. This may include repeated navigation of areas of the environment, where knowledge of the environmental features or best methods of execution are an advantage. Overmars and Švestka (1995) introduce a probabilistic learning approach to motion planning that is highly effective in a closed environment where the vehicle is required to navigate repeatedly around the environment, by building up a profile of information about the probability of an obstacle occurring in any given path.

It is currently considered, accounting for the hardware and software complexity of learning methods and the configuration of the CCC, that a learning approach to the planning of a path is too complex a task for the expected results provided on the racetrack. The proof of this lies in the fact that the CCC will only “see” each section or aspect of the terrain once in a lap and therefore learning it would be a waste of time as there would be no benefit from knowing about it. This applies if the length of a vehicle run is only one lap of the track. If it were extended to multiple laps then the method may see some results (much like a real race car driver) however the complexity of the task is too great to warrant this as perfectly acceptable

5.2.2 Non-learning Methods

A far more suitable approach to motion control for this project is a specification of a non-learning vision based method (Taylor, 1995), operating in a combination of 2D and reflex modes (Chatlia 1995) most suitable to the tasks required by the CCC. Much of the work already done on the Monash CCC works this way – reacting only to what is seen as the data becomes available.

Bruton (1999) introduced a hybrid of this idea with what was called “improved path planning,” whereby the CCC looked further ahead of its current position to begin planning for corners before they occurred. The algorithm then “forgot” about the corner once it was gone. In essence this is still a reflex mode operation as the car only reacts to each corner individually each time, rather than retaining any information about it.

5.3 Control Software

Control software for autonomous vehicles does not consist solely of object or lane detection algorithms. The vehicle must also be able to make “intelligent” choices about navigating the environment. The earliest control structure for the Monash CCC shown by Bruton (1999) demonstrated the concept of parent / child architecture applied to periodic decision making by the car. This was improved on by Tung (2000) with the introduction of a multi-threaded structure.

Other autonomous vehicle projects have shown different architectures to be successful also. Jochem and Baluja (1993) demonstrated a low cost massively parallel architecture using custom ICs. The sort of custom hardware used by Jochem and Baluja is not applicable to this project, but is still a valid and effective way of implementing the control software.

6 Hardware

6.1 Speed Detection Devices

One of the aims of this project is to develop a speed detection device for the CCC. Such a device will be a piece of hardware possibly with some software, which will allow the host PC to measure the speed of the car in some way. Several approaches common in the marketplace today are outlined below.

Using ideas taken from automotive parts, several different types of speed detection are available. The bulk of these can be classified into two main technology groups, magnetic and optical sensors. Magnetic sensors (Gilbert 2001, Law 1998) have been used in ABS systems, fuel systems and car valve timing systems for many years and are a proven technology. These include variable reluctance sensors, Hall effect sensors and magneto-resistive sensors, which eliminate the problems of early variable reluctance type sensors (Gilbert 2001). Optical sensors are also widely used and are available from vendors such as Hewlett-Packard. Their accuracy and convenience make them perfect for use with the CCC, in fact Monash already uses them for third year micro mouse projects.

Due to the availability of optical packages it is proposed that this project will use optical sensors for speed detection. Optical sensors are more suited to this application due to improved accuracy at all speeds over magnetic sensors, as well as smaller size.

References

Adams, J. (2000). Artificial vision is bringing robots out of the dark age, *Red Herring*.

Apostolopoulos, D., Wagner, M. and Whittaker, W. (1999). Technology and field demonstration results in the robotic search for antarctic meteorites, *Proceedings of the International Conference on Field and Service Robotics*, pp. 185–190.

Balakrishna, S. (2000). Pthreads. http://www.coe.neu.edu/ signoril/op_sys/pthreads1.html, Last Accessed: (2/7/2002).

Bertozzi, M. and Broggi, A. (1998). GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection, *Image processing*.

Bertozzi, M., Broggi, A. and Fascioli, A. (1998). An extension to the inverse perpective mapping to handle non-flat roads, *IEEE Intelligent Vehicles Symposiym*.

Boissonnat, J., Devillers, O. and Lazard, S. (1995). *Algorithmic foundations for robotics*, A K Peters Ltd., chapter Motion planning for legged robots.

Broggi, A. (1995a). A massively parallel approach to real-time vision-based road markings detection, *Proceedings ICIP – Second IEEE International Conference on Image Processing*, pp. 532–535. Washington DC.

Broggi, A. (1995b). Vision-based road detection in automotive systems: A real-time expectation-driven approach, *Journal of Artificial Intelligence Research 3* pp. 325–348.

Broggi, A. (1996). The evolution of a massively parallel vision system for real-time automotive image processing, *Proceedings IEEE IPPS'96 - International Parallel Processing Symposium*, pp. 724–728. Honolulu, HI.

Broggi, A., Bertozzi, M. and Fascioli, A. (1999a). The 2000km test of the ARGO vision-based autonomous vehicle, *IEEE Intelligent Systems*.

Broggi, A., Bertozzi, M., Fascioli, A. and Conti, G. (1999b). *Automatic vehicle guidance: Experience of the ARGO autonomous vehicle*, World Scientific.

Bruton, T. (1999). Computer controlled model car. Monash University, CSSE. Honours Thesis http://www.csse.monash.edu.au/hons/projects/1999/Tim.Bruton Last Acessed: (30/4/02).

Chatlia, R. (1995). *Algorithmic foundations for robotics*, A K Peters Ltd., chapter Mobile robot navigation algorithms, pp. 97–108.

Cox, I. and Wilfong, G. (eds) (1990). *Autonomous robot vehicles*, Springer-verlag.

Dickmanns, E., Behringer, R., Hildebrandt, D., Maurer, M., Thomanek, F. and Schiehlen, J. (1994). The seeing passenger car 'VaMoRs-P', *Proceedings of the IEEE Symposium on intelligent vehicles*, pp. 68–73. Paris.

Duggins, D., McNeil, S., Mertz, C., Thorpe, C. and Yata, T. (2001). Side collision warning systems for transit buses: Functional goals, *Technical Report CMU-RI-TR-01-11*, Carnegie Mellon University.

Franke, U. and Gavrilla, D. (1999). Autonomous driving goes downtown, *IEEE Intelligent Systems*.

Gilbert, D. (2001). Magneto-resistive wheel speed sensors, *AutoInc Magazine*.

Goldberg, K. (1995). *Algorithmic foundations for robotics*, A K Peters Ltd.

Graefe, V. and Kuhnert, K. (1991). Vision based autonomous road vehicles, *Vision based vehicle guidance* pp. 1–29.

Hough, P. (1962). *Method and means of recognising complex patterns.* USA Patent #3069654.

Jochem, T. and Baluja, S. (1993). A massively parallel road follower, *Proceedings of computer architechtures for machine perception*, pp. 2–12.

Jochem, T., Pomerleau, D. and Thorpe, C. (1993). MANIAC: A next generation neurally based autonomous road follower, *Proceedings of the International Conference on Intelligent Autonomous Systems*.

Kluge, K. and Thorpe, C. E. (1990). *Vision and navigation: The Carnegie Mellon NAVLAB*, Kluwer Academic Publishers, chapter Explicit models for robot road following, pp. 25–38.

Law, L. (1998). Magnetic sensors and timing applications, *Sensors Magazine*.

Leu, J. G. (1992). Boundary-based shape normalisation technique, *Intelligent robots and computer vision XI: Algorithms, techniques and active vision*, SPIE.

Martin, M. (1998). Breaking out of the black box: A new approach to robot perception. Doctoral dissertation, Carnegie Mellon University http://www.ri.cmu.edu/pub_files/pub2/martin_martin_c_1998_1/ martin_martin_c_1998_1.pdf Last Accessed: (31/7/2002).

Matthies, L. (2000). A portable, autonomous, urban reconnaissance robot, *6th international conference on intelligent autonomous systems*.

McDonald, J., Franz, J. and Shorten, R. (2001). Application of the hough tranform to lane following in motorway driving scenarios, *in* R. Shorten, T. Ward and T. Lysaght (eds), *Proceedings of the Irish Signals and Systems Conference*.

Muys, A. (2000). Pthread tutorial. http://www.cs.nmsu.edu/ jcook/Tools/pthreads/pthreads.html , Last Accessed: (2/7/2002).

Navarro-Serment, L., Grabowski, R., Paredis, C. and Kholsa, P. (1999). Modularity in small distributed robots, *Proceedings of the SPIE conference on Sensor Fusion and Decentralized Control in Robotic Systems II*.

Overmars, M. and Švestka, P. (1995). *Algorithmic foundations for robotics*, A K Peters Ltd., chapter A probabilistic learning approach to motion planning.

Overmars, M. and Švestka, P. (1996). A paradigm for probabilistic path planning, *Technical Report UU-CS-1995-22*, University Utrecht.

Pomerleau, D. (1992). Progress in neural network-based vision for autonomous robot driving, *Proceedings of the 1992 Intelligent Vehicles Conference*, pp. 391–396.

Pomerleau, D. (1995). RALPH: Rapidly adapting lateral position handler, *IEEE symposium on intelligent vehicles*, pp. 506–511.

Quenot, G., Pakleza, J. and Kowalewski, T. (1997). *Experiments in fluids*, Springer-Verlag, chapter Particle image velocimetry with optical flow, pp. 177–189.

Richardson, J., Ward, N., Fairclough, S. and Graham, R. (1996). Prometheus/drive aicc safety assessment: Basic simulator. DoT Vehicle Standards and Engineering Division Contract DPU 9/81/1 , HUSAT Research Institute, UK.

Sun, G. (1992). Improvement algorithm about corner detection in object feature extraction, *Intelligent robots and computer vision XI: Algorithms, techniques and active vision*, SPIE.

Taylor, C. (1995). *Algorithmic foundations for robotics*, A K Peters Ltd., chapter Vision-based motion planning and exploration algorithms for mobile robots, pp. 69–83.

Thorpe, C. (1990). *Vision and navigation: The Carnegie Mellon NAVLAB*, Kluwer Academic Publishers.

Thorpe, C. and Herbert, M. (1997). *Intelligent unmanned ground vehicles*, Kluwer Academic Publishers.

Thorpe, C., Duggins, D., Gowdy, J., MacLachlan, R., Mertz, C., Siegel, M., Suppe, A., Wang, C., and Yata, T. (2002). Driving in traffic: Short-range sensing for urban collision avoidance, *Proceedings of SPIE: Unmanned Ground Vehicle Technology IV, Vol. 4715.*

Tribe, R. (1996). *Advanced robotics and intelligent machines*, The Institution of Electrical Engineers, chapter Intelligent autonomous systems for cars, pp. 165–176.

Tung, D. (2000). Computer controlled model car. Monash University, CSSE. Honours Thesis http://www.csse.monash.edu.au/hons/projects/2000/Daniel.Tung Last Acessed: (30/4/02).

Tunstel, E., Huntsberger, T., Aghazarian, H., Backes, P., Baumgartner, E., Cheng, Y., Garrett, M., Kennedy, B., Leger, C., Magnone, L., Norris, J., Powell, M., Trebi-Ollennu, A. and Schenker, P. (2002). Fido rover field trials as rehearsal for the 2003 mars exploration rover mission, *9th Intl. Symp. on Robotics & Applications, 5th World Automation Congress.*

Wang, C. and Thorpe, C. (2002). Simultaneous localization and mapping with detection and tracking of moving objects, *IEEE International Conference on Robotics and Automation.*