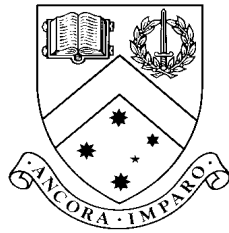


A smarter computer controlled model car

Brian W. Grasby

July 4, 2002



Monash University School of Computer Science and Software Engineering

Bachelor of Digital Systems (Hons), Clayton Campus

Research proposal - 2002

Supervisors Dr. Lloyd Allison, Dr. Ronald Pose

Contents

1	Introduction	3
2	Research context	4
2.1	Vision based CCC's	4
2.2	The CyCab	5
2.3	Road going vehicles	5
2.4	Other research vehicles	6
3	Research plan and methods	7
3.1	Construction goals	7
3.1.1	Resurrecting the car	7
3.1.2	Porting the hardware	8
3.1.3	Speed detection	8
3.1.4	Bringing the software up to date	8
3.1.5	Finalising	8
3.2	Research goals	9
4	Relevance	10

1 Introduction

The Computer Controlled Car (CCC) is a project that began its development at Monash University in 1999 as part of an honours thesis by Tim Bruton [2]. In simple terms, the CCC is a radio controlled (RC) model car with a small video camera mounted on the top. The car's handheld controller has been modified so that a computer can control the motion of the car, using the video output from the camera for guidance. Video signals are sent back to the host computer via a video transmitter. These signals are picked up by a video card in the host computer. The host computer takes the visual signal from the camera and performs image processing to locate borders of a track set out for the car, usually with parallel black fabric strips. Algorithms implemented in the control software guide the car around the identified track, attempting to stay within the borders. The purpose of the car is to be able to negotiate a complete lap of an arbitrary track laid out in the above manner as quickly as possible.

Initially Tim Bruton began development on the CCC from scratch for his thesis in 1999. He chose the current car and video setup and designed some good control software for use on a Silicon Graphics Indy machine. His setup comprised the car, a CAM-COL 554 miniature video camera and Arista AVS30 UHF transmitter, the car's radio controller with a custom circuit allowing a computer to control the transmitter, a VCR tuned to channel 30, and an Indy. The choice of the Indy as the host was an obvious one because Indy computers have a built in composite video input and video processing libraries. At that stage, video signals could be picked up with a standard VCR tuned appropriately and fed directly into the Indy.

Later the CCC was modified for a second honours thesis [12]. The main feature of those additions was the re-implementing of the host control software on an Intel PC platform, making the project more readily usable on any computer. The PC implementation required the addition of a video card; as PC's do not have the appropriate built in video capture hardware. The CCC itself was not changed greatly.

This project aims to further enhance the CCC by redesigning the existing hardware and re-implementing the system on a new, more advanced car, as well as adding additional features to the software as needed. Features that require implementation at this stage include support for proportional steering and speed control as well as speed detection. Other features for later consideration include obstacle detection and obstacle identification / avoidance. These features will be useful in building up more complex behaviour from the car, in an effort to reduce the lap times around any unknown track and make the CCC do new and useful things.

2 Research context

Development of autonomous vehicle control has long been a goal for researchers, visionaries and manufacturers alike [7] [3]. Realisation of commercial vehicle autonomy would (and perhaps will) be a revolutionary step that would certainly change both the car industry and the way humans travel forever. It would be possible to use vehicles that drive themselves in all the situations where we currently use manual vehicles, and many more. For example, autonomous cars could be used for mundane tasks such as taxi, courier and freight services, or for dangerous situations where humans cannot possibly fulfill the needs of the situation, or where the risk is too great, perhaps in fire fighting and so on.

Another issue is that cars need not drive themselves all of the time, but may have the capability to take control in case of driver negligence, incapacitation or inattention. The same technology developed for complete vehicle autonomy has implications even in current production cars, with many manufacturers wanting to be the first to come up with an automatic collision avoidance mechanism for public use.

There are several classes of Computer Controlled Car. Many current projects [8] [1] [10] investigate aspects of vision based navigation in CCC's, where as other of projects look at other ways, such as GPS, laser and radar. This project aims to continue and possibly improve the vision based system originally implemented in 1999, so most attention will be paid to vision based projects and research.

2.1 Vision based CCC's

Several groups are looking into the aspects of vision based CCC's. Some of these aspects include image processing, motion detection, speed detection, path tracking, object detection and object tracking to name a few. Research papers from Daimler-Chrysler [4] outline many more issues in the application of autonomous driving to urban environments. The vision system used by their prototype car has been proven to work very effectively to detect not just the road but also cars, obstacles, traffic signals and signs.

As the research here will be based on a platform that will never make it out onto the road, the control software will not have to be general enough to cater for as many cases (see Research plan) as a fully fledged highway vehicle. It will also have to be specific to the type of "road" that it will be used on, namely a track comprised of two parallel black lines, rather than a real road.

At the moment the Mercedes-Benz prototype [4] and other vehicles are not up to the state of development and sophistication where the level of assurance in the systems is high enough to warrant placing much human life in the hands of the computers. The systems in place would need to be

tried, tested and fool proofed before road-going cars were allowed to drive themselves. It will only be through experiment and testing that such reliable and trustworthy systems are developed and made available for commercial use.

2.2 The CyCab

In France the INRIA Institute has studied intelligent car concepts for the past decade, investigating several computer techniques to make them drive automatically. They have created the CyCab project¹, small 2-4 person electric vehicles limited to short distance runs at 30km/h. These CyCabs are to run in the car-free “cities of the future” as put by INRIA. The vehicles have been used to demonstrate the so-called platooning technique whereby cars can be ordered to automatically follow some lead vehicle with only a single human driver. This is interesting as it allows for the creation of virtual “trains” of cars all following one another automatically via the use of internal CCD cameras. While completely autonomous, the CyCab provides only the limited functionality of following other CyCabs and is not designed to work on roads with other cars.

2.3 Road going vehicles

Toyota, Jaguar and Nissan are all companies who have been using parts of this autonomous driving technology in their production vehicles, with Saab, Volvo, and Opel soon to follow. At this stage these manufacturers have taken the best parts that they can use (with current technology) and have implemented it as what is now being referred to in the industry as Adaptive Cruise Control (ACC). Using a variety of methods, including vision based systems, laser, radar etc., the autonomy is linked to the car’s cruise control so that, while enabled, the cruise control speed - which is already under the control of the car - is continuously changed to reflect the conditions of the road immediately ahead.

Techniques such as ACC focus on speed, but have no impact on the direction of the car. Currently Mercedes-Benz has done extensive research into autonomous vision based navigation [4] [5] for the commercial market. They have developed a prototype car based on one of their existing models, which is capable of tracking the car in front of it and matching speeds as well as driving behaviour with it and stopping to avoid obstacles such as pedestrians. Mercedes have integrated this system into one of their production cars simply as a prototype “driving aid” which amounts to an intelligent cruise control, dubbed “intelligent stop-and-go” by its creators. The system requires a lead vehicle to be detected and locked on to, which then dictates the motion of the following vehicle. The limitations of this system are that

¹<http://www-lara.inria.fr/cycaba/>

it would not be of any use where there are few or no other cars on the road to follow, such as in country areas of Australia. It is much more suited to the more crowded cities of Europe.

The open road is also an ideal testing ground for vehicles that are purely research focused, such as the Carnegie Mellon Robotics Institute's intelligent vehicle projects [11]. The researchers at Carnegie Mellon have developed a series of hardware and software projects that investigate many different aspects of "intelligent" vehicle control. Their vehicles are able to operate reliably in rugged environments using different techniques including vision based control. Unfortunately, they use an extremely wide variety of expensive components that are neither suitable nor relevant to this project, such as sonar and dedicated high speed computers running neural network controllers.

2.4 Other research vehicles

Many other projects are of interest in this research area, not only due to similarities in the hardware design and research goals, but also in terms of the approaches used for software design. One such project [9] at Monash University's Robotics and Mechatronics research laboratory investigated ways to simplify and therefore speed up the tracking algorithms for vision based autonomous vehicles. It worked by predicting the future path of the vehicle from the previous position frames and therefore searching less of each frame for a successful result.

It is important to recognise the achievements of Tim Bruton and Daniel Tung in their work to date on the CCC. Tim implemented the first ever version of the basis for this project and had the car detecting and navigating the track. The limits to his hardware are obvious from watching one of his trial runs, as the car came off on a tight corner, due to a lack of sharp steering ability, and had to find its way back. All of his goals were completed quite successfully and forms a solid basis to work from.

Daniel did extensive work into getting video hardware to work under Linux with the car and the results of his thesis provide a framework for coding the visual interface on a PC in Linux. With this in mind this project will not seek to reinvent the wheel where it is not necessary, as the foundations are quite solid and usable already. The aims of this project mandate changes that need to be made to these foundations in order to progress.

3 Research plan and methods

The current hardware platform has been carried over from 1999 and has various limitations. Primarily the drive system of the car is very coarse, only allowing the host PC a choice of full left turn, full right turn, reverse, and two forward speeds. Also, the video camera has a limited field of view, only a 90-degree field of vision in front of the car. Looking at a video taken from the perspective of the car in 1999 it can be seen that while the car does find its way around the circuit and even recover from a crash where it strays from the track, it still does not get around the track without incident. Finally the host has no way of determining the actual speed of the car at a given time, it can only make an estimate based on the current velocity command and the time elapsed since that command was given.

In terms of software, this project will endeavour to fit the existing code from previous theses [2] [12] to the new hardware and refine it where possible. The navigation system will remain true to the original intent of the project and as such will reuse whatever is available of the structure of previous code by Tim Bruton and Daniel Tung.

3.1 Construction goals

There are certain physical requirements that must be met if this project is to be successful and useful.

3.1.1 Resurrecting the car

The first goal before anything else is to get the current car working under some rudimentary computer control, most likely using the previous work of Daniel Tung and Tim Bruton. The car has not been in use for one year now and may have suffered damage as it has fallen into neglect. At this stage it has been determined that the car is able to move under manual control via the handset, but very quickly runs out of battery power. The transmitter has been tested and runs off external power but will not run off battery power. The camera does indeed return a video signal, but even under the best circumstances the signal is very noisy — this may indicate damage. The custom board that interfaces a PC's serial port to the handheld controller has not as yet returned any positive response to testing. Further investigation will show if there are parts that need to be replaced.

Once this setup has been verified to work, the software will be ported over to a new car. Under this control setup, the old detection and steering algorithms should still work and the results of previous work [2] [12] can be replicated. From this, parts such as the computer video input routines and manual keyboard steering functions can be reused or redesigned to fit in with new software components.

3.1.2 Porting the hardware

Therefore, the second major goal is to acquire a new car suitable for this project. The new car should have proportional steering and speed control, allowing a much better resolution for control and a wider range of control functions that can be implemented later in software.

3.1.3 Speed detection

Once the car is moving reliably it requires some form of speed detection, most likely to be implemented as a piece of hardware. Investigation will reveal the optimal method to fit in with the design of the car and provide a degree of accuracy that is useful. Jones [6] describes a solution to this which may be investigated further. Some proposed techniques for detecting speed include measuring the back EMF at the motor, counting the revolutions of an encoder disk and using software analysis of the video image to detect the rate of flow of the image and therefore the speed of the car. An efficient solution should be one which does not place additional load on the PC and still produces accurate results which are easily accessible.

3.1.4 Bringing the software up to date

The control software must then be extended to handle all of the new functions provided by the redesigned car. Also, the existing software may have to be tweaked to lower the tendency of the car to crash or stray from the course. This can be achieved through more complex image processing and analysis, but only at the cost of higher track identification latency and processing overheads. The way to combat this is to make the code as efficient as possible for the functions provided and only implement functions which are proven to be useful through testing. An important design issue is then to bring the individual parts together to create a “racing mode” whereby two identical cars controlled by similar software could be placed on the track and race against one another. Steps towards achieving this will include generalising the control algorithms to handle states involving another “competitor” car, recognition of this car and path planning to avoid or overtake the car. This might easily be developed as an add-on routine which may or may not be run when the car is on the track.

3.1.5 Finalising

As a final goal extra features can be built into the system to extend the control program’s ability to handle situations (albeit in a controlled environment). Functions such as static obstacle detection and avoidance, as well as further track generalisations may be investigated. These improvements however all come at a cost to the speed and efficiency of the system. It may

be discovered that there is an upper limit to the number of features that can be incorporated into the program, based on the processing power of the PC that is running the control software. Further aims of this project are to expose such limits in the system (see below).

These goals require investigation into their feasibility on the available hardware and results of implementation of different strategies and algorithms during the course of developing the final solution.

3.2 Research goals

There are philosophically two parts to this project. The first is the development and construction of an end product — in this case a model car. The second is the theoretical conclusions and observations that come along with having done the former. This research primarily intends to determine the optimal hardware and software configuration (out of what is available) for the model car to fulfill its purpose, to optimise this configuration (hardware and software) for maximum effectiveness and efficiency. Secondly to investigate the “keys” to smarter systems in relation to this project ie. more robust track recognition under a wider variety of circumstances and higher achievable speeds as well as better and faster error detection and correction or recovery. Hopefully some practical limits will be determined upon completion. It would be useful to determine the exact limit of a particular processor in terms of car speed, the limits of the available hardware in terms of features, speed and reliability, the limits of the software and how variables interact with the car’s environment as a whole.

4 Relevance

The relevance of this project is that it will provide a series of enhancements to the existing structure of the computer controlled car (CCC) developed at Monash for honours theses from 1999. These enhancements are targeted, and will be specifically designed, to meet the most serious flaws or shortcomings in the existing model. The final product should provide something that more closely models the way real cars drive on the road under human control and allow the computer controlling it to make more “intelligent” movements. It should make the overall unit do what it does better than before. It will also provide new modes of operation that will make the car do more useful and interesting things that may possibly be extended upon in the future.

While this area of research has implications in the fields of image processing and self-driving (autonomous) passenger vehicles, the specific aims of this project are limited to enhancing a track-bound model vehicle for the purposes of demonstration and later racing. It is still a good exercise in vehicle autonomy but not with any specific goals toward road-going autonomous cars.

References

- [1] M. Bertozzi and A. Broggi. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *Image processing*, 7(1), January 1998.
- [2] T. Bruton. Computer controlled model car, 1999. Monash University, CSSE.
<http://www.csse.monash.edu.au/hons/projects/1999/Tim.Bruton>
Last Accessed: (30/4/02).
- [3] A. L. Dorie. Computer controlled cars used for automated delivery, 1997. Stanford University.
<http://www-leland.stanford.edu/~tdorie/essay2.html> Last Accessed: (22/4/02).
- [4] U. Franke and D. Gavrilla et al. Autonomous driving goes downtown. *IEEE Intelligent Systems*, 13(6), 1999.
- [5] D. Gavrila. Computer vision: Autonomous driving, 2001.
http://www.gavrila.net/Computer_Vision/Smart_Vehicle/Media_Coverage/Autonomous_Driving/autonomous_driving.html
Last Accessed: (1/5/02).
- [6] J. Jones and A. Flynn. *Mobile robots: Inspiration to implementation*. Natick, 1999.

- [7] J. McCarthy. Computer controlled cars, 1996. Stanford University.
<http://www-formal.stanford.edu/jmc/progress/cars/cars.html> Last Accessed: (22/4/02).
- [8] H. Schneiderman and M. Nashman. Visual processing for autonomous driving. In *Proceedings of IEEE workshop on applications of computer vision*. IEEE, December 1992.
- [9] B. Shirinzadeh. Vision based tracking and control of remotely controlled vehicles. *Industrial Robot*, 24(4):297–301, 1997.
- [10] C. Thorpe. *Vision and navigation: the Carnegie Mellon Navlab*. Kluwer Academic Publishers, 1990.
- [11] C. Thorpe and M. Herbert. *Intelligent unmanned ground vehicles*. Kluwer Academic Publishers, 1997.
- [12] D. Tung. Computer controlled model car, 2000. Monash University, CSSE.
<http://www.csse.monash.edu.au/hons/projects/2000/Daniel.Tung>
Last Accessed: (30/4/02).