Vision Based Tactical Driving

Todd M. Jochem

CMU-RI-TR-96-14

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890

January 1996

© 1996 by Todd M. Jochem. All rights reserved.

This research was partly sponsored by DARPA, under contracts DACA76-89-C-0014 and DAAE07-90-C-R059 respectively monitored by the US Army Topographic Engineering Center and TACOM, and through a DARPA Research Assistantship in Parallel Processing administered by the Institute for Advanced Computer Studies, University of Maryland. Additional sponsorship was provided by USDOT under contracts DTNH22-93-C-07023 and DTFH61-94-X-00001. Finally the author would like to thank Delco Electronics for providing the testbed vehicle on which much of this work was conducted. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

Keywords: Mobile robots, autonomous driving, active vision, virtual cameras, lane transition, intersection navigation.

For Barbie, Mom, and Dad

Thanks for your love and support throughout the years.

You have blessed my life and made this possible.

With all my love, Todd.

Abstract

Much progress has been made toward solving the autonomous lane keeping problem using vision based methods. Systems have been demonstrated which can drive robot vehicles at high speeds for long distances. The current challenge for vision based on-road navigation researchers is to create systems that maintain the performance of the existing lane keeping systems, while adding the ability to execute tactical level driving tasks like lane transition and intersection detection and navigation.

There are many ways to add tactical functionality to a driving system. Solutions range from developing task specific software modules to grafting additional functionality onto a basic lane keeping system. Solutions like these are problematic because they either make reuse of acquired knowledge difficult or impossible, or preclude the use of alternative lane keeping systems.

A more desirable solution is to develop a robust, lane keeper independent control scheme that provides the functionality to execute tactical actions. Based on this hypothesis, techniques that are used to execute tactical level driving tasks should:

- Be based on a single framework that is applicable to a variety of tactical level actions,
- Be extensible to other vision based lane keeping systems, and
- Require little or no modification of the lane keeping system with which it is being used.

This thesis examines a framework, called Virtual Active Vision, which provides this functionality through intelligent control of the visual information presented to the lane keeping system. Novel solutions based on this framework for two classes of tactical driving tasks, lane transition and intersection detection and traversal, are presented in detail. Specifically, algorithms which allow the ALVINN lane keeping system to robustly execute lane transition maneuvers like lane changing, entrance and exit ramp detection and traversal, and obstacle avoidance are presented. Additionally, with the aid of active camera control, the ALVINN system enhanced with Virtual Active Vision tools can successfully detect and navigate basic road intersections.

Acknowledgments

I would like to thank my two advisors, Dr. Chuck Thorpe and Dr. Dean Pomerleau for their valuable guidance, support and friendship over the past 5+ years. Chuck and Dean have been involved in every aspect of my graduate career, from teaching me how to use email and C compilers to providing technical insight throughout this dissertation. I could not have asked for a better duo to guide me through my Ph.D.

I would also like the thank all of the members of the Navlab and VASC groups - there are too many to name individually, but their friendship has made my graduate career more than just an academic endeavor.

All of the good things that have happened in my life can in some way be attributed to the love, support, and encouragement my Mom and Dad have given me and the sacrifices they have made for me. I hope I can be as good a parent to my children as they have been to me.

I would like to thank my wife, Barbie, for the unconditional love and support she has given me. She has always sacrificed so that I could do what I thought necessary to complete this degree. My life would not be complete without her. I love you, Barbie.

Finally, I would like to thank God for watching over me, for all my family and friends, for my Mom and Dad, and for my wonderful Wife. I feel as if my life has been blessed.

Todd Jochem

May 15, 1996

Table of Contents

Chapter 1 Introduction 1
1.1 Thesis Overview 1
1.2 Thesis Motivation 1
1.2.1 Implementing Tactical Functionality 2
1.3 Thesis Statement 4
1.4 ALVINN Overview 5
1.4.1 ALVINN Confidence Estimation 7
1.4.2 Towards a Comprehensive System 7
1.4.3 Integrating ALVINN 8
1.5 The Virtual Camera 10
1.6 System Overview and Comparison 12
1.6.1 The Old ALVINN System 12
1.6.2 The Current ALVINN System 13
1.7 Other Similar Systems 14
1.8 Research Overview 17
1.8.1 Improving Performance Through Optimal View Selection 18
1.8.2 Lane Transition 18
1.8.3 Intersection Detection and Traversal 19
1.8.4 Other Similar Systems 20
1.8.5 Vehicle Development 20

Chapter 2 Optimal Camera Placement 21
2.1 Introduction 21
2.2 ALVINN View Selection 21
2.3 Experimental Procedure 23
2.4 Paved Path Experimental Results 24
2.5 Highway Experimental Results 27
2.6 Discussion 32

Chapter 3 Lane Transition 33
3.1 Introduction 33
3.2 Road Model 34
3.3 Other Work 34
3.4 Lane Change 36
3.4.1 Control Scheme 36

3.4.2 Lane Transition by Network Switching 37
3.4.3 Incremental Network Switching 37
3.4.4 Incremental View Lane Transition 43
3.4.5 Dual View Lane Transition 49
3.4.6 Lane Transition Experimental Parameters and Procedures 56
3.5 Lateral Offset Driving 57
3.5.1 Swerve Experimental Results 58
3.5.2 Offset Driving Experimental Results 59
3.6 Exit Ramp Detection and Traversal 62
3.6.1 Exit Ramp Detection and Traversal Experimental Results 64
3.6.2 Mainline Detection and Traversal 66
3.7 Lane Detection 68

Chapter 4 Intersection Navigation 71
4.1 Introduction 71
4.2 Detection Philosophy 71
4.2.1 Data vs. Model Driven Detection 72
4.2.2 Detection Signaling 73
4.2.3 Active Search 73
4.2.4 ALVINN 74
4.3 Other Systems 74
4.3.1 ALV Project Intersection Detection 74
4.3.2 SCARF 75
4.3.3 YARF 75
4.3.4 Old ALVINN 76
4.3.5 VaMoRs Intersection Detection and Traversal 76
4.3.6 IITB Intersection Detection and Traversal 76
4.3.7 Intersection Detection for the Driver's Warning Assistant 77
4.4 Experimental Overview 77
4.5 Intersection Detection and Traversal Experiments on the Navlab 2 78
4.5.1 Road Detection 78
4.5.2 Application of IRRE to Road Detection 79
4.5.3 Accuracy of Approach to Road 84
4.5.4 Alignment 85
4.5.4.1 Simple Road Alignment 85
4.5.4.2 Alignment by Projecting Along Road 87
4.5.4.3 Image-based Alignment and Traversal Using Active Camera Control 89
4.5.5 Initial "Y" Intersection Detection Experiments 90
4.6 Intersection Detection and Traversal Experiments on the Navlab 5 91
4.6.1 Detection with Known Geometry and Unknown Location 92
4.6.2 Detection with Unknown Geometry and Known Location 94
4.6.3 Intersection Localization and Traversal using Active Camera Control 100
4.6.3.1 Preliminary Lateral Offset Adjustment 101
4.6.3.2 Alignment to the Intersection Branch 101
4.6.3.3 Traversal 105

4.7 Results and Discussion 112

Chapter 5 Other Systems 115
5.1 Introduction...115
5.2 ROBIN 115
5.3 RALPH 118
5.4 Discussion 120

Chapter 6 The Testbed Vehicle 123
6.1 Introduction...123
6.2 Motivation...123
6.3 PANS Overview 124
6.4 Position Estimation 129
6.4.1 Global and Local Positioning..130
6.4.2 X Y Position...131
6.4.3 Heading 132
6.4.4 Turn Radius...132
6.4.5 Velocity 133

Chapter 7 Conclusions..135
7.1 Summary 135
7.2 Contributions 135
7.3 Future Work 138

Chapter 8 References..141

List of Figures

Figure 1 ALVINN network architecture. 6
Figure 2 Relationship between turn radius, displacement, and lookahead distance. 9
Figure 3 Typical virtual camera scenes. 11
Figure 4 ALVINN system diagrams. 12
Figure 5 Information and control flow in a typical robotic system. 13
Figure 6 Detecting exit ramps. 19
Figure 7 Intersection navigation. 20
Figure 8 Paved path test road. 24
Figure 9 Peak-to-peak error for different virtual camera locations on the paved path. 25
Figure 10 Virtual vs. actual error metrics 26
Figure 11 Right lane of the rural interstate test road. 27
Figure 12 Peak-to-peak error for different virtual camera locations on the interstate. 28
Figure 13 Average error and confidence values from 30 degree HFOV virtual cameras. 29
Figure 14 Highway scene after exit lane appeared. 30
Figure 15 Combined error graphs for 30 and 50 degree HFOV. 31
Figure 16 INS processing 39
Figure 17 Average vehicle path during Incremental Network Switch Lane Transition. 40
Figure 18 IRRE values during INS Lane Transitions. 42
Figure 19 Driving and Test Views for a right to left lane transition 45
Figure 20 IVLT processing. 46
Figure 21 Average vehicle path during Incremental View Lane Transitions. 47
Figure 22 IRRE values during IVLT's. 48
Figure 23 Initial virtual view placement for the DVLT technique. 50
Figure 24 Movement of the vehicle and the views in the DVLT method. 51
Figure 25 DVLT processing. 52
Figure 26 Average vehicle path during Dual View Lane Transitions. 54
Figure 27 IRRE values during DVLT's. 55
Figure 28 Pennsylvania Turnpike Commission STAR Site near Breezewood, PA. 57
Figure 29 Vehicle movement while avoiding an obstacle. 59
Figure 30 Vehicle and virtual camera locations during SODA maneuver. 60
Figure 31 In-vehicle view of SODA maneuver. 61
Figure 32 Exit ramp detection diagram. 63
Figure 33 Exit ramp detection images. 65
Figure 34 Entrance ramp detection images. 67
Figure 35 Lane detection scenario. 70
Figure 36 Road detection scenario. 79
Figure 37 Virtual cameras imaging an upcoming road. 80

Figure 38 Virtual camera spacing. 81
Figure 39 IRRE response to different camera images. 82
Figure 40 IRRE values for typical detection images. 84
Figure 41 Transformed IRRE response curves. 85
Figure 42 Vehicle to road misalignment. 86
Figure 43 P2 point projection diagram. 87
Figure 44 “Y” intersection geometry. 90
Figure 45 Camera on pan-tilt mount. 91
Figure 46 Intersection branch detection from a moving vehicle. 93
Figure 47 Hypothesis camera views. 95
Figure 48 Searching for “Y” intersection branches. 96
Figure 49 Searching for “T” intersection branches. 97
Figure 50 Images and IRRE values for the “Y” intersection. 98
Figure 51 Images and IRRE values for the “T” intersection. 99
Figure 52 Lateral and angular branch localization. 102
Figure 53 Possible alignments with zero output displacement. 103
Figure 54 Orientation localization. 104
Figure 55 Orientation and lateral offset error correction. 106
Figure 56 Lateral misalignment after rotation. 107
Figure 57 Initial “Y” intersection traversal images. 108
Figure 58 Subsequent “Y” intersection traversal images. 109
Figure 59 Initial “T” intersection traversal images. 110
Figure 60 Subsequent “T” intersection traversal images. 111
Figure 61 Traversal turn radius determination. 113
Figure 62 ROBIN network architecture. 117
Figure 63 Ralph display. 119
Figure 64 The Navlab 5, a 1990 Pontiac Trans Sport. 125
Figure 65 PANS components inside the Navlab 5. 127
Figure 66 Two runs on the 100 km course. 130
Figure 67 Start/end points for the two 100 km runs. 132

Chapter 1 Introduction

1.1 Thesis Overview

This dissertation documents the results of an investigation into executing tactical level driving tasks using a vision based lane keeping system. Algorithms to accomplish lane transition, exit and entrance ramp detection and traversal, obstacle avoidance maneuvers, and intersection navigation were developed and tested on real roadways using Carnegie Mellon University's autonomous testbed vehicle, a Pontiac Trans Sport minivan called the Navlab 5.

Each of the algorithms use a new framework for extracting relevant regions in the road scene. The selection of scene regions is driven by the geometric constraints of the task being executed as well as by the requirements of the lane keeping system being used. This framework, called Virtual Active Vision, does not compromise the performance or flexibility of the underlying lane keeping system, yet adds the capability to execute the tactical driving tasks mentioned above.

1.2 Thesis Motivation

The field of autonomous driving has matured to the point where there are many competent systems which can perform parts of the problem very robustly and reliably. Systems have been developed using a wide range of techniques which can drive autonomous vehicles on roads, avoid obstacles, maintain safe headway, and provide global, map-related information to other complementary systems.

A key component of any complete driving system is a robust lane keeping module. Lane keeping modules have been demonstrated which can drive robot vehicles at high speeds for long distances. These systems are based on varying methods. Some use road models, which incorporate knowledge about important road features like road color and markings, to determine where the road [6][21][54][58][59] or edge/lane markings [8][16][25][27][29][35][53][57][60] are located. Others are adaptive and do not require an a priori road model. These systems learn the salient features required for driving on a particular road type [20][40][44]. Still others fall somewhere in between [1][26][42].

The current challenge for vision-based on-road navigation researchers is to create systems that maintain the performance of the existing lane keeping systems, while adding the ability to execute tactical driving tasks like lane changing and intersection detection and navigation. Additionally, the existing lane keeping systems should be used, if possible, to provide vehicle environment information such as the current road type and the number and location of lanes, to tactical level decision modules [TLDMs] for situational reasoning.

Tactical actions like lane changing and intersection navigation fall between low level operations like lane keeping and obstacle detection, and strategic functions like goal selection and route planning. Tactical actions are assuming increased importance for several reasons. First, low level reactive systems such as lane keepers, have matured to a level where tactical control algorithms, previously demonstrable only in simulation, can be realistically tested on vehicles which operate in the real world. Second, tactical reasoning algorithms have improved to the point where characterization of and operation in the real world is feasible. Finally, specifications and concepts of new intelligent vehicle programs in Europe, Asia, and the United States require the ability to execute tactical actions.

1.2.1 Implementing Tactical Functionality

One way to add tactical functionality is to create new, independent, software modules which are designed to perform tasks requested by or provide information to TLDMs. These independent modules are typically tailored to one particular task, like intersection detection or lane changing [46]. They may or may not use information, like road position, provided to them by other modules which could help them execute their task. Because they operate independently of the lane keeping module, a separate arbiter must be used to merge information with that of the lane keeping module. Also, adding modules increases the complexity of the system and leads to difficulty in debugging. Finally, because the lane keeping and task specific modules are not combined, information generated by one may be of little use to the other.

Another way to add tactical level capabilities is to add functionality to the lane keeping system itself. This idea overcomes many of the drawbacks associated with using a separate module to service requests of TLDMs. No arbitration is necessary and decisions are made using common information. But because the tactical level functionality is tailored to working with a single lane keeping system, it may not be easily extensible for use with others [28][40]. This is especially true when trying to adapt techniques from systems which are strictly model based to ones which are not, and vice versa. Finally, adding functionality may decrease the performance of the lane keeping system, because the requirements of the tactical task may not match those established by the system's primary function, lane keeping.

A more desirable solution is to develop a robust, lane keeper independent, control scheme that provides the functionality to execute tactical actions and concurrently deliver relevant information to TLDMs. Based on this hypothesis, techniques that are used to execute tactical driving tasks and provide information to TLDMs should:

1. Be based on a single framework that is applicable to a variety of tactical level actions,
2. Provide TLDMs with a common interface to the lane keeping module,
3. Be extensible to other vision-based lane keeping systems, and
4. Require little or no modification of the lane keeping system with which it is being used.

This thesis examines a way to provide this functionality through intelligent control of the visual information presented to the lane keeping system. Specifically, the techniques described in the following chapters use the inherent ability of a neural network lane keeping system to perform tactical level driving tasks. In addition, because the techniques are grounded geometrically, application of them to other systems is less difficult.

The lane keeping system that was used for all of the experiments described in this thesis was ALVINN (Autonomous Land Vehicle In a Neural Network). ALVINN is based on an artificial neural network that is trained to take a subsampled, preprocessed video image as input, and produce the appropriate steering wheel position as output. ALVINN has demonstrated robust lane keeping performance in a wide variety of situations, but is generally limited to this task due to its lack of geometric models that are typically required to execute tactical actions. Grafting geometric reasoning onto a non-geometric base would be difficult and would dilute ALVINN's capabilities. A much better approach is to leave the basic neural network intact, preserving its real-time performance and generalization capabilities, and to apply geometric transformations to the input image and the output steering vector. These transformations form a new set of tools and techniques called **Virtual Active Vision**. **Virtual** because all the techniques use artificially created imaging sensors which can be robustly manipulated to suit the needs of the task and **Active** because the techniques move sensors to locations where the images they create will enhance system performance.

This thesis focuses on several different tactical level driving tasks relating to lane transition and intersection navigation and demonstrates the applicability of the ALVINN lane keeping system, enhanced with Virtual Active Vision tools, to them. While development was done in a simulated environment, demonstration of and experimentation using these capabilities was completed on real roads and highways using Carnegie Mellon University's testbed vehicles.

1.3 Thesis Statement

The goal of this research was to develop methods that allowed reliable and robust performance in complex, real world driving tasks. Such tasks inherently require information from high level, goal directed modules as well as stable performance from low level, reactive control systems.

This goal was met through the development of techniques which provide a framework for executing tactical driving tasks as well as by merging tactical knowledge sources with low level, reactive modules. This merger can be bidirectional, with ALVINN providing feedback to high level modules about how it is currently performing. Information contained in TLDMs that is required for goal directed behavior is easily accessible to ALVINN. For these reasons, autonomous operation in new domains was possible.

In this thesis, emphasis was placed on developing a suite of techniques, based on Virtual Active Vision tools, which can be used by TLDMs to execute tactical level driving tasks. Developing these TLDMs was not investigated and interfacing with existing TLDMs proved infeasible because of differing research agendas and constraints. For most of the experiments presented in this document, the safety driver served as the TLDM, instructing the low level driving system to execute specific tasks. To reiterate, this thesis focused on developing techniques for autonomous vehicle control in tactical driving scenarios. These methods were based on Virtual Active Vision Tools and this thesis will show that:

> **Virtual Active Vision tools, when applied to systems for vision based lane keeping, expand the capabilities of these systems without compromising performance, to also include lane changing, intersection navigation, and other tactical driving functions.**

1.4 ALVINN Overview

ALVINN has shown that neural techniques hold much promise for the field of autonomous road following. Using simple color image preprocessing to create a grayscale input image and a 3 layer neural network architecture consisting of 960 input units, 4 hidden units, and 30 steering output units, ALVINN can quickly learn, using back-propagation, the correct mapping from input image to output steering direction. See Figure 1. This steering direction is then used to control the testbed vehicle. (See Chapter 6 for details about the vehicle used for most of the experiments presented in this dissertation.)

ALVINN has many characteristics which make it desirable as a robust lane keeping system. They include:

- ALVINN learns the features that are required for driving on the particular road type for which it is trained.
- ALVINN can estimate its own confidence.

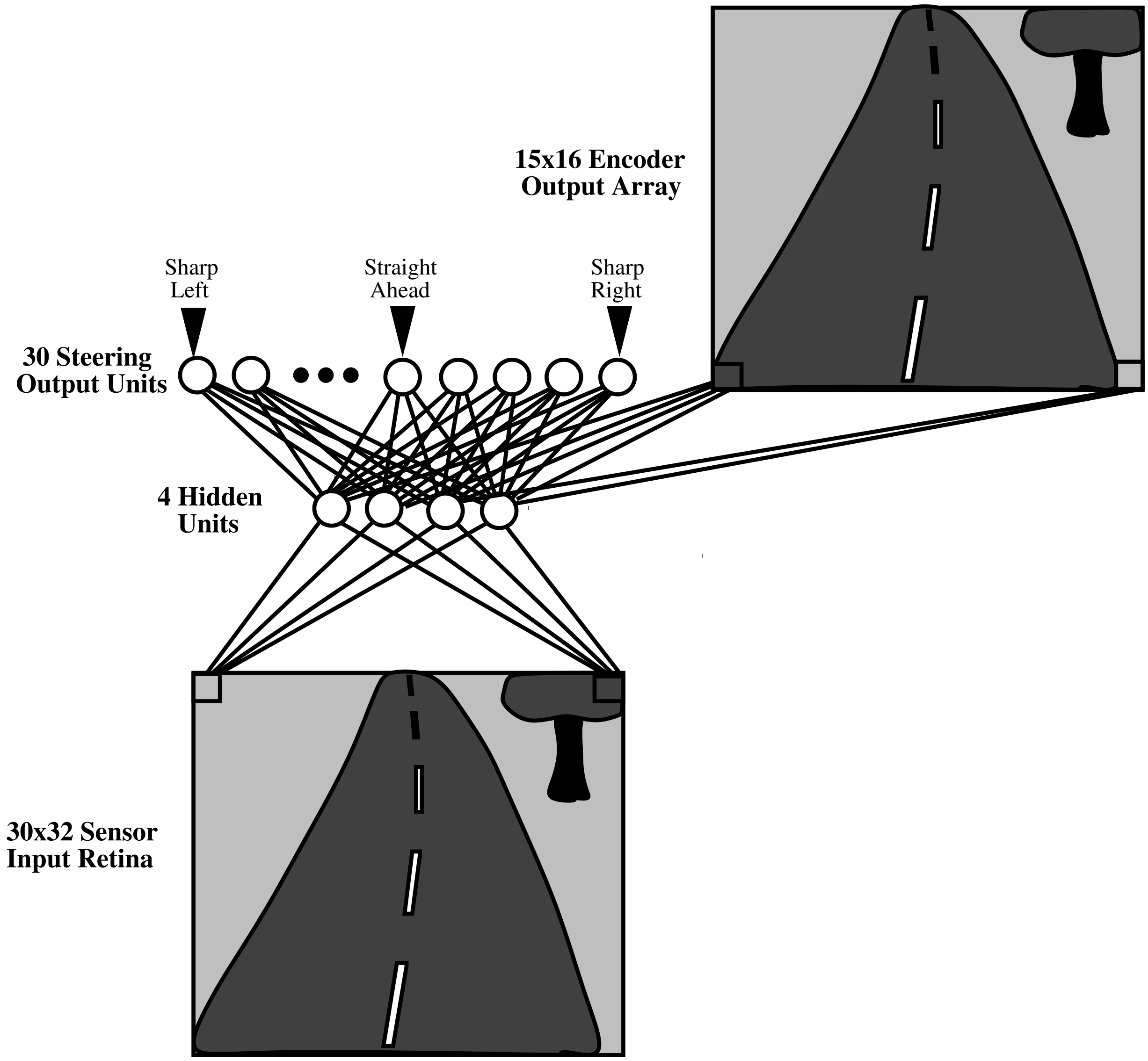


Figure 1 ALVINN network architecture.

- ALVINN is computationally simple.
- ALVINN has been shown to work in a variety of situations.
- ALVINN learns feature detectors that are intuitively plausible when viewed by a person.

ALVINN can learn to drive on lined city streets, jeep trails, and interstate highways and has successfully driven one of our testbed vehicles for over 90 continuous miles at speeds exceeding 50 miles per hour. Because it has proven to be reliable over a wide range of road types and because it uses very basic input to produce its output, it is a natural choice on which to build more elaborate and comprehensive autonomous driving systems.

1.4.1 ALVINN Confidence Estimation

ALVINN's ability to estimate its own confidence is a characteristic that is used in many of the experiments presented in this dissertation. The confidence metric that is used is called Input Reconstruction Reliability Estimation - IRRE. IRRE is a measure of the familiarity of the input image to the neural network. In IRRE, the network's internal representation is used to reconstruct the input pattern being presented. The more closely the reconstructed input matches the actual input, the more familiar the input and hence the more reliable the network's response.

IRRE utilizes an additional set of output units to perform input reconstruction, as depicted in Figure 1. This second set of output units is half the size of the input retina - 15 rows by 16 columns. The desired activation for each of these additional output units is the average of the activation on four corresponding input units. For example, IRRE unit (0,0) contains the average activation of input units (0,0), (0,1), (1,0) and (1,1). In essence, these additional output units turn the network into an autoencoder.

The network is trained using backpropagation both to produce the correct steering response on the steering output units, and to reconstruct the input image as accurately as possible on the reconstruction outputs.

During testing, images are presented to the network and activation is propagated forward through the network to produce a steering response and a reconstructed input image. The reliability of the steering response is estimated by computing the correlation coefficient between the activation levels of units in the actual input image and the reconstructed input image. The higher the correlation between the two images, the more reliable the network's steering response is estimated to be. A more complete description of IRRE, as well as the ALVINN system in general, can be found in [40].

1.4.2 Towards a Comprehensive System

Previously, ALVINN aided TLDMs in determining appropriate vehicle action in a very minimal way. Its primary function, besides staying on the road, was to change networks when the TLDM requested. Some other module was responsible for determining when intersections and other road transitions occurred, determining which pre-trained ALVINN network was required for the new road type, and relaying this information to ALVINN. ALVINN provided little [40][52] or no [39] feedback to the TLDM about where it thought the road transition was or how it was performing. This should not be the case. Knowledge from TLDMs should be used to guide, rather than force, lower level modules in the execution of driving tasks as well as in their search for information relevant to satisfactory completion of high level goals.

To aid in implementation of this idea, a new sensor called a virtual camera was developed. Virtual cameras can be used to enhance the performance of existing vision based lane keeping systems as well as add new, more advanced, functionality to these systems. This is possible because virtual cameras image relevant areas of the scene in a way that is usable to the lane keeping system. In essence, they force the lane keeping system to attend to areas in the image which are known to contain information, like road features, that is required for successful completion of the task at hand. Virtual cameras are the fundamental tool upon which all other Virtual Active Vision tools and techniques presented in this dissertation are based and create a natural link to TLDMs and the important information that they contain. They provide a mechanism for determining appropriate vehicle action and create a framework in which the execution of tactical driving tasks is possible.

For ALVINN, virtual cameras are used to create images which resemble those that ALVINN's neural network was trained on and to which it can properly respond. They do not compromise the robust driving performance of the original ALVINN system, and add to its capabilities by facilitating the execution of tactical driving tasks like lane transition and intersection navigation.

1.4.3 Integrating ALVINN

In order to more tightly integrate virtual cameras into a high performance driving system, one change to the basic ALVINN system was needed. This change was moving away from training ALVINN to output a driving arc or steering wheel position to having it learn to produce the location of the center of the road or driving lane at a pre-specified distance in front of the vehicle. In effect, the new system produces a point to drive over rather than an arc to drive. By using this point-based approach, ALVINN and TLDMs have a common reference frame in which to communicate.

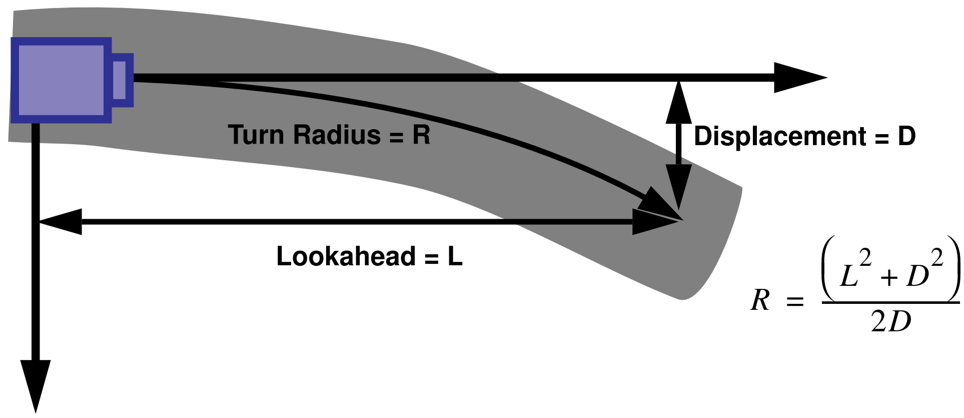


Figure 2 Relationship between turn radius, displacement, and lookahead distance.

This change has a minor affect on the training procedure. As before, ALVINN is trained by watching a person drive for about 5 minutes. Training is still done on-the-fly, with shifted and rotated images (which are created using virtual camera views) used to improve driving ability by supplementing the untransformed image in the training set [37]. Instead of tagging each image with the current turn radius of vehicle, it is tagged with the displacement from straight ahead at the lookahead distance. This change is actually just another representation of the circular steering arc used as the training signal in the original system. Figure 2 shows the relationship between these three parameters. Although the difference between turn radius and displacement at the lookahead distance can be viewed as just a different interpretation of ALVINN's output, using a two dimensional representation like displacement at the lookahead distance simplifies the way driving tasks can be formalized. For the lane transition experiments presented later in this dissertation, the network was trained with a lookahead value of 35 meters, while for intersection navigation experiments, the lookahead was reduced to 15 meters.

Geometry can be thought of as a common language, thus it is much more useful when aspiring toward a robust, integrated, driving system, to express the location of the road directly in cartesian coordinates as opposed specifying how the vehicle should steer. Finally, this representational change means that ALVINN's output and the location of virtual cameras are expressed in the same manner - geometrically.

1.5 The Virtual Camera

A virtual camera is simply an artificial imaging sensor which can be placed at any location and orientation in the world reference frame. It creates images using actual pixels imaged by a real camera that have been projected onto some world model. By knowing the location of both the actual and virtual camera, and by assuming a flat world model, accurate image reconstructions can be created from the virtual camera location. A flat world model has been chosen as a first approximation of the actual world because in many driving scenarios, it accurately represents the world near the vehicle. Virtual camera views from many orientations have been created using images from several different actual cameras. The images produced by these views have proven to be both accurate and usable by the ALVINN system to navigate successfully. Figure 3 shows some typical virtual camera views and the images that they create.

A general theme of this thesis is showing that virtual cameras allow systems with weak geometric models, like ALVINN, to accomplish traditionally geometrically grounded tasks. Virtual cameras can do this because they impose a model on the system. In our case, the model is not a feature in an image, but rather a canonical image viewpoint which ALVINN can interpret. This idea can be better understood by looking at a virtual camera from the point of view of both the TLDM, which has a specific task to be accomplished, and of ALVINN.

From the standpoint of the TLDM, a virtual camera isn't a camera, or even a sensing device, but instead a geometric tool that ALVINN uses to get its job done. The single most important characteristic of a virtual camera to the TLDM is that it can be placed at an arbitrary location in the real world. Location is a concrete concept which many TLDMs use as an integral part of their operation. Location can be relative ("past the grocery store") as well as absolute ("sail to 10 latitude and 5 longitude"), and need not necessarily be specified by a number ("at the grocery store"). These geometric concepts are all ones which tactical level systems handle well.

To ALVINN, the virtual camera is a sensing device. It is ALVINN's only link to the world in which it operates. ALVINN doesn't care where the virtual camera is located, only that it is producing images which can be used to locate the road. This interpretation may seem to trivialize ALVINN's functionality, but in reality, finding the road is what ALVINN is designed to do best.

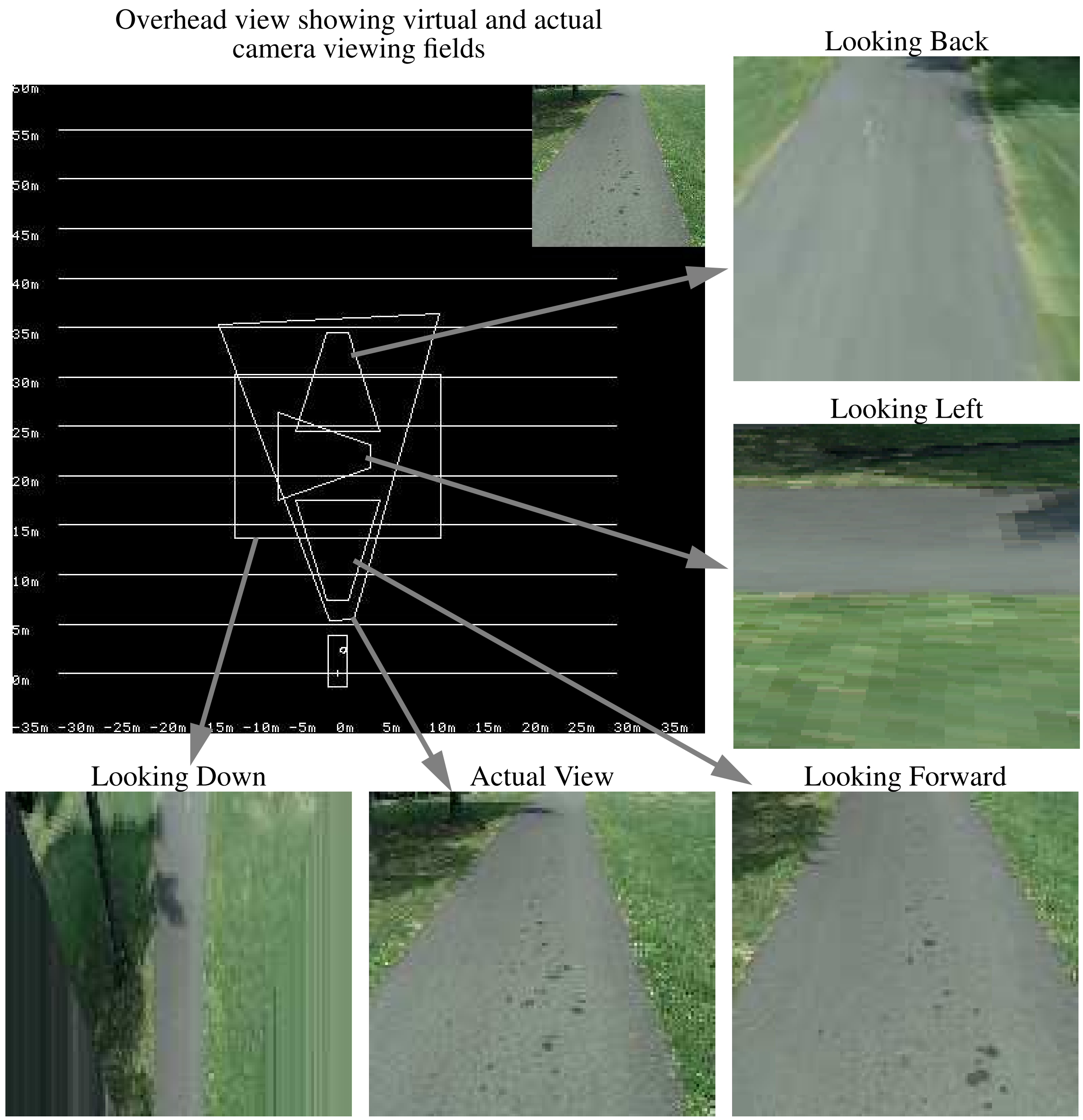


Figure 3 Typical virtual camera scenes.

The virtual camera, guided by tactical level control, insures that ALVINN gets images which will let it do its job to the best of its ability.

So in essence, the virtual camera imposes a geometric model on the neural system without the neural system knowing, or even caring, about it. The model helps both the tactical level system as well as the lane keeping system do their respective jobs better and allows them to seamlessly cooperate to exhibit goal directed behavior.

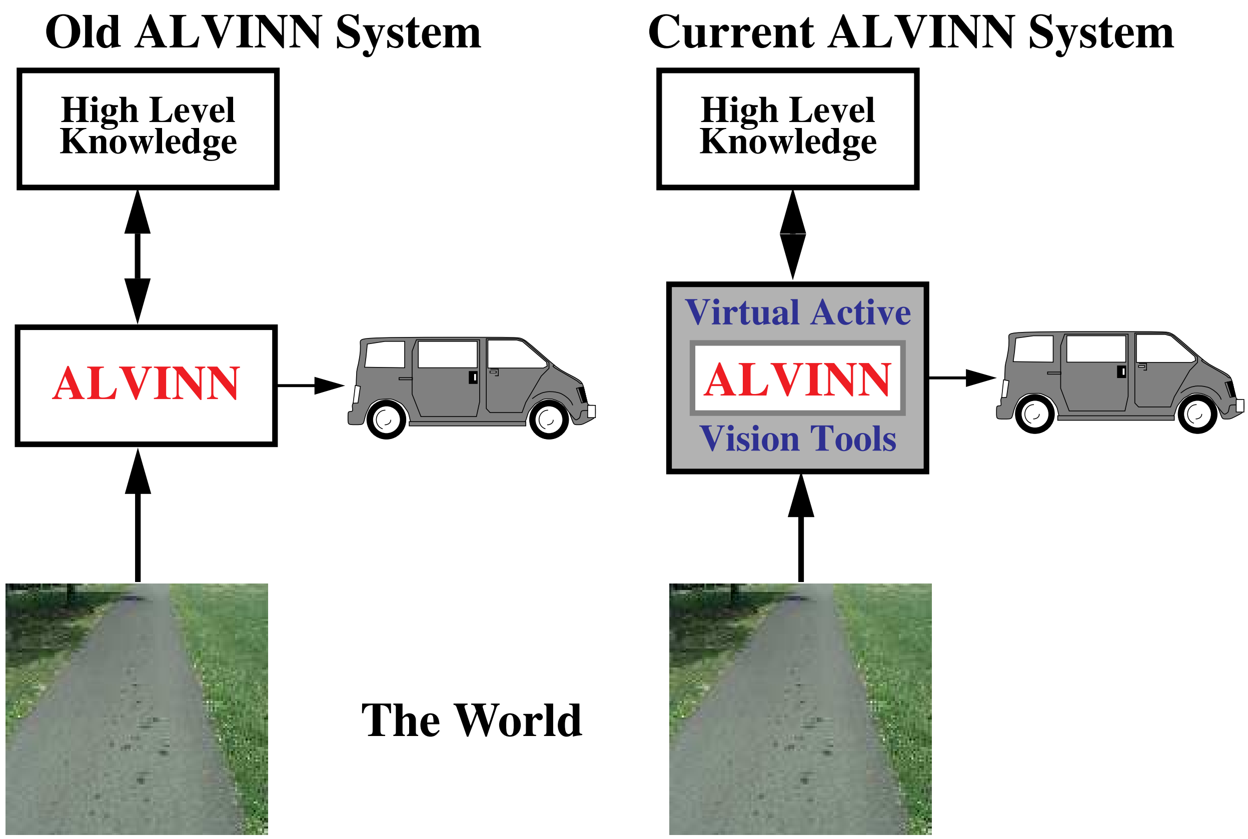


Figure 4 ALVINN system diagrams.

1.6 System Overview and Comparison

A graphical comparison between the old ALVINN system and the one used for the experiments described in this thesis is shown in Figure 4. The main differences that should be noted are:

- ALVINN is coupled with the outside world in the current system by Virtual Active Vision tools.
- There is the potential for more information flow from ALVINN to TLDMs in the current system.

These differences and the reasons for them are described in the next paragraphs.

1.6.1 The Old ALVINN System

The old ALVINN system was a closed system. ALVINN directly sensed the world and sent steering commands to the vehicle controller. Although this closed loop was necessary to achieve real-time performance, it made interacting with ALVINN difficult. As a consequence, providing

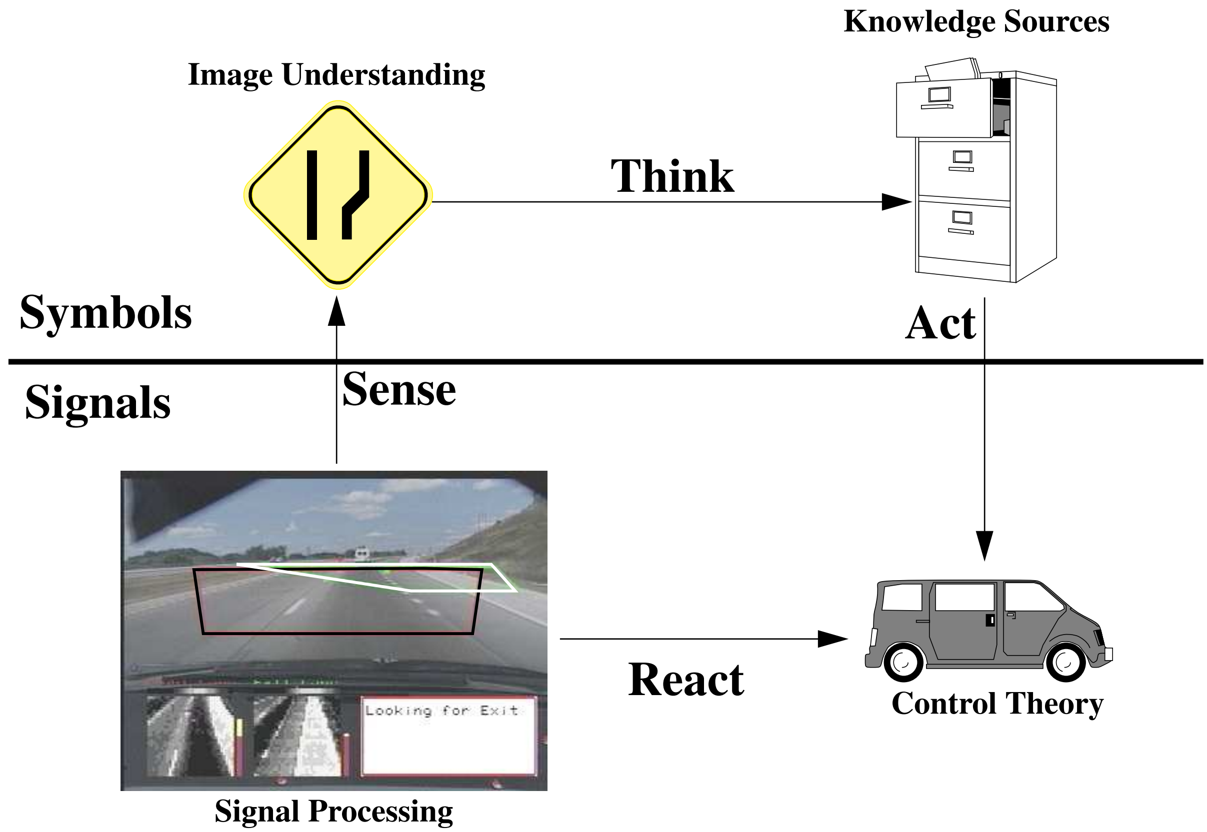


Figure 5 Information and control flow in a typical robotic system.

ALVINN with information that allowed it to exhibit intelligent, goal directed behavior was done in a somewhat ad-hoc manner. New ideas about integration with other systems and improving performance were tacked onto the sense-react cycle instead of integrated with it. Extensions to ALVINN were mostly based on interpreting its output in a different manner. Different components were added which allowed ALVINN to do things like detect simple “Y” intersections and keep the camera it used pointed at the road during sharp turning maneuvers [39][52]. The techniques developed as part of this thesis research allow ALVINN to do all of these things, and more, using a single, consistent framework for interacting with the world.

1.6.2 The Current ALVINN System

Figure 5 shows a typical robotic system. In most systems, there are low level modules (shown with dashed borders in Figure 5), like ALVINN, which interact with the world. This type of module is responsible for sensing the current world state and determining appropriate robot action from this state. There are other modules (shown with solid borders in the figure), which provide feedback and guidance about how the robot is performing. In order for the robot to exhibit goal directed behavior, these disparate modules must work together. A common language at the interface is necessary for interaction to take place. Virtual Active Vision tools sit squarely on this interface between signals and symbols, as well as between ALVINN and the world, and provide a geometrically based mechanism for inter-module communication.

In the ALVINN system used for the experiments described in this dissertation, the tightly closed sense-react loop is maintained, but isolated from the world and TLDMs by Virtual Active Vision tools. See Figure 4. These tools allow ALVINN and other high level systems to work in the common language specified earlier - geometry. Data mapping problems associated with heterogeneous systems are overcome using this language. Virtual Active Vision tools still allow ALVINN to perform in real-time, but serve as a useful, bidirectional gateway to high level systems and the knowledge they contain. They allow ALVINN to focus its attention on the tasks which are critical for achieving goal directed behavior while still maintaining the robust performance of the original ALVINN system. Using Virtual Active Vision tools, all of the tasks that the old ALVINN system could perform can be accomplished within a single framework, as opposed to using a collection of add-on components [20][39][40].

1.7 Other Similar Systems

Other mobile robot researchers have experimented with altering the image, or image derivatives, to make navigation tasks easier. These systems were able to perform the functions for which they were designed, but none were tested in complex, real world scenarios or attempted to address as wide a variety of tasks as the work described in this dissertation. These systems do share the philosophy that navigation tasks can be made easier by manipulating the image using the available constraints. In addition to the systems presented in the following paragraphs, Chapter 5 describes in more detail two recently developed systems which use many of the tools and concepts developed in this dissertation.

An early attempt to use an artificial viewpoint to more easily interface low level robot control with high level functionality was reported by Wallace [58]. This work used a local flat world assumption to create a bird's-eye view (looking straight down) of the area in front of a mobile robot. From this viewpoint, the actual road structure became explicit, and when road edges were extracted, they could be easily matched to an actual map of the area.

More recently, Meng and Kak [32] have presented work on a system for an indoor mobile robot that couples low level, neural network based navigation systems with higher level, semantically based planning. Their system uses neural networks to identify when landmarks such as hallway junctions and dead ends are reached, as well as to navigate correctly down the center of the corridor. They use parts of a Hough transform created from an edge image of the current scene as a training signal for each of the networks. The part of the Hough transform which each network sees is determined a priori by the researchers and corresponds to where relevant features in the Hough space are likely to appear. Each of their landmark detecting networks is trained to create a "near," "far," or "at" output. These outputs are then directly fed to a planner so that the robot can localize itself and replan if necessary. An interesting point about this work is that absolute position is never used; only relative, semantic positioning ("near the junction"). The claim is that humans use only relative positioning to navigate, but by giving up absolute locational information, some tasks may become more difficult, if not impossible.

Work done by Lotufo [31] uses a bird's-eye view to simplify processing which must be done to detect road edges. In this work, an input image of a road is transformed onto the ground plane where specialized vertical edge detectors are applied. Because the road edges in the transformed image appear to be parallel vertical lines, the edge detector can quickly and accurately find them. This work was later expanded to explicitly model curving roads, which had previously caused difficulties because the edge lines in the overhead image of the curved road were no longer vertical. This situation was handled by rotating the overhead view, based on earlier road curvature estimates, so that it aligned with the road orientation and again imaged the road edges as vertical lines [5].

Like the work described in the last paragraph, the lane keeping system which drives the MOB-LAB vehicle reprojects the image into a bird's-eye view using a flat world model. This system uses a massively parallel processor to achieve 17Hz cycle rates on 128x128 input images. This system uses a simple template matching scheme, along with image enhancement algorithms, to locate white lines on the black pavement. The system has successfully driven the MOB-LAB vehicle at speeds up to 50 k.p.h in a number of different road conditions [4].

Work being done at Universitat der Bundeswehr Munchen illustrates an imaging framework similar to Virtual Active Vision. In this work, a detailed spatio-temporal model of where lane edges are located is used to determine which areas in the image should be attended to and sampled. For example, if the task being executed is lane keeping and a lane edge was found at a particular location in the image, the model would predict where in the next image a sampling window should be placed to image the same edge. This system uses many imaging and control techniques which are similar to the Virtual Active Vision ideas presented in this dissertation. Like virtual cameras, the imaging framework in this work allows for identical image processing techniques to be used to detect feature at different locations in the input image. In addition, a separate control algorithm determines which areas of the scene contain information necessary for executing the current task and instructs the imaging portion of the system to sample from those areas. The principle difference between this work and Virtual Active Vision tools is that the imaging system is closely tied to the model-based lane keeping system with which it is being used. Although this system has been used to execute several tactical tasks, like lane changing and intersection navigation, task specific modules, in addition to the basic lane keeping system, have been required to do each [9][33].

A stereo-based system which uses image reprojection into a canonical viewpoint to accomplish obstacle detection for on-road navigation is VisionBumper. In this system, differences between two stereo images, after they have been mapped using a flat world model onto a new camera viewpoint, are used to estimate the location of obstacles in the scene. Because both stereo images are mapped to the same viewpoint, any differences between them must be a result of a violation of the flat world projection model. For most on-road driving scenarios, the violation is a result of a vehicle or other obstacle. The viewpoint used for detection is changed based on the area in the scene which is appropriate for the current driving speed and task being executed. VisionBumper has been used to detect obstacles at distances of up to 35 meters ahead of the vehicle and at speeds of up to 50 km/hr [3].

The original ALVINN system used a form of Virtual Active Vision to improve its robustness and generalization capabilities. Transformed images in which the vehicle appeared shifted from the road center or rotated from the proper direction of travel were created from the original input image during ALVINN's training phase. These transformed images were necessary for two reason. First, because the person driving the vehicle always stayed on the road, ALVINN never saw how to properly recover from driving mistakes. Second, if the section of road on which ALVINN was trained did not contain a particular road geometry, like left turns, it would not be able to handle that geometry when it was encountered during testing. The transformed images simulated both of these conditions and allowed ALVINN to learn to properly respond to them when they were experienced during testing [37]. In the current ALVINN system, this functionality is maintained using virtual views to created shifted and rotated images.

An early attempt to improve ALVINN's driving performance and capabilities by forcing ALVINN to attend to relevant scene locations was the Panacea system [52]. Panacea provided a mechanism for coupling ALVINN's output, which is an indication of the position of the road ahead, to a moveable camera platform. For each image, Panacea examined ALVINN's output and moved the platform so that the camera pointed at the center of the road. This was beneficial for at least two reasons. First, by imaging the center of the road, the important scene features required for driving remained in view. Second, because images with the road centered in the field of view look the most like those seen during training, ALVINN's output accuracy increased. For these reasons ALVINN's performance improved, especially on winding roads. Because the camera platform was only panned though, Panacea was still restricted to creating images from the actual camera viewpoint. This limited the tasks that could accomplished using it in conjunction with ALVINN.

1.8 Research Overview

The goal of the research described in this thesis is to develop methods that will allow autonomous vehicles to exhibit reliable and robust performance for tactical driving tasks in complex, real world driving scenarios. The work to accomplish this goal falls in the following areas:

- Improving Driving Performance

- Lane Transition
- Intersection Navigation
- Vehicle Development

The following sections briefly describe each of the subsequent chapters.

1.8.1 Improving Performance Through Optimal View Selection

One way that virtual cameras can improve ALVINN's driving performance is that they allow ALVINN's neural network to be trained on a camera view from which it can learn the image to road location mapping most quickly and accurately. Consider the case when the actual camera is oriented in such a way that the horizon line is in the image. Parts of the image which would ideally contain useful information about the road, now contain data which is not necessary for lane keeping, such as trees and sky lines. This extra, and often contradictory, information makes it more difficult for ALVINN to learn the correct mapping. By using a virtual camera, it is possible to create a view which contains only information that is useful for driving. Chapter 2 describes two sets of experiments to determine the optimal camera view for driving on a two lane interstate highway as well as on a single lane, paved bike path.

1.8.2 Lane Transition

Chapter 3 describes several tasks which require moving the vehicle from one lane to another. The most basic of these is lane changing. Three techniques are described which transition the vehicle between lanes. These techniques are based on geometrically interpreting the output of ALVINN's networks and for two of the methods, controlling the placement of virtual camera views based on this geometric interpretation. Another similar task is exit ramp detection and traversal. A technique is described which allows ALVINN to find and take exit ramps like those found on interstate highways. ALVINN is supplied with general information about where the exit ramp is located (like within the next 1/4 mile) along with other relevant information like the network to use to detect the ramp. ALVINN then uses this information to create a virtual camera view that is positioned so that it will image the area to the right of the interstate - where exit ramps usually occur. At the same time, another virtual camera is used to keep the vehicle on the road until the exit ramp is found. See Figure 6. With an approach such as this, ALVINN can simulta-

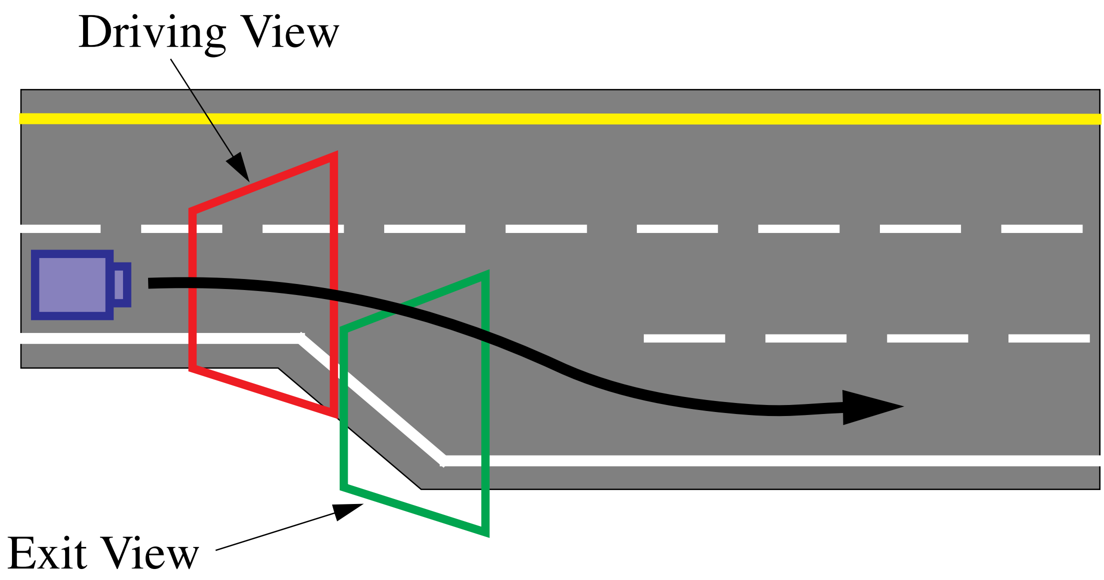


Figure 6 Detecting exit ramps.

neously follow the road as well as carry out goal directed behavior requested by TLDMs. Once the exit ramp has been found, virtual cameras are used to move the vehicle off the interstate and onto the exit lane. Additionally, the exit's location can be refined in the TLDM's map for more precise specification later.

1.8.3 Intersection Detection and Traversal

Chapter 4 presents another example where virtual cameras can be used to allow ALVINN to exhibit goal directed behavior. The scenario is that of detecting and navigating through road intersections. For this scenario, ALVINN enhanced with virtual cameras, is used to detect and traverse intersections with known location and unknown geometry, and those with known geometry and unknown location. By placing virtual cameras in a manner similar to that shown in Figure 7, this task can be completed. The Virtual Active Vision techniques presented made it possible to navigate through intersections without having to rely solely on blind, dead reckoning methods. Because a single camera was not sufficient to image the entire intersection, traditional active vision techniques were integrated.

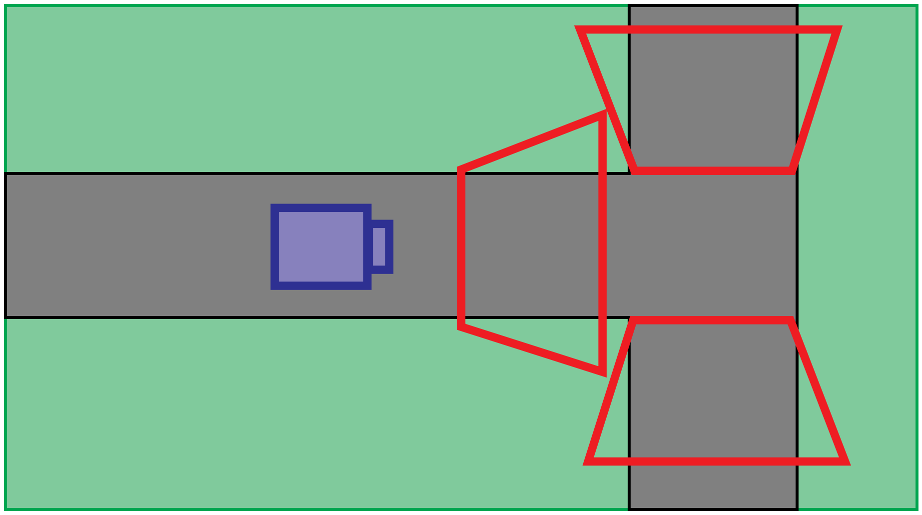


Figure 7 Intersection navigation.

1.8.4 Other Similar Systems

Chapter 5 describes two additional vision based lane keeping systems which not only use some of the tools described in this dissertation, but also contain new ones. The systems are compared to each other, as well as to ALVINN, and common characteristics are discussed. Based on this comparison, as well as other non-navigation examples, the chapter concludes with a discussion of the exact nature of Virtual Active Vision tools.

1.8.5 Vehicle Development

In parallel to the research into the algorithms which allow tactical driving tasks to be executed, work was done on developing a new testbed vehicle, tailored for on-road navigation experiments. The previous testbed vehicle, a converted U.S. Army HMMWV, was useful for early research into on-road driving, but its complexity and size soon became a limiting factor for conducting extensive experiments. With this in mind, a new testbed vehicle called Navlab 5 was developed. The Navlab 5 is the primary vehicle on which the experiments presented in this dissertation were conducted. The main components on the vehicle are a Sparc laptop computer with video digitizer, a color LCD monitor, a color camera which can be mounted on a pan-tilt platform, a GPS, a fiber optic gyroscope, and a steering motor. Chapter 6 describes each of these components, as well as the safety hardware, in greater detail.

Chapter 2 Optimal Camera Placement

2.1 Introduction

A fundamental characteristic of all Virtual Active Vision tools is that they focus the attention of the system which is using them on a specific area of the image. The primary application of this ability that has been explored in this dissertation is enabling the execution of tactical driving tasks. But to reach this goal, basic lane keeping performance should not be sacrificed. This chapter presents a series of experimental results which suggest that, at a minimum, virtual cameras do not impact the basic lane keeping performance of the ALVINN system. Additionally, the results show that for some virtual camera views, lane keeping performance improves.

2.2 ALVINN View Selection

In order to reliably drive, ALVINN must learn which image features are correlated with proper driving and how changes in the image location of these features relate to changes in road geometry. Virtual cameras can facilitate this process by imaging only areas of the scene which contain relevant information. Spurious features, like trees, other cars, the dashboard or the roof, which are not related to driving, but that often appear in images, can be removed. In this way, ALVINN can concentrate on learning only those features which relate to the driving task.

It is possible to find the optimal virtual camera view by training several networks, each using images created by different virtual views. Virtual camera pose parameters which can be modified are the lateral offset of the virtual camera from the center line of the vehicle (x), the longitudinal offset from the center of the vehicle's rear axle (y), the virtual camera's height (z), the virtual camera's pitch, the virtual camera's yaw, and the virtual camera's horizontal and vertical fields of view.

After a network has been trained, the mean error can be computed by comparing human driving performance with the output of the network. The virtual camera view which has the lowest mean error is judged as optimal for driving on the current road type. It cannot be judged as optimal for all road types because the location of important features for driving may change if road types change. For example, on an unlined bike path, it is reasonable to assume that the optimal camera offset is zero (i.e. centered) because both edges of the road are important for driving. But for driving on a lined city street, it may be more important to keep the yellow center line in the field of view at all times while sacrificing some of the outside road edge.

Additionally, selecting an optimal camera view is closely related to eliminating the need for adding structured noise to images during ALVINN training [38]. (Structured noise refers to spatially coherent features in the image which should not be, but often are, mistaken for the real features which are required for driving.) Adding structured noise is required for driving in situations where transitory features could be mistaken for the actual features required for correct driving. For example, suppose a network is trained to drive on a typical four lane highway, having broad shoulders and a grassy median. During testing, it is observed that the network becomes confused when driving over a bridge that has concrete jersey barriers at the road side. Even though the lines marking the lanes (the real features required for correct driving) do not change, system performance degrades. To prevent this degradation, structured noise is added during training to the areas of the image where this type of transitory feature is likely to occur. By adding noise in these areas, the network will learn to ignore those areas and only key off the important features required for driving. An alternative solution to this problem using virtual cameras is to pick a virtual camera view which does not include the area where transitory features occur.

2.3 Experimental Procedure

The experimental results presented in Section 2.4 and Section 2.5 were both conducted in simulation in the laboratory using image sequences taken from real roads. For each experiment, two image sequences were collected while the vehicle was driving on the specified road type. One of these sequences served as the training sequence while the other was used for testing. After the sequences were collected, the center of the road or driving lane was hand labeled in each image.

For the single lane road experiments, the camera was mounted 2.05 meters high, 1.23 meters forward from the rear axle, and tilted downward at an angle of 11.0 degrees. For the highway experiments, it was located 1.36 meters high, 2.11 meters forward from the rear axle, and tilted downward at an angle of 10.6 degrees. Both cameras were very close to the centerline of the vehicle (x = 0.0 meters) and were facing straight ahead in the direction of vehicle travel (yaw = 0.0 degrees).

For each experiment, many virtual camera views were created by adjusting their x, y, z, and pitch parameters across a range of reasonable values. In addition for the highway experiment, two horizontal fields of view - 30 and 50 degrees - were tested for each virtual camera pose. For each view, ALVINN's neural network was trained on the same images. After the network had been trained, the learned weights were saved for the virtual view pose and the test sequence was loaded. Using the same view and the saved weights, the system computed three error metrics for each virtual camera image in the test sequence. The most important of these metrics is the peak-to-peak error. This metric represents the difference at the lookahead distance between ALVINN's estimate of the road center location and the hand labeled road center location. The other metrics which were computed were the IRRE confidence value and the sum-squared error between the actual and target output displacement vectors. The IRRE confidence value is a measure of the image's familiarity to the network while the sum-squared error is a traditional neural network error criteria.

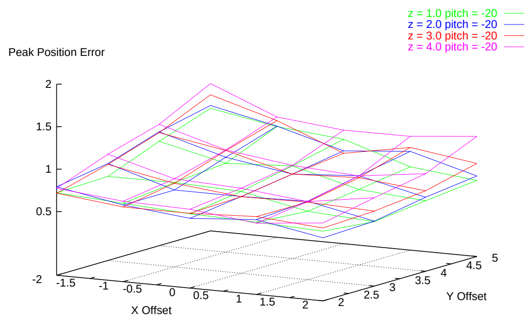
Figure 8 Paved path test road.

2.4 Paved Path Experimental Results

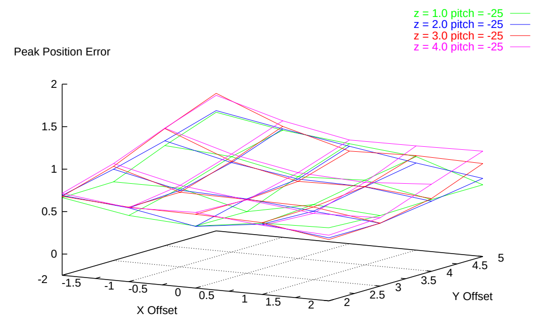
This set of experiments used images from a single lane, paved path near the Carnegie Mellon campus. See Figure 8. The path is 3.1 meters wide and has no lane markings. Additionally, only one small intersection is located along this path.

The summarized results of this set of experiments are shown in Figure 9. Each plot in Figure 9 represents a different virtual camera pitch angle and shows the composite output peak-to-peak errors at different x, y, and z virtual camera locations for the specified pitch angle. Specifically, each surface in a plot represents virtual camera locations which have the same z value. The x, y coordinates of a point on that surface represents the x, y location of the virtual camera view while the z value of the surface at that point represents that peak-to-peak error. The x, y, and z values are all in meters and measured from the origin of the vehicle coordinate system.

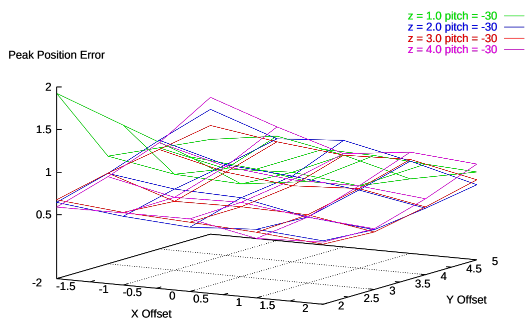
The most important characteristic to notice about these plots is that there is no distinctly best virtual camera location. The surfaces smoothly vary and the difference in error between surfaces is also very small. This means that there are many locations which should yield adequate driving performance. Another characteristic that is fairly evident is that in nearly every case, the peak-to-



Pitch Angle: -20



Pitch Angle: -25



Pitch Angle: -30

Figure 9 Peak-to-peak error for different virtual camera locations on the paved path.

peak error slowly increases as the virtual camera location moves farther away from the vehicle. A likely explanation for this effect is inconsistent road geometry between the local road segment and the portion of the road that the virtual view is imaging. This leads to contradictory information in the training set.

For example, if the vehicle is on a long straight section of road during training, using a view which is farther in front of the vehicle will have little negative effect. This is because the training driver's steering wheel position will still be correct - the road geometry is consistent. But when the vehicle approaches a curve, the distant virtual camera will image it before the safety driver actually begins turning. In this case, the steering wheel direction is still straight ahead, but the image created by the virtual view shows a curving road.

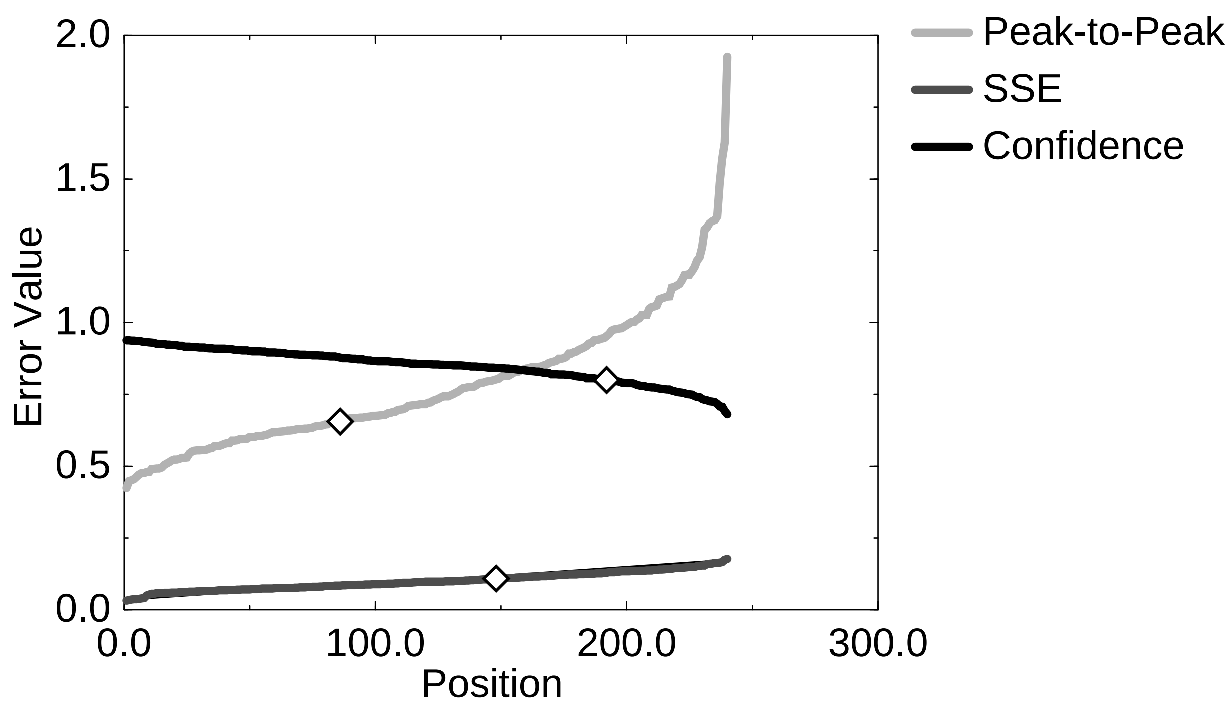


Figure 10 Virtual vs. actual error metrics

The last subtle characteristic about many of the curves is that the peak-to-peak error forms a slight valley around which the error slowly increases. This is intuitively plausible and is supported by previous results which show that the features the network learns to use for driving on this type of road are the symmetric road edges and road area [41]. This suggests that for this particular road type, the virtual camera view should be centered around these features.

In addition to showing that many virtual camera positions can yield good results, it would be useful to determine if any of the virtual camera locations create images which yield better driving performance than ones created from the entire actual camera image. (The original ALVINN system used the whole image during training and testing.) Each line in Figure 10 shows the profile of values for one of the error parameters. The values have been sorted so that better values are at earlier positions in the graph. The diamonds on each graph represent where each metric from the actual camera location image falls. In each case, the actual camera view does not yield the best results. In two of the cases, the metrics for the actual camera location are not even in the top half of all values.

Figure 11 Right lane of the rural interstate test road.

Although statistically this may seem significant, care must be taken when trying to draw conclusions about expected driving performance from a particular view. In actuality, except for some of the extremes, most of the views would yield performance that would not be noticeably different to the safety driver. This is supported by examining the peak-to-peak error curve. Although the actual camera position on this curve is not in the top third of all views, the actual distance error between the best view and the actual driving view is only about 5 cm.

2.5 Highway Experimental Results

This experiment was conducted in the right lane of a rural interstate highway outside of Pittsburgh, PA. See Figure 11. (The dark regions in the upper left and right corners of the image are caused by the shade placed in front of the camera to shield it from direct sunlight.) The lane was 3.65 meters wide and had typical highway markings - a solid yellow line on the left edge of the left lane, a broken white dashed line between lanes, and a solid white line on the right edge of the right lane. On this section of road several exits were present.

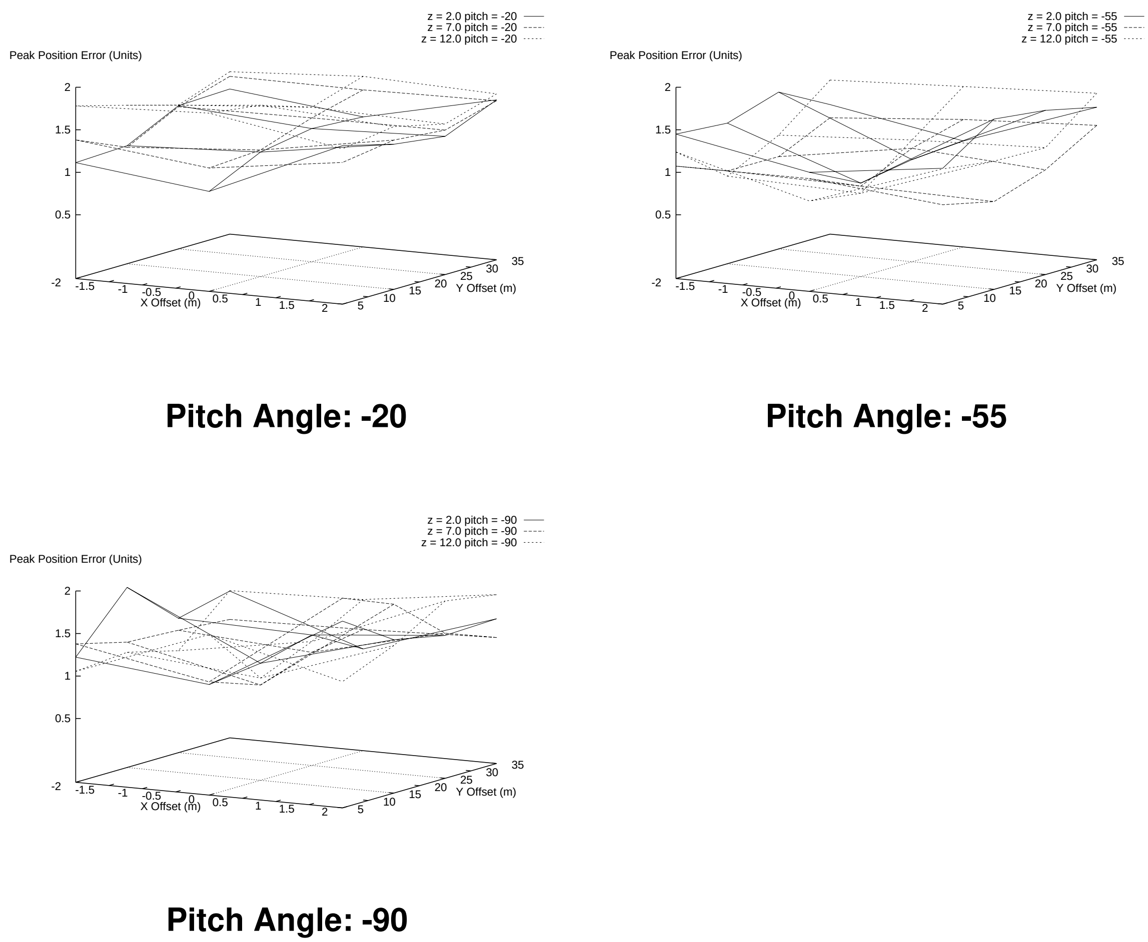


Figure 12 Peak-to-peak error for different virtual camera locations on the interstate.

The results of this experiment are shown in Figure 9. As with the paved path experiments, no best view is evident from these summarized results. Also like in the paved path experimental results, there is a slight upward trend in the peak-to-peak error as the virtual view moves further away from the vehicle, and a slight valley in the peak-to-peak error surface as the virtual view becomes centered in front of the vehicle.

More interesting insight about optimal virtual view placement for this road type can be obtained by comparing the peak-to-peak error and confidence values produced by networks trained using virtual cameras having 30 and 50 degree horizontal fields of view. In order to determine the full effect of the different fields of view, the testing sequence must be analyzed on a per image instead of per sequence basis. This means that for a particular image in the testing

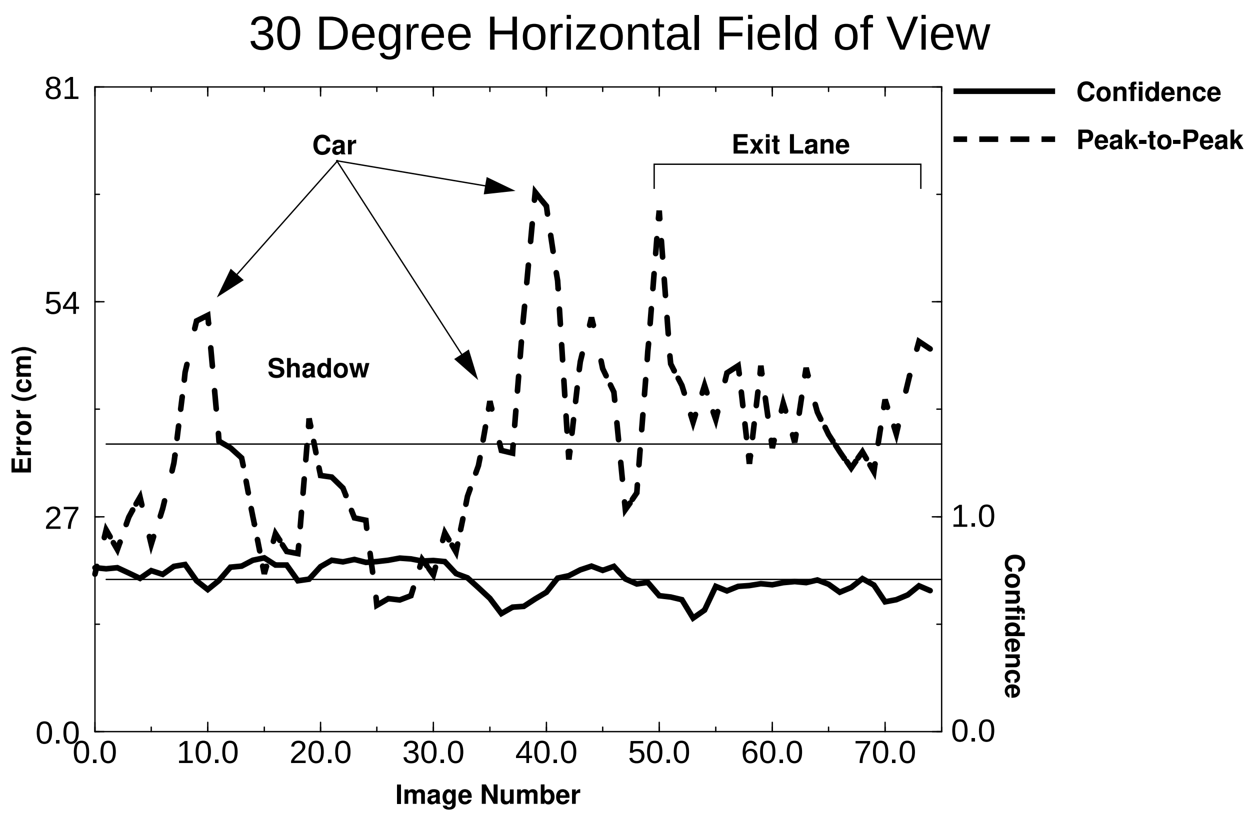


Figure 13 Average error and confidence values from 30 degree HFOV virtual cameras.

sequence, the peak-to-peak error and confidence value is computed by averaging the response of all virtual camera views on that image which have the specified HFOV. These average values are graphed in Figure 13 and Figure 15.

Figure 13, shows the average confidence and peak-to-peak error values for all tested virtual views with a 30 degree horizontal field of view on each image in the test sequence. Also shown, as the two straight lines, is the average value of the peak-to-peak error and confidence value graphs.

In Figure 13, upward spikes in the peak-to-peak error graph are marked with the cause of the degraded performance. The primary cause of the largest upward spikes in the error graph is vehicles in adjacent lanes entering the field of view of the virtual camera. In these cases, after the vehicle passes, the error value quickly returns to normal. But for images 50 to 75, the error value stays consistently high. This would indicate a feature which is not transient, like the vehicle. In fact, in these images, an exit lane had appeared to the right of the driving lane. See Figure 14. Because the

Figure 14 Highway scene after exit lane appeared.

lane the vehicle was traveling in was no longer the right lane of the highway, some of the features the network had learned to use for driving were no longer present, or at a different location in the image, and system performance dropped. In this case, the problem was that the network had learned to rely on features which were outside the typical boundaries of the driving lane. These features were scene elements like the far left road to shoulder boundary and the right shoulder to ground boundary.

Because allowing the network to learn to key off features which are not strictly tied to proper driving can be problematic when those features change, it is beneficial to develop a method to focus the network's resource only on those consistent features which will always delineate the proper driving lane. One method to do this is to add structured noise [38] to the input image. Another is to use previously consistent feature to determine what features the network should rely on[1]. The technique explored in this dissertation to alleviate this problem was to use a a virtual camera view which images only the driving lane and therefore eliminates spurious features from the network's input. The effect of trying to limit the imaging area of a virtual camera to the driv-

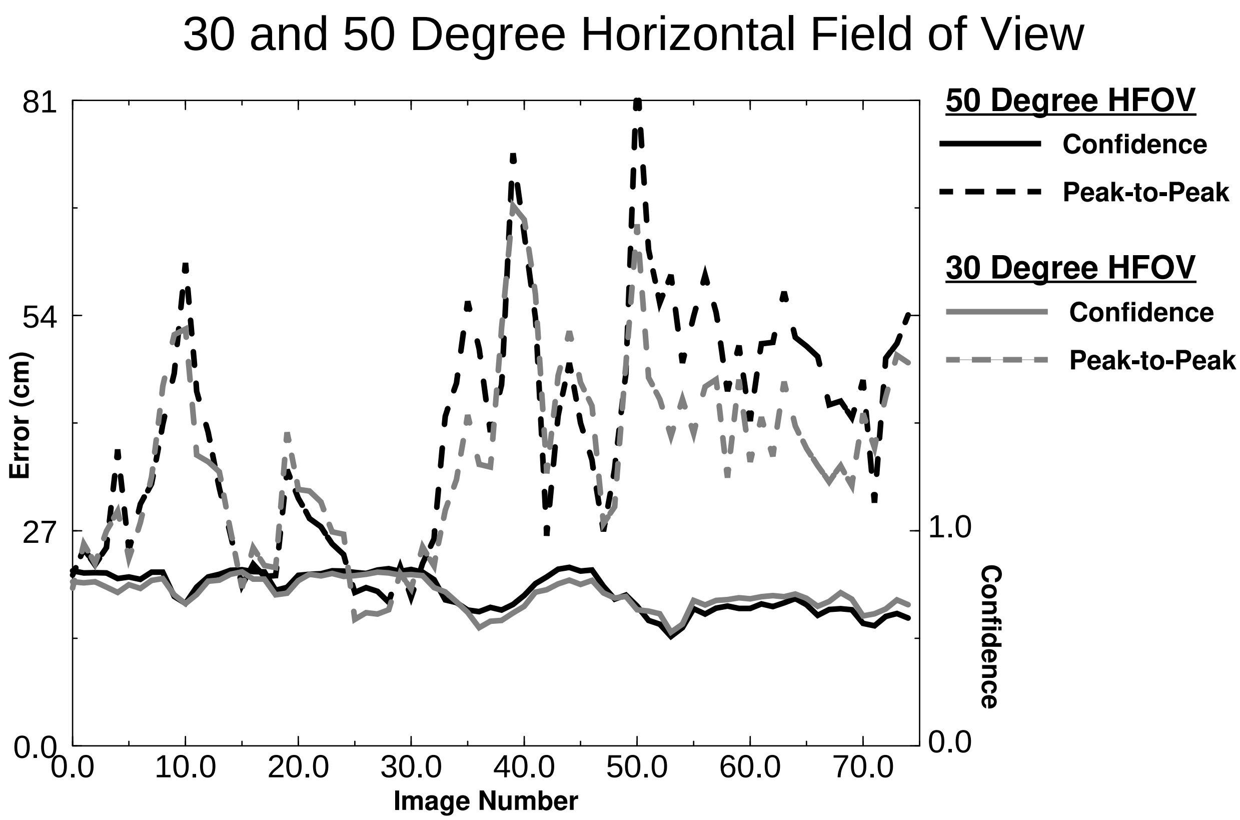


Figure 15 Combined error graphs for 30 and 50 degree HFOV.

ing lane can be seen in Figure 15. In this figure, the peak-to-peak error and confidence values for the 30 and 50 degree field of view virtual views are plotted together. As expected, the graphs for the networks trained using 30 degree horizontal field of view images have a generally lower peak-to-peak error value than those trained using 50 degree horizontal field of view images. This is especially evident in images 50-75, where the exit ramp is located. For nearly all of these images, the peak-to-peak error of the network trained using a 50 degree horizontal field of view is greater than the one trained with a 30 degree horizontal field of view. Although the magnitude of the difference in this example is relatively small - around 3.5 centimeters across the entire image sequence and about 9 centimeters for images 50-75 - it does provide some support for the claim that by selecting only consistent features, which the 30 degree horizontal field of view virtual views do better than 50 degree ones, driving performance can be improved.

2.6 Discussion

Several basic conclusions can be drawn from the results presented in this chapter. First, images from virtual cameras do not degrade the performance of the ALVINN system. Next, for a particular road type, there are likely to be many different virtual camera views that provide acceptable performance. This means that algorithms do not need to be too concerned with using a limited set of views, but have a wide variety available to them. Finally, from the limited evidence presented, it is plausible to conclude that virtual camera can be used to increase driving performance by limiting spurious and inconsistent features, just as structured noise does.

Chapter 3 Lane Transition

3.1 Introduction

It is obvious to think about lane transition only in terms of moving a vehicle from one driving lane to another adjacent lane. While this task is the most evident, there are many others that are similar in nature which require a subset of or minor extension to the capabilities required for lane transition. In addition to presenting three techniques for lane transition, this chapter introduces algorithms for exit and entrance lane detection and traversal, short distance obstacle avoidance maneuvering, and extended driving outside of the typical lane boundaries.

All of these tasks are assuming increasing importance because of the requirements for new advanced traffic control projects like the Automated Highway System (AHS) in the United States and PROMOTE in Europe. For the AHS, moving vehicles into and out of automated lanes and maneuvering to avoid disabled or non-automated vehicles or other obstacles are the primary situations where lane transition and related maneuvers are needed.

Technically, these tasks are important because they represent tactical level driving tasks which have not been studied as closely as low level tasks like lane keeping and obstacle detection. In order to be successfully accomplished, all of the tasks require a sequence of actions to be taken - a plan. This type of behavior is difficult to coax from low level systems, which are characterized by fast response to features in their surroundings. Yet traditional planning is often rigid and difficult to integrate into autonomous systems which function in the real world. The algorithms presented in this chapter circumvent the problem by framing the task in a way that preserves the flexibility and robustness of the low level lane keeping system, yet exhibits enough control to permit successful execution.

This behavior is implemented using active placement and control of virtual cameras, intelligent interpretation of the lane keeping system's response to the images created from the virtual cameras, and a simple road model. Using the information contained in the model, the virtual camera views [22] can be positioned which allow existing lane keeping systems to be used for these maneuvers. The techniques presented are based on controlling the lateral placement of virtual camera views using the road model and control scheme for guidance. They are not strictly tied to any specific lane keeping system, only requiring that the system take images as input and produce a point on the road to drive over and a measure of its internal confidence in this point as output. When implemented using a slightly modified version of the ALVINN [40] lane keeping system, the techniques have been able to autonomously navigate our testbed vehicle, the Navlab 5 [24], between lanes of a rural interstate highway, onto exit ramps and around obstacles, as well as to offset the vehicle within the current driving lane.

3.2 Road Model

All of the techniques described in this chapter require a very simple lane model. The model that the techniques use is that the lane centers have constant separation. This implies that the lanes are parallel and have a constant width. Because this model is so simple, it can either be pre-specified at start up or can be passed to the system from some other, higher level module.

3.3 Other Work

There has been a significant amount of research published describing how people change lanes as well as identifying theoretically optimal control strategies which could or should be used to autonomously control a vehicle in a lane change maneuver [15][17][18][19][36][49]. Unfortunately, many of the researchers didn't have the facilities or equipment to test their results outside of the lab.

An early and successful approach to lane transition is presented in [2]. In this approach, a vision-based model-driven lane keeping system is used to find lane markings [8]. When a lane change is initiated, instead of driving exactly between the left and right lane markings, the system offsets the vehicle in the direction of the destination lane. The source lane markings are tracked until either they leave the camera field of view or when the vehicle crosses the center line of the road. When either of these occur, the system looks for the destination lane markings, using a road model like the one mentioned in Section 3.2. Once it acquires the destination lane markings, the system recenters the vehicle in this new lane. This technique is very similar to the Incremental View Lane Transition algorithm, which will be presented in Section 3.4.4, except that it is more closely tied to their lane keeping system.

Another system which executes lane change maneuvers is ROBIN. (See Section 5.2 for a detailed description of ROBIN.) Although ROBIN uses virtual cameras to enhance its driving capabilities, it does not use any of the active virtual camera control techniques presented later in this chapter. The method that it uses is very similar to the network switching method described in Section 3.4.2. The ROBIN system uses two radial basis function networks which are trained to drive in the right and left lanes of a two lane interstate highway. When the system is transitioning lanes, a slide behavior is activated. When this behavior is activated no network is controlling the vehicle and it is essentially driving blind between the lanes. When the slide is complete the destination lane network is activated. The lateral slide distance is set so that when the destination lane network is activated after the slide is complete, it will be able to recognize the situation and finish reentering the vehicle in the new lane. If the lateral slide distance is too short, the destination lane network becomes confused or acts abruptly when it was activated. This method works well at low speeds on straight sections of highway, but does not perform adequately on curves. It has not been tested at typical highway speeds [45].

In a related area, experimental work on lane structure detection was reported in [56]. The system uses video images to detect the presence of additional lanes. This system was not used to control the vehicle, but rather to provide geometric information about the environment around the vehicle.

3.4 Lane Change

Many systems have been created which can keep a vehicle within a driving lane, but little experimental work has been reported that describes methods to transition an autonomous vehicle between lanes. This task is assuming increasing importance because of the requirements for new advanced traffic control projects like the Automated Highway System (AHS). For the AHS, moving vehicles into and out of automated lanes is the primary situation that lane change maneuvers are needed and one which is on the critical path for deployment of full scale AHS.

Three techniques to accomplish lane transition have been investigated. The most basic involves intelligently switching between two trained ALVINN networks. The other techniques use active control of virtual camera views to move the vehicle into the destination lane.

3.4.1 Control Scheme

All of the lane transition techniques presented in this chapter are implemented using a linear control scheme to determine the appropriate way to move to the next step of the lane transition maneuver. This scheme was chosen because it provides a simple, yet effective, control mechanism for autonomous lane transitions using the Navlab 5. The techniques presented in this chapter are not constrained to using a linear scheme, but because it leads to reliable, intuitive lane transitions when used with the ALVINN lane-keeping system, other schemes were not investigated.

When using this scheme with the Incremental Network Switching technique it defines how to integrate two network output displacements into a single displacement which can be used to compute a transition point to drive over. Briefly, if a transition of this type has 10 steps, during the first step 90% of the output displacement of the driving lane network will be combined with 10% of the destination lane output displacement to compute a final displacement. On the second step, the percentages would be 80% and 20%. See Section 3.4.3 for a more detailed explanation of this technique. For the Incremental View and Dual View Lane Transition techniques, described in Section 3.4.4 and Section 3.4.5, the scheme defines how virtual camera views are to be moved at each step in the transition. For these techniques, the lane separation distance is divided by the desired number of lane transitions steps. The resultant step distance is then used to update the virtual camera views appropriately.

All of the lane transition techniques except those presented in Section 3.5, Lateral Offset Driving, require two trained networks. This requirement is not a function of the techniques, but rather of the underlying lane keeping system. Because ALVINN learns to key off features at particular locations in the input image, a single network can only drive in lanes or on roads whose feature are consistent with those learned. For the left and right lanes of an interstate highway, this is not the case, thus the need for two networks.

3.4.2 Lane Transition by Network Switching

Because ALVINN has the ability to recenter the vehicle in the driving lane even if it is displaced by a large distance, it is possible to transition from the current driving lane to the destination lane by simply switching to a network that was trained to drive in the destination lane. When presented with an image of the road while in the driving lane, this network would recognize that the vehicle was offset by a large amount from the proper driving position (in the destination lane) and would produce a sharp turn to recenter the vehicle. This type of lane transition is inadequate for two reasons. First, it is a very unsafe maneuver. At 25 meters/second, the lateral acceleration resulting from the sharpest turn ALVINN typically produces for highway driving is approximately 4.0 m/s^2. Even though the system cycles at 12-15 Hz, it may not recover quickly enough to prevent the vehicle from driving through the destination lane and off the other side of the road. Second, this type of maneuver would be uncomfortable and frightening to any passenger in the vehicle. A more controlled approach is desired in which the vehicle slowly transitions from one lane to the other, monitoring its performance along the way. To accomplish this goal, a modification of the network switching idea, along with two techniques based on controlled placement of virtual camera views, have been developed.

3.4.3 Incremental Network Switching

A method to transition lanes which is based on the simple network switching idea described above is called Incremental Network Switching (INS). Instead of immediately switching from the driving lane network to the destination lane network, this method combines the output displacements of the driving and destination lane networks to slowly transition the vehicle from the driving to the destination lane.

The INS method requires two trained networks, one for each lane. When a lane transition is initiated, road images are passed not only to the current driving lane network, but also to the network trained to drive in the destination lane. Each network responds by producing the displacement at the lookahead distance that would be required to recenter the vehicle in its respective lane.

Initially, the destination lane network's displacement is large, indicating that a hard turn is necessary. But instead of using this displacement alone, the driving lane network's displacement is also utilized to compute a target displacement, which will determine how to steer the vehicle. This is done by incrementally adjusting how much contribution the displacement associated with the driving and destination lane networks have to the target driving displacement.

Initially, the target displacement is very similar to the driving lane displacement. This prevents the sharp transition into the destination lane which is present in the basic lane switching method. But because a small part of the destination lane displacement is used, the vehicle does begin to move toward the destination lane. In subsequent iterations, the amount of the displacement contributed by the destination lane increases and the amount contributed by the original driving lane decreases. When the amount contributed by the destination lane becomes significant, its output displacement is smaller because the vehicle is much closer to the center of the destination lane. This means that sudden, sharp movements are again averted. As the transition ends, the destination lane output displacement is the major contributor to the target driving displacement. Although the original driving lane network is producing a displacement which would cause the vehicle to return sharply to the original driving lane, its contribution is small and has only a limited effect. For these experiments, the lane transition was specified to be accomplished in 30 increments. This means that at each step, the contribution of the driving lane displacement was reduced by 1/30 while the contribution of the destination lane displacement was increased by 1/30.

Two constraints were used to ensure that both networks were performing adequately during the lane transition maneuver. The IRRE confidence measure [40] produced by each network was the first constraint. This measure is an indicator which is strongly correlated to satisfactory driving behavior. In order for an INS lane transition to proceed, the IRRE measure produced by either

Left Lane **Right Lane**

Left Lane **Right Lane**

Figure 16 INS processing.

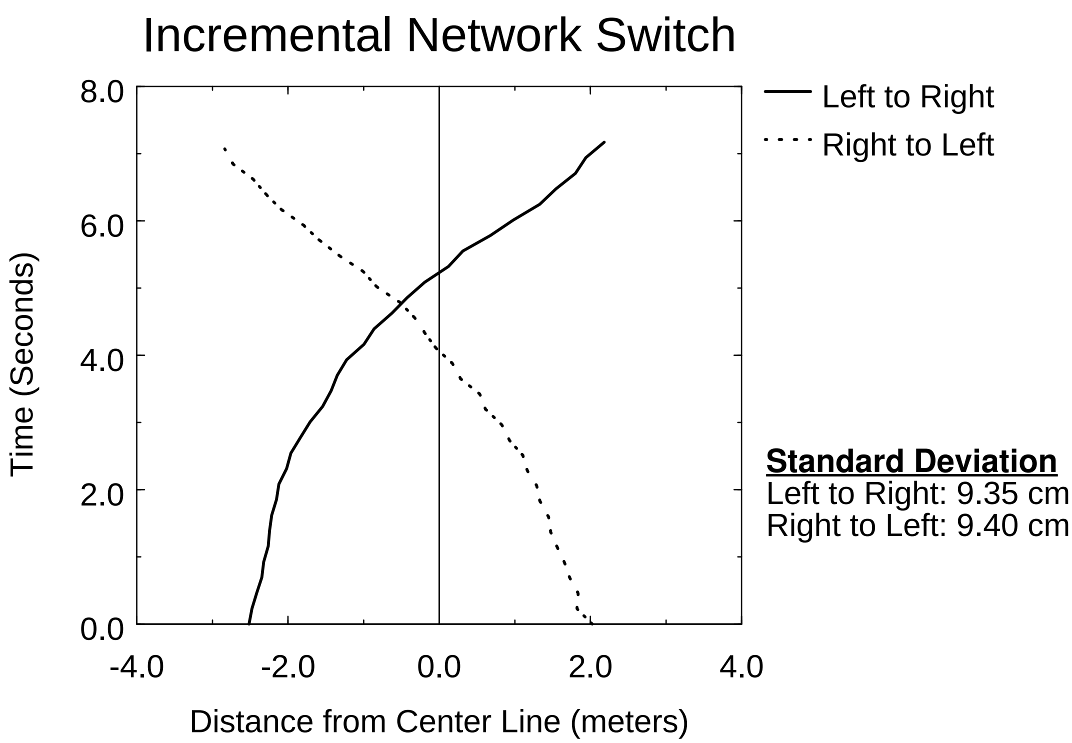


Figure 17 Average vehicle path during Incremental Network Switch Lane Transition.

of the networks was required to be greater than a threshold value. The reason only one network was required to produce a high IRRE value was the initially large lateral offset from the destination lane. This offset could be significant enough to cause the network trained for the destination lane to have difficulty correctly reconstructing the input image, but at the same time produce a reasonable steering response. Basically, it is possible that the network was never presented with images shifted similarly during training. Typically however, both networks produced satisfactory measures, with the measure increasing (or decreasing) as the vehicle moved into (or out of) the lane a network was trained to drive in.

The second constraint was related to the geometric relevance of the network's output displacements. In general, the difference between the displacements should be equal to the lane separation distance, specified in the road model. The driving lane network should specify the center of the driving lane while the destination lane output specifies the center of the destination lane. In order for this constraint to be satisfied, the difference between displacements was required to be within 40% of the lane separation distance. This figure allowed for inadequacy in the network's output to represent all displacements while still prohibiting obviously false combinations of displacements.

Figure 16 shows two images taken at the beginning and end of a left to right INS lane transition. In both images, the virtual camera view footprint on the ground plane is shown immediately in front of the vehicle. Items associated with each of the networks used for the INS are grouped in the lower left corner of each image. Each grouping contains a preprocessed ALVINN image created from the virtual camera view, ALVINN's output steering direction vector (shown above the preprocessed image), a bar graph representation of the IRRE confidence metric (shown to the right of the preprocessed image), and the network name (shown above the output steering direction vector). The grouping for the left lane network is on the left, while the one associated with the right lane is on the right. Also shown in each image as a grey box inside the virtual camera view footprint is the point on the road that the system is currently steering toward.

In the top image, the steering output of the left lane network - the one the vehicle is currently in - is almost straight ahead, while the output of the right lane network indicates a sharp right turn is required to recenter the vehicle. In the bottom image the situation is reversed. The output of the right lane network is nearly straight ahead while the output steering direction of the left lane network signifies that a hard left turn is necessary to recenter the vehicle.

Of the 38 lane transitions (19 in each direction) that were attempted, 37 succeeded. The results of these transitions are shown in Figure 17. This figure shows the average vehicle position with respect to the center line of the road versus time. (The average distance for completion of an INS lane transition was 156 meters, while the average vehicle speed was about 22.2 meter/second. Also note that the standard deviation for each transition type is the average deviation across all time slots within the transition.) The most obvious feature in this graph is that the two trajectories do not cross at the center line of the road. Initially it was suspected that this was due to the vehicle not being properly centered during training of the left lane network. But as will be shown later, this assumption proved to be false. The exact cause of the anomaly is not known, but it is suspected that it is related to some small difference in how the left and right lane networks respond to images which produce large displacements.

A somewhat subtler feature that is visible in the figure is the that the trajectories cross later in time than would be expected. If the transition was perfect, the trajectories would cross at about the 3.5 second mark, but using this method, they cross much later. This characteristic manifests itself

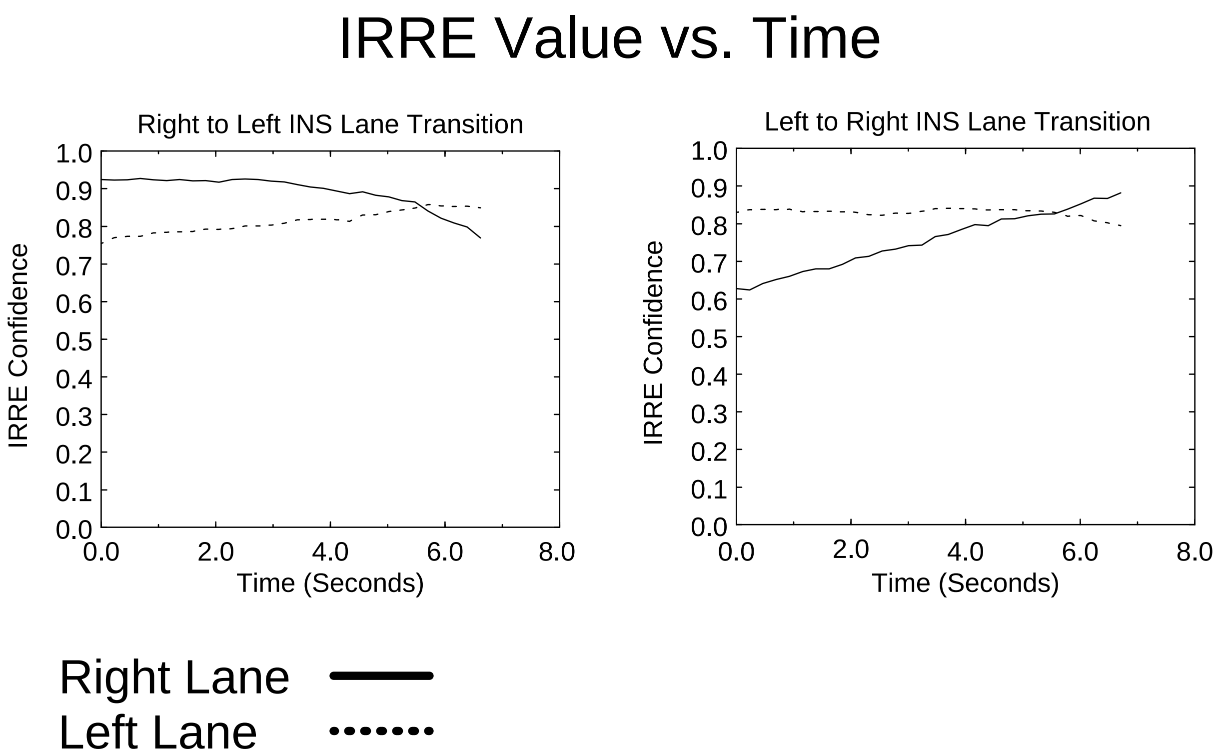


Figure 18 IRRE values during INS Lane Transitions.

during experiments by causing the vehicle to “dive” into the destination lane. Specifically, instead of moving smoothly into the destination lane, the vehicle’s lateral velocity increases as the lane transition continues. This leads to overshoot in the destination lane after the maneuver has completed because of the somewhat long correction time required for smooth driving at high speeds. One explanation for this behavior relates to each network’s inability to produce the large displacements that are required to locate the center of each lane during the beginning and end of the INS. Initially, the destination lane network produces the largest steering response it is capable of creating in the direction of the destination lane. Although this response is in the correct direction, it is not large enough to accurately locate the destination lane center. The result is that the vehicle initially moves slower toward the destination lane than would be anticipated. As the transition continues and the vehicle nears the center of the road, both networks begin to produce accurate steering responses and the vehicle moves laterally as expected. But once the vehicle has moved far enough from the original driving lane, the network associated with that lane cannot produce a large enough steering response to accurately locate the center of the lane. In this case, the target displacement is greater than what the geometry dictates, and the vehicle accelerates into the destination lane.

Figure 18 shows the average IRRE confidence values for the right and left lane ALVINN networks during right to left and left to right Incremental Network Switch lane transitions. Note that the confidence value of the destination lane network increase throughout the switch, while the original driving lane network's confidence decreases. Also, throughout the transition, the average IRRE confidence value was high for each network, even when the vehicle was not in the lane that the network was trained to drive in. This was a result of both road/shoulder boundaries being visible in the virtual camera image ALVINN used. Because the boundaries were partially visible in the image when the vehicle was in either lane, a high IRRE confidence value resulted. But as the graph shows, the network's ability to reconstruct the input image (which leads to high IRRE values) was beginning to diminish as the vehicle neared the end of the lane transition.

3.4.4 Incremental View Lane Transition

The Incremental View Lane Transition (IVLT) method uses two virtual camera views to move the vehicle from the driving lane to the destination lane. Unlike the INS method, the IVLT technique accomplishes lane transition by intelligently controlling the placement of these views and using ALVINN's response to the images created from them. The central idea behind this technique is that virtual camera views can be used as a tool to check if the upcoming driving situation can be correctly handled by ALVINN before actually encountering the situation. If it is determined that the situation can be handled, ALVINN's network output, produced from the images created by the virtual camera views, is used to control the vehicle in a manner which produces the desired lane transition.

Two virtual camera views are created for the IVLT technique. During normal operation before a lane transition has begun, only one virtual camera view, the Driving View, is used to keep the vehicle in the source driving lane. Initially, the Driving View is at the same location as the view which was used during training of the source lane network. When an IVLT is initiated, an additional view, called the Test View, is created. The only difference between the Driving View and the Test View is that the Test View is offset a small lateral distance, in the opposite direction of the desired lane change, from the Driving View. Lane transition is accomplished by switching the source of the images which ALVINN uses for vehicle control from the Driving View to the Test View. The lateral offset of the Test View leads to images in which the vehicle seems to be improperly aligned with the driving lane. Because ALVINN's neural network is trained to recover from these situations, it responds with steering commands that attempt to recenter the vehicle. Once the vehicle has been recentered in the virtual image, a new, shifted, Test View is created. As a side effect of the shifted views and resultant steering commands, the vehicle is incrementally moved through several lane transition locations and into the destination lane. The relative location of the Driving and Test Views for a right to left lane transition is shown in Figure 19.

The IVLT method requires two networks be trained, one network for driving in the source lane and another for the destination lane. The source lane network is used by the IVLT algorithm when the vehicle is moving from the center of the source lane to the boundary between lanes. When the vehicle reaches this point in the transition, the IVLT algorithm uses the destination lane network to finish the transition. Although the IVLT method requires two networks, only one is used to control the vehicle at any time during the transition. While the system is using the image created from the Driving View to servo the steering wheel, it uses the Test View image to determine if the system will be able to reliably move the vehicle to and continue control at the next transition location.

To determine if the next transition location is satisfactory, the Test View is used by the IVLT technique to compute both internal and external confidence metrics. These metrics can be used to determine if moving to the new lane location is advisable before actually executing the move. The IRRE confidence computed from ALVINN's output when presented with an image created from the Test View is used as an intrinsic measure of how well the network thinks it will be able to drive if the Test View is switched to the Driving View. Geometric knowledge about the lateral offset distance of the Test View from the Driving View, along with the output displacement of the Test and Driving Views, is used as the external metric. Because the system uses this geometric constraint in addition to the purely intrinsic IRRE measure, it is less susceptible to transitory features in the input image which can cause improper driving behavior yet not produce substantially lower IRRE values.

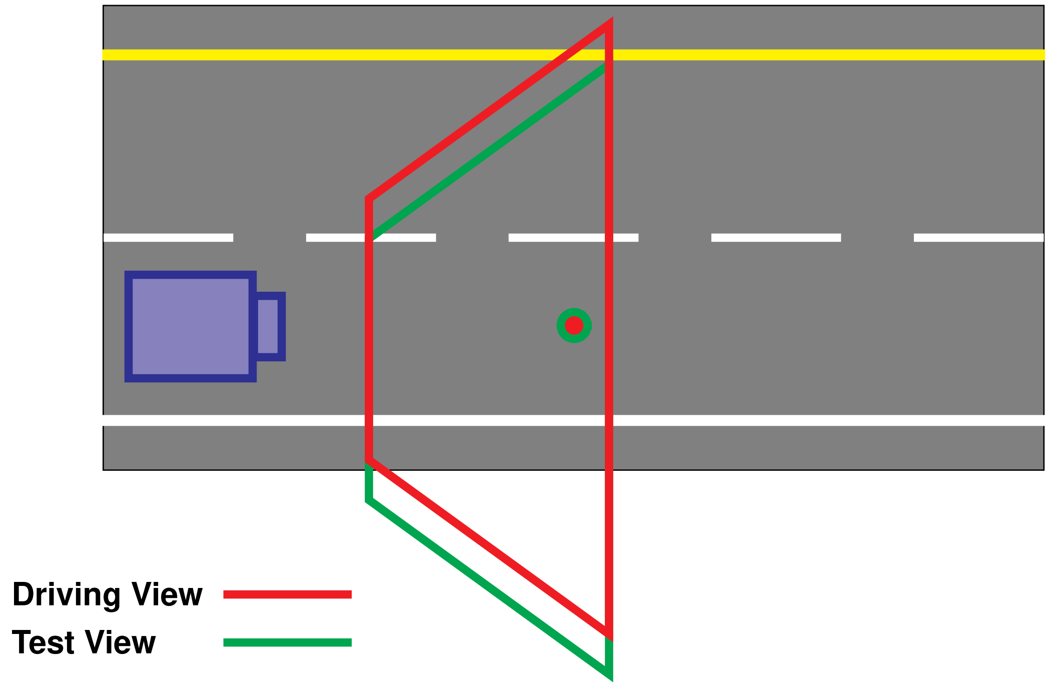


Figure 19 Driving and Test Views for a right to left lane transition.

Before the Test View can be created, the appropriate lateral offset distance is computed. The lateral offset distance applied to the Test View is determined by the lane model as well as the desired number of intermediate lane transition positions. Because the view associated with each lane transition position must be used for a fixed period of time - while the system is determining if moving to the next location is advisable - the selection of the number of lane transition locations is proportional to the transition rate of the vehicle. Typical lane separation distances of 3.2 to 4.0 meters, along with 10-20 intermediate lane transition positions, lead to controlled and comfortable IVLT's.

If the vehicle is to transition into a lane to its left, the Test View is offset to the right of the Driving View. The opposite holds for lane transitions to the right. An example of initiating a right to left lane transition in shown in Figure 19. The network associated with the Test View is in most cases the same as the one associated with the current Driving View. Because the Test View is laterally offset, the image it creates looks as if it was taken from a vehicle that was not centered in the driving lane. Because of this, when the Test View becomes the Driving View after the internal and external reliability constraints have been met, ALVINN will compensate by steering the vehi-

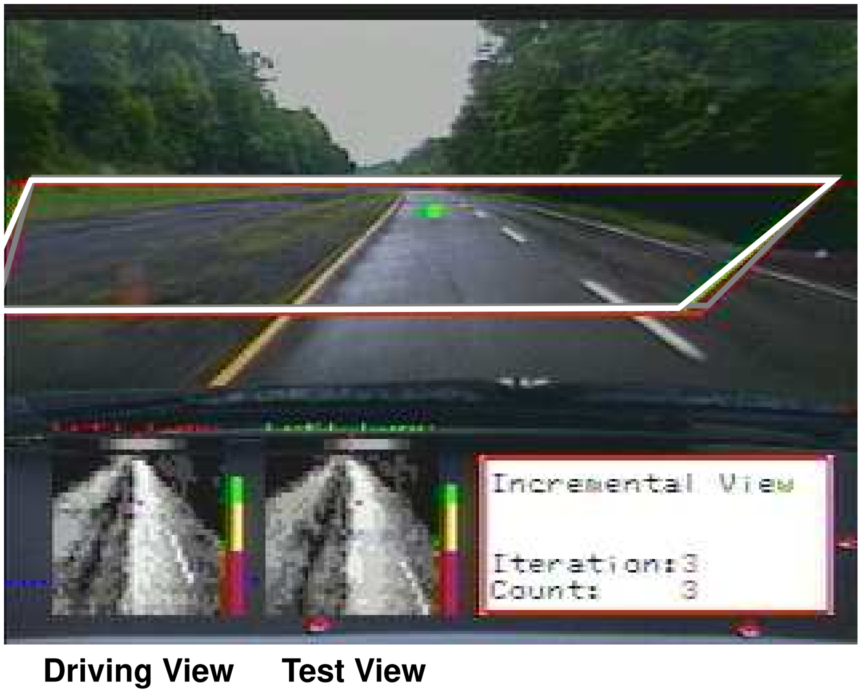


Figure 20 IVLT processing.

cle so that the road is centered in the image. The effect is to move the vehicle by the offset amount in the opposite direction of the offset.

Before actually moving from the current lane transition position, the IVLT method estimates ALVINN's ability to drive at the next location. It does this by monitoring the internal and external reliability metrics produced by ALVINN's neural network and the geometric constraints. To compute the external metric, the system computes the difference between the output displacement of the Test View network and the Driving View network. This difference is compared to the lateral offset distance of the Test View. If the difference between these figures is below some threshold (typically 40% of the Test View Offset), and the IRRE confidence is above another threshold (typically 0.40 out of 1.0) for a few (typically 2) Test View images, the system switches to using it as the Driving View and creates another Test View, again offset by the same distance from the new Driving View. This cycle repeats until the vehicle has reached the midway point of the lane transition. At this location, instead of using the same network as the Driving View, the Test View uses

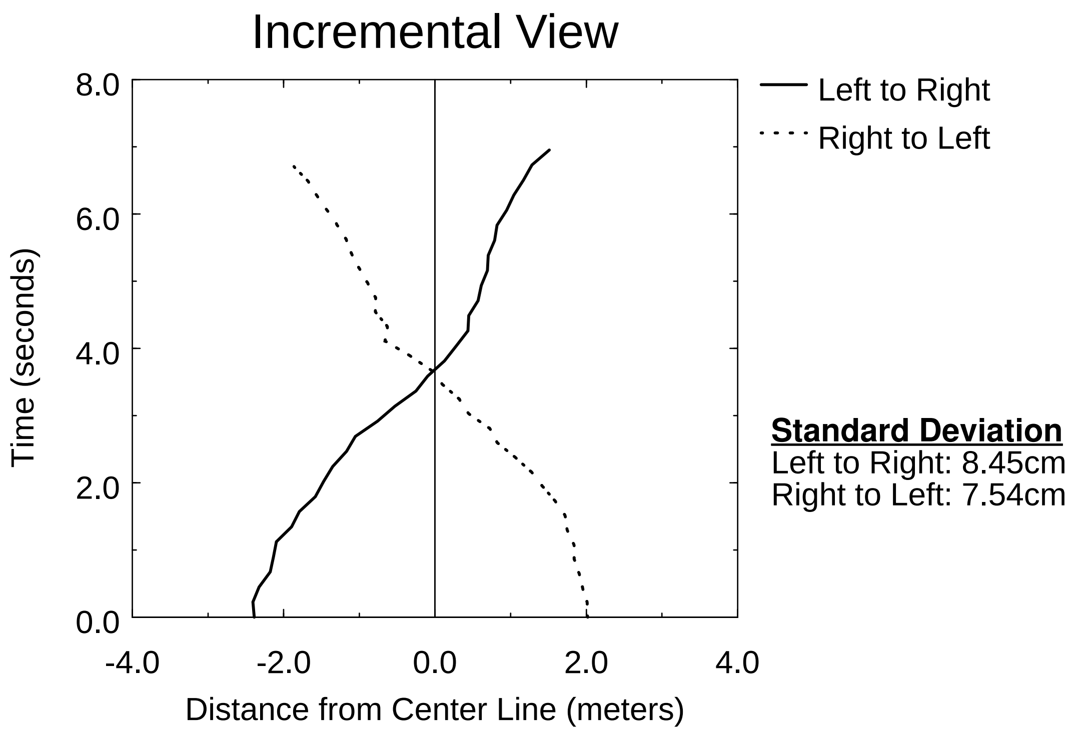


Figure 21 Average vehicle path during Incremental View Lane Transitions.

the network associated with the destination lane. Also at this step, the Test View offset distance from the Driving View is modified so that it correctly images the destination lane.

Figure 20 shows an image taken at the beginning of a left to right IVLT. The Driving View footprint is visible as the darker trapezoid, while the Test View footprint is displayed as the lighter trapezoid, offset a small distance to the left of the Driving View footprint. The preprocessed image, steering direction, and IRRE confidence groups are present in the lower left corner of the image, with the Driving View group positioned on the left.

ALVINN's steering response to the preprocessed image created from each view is nearly identical. But upon close examination, the steering output vector associated with the Test View is shifted slightly right of its position for the Driving View. This indicates that a a shallow right turn will be executed when the Test View is switched to the Driving View, leading to a left to right lane transition. Also note that the IRRE confidence associated with both preprocessed images is very high, indicating that ALVINN believes that it has encountered similar scenes during training.

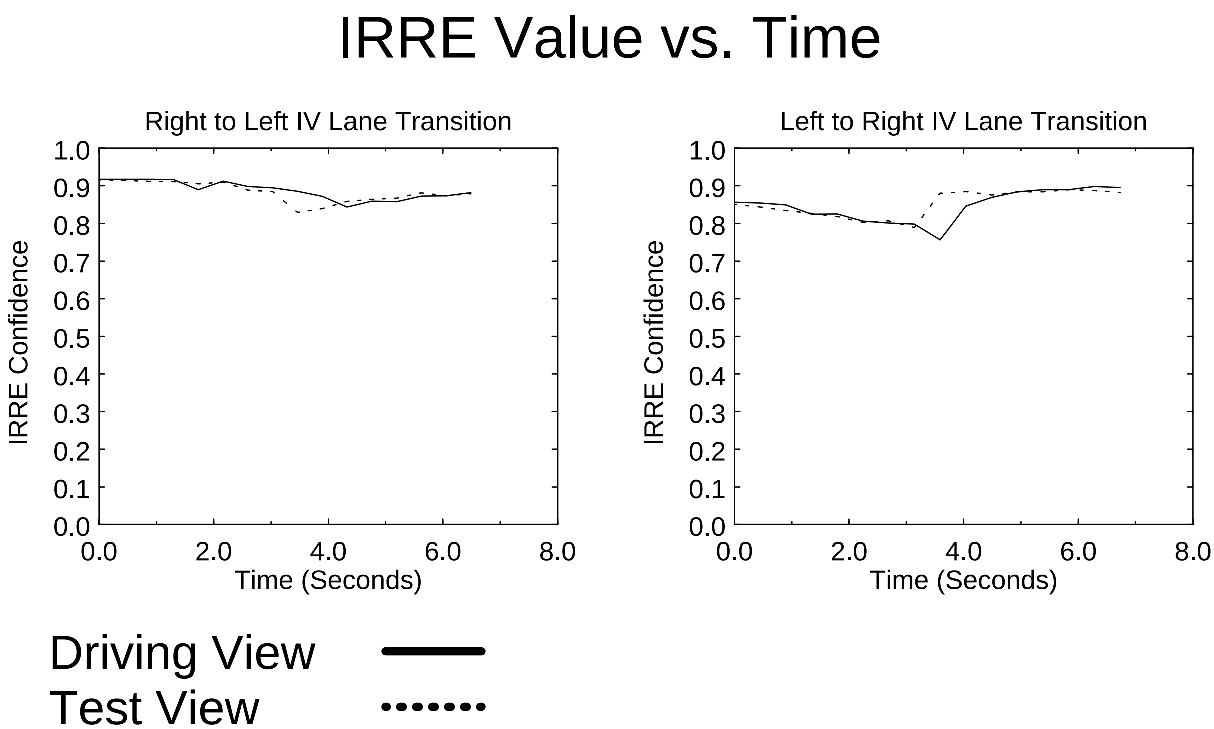


Figure 22 IRRE values during IVLT's.

All 32 IVLT's that were attempted to verify this lane transition method were successfully completed. Like the INS method, half were left lane to right lane transitions. The same two networks that were used for the INS method experiments were for these tests. The results of these transitions are shown in Figure 21. (The average distance for completion of an IVLT was 151 meters, while the average vehicle speed was about 21.6 meter/second.) There are three things to note about this graph. First, the average trajectory lines cross at the center line of the road. This is contrary to what occurred for the INS technique, which was thought to indicate improperly trained networks. Second, the trajectories are not as smooth as those in the INS method. This is a result of intermittent steering output due to the reduced cycle rate when a Test View is being processed. The last characteristic to note is that the IVLT method relinquishes control before the vehicle is at a typical driving position in the destination lane. This characteristic was unanticipated, but upon closer inspection it was determined to be the result of at least two contributing algorithmic factors. The first factor was the relaxed constraint on the required agreement of the output displacements of the Driving View and the Test View. Basically, because the last transition of a Test View to a Driving View was executed when the difference between their respective output displacements was within 40% of their lateral offset amounts, the IVLT technique relinquished control before the vehicle was fully centered in the destination lane. This means that control is relinquished when the vehicle is at a distance of up to 140% of the lateral offset distance away from the typical driving location in the destination lane. The second factor was the relatively loose closed loop vehicle control with respect to the Driving and Test View locations. Specifically, because the networks' output steering response to images created from Driving View and Test View were only compared to each other, and not to ground truth, transitions occurred before the vehicle had recentered itself in the Driving View image. For example, it was allowable for a transition to occur while driving on a straight road even if the Driving View and the Test View both responded with a moderate turn command. The reason that both networks produced this response was that the vehicle hadn't fully transitioned to the new driving location. i.e. The lane hadn't been recentered in the Driving View image. But because the output's were only compared to each other, the transition was allowed.

Figure 22 shows the average IRRE confidence values for the right and left lane ALVINN networks during right to left and left to right IVLT's. Like the confidence graphs associated with INS lane transitions, the IRRE confidence values associated with both the Driving and Test Views remained high throughout the transitions. The main feature that is visible in these graphs is the jump in confidence values, first visible for the Test View, and a short time later in the Driving View. This jump corresponds to the midway point of the IVLT, where the system changes which network is associated with the Test and Driving Views. For example, for a right to left lane transition, the right lane network is initially associated with the Driving View and the Test View. When the IVLT reaches its midway point, the left lane network first becomes associated with the Test View and then, if the constraints are satisfied, with the Driving View. This change corresponds to the jumps.

3.4.5 Dual View Lane Transition

Like the IVLT method, the Dual View Lane Transition method (DVLT) uses two virtual camera views to move the vehicle from the driving lane to the destination lane. While the IVLT method tracks only the original driving or destination lane to execute the lane transition, the DVLT method tracks them both. The idea behind the DVLT technique is to use a geometric road

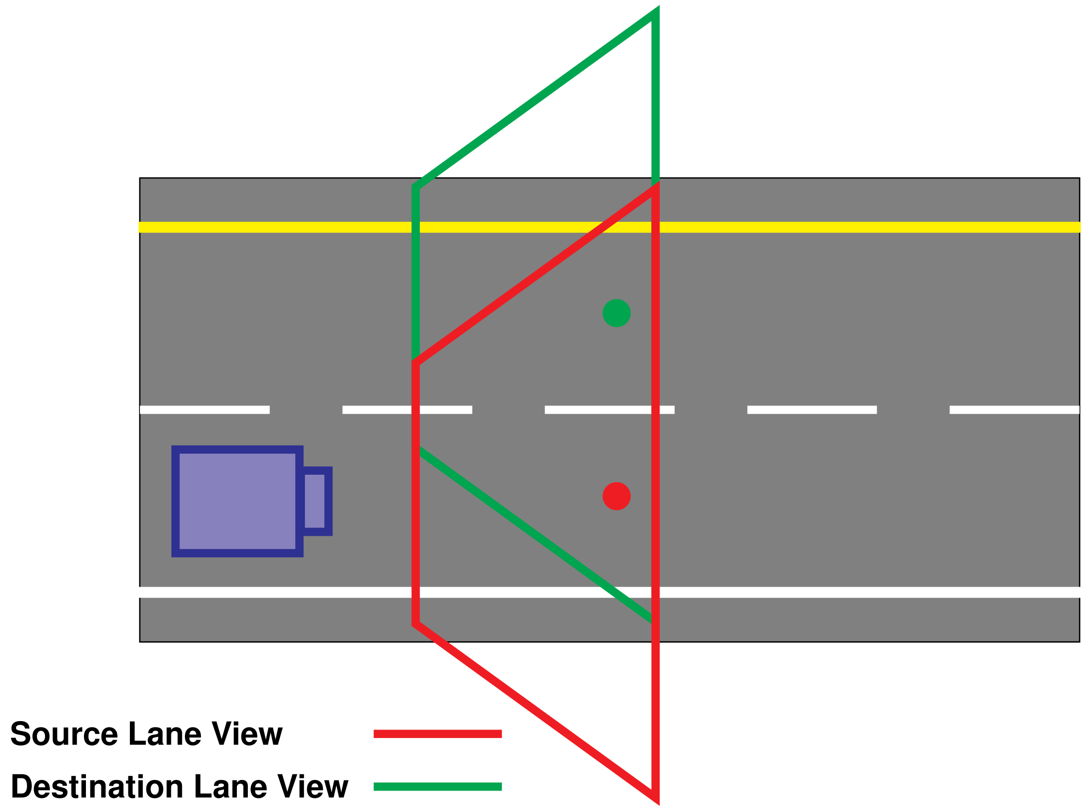


Figure 23 Initial virtual view placement for the DVLT technique.

model along with accurate knowledge of the location of the center of each lane to smoothly servo the vehicle between lanes.

This technique is a bottom up approach to lane transition. Although it can use input from high level modules to initiate the lane transition, the geometry of the situation is what drives this method. The system locates the center of both lanes and then moves the vehicle based on these locations. This is in contrast to the IVLT method where ALVINN's response to the movement of virtual camera views is what causes the lane transition to proceed.

Like IVLT, the DVLT method requires networks to be trained for the driving and destination lanes. When a DVLT is initiated, a second virtual camera view is created. The road model is used to laterally offset this view, called the Destination Lane View (DLV) so that it is centered over the destination lane. The network that was trained to drive in the destination lane is associated with

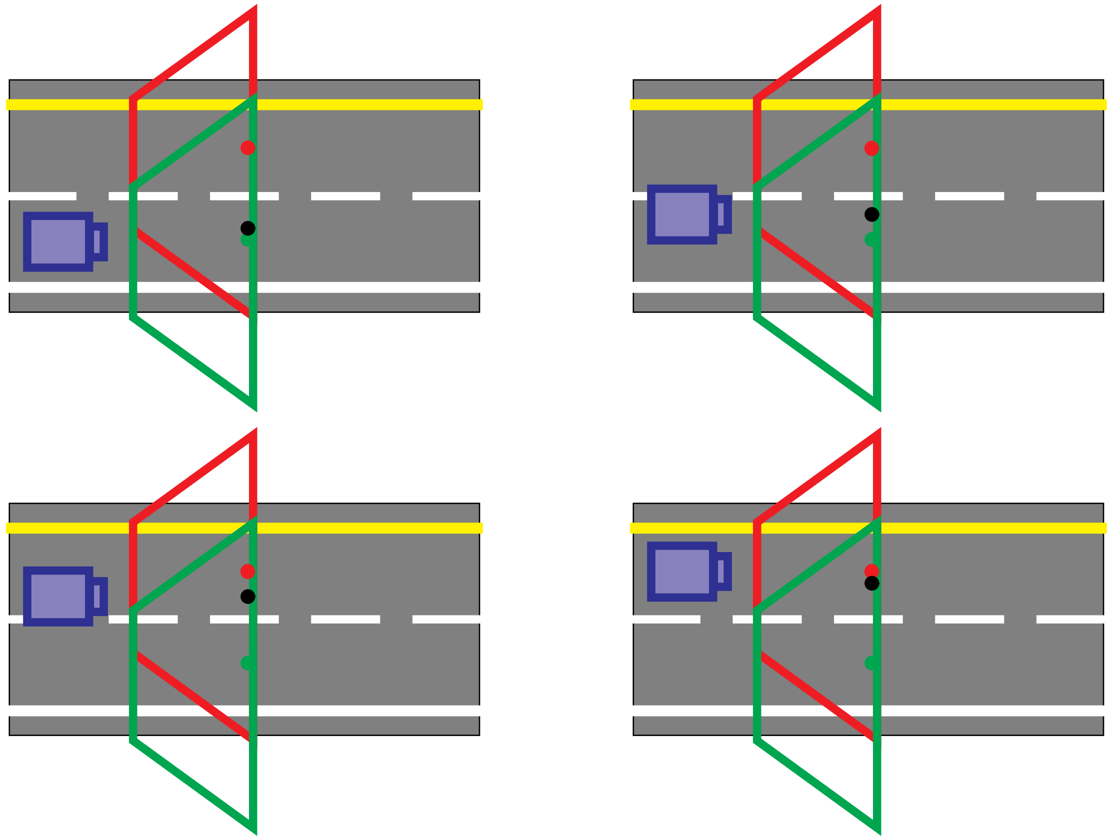
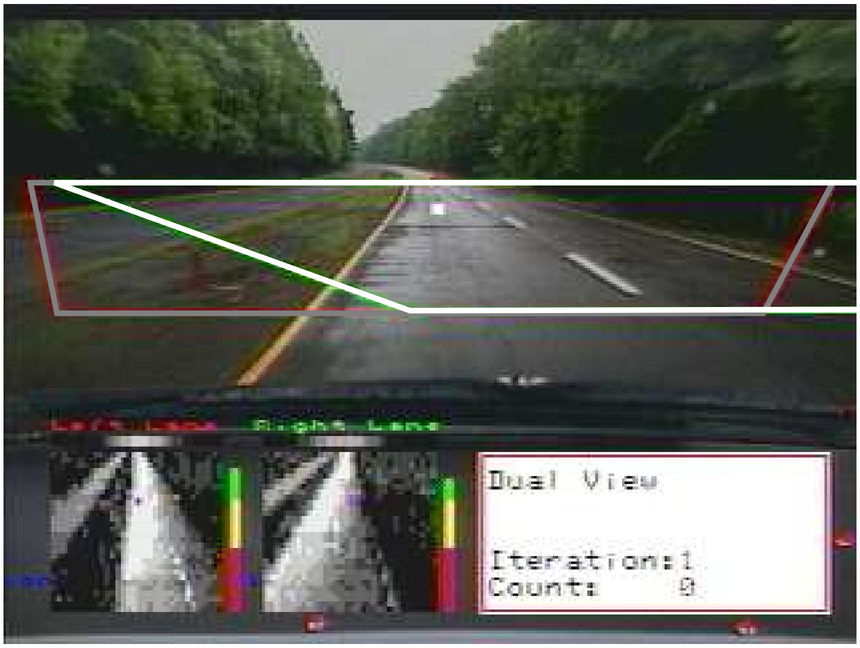


Figure 24 Movement of the vehicle and the views in the DVLT method.

the DLV. Contrary to the IVLT, the DLV is not treated as a Test View, but rather as a tracking device. The driving, or Source Lane View (SLV) is treated in the same manner. See Figure 23.

After the DLV has been created, images are generated from both the DLV and the SLV and are passed to their respective networks, which produce an output displacement along with an IRRE confidence value. Both output displacements are converted to vehicle relative points. These points, called Lane Center Points (LCP's), specify where ALVINN believes the center of each lane is located. Because each network has been trained using the same lookahead distance, and because the DLV is only offset laterally from SLV, the LCP's fall on a line which is perpendicular to the direction of travel.

At this point, the LCP's are used to calculate the Modified Lookahead Point (MLP). The MLP is the point that the vehicle will actually drive toward. The MLP is between the two LCP's, along the line connecting them. The distance between the MLP and either of the LCP's is related to the



Left Lane **Right Lane**

Left Lane **Right Lane**

Figure 25 DVLT processing.

step in the DVLT process. For example, in DVLT experiments which use 16 iterations to transition between lanes, the first MLP would be 1/16 of the total distance (along the line connecting the LCP's) away from the driving lane LCP and 15/16 from the destination LCP. The second MLP would be 1/8 and 7/8, respectively. Because the vehicle is instructed to steer toward the MLP, the virtual views become misaligned with their respective lanes and must be updated. This process is similar to the one used for the IVLT method except that both views are changed. If the vehicle is moving to the left, both views are updated by moving them to the right. This process is continued until the vehicle has transitioned completely into the destination lane. See Figure 24. The black dot in each of the diagrams in this figure is the MLP. The combination of moving toward a MLP which is continually closer to the destination lane center and of updating the location of the virtual camera views results in a smooth, controlled transition.

During the lane transition, the IRRE confidence metric and the constant lane separation constraint are used to determine if the system is confident in its current driving ability. This is done by checking the IRRE confidence value associated with the images created by the views and by comparing the distance between the two LCP's. For the transition to continue, the IRRE confidence measure is required to be above a threshold value, typically 0.40, and the distance between the LCP's is required to be within 40% of the lane separation distance, specified by the lane model. Unlike the IVLT, this method does not check to see if the system will be able to drive at the next lane transition position.

Figure 25 shows two images taken at the beginning and end of a left to right DVLT. In the top image, the vehicle is still in the initial driving lane. The SLV footprint is visible as the darker trapezoid immediately in front of the vehicle. The preprocessed image associated with the SLV is shown in the lower left corner. The DLV footprint is also visible as the lighter trapezoid, offset to the right of the SLV footprint. The preprocessed image associated with it is shown to the right of the preprocessed image from the SLV. Just above each preprocessed image is ALVINN's driving response to that image. For each view, the driving response is almost the same, indicating that the constant separation distance road model and virtual camera imaging were accurate. Also shown in front of the vehicle is a grey dot which represents the MLP. Next to each preprocessed image is a bar graph representation of the IRRE confidence value associated with each. For both the SLV and DLV, the confidence is high - above 0.70.

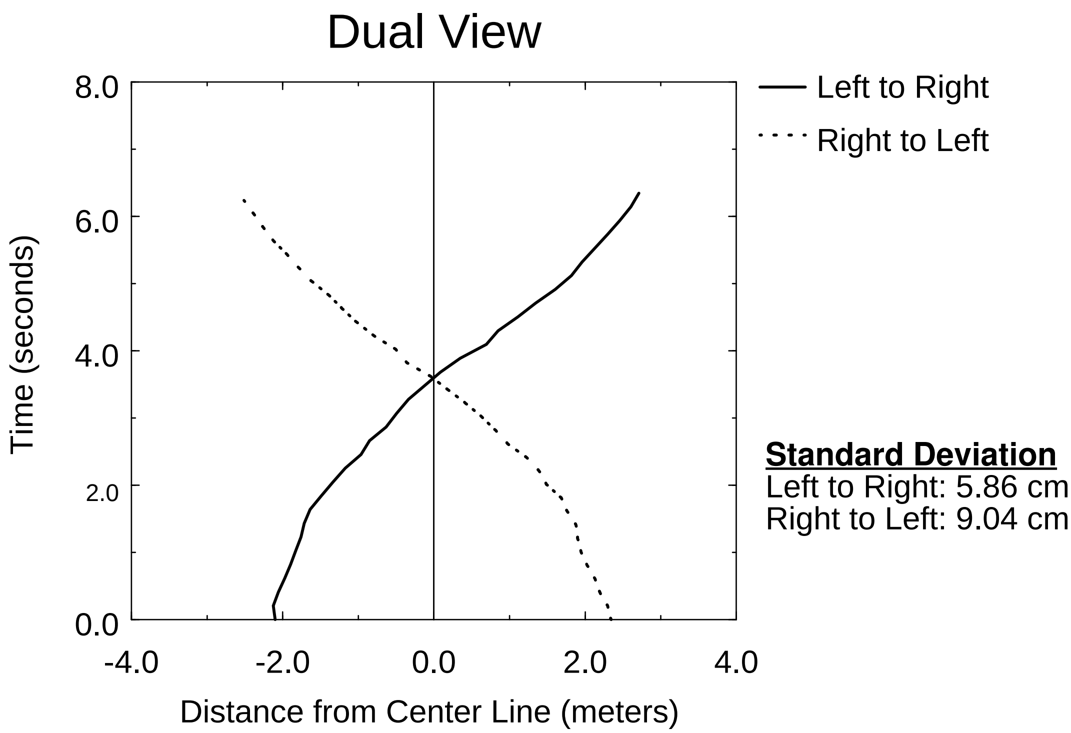


Figure 26 Average vehicle path during Dual View Lane Transitions.

The image on the bottom was taken as the vehicle approached the destination lane. In this image, the DLV footprint is almost centered in front of the vehicle while the original SLV is offset to the left. The preprocessed images are shown in the same location, along with ALVINN's driving responses and IRRE confidences values. The responses and confidence values are similar to those in the top image, even though the vehicle is now in the destination lane. The MLP is also shown.

All 42 DVLT's that were attempted to verify this lane transition method were successfully completed. Like the other methods, half were left lane to right lane transitions while the rest were right to left. The results of the experiments are shown in Figure 26. (The average distance for completion of an DVLT was 138 meters, while the average vehicle speed was about 21.9 meter/second.)

There are two characteristics to notice in this figure. First, the lane transitions are symmetric across the center line of the road as well as with respect to the time in the lane transition. Second, the DVLT method doesn't relinquish control until the vehicle is at the proper driving position in

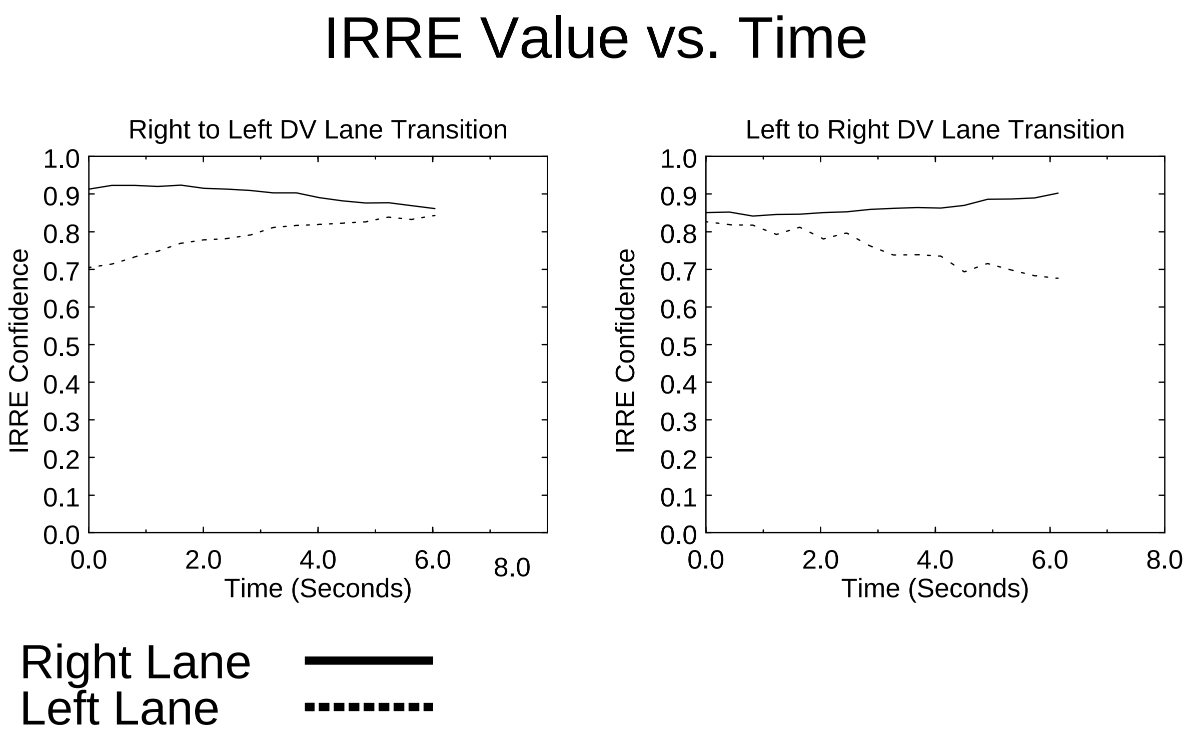


Figure 27 IRRE values during DVLT's.

the destination lane. Also notice that at the end of the transition the vehicle trajectory is still moving somewhat outward. A small amount of overshoot did occur and can be attributed to the existing lateral movement of the vehicle coupled with the moderately long response time that is a side effect of the large lookahead distance required for highway driving. But empirically, the DVLT method yielded the smoothest and most realistic transitions into the destination lane.

Figure 27 shows the average IRRE confidence values for the right and left lane ALVINN networks during right to left and left to right DVLT's. Similar to the graphs in Figure 18, the IRRE confidence values produced by an ALVINN network increase as the vehicle moves into the lane for which it was trained and decrease as the vehicle moves out of it. In both graphs though, the IRRE values remains well above the low confidence threshold. One cause of the increasing or decreasing confidence values is the extreme location of the SLV and DLV. Near the start and end of the transition, because each view remains centered over the proper lane, the actual camera viewing field may not overlay with the virtual camera viewing field. Although the missing virtual camera pixels are filled with the best actual camera pixel, the image is not quite consistent with what the network was trained with.

3.4.6 Lane Transition Experimental Parameters and Procedures

The same base virtual camera view was used to train all of the networks in each of the methods described. This means that each network "saw" the road or lane in the same manner. The virtual camera view used for these experiments was located 5.0 meters forward from the rear axle, 5.0 meters high, on the midline of the vehicle, and pitched downward from horizontal at an angle of 22 degrees. The actual camera was mounted inside the cab on the windshield, using the rear view mirror mounting post. It was located 2.11 meters forward from the rear axle, 1.36 meters above ground level, on the midline of the vehicle, and pitched downward at an angle of 10.67 degrees. Both the virtual and actual cameras were pointed straight ahead, i.e. they both had a yaw angle of 0 degrees. Also, all the networks were trained to drive with the same lookahead distance - 35 meters. Although using the same virtual view is not a requirement for any of the method to be described, training each network using the same lookahead distance is necessary.

The test site for all of the lane transition experiments presented here was the Pennsylvania Turnpike Commission's Safety Testing and Research (STAR) Facility in Breezewood, PA. The site is an abandoned, four lane section of the Pennsylvania Turnpike. For the experiments presented here, two lanes of the site were swept and repainted to look like a typical suburban interstate. Typical image of this road taken during a lane change maneuver are shown in Figure 28. All experiments were conducted on the Navlab 5 using the PANS platform. The vehicle speed was between 40 and 60 m.p.h.

To facilitate quantitative analysis of the lane transitions experiments, the live video signal used by ALVINN was recorded to videotape during the experiments. Overlayed on the video image was information regarding the type of lane transition and the step within each transition that was currently being executed. Network confidence and output displacement were recorded to disk along with the corresponding transition step. To determine the absolute position of the vehicle with respect to the center line of the road, the videotaped transitions were used. During each transition, the center line of the road was hand labeled.

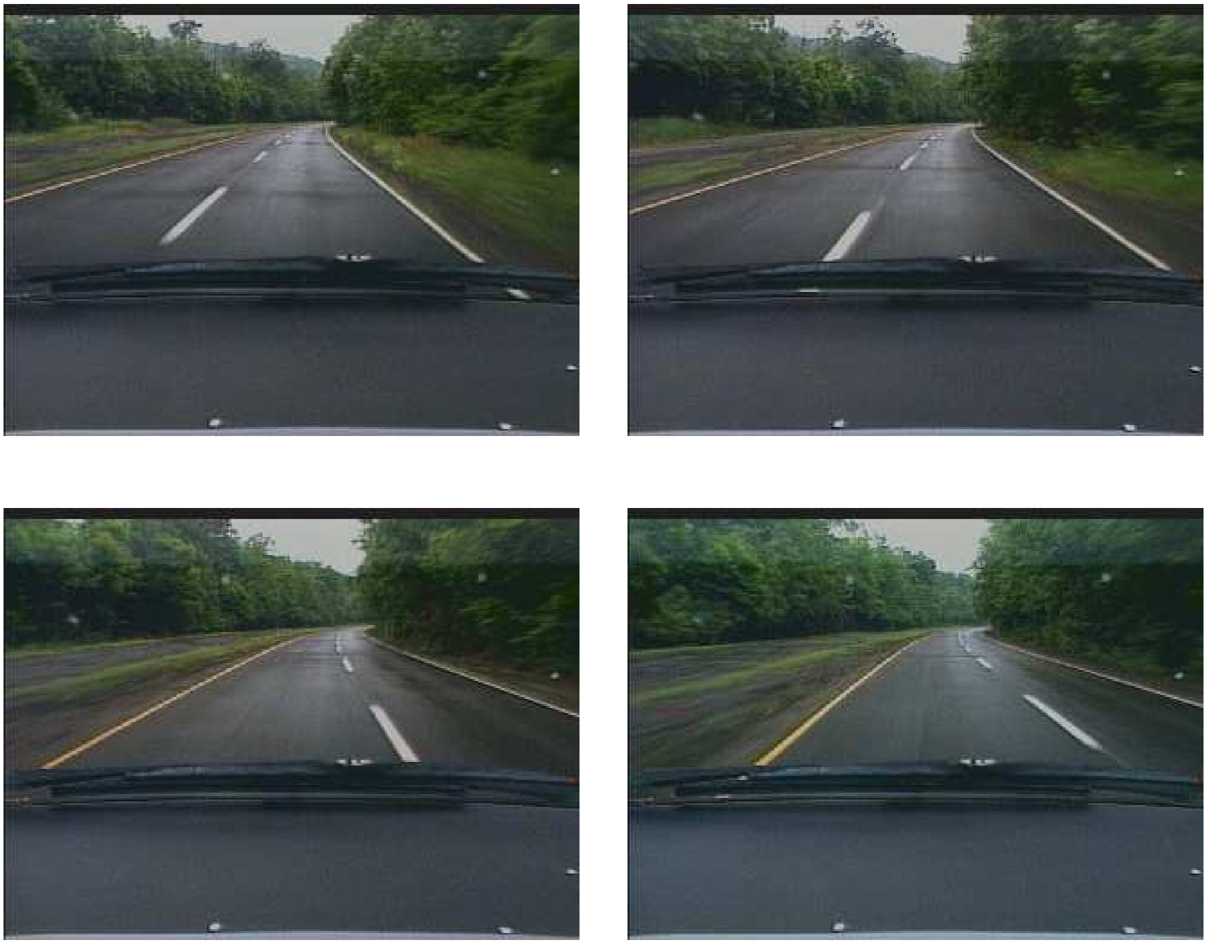


Figure 28 Pennsylvania Turnpike Commission STAR Site near Breezewood, PA.

3.5 Lateral Offset Driving

Although most systems are designed to keep the vehicle in the center of the driving lane, there are times when this behavior is not desirable. Obstacle avoidance is one such scenario. There are at least three reasons why moving the vehicle from the center of the driving lane when an obstacle is detected is important. The reasons relate to the capabilities of the obstacle detection system, the appearance of obstacle itself, and the capabilities of other cars on the roadway.

Many obstacle detection systems are currently being developed for use in the next generation passenger car. In these systems, which are mainly radar-based, the maximum detection distance of reasonably sized obstacles is very close to the minimum stopping distance for vehicles going the legal highway speed in the United States. If the systems do not detect the obstacle immediately and begin braking, the vehicle will not be able to avoid the obstacle by stopping.

The second reason why moving the vehicle from the center of the lane is important is that many obstacles will enter the driving lane at a distance closer than the minimum stopping distance of the vehicle. In this case, even immediate detection will not allow the vehicle to avoid a collision by stopping. In these cases, the only way to miss the obstacle is to swerve out of the current driving lane.

The final reason lane offset is important relates not to the capabilities of the vehicle equipped with the obstacle detection system, but rather to following vehicles which are not equipped with the systems. This type of “mixed-mode” traffic may be very common in the initial stages of a feasible Automated Highway System. In these situations, even if the detection system finds an upcoming obstacle in time to slow the vehicle to a stop, it may not be the safest maneuver because following vehicles may not have the same capability. They would not be able to stop and would either rear-end the stopped vehicle, or swerve out of the driving lane in order to avoid hitting it.

3.5.1 Swerve Experimental Results

Separate swerve systems do not need to be created to implement obstacle avoidance maneuvers. Active control of virtual camera views, similar to the techniques used for lane transition, can be used with current lane keeping systems to accomplish lane offset and swerve maneuvers.

To test this hypothesis, the IVLT technique presented earlier was modified into what is called the Swerve and Offset Driving Algorithm (SODA). The IVLT technique was selected as the base because it does not require two trained networks to initiate lateral movement of the vehicle, unlike the INS and DVLT techniques.

Instead of moving the vehicle completely into the destination lane like the IVLT technique does, SODA only offsets the vehicle from the center of the driving lane by a pre-specified amount, and then returns it to the normal driving position. This maneuver effectively allows the vehicle to swerve to miss an obstacle in the driving lane and is shown in Figure 29.

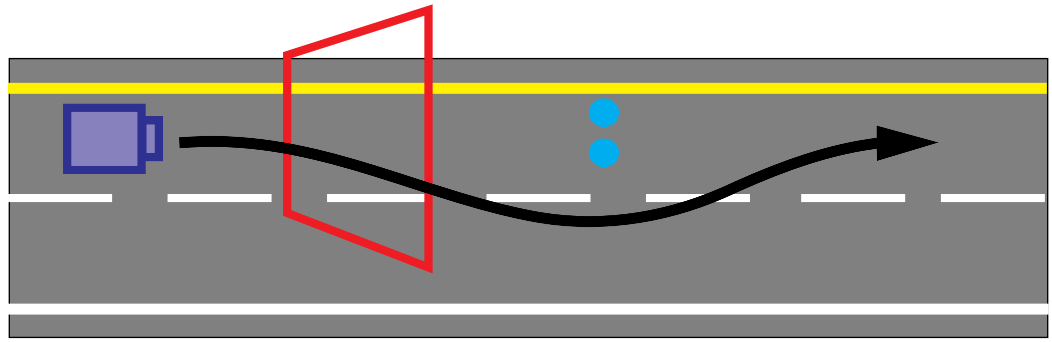


Figure 29 Vehicle movement while avoiding an obstacle.

SODA accomplishes the offset and return using a virtual camera control process, depicted in Figure 30, that is very similar to the one employed in the IVLT technique. The key differences are:

- Internal and external constraints are not enforced. This is done so that the maneuver takes place as quickly as possible. Although there is some lag between the virtual camera view location and the expected vehicle location due to the lack of coupling imposed by the constraints, the vehicle executes a smooth, stable maneuver.
- After the vehicle has reached the "apex" of the maneuver, it does not begin to return to the destination lane immediately, but rather drives with the specified offset for a predetermined distance. This "hold time" allows for some margin of error with respect to the actual location of the obstacle.

Approximately 10 left to right and 10 right to left swerves were attempted at the Breezewood test site. All were successfully completed. The obstacle was "detected" at distances between 40 and 60 meters by the test driver. After the test driver indicated an obstacle was present, the system moved the vehicle to an offset of 2.25 meters from the driving lane position using 8 virtual camera views. The system held the vehicle at this position for 0.5 seconds, and then returned it to its normal position in the driving lane. A condensed version of the maneuver, showing the virtual camera views and the vehicle's position relative to them, is shown in Figure 31.

3.5.2 Offset Driving Experimental Results

An additional experiment was conducted at the Breezewood site to test the feasibility of controlling the vehicle for an extended period while driving with a large offset from the lane which

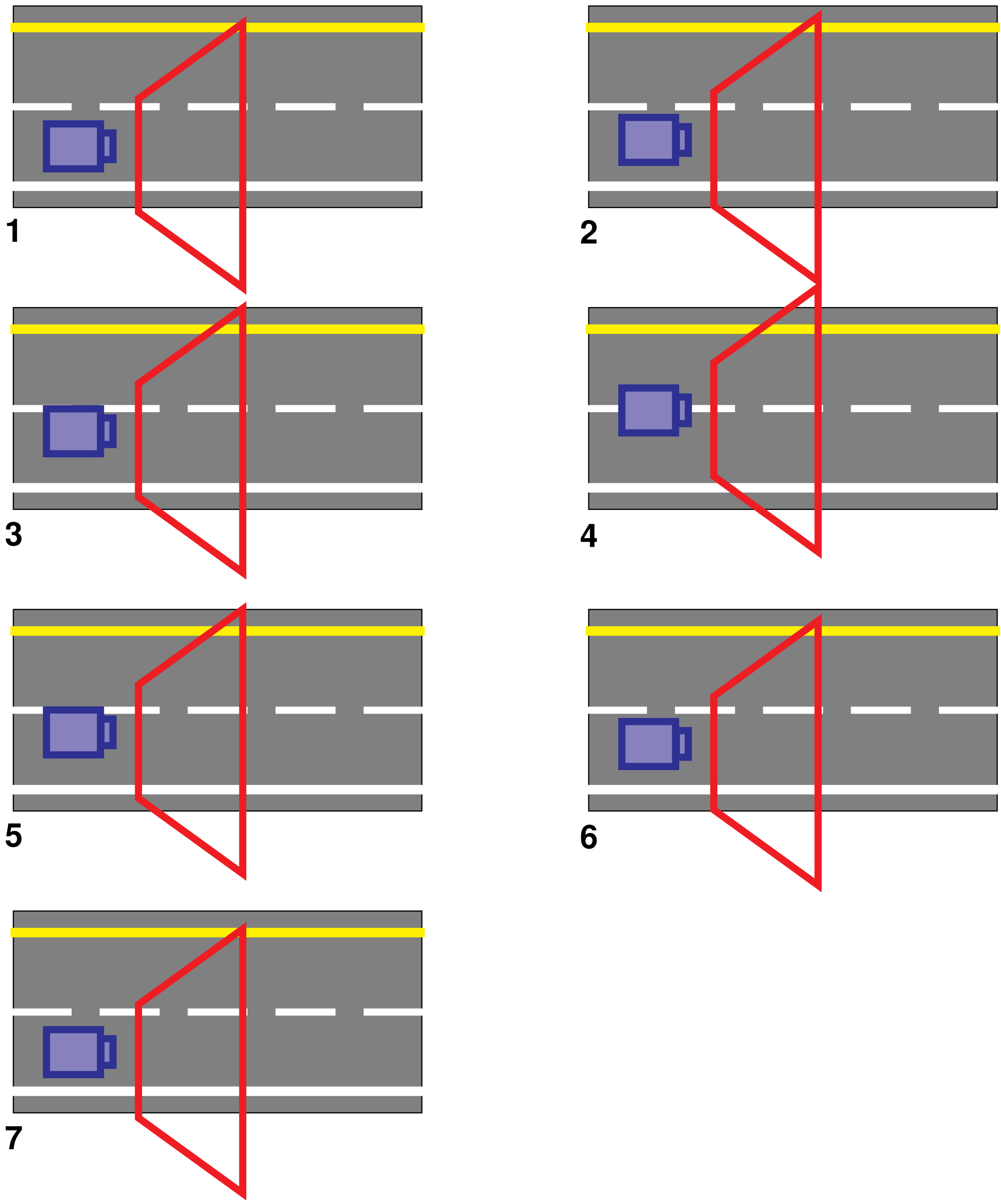


Figure 30 Vehicle and virtual camera locations during SODA maneuver.

the ALVINN network was originally trained. This type of behavior would be useful in circumstances where an emergency maneuver out of the driving lane and to a stop is necessary, such as when an accident is blocking the road ahead and the shoulder is the only escape route. The tech-

Image 1

Image 2

Image 3

Image 4

Image 5

Image 6

Figure 31 In-vehicle view of SODA maneuver.

nique could also be used in situations like road construction, where driving in the breakdown lane is necessary.

This maneuver was accomplished using the SODA algorithm, but by increasing the hold time to about 25 seconds. In this mode, the system was able to control the vehicle for over 580 meters (until the hold time had expired) and return it to the driving lane. Although performance was not as stable as when driving in the normal lane, it was well within acceptable limits, especially considering that this type of maneuver would be executed only as a last alternative.

3.6 Exit Ramp Detection and Traversal

A scenario which requires similar vehicle control schemes is exit ramp detection and traversal. This technique differs from simple lane transition because the exact location of the exit ramp is not usually known with enough precision to blindly move the vehicle onto it. The philosophy that has been adopted to accomplish this task is to:

1. Depend on other modules with higher level knowledge, like a map, to signal ALVINN when an exit ramp is approaching.
2. Use a virtual camera to preview the area of the scene where the exit ramp is expected.
3. Monitor the internal and external confidence measurements of the driving and exit lane networks which indicate the presence of an exit ramp.
4. Move the vehicle onto the exit ramp using lane transition techniques similar to those described earlier.

To accomplish this task using ALVINN, at least two trained networks are needed - one for the driving lane and another for the exit lane. (A third network may be needed to drive the vehicle down the exit lane once the vehicle has left the highway.) The road scenario is a two lane interstate highway where the exit lane is situated to the right of the right lane. ALVINN is driving the vehicle in the right lane.

Detection of the exit lane is the first sub-task which needs to be accomplished. Some distance before the exit lane begins, ALVINN receives a signal from a high level module informing it that the exit is approaching. Information about the exit lane's location relative to the vehicle, and type is passed to ALVINN. The information is used to create an appropriate virtual camera view, the

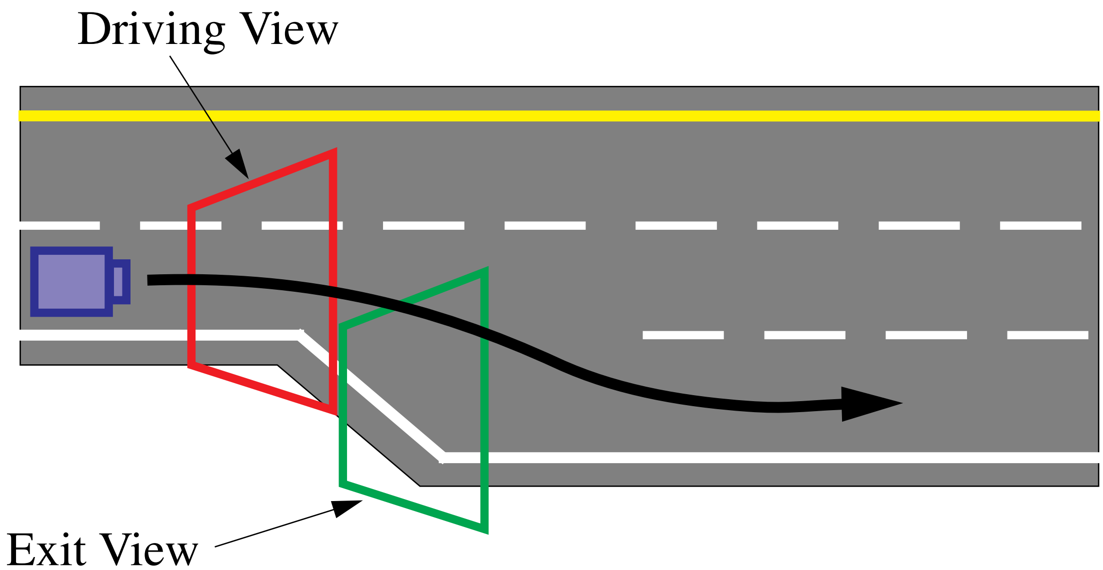


Figure 32 Exit ramp detection diagram.

Exit View, and associate the correct network with it. In addition to being laterally offset from the Driving view, the Exit View is also shifted forward by 10 meters. This is done so that the exit lane can be detected before the vehicle is adjacent to the lane and is important because in some situations the driving lane network learns to key off features, like the shoulder to road boundary, which change significantly when the exit lane appears. To maintain proper control in these situations it is important to detect the lane before it is imaged by the Driving View. A diagram of the roadway near the exit and virtual camera view locations is depicted in Figure 32.

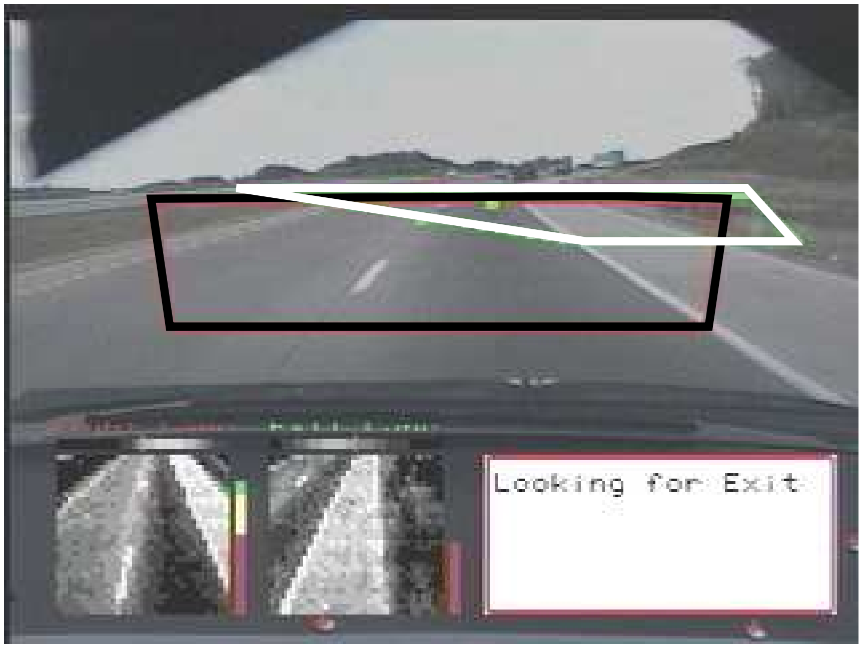
While the system uses the Driving View to keep the vehicle in its lane, images are created from the Exit View and passed to the associated ALVINN network which creates an output steering displacement and IRRE confidence measure. By monitoring the confidence and output steering displacement of the exit lane network using the methods described in earlier sections of this chapter, the system is able to determine when the exit lane begins. The IRRE confidence of the exit lane network will become high and its output displacement will be the same as that of the driving view when the exit ramp is present. (The displacements will match because the driving and exit lanes are parallel and the Exit View is not rotated with respect to the road.)

Note that for this scenario, high level knowledge does not tell the system where or how to exit, but only what type of exit is approaching. Figure 33 shows the scene, virtual camera image, and network response when the exit lane is and is not present. The Driving View is outlined in black and the preprocessed image created from it along with its associated output displacement vector and bar graph representation of its IRRE measure is shown in the lower left corner of each image. The Exit View is outlined in white and the items associated with it are to the right of those for the Driving View. In the top image, where the exit lane is not present, both the internal and external constraints are not satisfied, namely the IRRE confidence value is low and the output displacement vector does not match that of the Driving View. In the lower image where the exit lane is present, both constraints are satisfied, which is indicated by the higher Exit View IRRE value and the nearly identical output displacements from both networks.

After the exit lane has been detected, the driving lane network is no longer used and the vehicle is controlled using only the output of the exit lane network. The Exit View is incrementally shifted both laterally and longitudinally toward its standard location in front of the vehicle. The lateral component of each shift results in an image in which it appears that the vehicle is offset to the left of its proper driving position in the exit lane. To recenter the vehicle, the network produces an output indicating that the vehicle should steer to the right, recentering it with respect to the image and moving it further into the exit lane. Like for IVLT's, the network confidence for each new virtual view is required to be above a threshold for two iterations before the view is shifted again. This process continues until the Exit View has reached its standard position and the vehicle has completely transitioned into the exit lane.

3.6.1 Exit Ramp Detection and Traversal Experimental Results

Of the 20 attempts at detecting and traversing exit lanes on a rural interstate highway, 19 were successful. The attempts were approximately evenly spread across three different ramps. The same driving and exit lane networks were used for every attempt, except for the single failure, in which a different exit lane network was used (and was the cause of the failure). Exit ramp detection occurred at approximately 55 m.p.h., while traversal was done at speeds between 35 and 50 m.p.h.



Driving Lane Exit Lane

Driving Lane Exit Lane

Figure 33 Exit ramp detection images.

Exit lane detection was robust and consistent, occurring immediately after the lane appeared centered in the Exit View. The system was instructed to begin looking for the exit lane between 100 and 800 meters before it actually occurred, and to begin transitioning onto it as soon as it was found. Finally, there were no false positives, in which the system incorrectly believed that the exit lane was present.

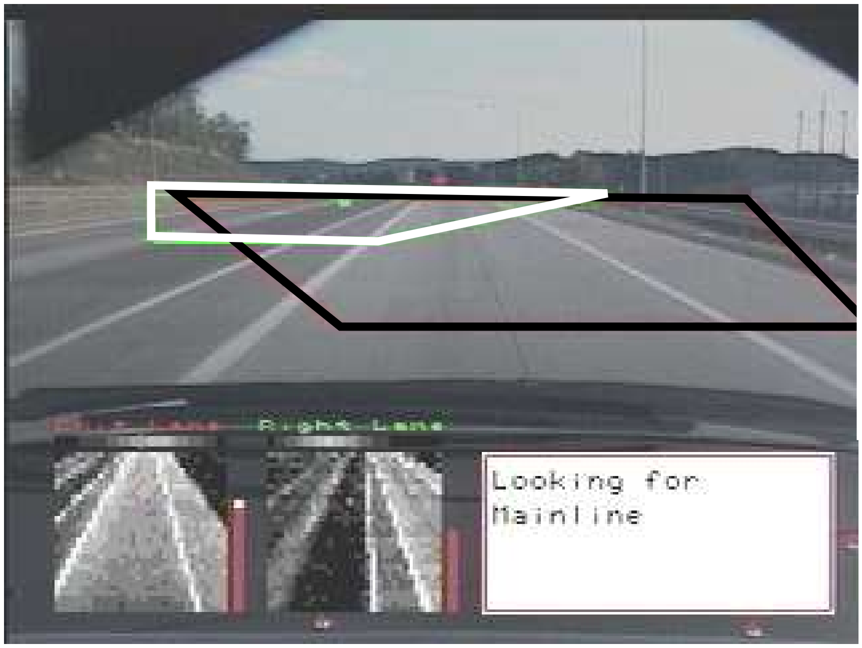
The system performance differed the most in the traversal part of the task. This was a result of the exit lane network and view that was used. This view was positioned so that the network would learn the most visible feature in the scene - the shoulder to offroad boundary. (The white lines on the concrete roadway were not a large enough feature and the exit lane was too lightly travelled for noticeable oil spots to be created.) The result of this choice of view meant that when the shoulder width changed, navigation performance would suffer. In practice, this lead to consistent overshoot during navigation of one of the three exits in which the shoulder was wider than that on which the network was trained.

3.6.2 Mainline Detection and Traversal

A task which is almost identical to exit ramp detection and traversal is mainline detection and traversal. This task refers to the act of moving along an entrance ramp, locating the right hand driving lane of the highway and transitioning onto it. It differs from exit ramp detection and traversal in that the view which is looking for the right hand lane is offset to the left of the diving view instead of the right and that after transition has occurred, the vehicle will be in a lane which has neighboring lane to the right and left.

Experimentation with this maneuver was conducted on a single entrance ramp using the same networks that were used for the exit ramp experiments. The exit lane network was used to drive the vehicle up the entrance ramp and the driving lane network was used to detect and travel onto the right lane of the interstate highway. The same algorithm used for exit ramp detection and traversal was used to find and move the vehicle into the right hand driving lane.

Figure 34 shows two images both before and after the right hand lane of the interstate highway was detected.



Driving Lane   Entrance Lane

Driving Lane   Entrance Lane

Figure 34 Entrance ramp detection images.

3.7 Lane Detection

In addition to being used to control the vehicle, virtual camera views can be used with ALVINN to signal the presence or absence of adjacent driving lanes. The lane transition and exit/entrance ramp scenarios described earlier in this chapter are two examples where this capability was used to determine if an action was appropriate. This idea is based on using the two metrics that are available which measure the reliability and confidence of ALVINN's output and was partially explained in earlier sections of this chapter.

To look for another driving lane, a network trained for the lane to be checked is needed and a virtual camera is placed so that it properly images the hypothesis road or lane location. An image is created from the virtual camera and is given to the network trained for that road or lane. The network produces an output displacement along with its associated IRRE confidence in this output. If the IRRE confidence is low, meaning that the image is not familiar to the network, two things are possible.

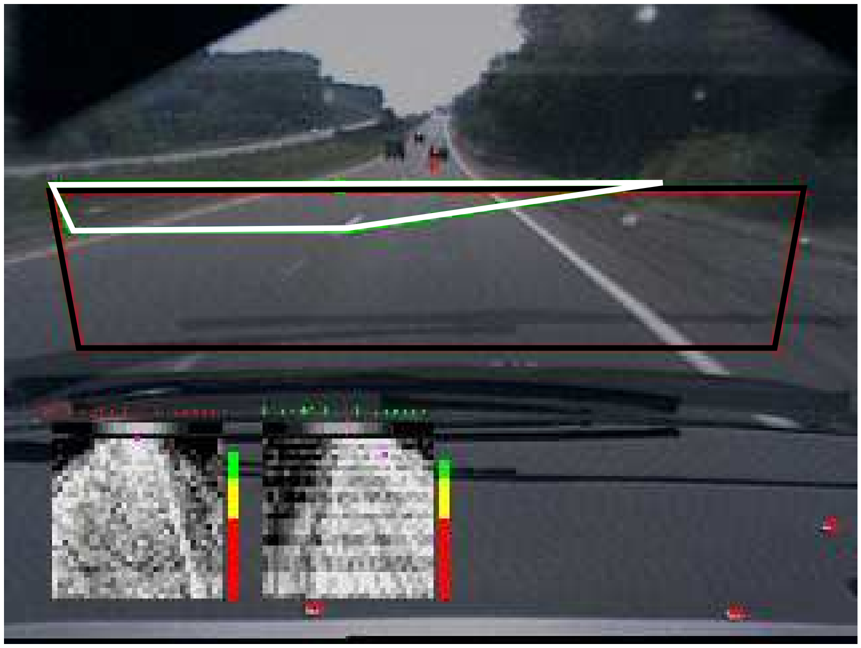
1. There is no lane present in the area that the virtual camera imaged.
2. There is a lane present, but the network used with the image created from the virtual camera view is not capable of driving in it.

In either case there is a mismatch between what was expected and what was present. Although, low confidence values are fairly indicative of the absence of another lane or incorrect network usage, high IRRE confidence values cannot be generally thought of as absolutely indicating the presence of another lane. An example of this phenomenon is shown in Figure 35. In this figure, a virtual camera view is being used to determine if another lane is present to the left of the driving lane. The top image shows the scene and the preprocessed driving and preview images created from the virtual camera views. In this case, the lane is present and as expected, the IRRE confidence value, shown to the right of preview image, is high. The bottom image of the figure shows a similar scene, except that now no lane is present. In this case, the network still responds with a high IRRE value. To overcome this inconsistency, the geometric constraints of the problem can be used. The constraints are:

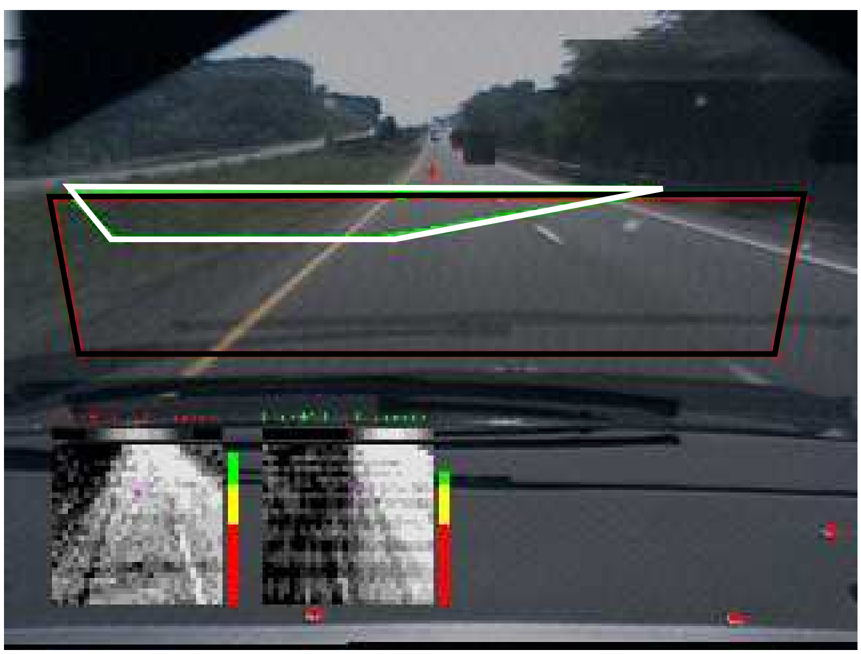
1. The location of the virtual camera view with respect to the vehicle is known.
2. The lane separation distance is known and assumed to be constant.

3. The road curvature is known from the current driving view.

These constraints mean that the output displacement of the network associated with the lane that is being checked can be validated against the actual driving displacement. Also visible in Figure 35 are the network output steering vectors for the driving and preview networks. For both images, the output steering vector of the associated network is shown immediately above the preprocessed image. In the top image, the output steering vector associated with the preprocessed image that contains the road is almost identical to the output of the network which is controlling the vehicle. But in the bottom image, in which the preview virtual camera is not imaging the road, the output displacement is very different. Because the shoulder looks similar to the road, the network responds with a high confidence value, but by examining the output displacement vector of the preview view, it is possible to determine that no lane is actually present.



Driving Lane Preview



Driving Lane Preview

Figure 35 Lane detection scenario.

Chapter 4 Intersection Navigation

4.1 Introduction

A next step in the evolution of autonomous driving systems is the intelligent handling of road junctions and intersections. The techniques presented in this chapter are based on a data driven, active philosophy of vision based intersection detection and traversal. This chapter describes the application of Virtual Active Vision tools to this area and presents the algorithms that make autonomous intersection navigation possible. The capability is based on geometrically modelling the world, utilizing this model to accurately image the relevant parts of the scene using Virtual Active Vision tools in conjunction with active camera control, and monitoring ALVINN's response to the created images.

4.2 Detection Philosophy

There are three principles upon which vision based intersection navigation systems should be based. They are:

1. Detection and navigation should be data (image) driven.
2. Detection is signaled by the presence of features.
3. Traversal should occur by active tracking of the intersection branch.

These principles and their relationship to ALVINN, enhanced with Virtual Active Vision tools, as well as to other road and intersection detection systems, are examined in greater detail in the following sections.

4.2.1 Data vs. Model Driven Detection

Detection and navigation should be data directed, rather than model directed because the data contained in the image represents the vehicle's world now, rather than when the model was built. The pixels from the current image should be used to determine if an intersection is present. A data directed method allows the robot to use features from its surroundings to compensate for errors in any prior information it has been given or to discover new information about its environment. For example, in a strictly model directed system, position estimation must be very accurate - small errors can cause the system to miss the desired intersection. But in a data directed scheme, the system can use images of its current surroundings to locate the desired intersection.

A data driven method does not exclude using some type of model to assist in determining where a road or intersection is likely to occur. In the example above, a coarse model supplied by a TLDM could be used to inform the system that an intersection is likely to be reached in the next mile or two. When the system receives this information, it could begin searching for it. In this way, the model allows the system to focus its resources on the most important task - staying on the road - until they are needed for some other task, like finding the intersection. Another example is that a model could provide information about the type of intersection that is approaching, and thus help the system to determine which areas of the input image contain information required for detection. Concisely, the model only guides, rather than executes, the act of detection.

In contrast, in a purely model directed scheme, intersection and road location data from the model would be used exclusively to navigate onto the road or through the intersection. In this scheme the model would contain information like: at latitude -79 degrees, longitude 40 degrees, execute a turn of radius 50 meters. In this scenario, detection would not occur - the robot just monitors its position and executes the command associated with that position. This type of system depends heavily on accurate global positioning and model data which is typically very difficult to acquire.

4.2.2 Detection Signaling

Positive detection should be signaled by the presence of the intersection as opposed to the absence of some other feature. This means that positive detection is signaled when the system "sees" an intersection branch as opposed to when it does not "see," for example, the center line of the road. (An absent center line often indicates that the vehicle is at an intersection.) Human intersection detection is a good example of this principle. A person does not stare at the center line of the road, waiting for it to disappear, to determine when an intersection is present. A person scans the scene looking for features, like intersecting roads or street signs, which indicate that an intersection is present. In a computer vision based system this principle translates to using the capabilities of the system to derive useful information about the scene, perhaps by actively searching for it, from images, rather than relying on the absence of information.

4.2.3 Active Search

When a person turns onto a road or traverses an intersection, they do not take one look at the intersection, make a mental picture of where they want to go, close their eyes, and begin turning. If they did, no new data could be acquired to help correct any errors that occur. When a person finds the intersection they want to traverse, they continue to monitor it as they turn. In this way they can continually correct any errors in their initial judgement of how the situation should be handled. This is the model which should also be used in computer vision based intersection detection and traversal.

A person is able to monitor the road by turning their head so that the intersection branch that they are moving onto is continually in view. For computer vision based systems, this functionality can be accomplished in two ways. The first solution is to use a pan/tilt platform to actively move the camera as the vehicle traverses the road junction. A system which uses this method must be very robust in order to handle the changing road appearance as the vehicle moves through the intersection. A second solution is to not move the actual camera, but to use virtual cameras to intelligently image parts of the scene so that the important areas can be successfully processed by the system. If the virtual cameras are placed correctly, the road detection system can be much simpler because the appearance of the road does not change. This method assumes that the actual camera, from which the virtual views are created, images a sufficiently large area of the intersection. If it does not, a combination of the two methods can be used. In this joint method, the desired placement of the virtual cameras would guide the movement of the actual camera.

4.2.4 ALVINN

ALVINN, enhanced with Virtual Active Vision tools, can be used to detect and traverse road junctions and intersections in two different scenarios. In the first scenario, the system only has knowledge that an intersection is present in front of the vehicle. The system does not know the orientation of road branches that are intersecting at this road junction. In this scenario, the goal is to locate each intersection branch.

In the second scenario, ALVINN has a priori knowledge about an upcoming intersection. The information does not specify where the intersection is actually located, but only that it is approaching. Using this information, appropriate virtual camera views can be created and correct networks can be associated with each. The location and orientation of each virtual camera, and the type of network used with each, is dependent upon the type of road that is expected to be encountered. When the road or intersection to be detected is present, the virtual cameras will image it in a way that is meaningful to ALVINN's neural network. By continually monitoring the network's confidence for each virtual camera image, the system can determine when the intersection is present.

4.3 Other Systems

Several other researchers have built systems to study the road and intersection detection problem, but few have explored actual intersection traversal after detection. Many of them have adopted a data directed approach, but many also rely on the absence of features rather than the presence of them to indicate when an intersection is present. Few use active camera control. The following sections present a sampling of these systems.

4.3.1 ALV Project Intersection Detection

Early work in intersection detection was done at the University of Maryland as part of the Autonomous Land Vehicle project. The system developed was unique in that instead of being tested on an actual vehicle, it was tested on a road network map developed by the U.S. Army Engineering Topographic Laboratories. This system used a video camera to recognize road intersections by matching current road edges to a stored model which contained data about intersection branch orientation and branch width. This system was tested on variants of a basic “Y” intersection and good detection results were achieved. Unfortunately, the system was only tested in simulation [30].

4.3.2 SCARF

An early driving system which also examined the intersection detection problem was SCARF (Supervised Classification Applied to Road Following). SCARF classified pixels as belonging to either road or non-road classes based on their color. Using this information and a trapezoidal road model, it was able to drive on paved paths near the Carnegie Mellon campus. The method SCARF used to detect intersections closely matches the philosophy described earlier. It used image data to locate intersections by creating image plane masks which corresponded to the shape and orientation of the intersection branch to be detected. The mask was created from a priori knowledge about how branches intersected the main road. This system did not have an a priori model of the intersection structure, but rather searched for possible branches during every iteration [6].

4.3.3 YARF

YARF (Yet Another Road Follower) is a model based road following system. It uses yellow and white line detectors along with a local road model to drive on city streets and interstate highways. The system can also be used to detect intersections. In this mode, the system uses the absence of the current road center line to indicate when an intersection is present. Then, using model data about the current intersection type and a second fixed camera, it positions its detectors in locations where the intersecting road’s features are likely to be located. Because of the second camera, YARF should be able to detect the intersection of lined roads in any geometric alignment. In practice though, this was not the case, because the second camera was fixed at one orientation before the experiments. The detection phase of YARF was tested in real world situations, but actual traversal though intersections was done only in simulation [28].

4.3.4 Old ALVINN

In the original implementation of ALVINN, simple intersection detection and navigation was accomplished using a coarse map and by monitoring the OARE error of the system's neural network. When the system noticed a sharp increase in this error metric, indirectly caused by the absence of a consistent driving features, it signaled to a mapping module that an intersection was present. At this point, the output vector of the neural network was examined for a bi-modality. If one was found, the reply from the mapping module provided information about which intersection branch, indicated by one of the modes in the output, to follow. The system used a single, fixed camera and was constrained to detecting and navigating only intersections which produced an increase in the OARE error and a multi-modal output. This meant that the intersection branches must fall within the field of view of the camera and that the orientation of the branches must be within the neural network's range of response [40].

4.3.5 VaMoRs Intersection Detection and Traversal

This module is a component of a mission based navigation system which uses coarse maps to direct low level vehicle actions. It detects intersections by searching the image for pairs of nearly horizontal road edges whose separation is indicative of a road branch. After the intersection has been detected, the module computes a traversal path to follow using a simple intersection model. This path is symmetric and accounts for the static and dynamic properties of the vehicle. It is unclear whether this path is updated during traversal to reflect new branch position information. As a safety precaution as the vehicle turns, an obstacle detection system directs the motion of a moveable camera so that it always images the upcoming portion of the road [33].

4.3.6 IITB Intersection Detection and Traversal

This system combines tracking of road boundaries and a digital map to determine upcoming road and intersection geometry. It also uses this information to select the appropriate camera to use to image the road geometry for the task to be performed. Like in the YARF system, initial detection of intersections is signified by gaps in road features. But the main feature that this system uses to detect intersections is the boundary of the cross road whose orientation is known from the digital map. As the vehicle moves through the detected intersection, the system automatically switches to the most appropriate camera. It is not clear from the literature whether the system is actively controlling the vehicle during the traversal maneuver or whether a predefined steering angle is being executed [48][51].

4.3.7 Intersection Detection for the Driver's Warning Assistant

This system is designed to warn drivers when they are approaching intersections too quickly. It tracks road edges and uses this information to infer where an intersection is likely to appear in the image. This area is searched for horizontal edges which are matched to one of the system's two internal intersection models. The intersection is considered detected when the match value is greater than a threshold. Although this system looks for positive indicators of intersections, it is very reliant on models which are not based on geometric properties of the intersection, such as its branches and their orientation. Instead, it depends on markings painted in the roadway or in the center of the intersection [12][46][55].

4.4 Experimental Overview

Two sets of experiments were conducted to assess the usefulness of virtual cameras for autonomously detecting roads and intersections. The first set of experiments was performed on the Navlab 2. The site for these experiments was a single lane paved path near the Carnegie Mellon campus. The path was unlined with grass on either side and was 3.1 meters wide. The system used a single color camera mounted 2.1 meters high, 2.5 meters forward from the rear axle and 0.62 meters to the right of the vehicle center line. The goal of this set of experiments was to test the basic ability of virtual cameras to create images which were usable by the system for intersection detection and to assess fixed camera traversal techniques. For this, a single virtual view was used to detect an upcoming road and to navigate onto it.

The second set of experiments used the Navlab 5 vehicle. This vehicle was also equipped with a color camera, but this camera was located on a roof-mounted pan-tilt platform, approximately 1.86 meters high, 1.06 meters forward from the rear axle and 0.02 meters to the right of the vehicle center line. This experimental set was conducted both on the paved path described above and on a rural road in suburban Pittsburgh. In this set of experiments, the goal was to use active camera control to enhance the performance of the detection and traversal algorithms.

4.5 Intersection Detection and Traversal Experiments on the Navlab 2

This experimental set was designed to test the system's robustness for detecting and navigating onto single roads. For the first experiment in the set, the system knew that the vehicle was approaching a road perpendicularly. Its function was to detect the road, drive the vehicle onto it, and continue operating in a normal autonomous driving mode. For this experiment, the vehicle was not on another road as it approached the road to be detected. This scenario could correspond to ending a cross country navigation mission and acquiring a road to begin autonomous road following.

Initially, the vehicle was positioned approximately 35 meters off the road which was to be detected, and aligned perpendicularly to it. A virtual view rotated 90 degrees to the right of straight ahead was created. This view was placed 20 meters in front of the vehicle to allow enough time for turning onto the road after detection occurred. See Figure 36. The vehicle was instructed to move along its current heading until the system detected the road. At this point, the system instructed the vehicle to turn appropriately, based on the point specified by the neural network. Once the system had aligned the vehicle sufficiently with the road, it was instructed to begin road following. Results from the detection and alignment phases of this experiment are presented in more detail in the following sections.

4.5.1 Road Detection

The ability to detect the upcoming road was the first and most important requirement of the system. To accomplish this, every 0.3 second as the vehicle approached the road (at a speed of about 5 m.p.h.), a virtual image was created and passed to the system's neural network. The network produced an output vector, interpreted as a point on the road to drive over, and a confidence value using the Input Reconstruction Reliability Estimation (IRRE) metric. This metric was described in greater detail in Section 1.4. To determine when the system had actually located the road, the IRRE metric was monitored. When this metric increased above a user defined threshold value, which was typically 0.8 out of 1.0, ALVINN reported that it had located the road.

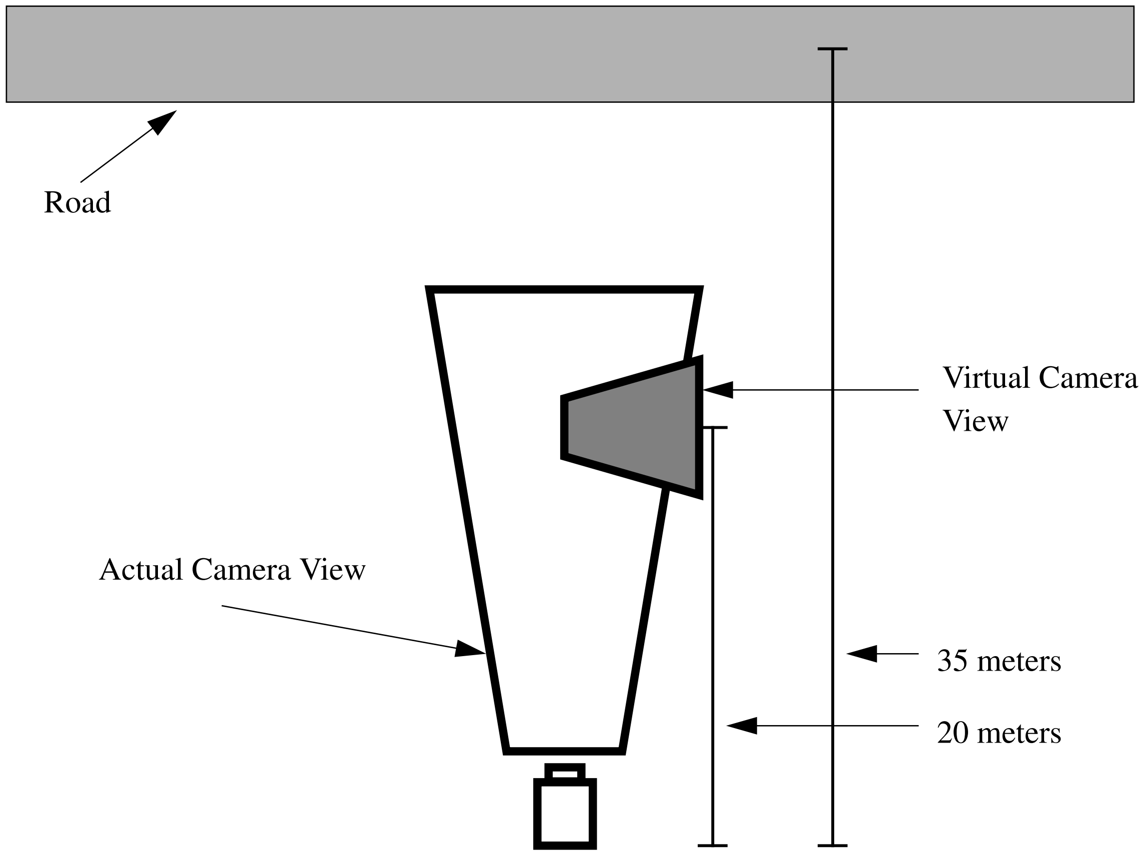


Figure 36 Road detection scenario.

4.5.2 Application of IRRE to Road Detection

Using the IRRE metric to indicate when roads are present in the input virtual image assumes that the metric will be low for images which do not contain roads and distinctly higher for those that do. For this assumption to hold, two things must occur. First, ALVINN’s neural network must not be able to accurately reconstruct images which do not contain roads, leading to a low IRRE measure. Second, images created by the virtual camera when a road is present must look sufficiently similar to ones seen during training, thus leading to an accurate reconstruction and a high IRRE response.

Figure 37 shows two actual camera scenes taken at different distances from the road. In these images, the virtual view is at the same distance in front of the vehicle. The top image shows the scene in front of vehicle as viewed by the actual camera just before the virtual view, outlined in red, images the road. The small image in the lower right is the preprocessed virtual camera view

Figure 37 Virtual cameras imaging an upcoming road.

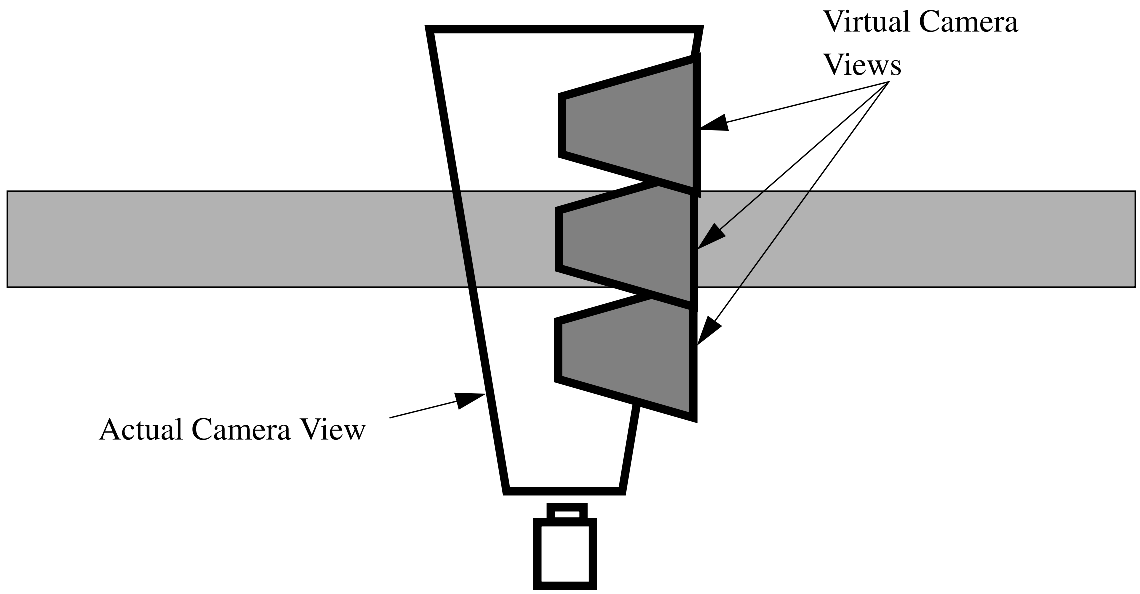


Figure 38 Virtual camera spacing.

that the system's network uses. The bottom image shows the same scene when the vehicle is a few meters closer to the road. In this scene the virtual view is aligned with the road and the image it creates looks very natural. Note that the actual view does not entirely contain the virtual view. By examining the two virtual images shown in this figure, it is reasonable to expect that the assumptions stated above will hold. The virtual image in the upper scene, which is imaging grass, looks nothing like a typical one lane, unlined road. Therefore, it is likely that the IRRE response would be low. For the bottom scene, the virtual image does look like a typical one lane road, and it would be reasonable to expect a much higher IRRE value.

To further test these assumptions several image were taken at various distances from the road as the vehicle approached. In each of these images, the location of the virtual camera was moved so that it imaged areas between the vehicle and the road, on the road, and past the road. Figure 38 illustrates the location of three typical virtual view locations created using a single actual image. Specifically, actual images were taken when the vehicle was at distances of 25, 20, 15, and 10 meters from the center of the road. Virtual camera images were created at 1 meter intervals on either side of the expected road location. For example, using the actual image taken 20 meters from the road center, virtual views were created every meter between the distances of 14 meters and 29 meters.

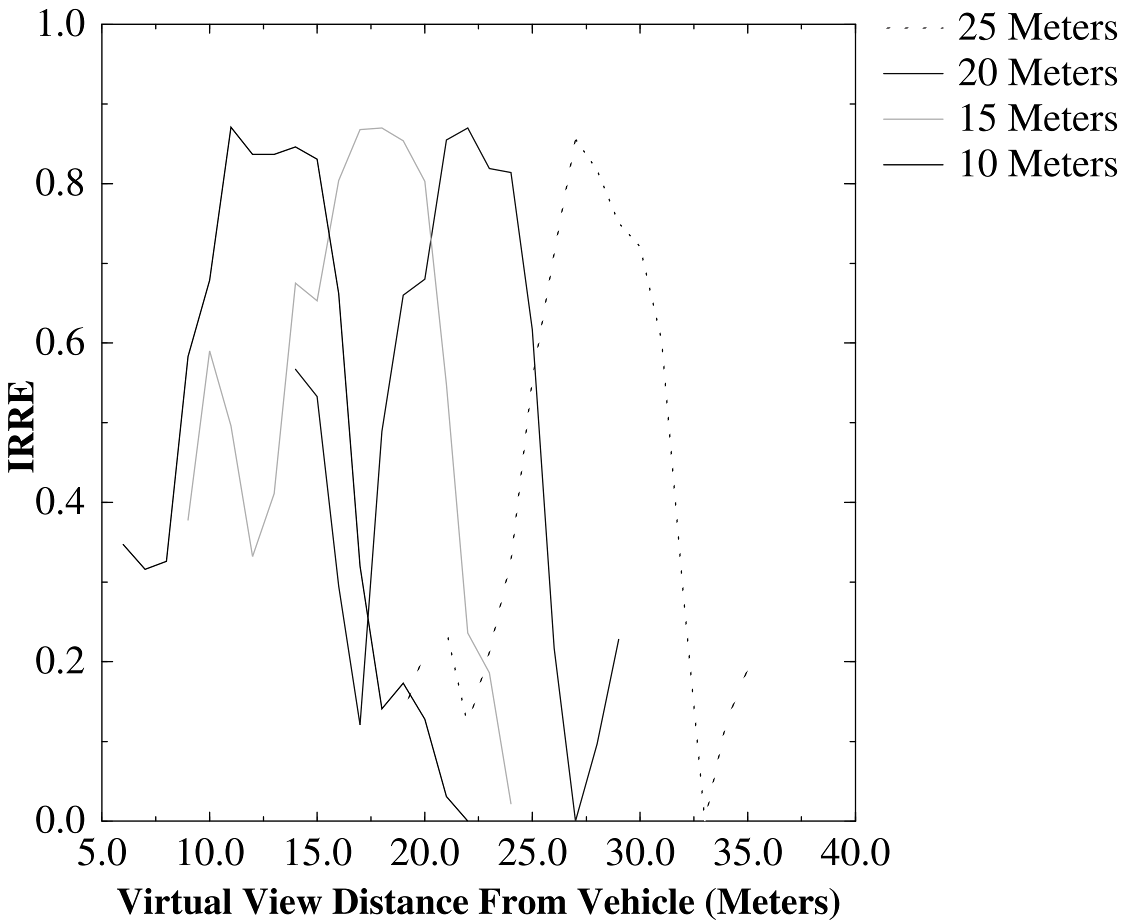


Figure 39 IRRE response to different camera images.

For each of the actual images, virtual camera images were created at the interval specified above and shown to a network previously trained to drive on the one lane road. The output road location and the IRRE confidence metric were computed. The results of this experiment are shown in Figure 39. This figure shows the IRRE response as a function of the virtual camera distance in front of the vehicle for several actual images taken at different distances from the road. (Data from the different actual images is represented by different curves in the graph.) For each actual image, the network's IRRE response clearly peaks near the expected road distance. As the virtual view moves closer to the road, the IRRE response increases, peaking when the virtual view is directly over the road. Response quickly falls again after the view passes over the road. The peaks in all of the curves have IRRE values greater than 0.80. For comparison, when the system is driving on a familiar road, the IRRE response is typically between 0.65 and 0.95. The peaks in each IRRE curve actually occur about 2 meters past the actual road center. This is due to three things: a violation of the flat world assumption, errors in camera calibration, and improper initial alignment to the road.

This graph shows that both assumptions are basically correct - the IRRE response when the network is not being presented road images is low, and the IRRE response is high when the network is being presented accurately imaged virtual views.

The relationship between the input virtual image and the IRRE value associated with that image can be better seen in Figure 40. It shows virtual images created at different distances in front of the vehicle along with the IRRE response they solicit. The images are all from an actual image that was taken when the vehicle was 20 meters from the road center. The image in the upper left corner is very similar to the top image in Figure 37. The road is barely visible in the top left corner of this image and, as expected, the IRRE response is very low. As the virtual view is moved forward, it begins to image more of the road, as shown in the upper right image. The IRRE value increases correspondingly. The trend continues until the virtual view is centered over the road, as shown in the lower right image. At this location, the IRRE value is at its peak.

Each of the IRRE response curves shown in Figure 39 clearly indicate that a road is present at some distance in front of the vehicle. Because it is generally better to detect a road at a greater distance, it is desirable to know if accuracy in detection decreases as the distance from the vehicle to the road increases. Insight to this can be gained by transforming all of the curves in Figure 39 into the same reference frame. This can be done for each of the virtual views associated with a single actual image by subtracting the distance between the vehicle and the road center from the virtual camera location. This results in a coordinate system whose origin is at the center of the road. The result of transforming each of the response curves in Figure 39 into this coordinate frame is shown in Figure 41. This graph shows that detection accuracy, at least when approaching the road perpendicularly, does not degrade as distance from the road increases. Negative numbers are closer to the road and positive numbers are further away.

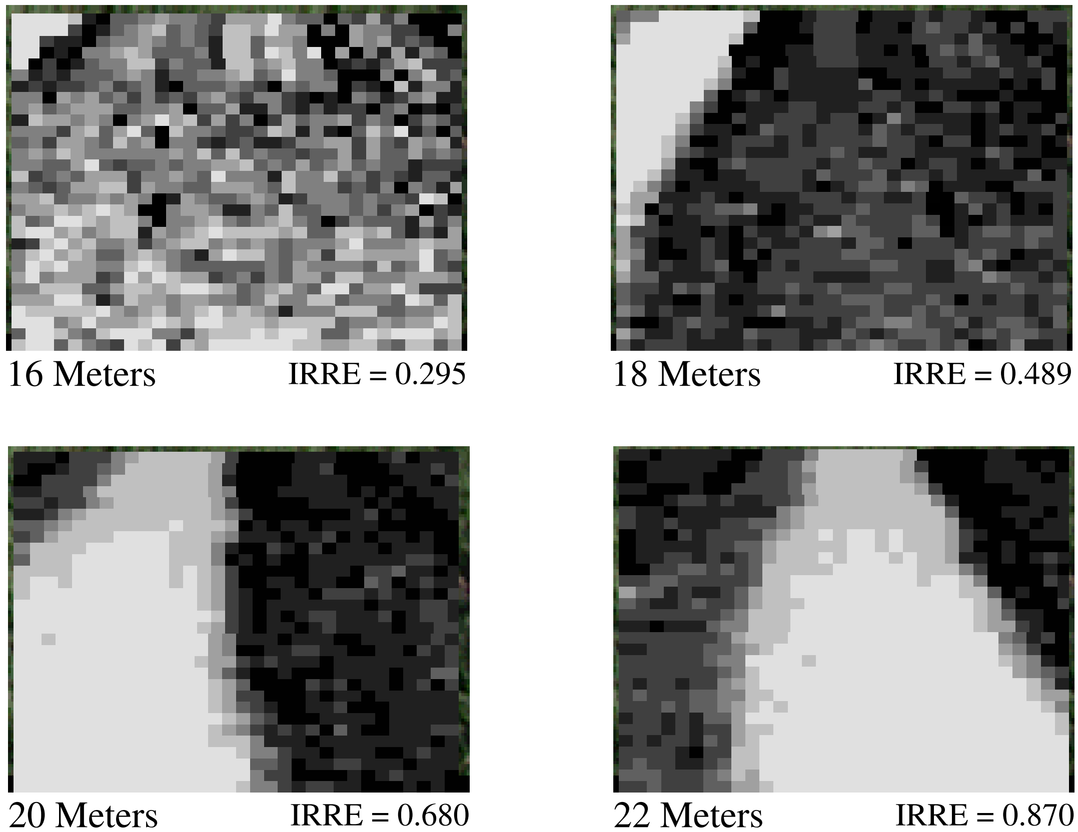


Figure 40 IRRE values for typical detection images.

4.5.3 Accuracy of Approach to Road

An important consideration in these experiments is that the vehicle does not need to approach the road at exactly the orientation used to create the virtual view for detection to occur. This is because ALVINN’s neural network is trained to produce the correct response even when the road is offset right or left of center of the vehicle, or when it is rotated with respect to the vehicle [37]. (This type of training is important for error recovery when the system is used to drive autonomously.) This is important because in real world situations, we rarely know the exact location of features, such as roads, with respect to the vehicle. This characteristic does have some drawbacks which will be explained in greater detail in Section 4.5.4.2.

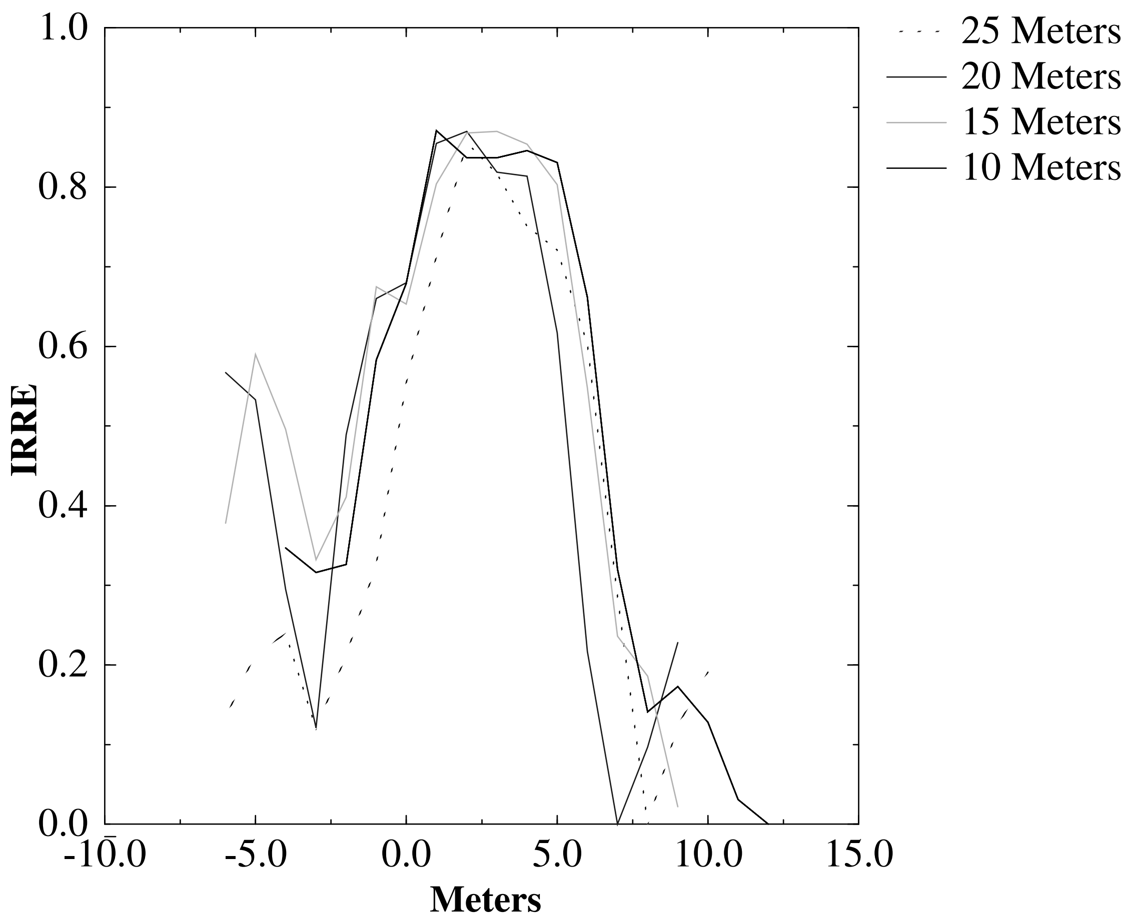


Figure 41 Transformed IRRE response curves.

4.5.4 Alignment

While testing the detection phase of the system, it became clear that the problem would not be detecting the road, but rather driving onto it after it was detected. The next sections detail the series algorithms used to drive the vehicle onto the road. The algorithms are presented in increasing order of robustness. The detection method described previously was used for finding the road for each method.

4.5.4.1 Simple Road Alignment

The first algorithm that was tested for moving the vehicle onto the road was to simply drive the vehicle over the point on the road which was specified by the system. For our vehicle, this meant that the center of the rear axle would pass over the specified road point. (The center of the

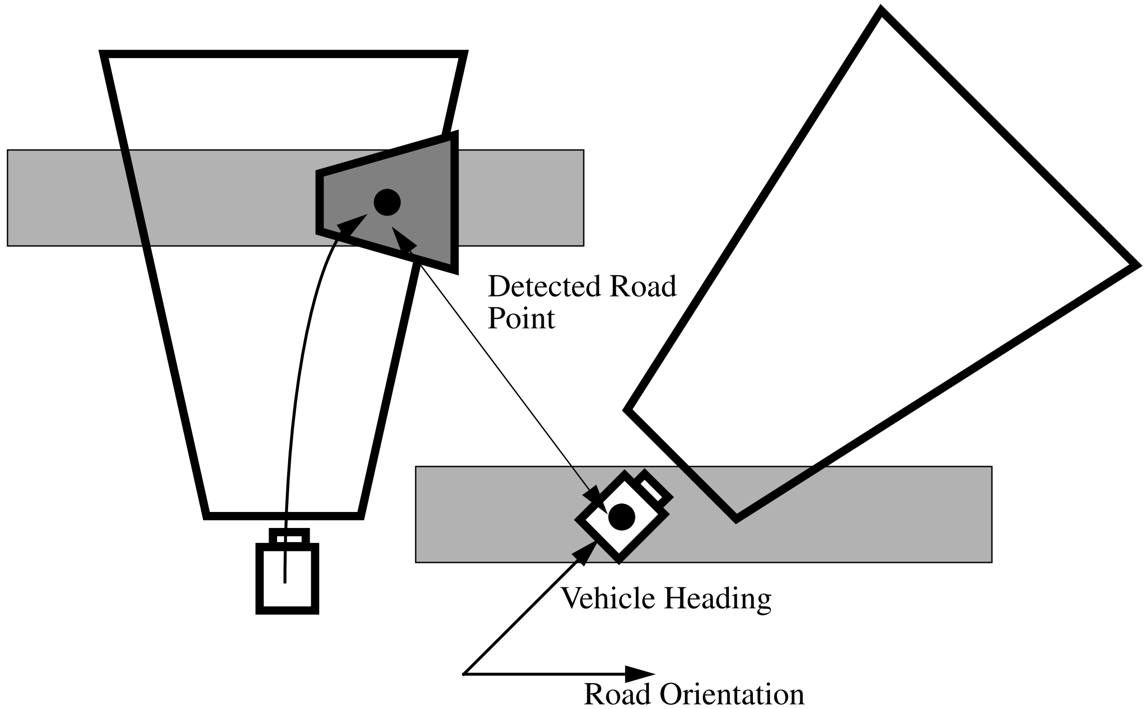


Figure 42 Vehicle to road misalignment.

rear axle is the origin of the vehicle coordinate system. Our point tracking algorithm uses this point as the location on the vehicle which should follow points to be tracked.)

The point tracking algorithm was able to reliably position the vehicle over the detected road point. The problem with this approach was that the vehicle heading was not matched with the road orientation. See Figure 42. This mismatch was a function of the angle at which the vehicle approached the road. Small angles led to minor misalignment, while larger ones caused significant deviations. Consequently, the vehicle was not able to begin road following after it had reached the road point because the road was no longer visible in the camera's field of view. One cause of this situation is that our point tracking algorithm, pure pursuit, does not attempt to match desired and actual headings. But even if it did, the combination of the computed road point location relative to the vehicle origin and the minimum turn radius of the vehicle would prevent proper alignment to the road in some cases. Basically, in many cases the road position computed using the virtual view does not provide enough lateral offset from the original direction of travel to allow the necessary turn to be executed and the vehicle to become properly realigned with the

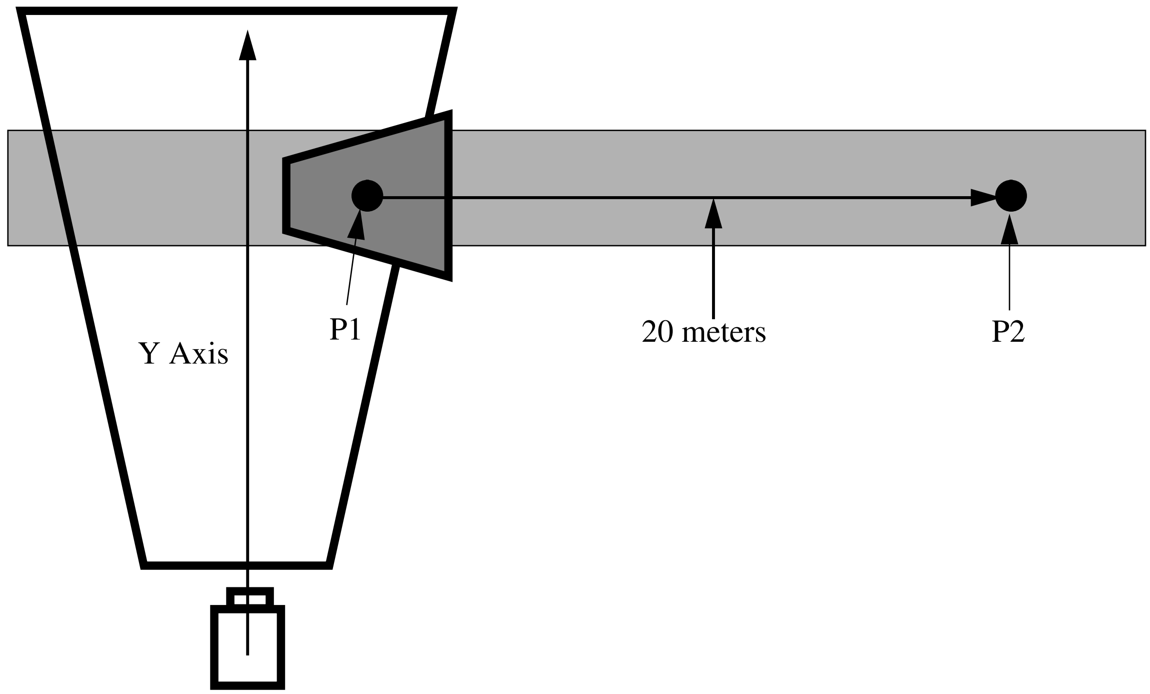


Figure 43 P2 point projection diagram.

road. The virtual view, and in turn, the computed road position, is constrained by the actual camera - a considerable portion of the virtual view must be within the actual camera's field of view in order for realistic images to be created. This result suggests that another point on the road, further along it, in the desired direction of travel, is needed.

4.5.4.2 Alignment by Projecting Along Road

To remedy the heading match problem encountered in the previous experiment, another point, P2, was created in addition to the network's output road location, P1. See Figure 43. P2 was created using information about the orientation of the virtual view with respect to the vehicle. By doing this, it is assumed that the orientation of the virtual view is consistent with the expected road orientation. For example, when the virtual view is at a 90 degree angle with respect to the vehicle, P2 was created by projecting from P1 at an angle of 90 degrees from the line the runs forward from the vehicle origin. (This is the line labeled "Y Axis" in Figure 43.) P2 is assumed to be on the road. The projection distance was typically 20 meters.

Whereas the first technique computed a single circular arc to drive to reach the road point, this technique requires more advanced interaction with our point tracking algorithm. Because the two points along with the vehicle location define a path to follow rather than just a point to drive over, other variables can effect system performance.

In this method, the most important factors which need to be considered are the lookahead distance of the point tracker, the projection distance from P1 to P2 and the detection distance. The lookahead distance determines the location of the point on the desired path which is used to compute the appropriate vehicle turn radius. Large lookahead distances (15+ meters) result in smoother turns which may in turn cause larger deviations from the desired path than when smaller lookahead distances are used. But smaller distances (6 - 10 meters) can cause jerky steering response, especially as speed increases.

The projection distance, as defined earlier, has a large effect on the likelihood of proper alignment of the vehicle to the road. A large projection distance is desirable because it increases the likelihood that the vehicle will be aligned correctly with the road (when using our point tracking method.) But it also makes more assumptions about the local straightness of the road - it assumes that the road continues in the virtual view heading for the projection distance, which may not be the case. Alternatively, a small projection distance increases the chance of misalignment but can model the local road geometry more accurately.

The detection distance is the distance from the vehicle origin to the detected road center. It is not known entirely beforehand, but is dependent upon when the system's network responds with a confidence greater than some threshold. This in turn is governed by when the virtual camera images a significantly large portion of the road. It is tempting to say that it is always better to detect the road from a greater distance. An argument could be made that by deferring detection until the vehicle is closer to the road, the virtual image is created using pixels from a more densely covered area in the actual view. Denser pixel distribution more accurately represents the real world and may therefore result in better detection reliability. Because the road was always detected correctly at a distance of 20 meters and because of the results indicating that detection accuracy does not significantly degrade as the distance to the road increases, this approach was not pursued. A certain drawback of detecting the road later is that it increases the likelihood that the vehicle will have to cross over the road to reach the projection point. (The lookahead distance and vehicle kinematics greatly effect this as well.)

The selection of these parameters is not independent - changing one will likely require changing others in order to maintain system performance. This made developing a set of consistently usable parameters very difficult. In some trials, the vehicle turned smoothly onto the road and was able to begin road following. Other times, it turned too sharply and could not locate the road at all. In still other instances, it would cross over the road in its path to reach the projected road point.

There are two main reasons why this approach was abandoned. The first is probably quite obvious: it is not a trivial task to develop a correct set of point tracking parameters. Different parameters would likely be needed for every detection scenario and although it could be done, it would be a very brittle and inelegant solution.

The second reason relates directly to determining P2. It was mentioned previously that a large projection distance when computing P2 is desirable, but may not accurately model the road. A similar situation can happen even with small projection distances if the virtual view is not oriented exactly with the road. This occurs because ALVINN's neural network is trained on images which are created so that they look as if the vehicle was shifted and/or rotated from its true location. These images are used so that the network can learn to correct any driving mistakes it makes. In the road detection scenario, this means that even if the road is not at the precise orientation defined by the virtual view, the network will respond with a high confidence. As a result, the road may not continue in the virtual view orientation and projecting to find P2 will yield a point off the road.

4.5.4.3 Image-based Alignment and Traversal Using Active Camera Control

The previous methods for detecting the road and travelling onto it only used information from a single image. During the act of traversal and alignment, the system was basically blind. No new data was acquired which could help the system correct any errors which had accumulated. Section 4.6.3 presents a new method in which the system continually tracks the road or intersection branch using knowledge from previous images about where it was located. This method uses active camera control to facilitate changing virtual camera views. By continually updating the

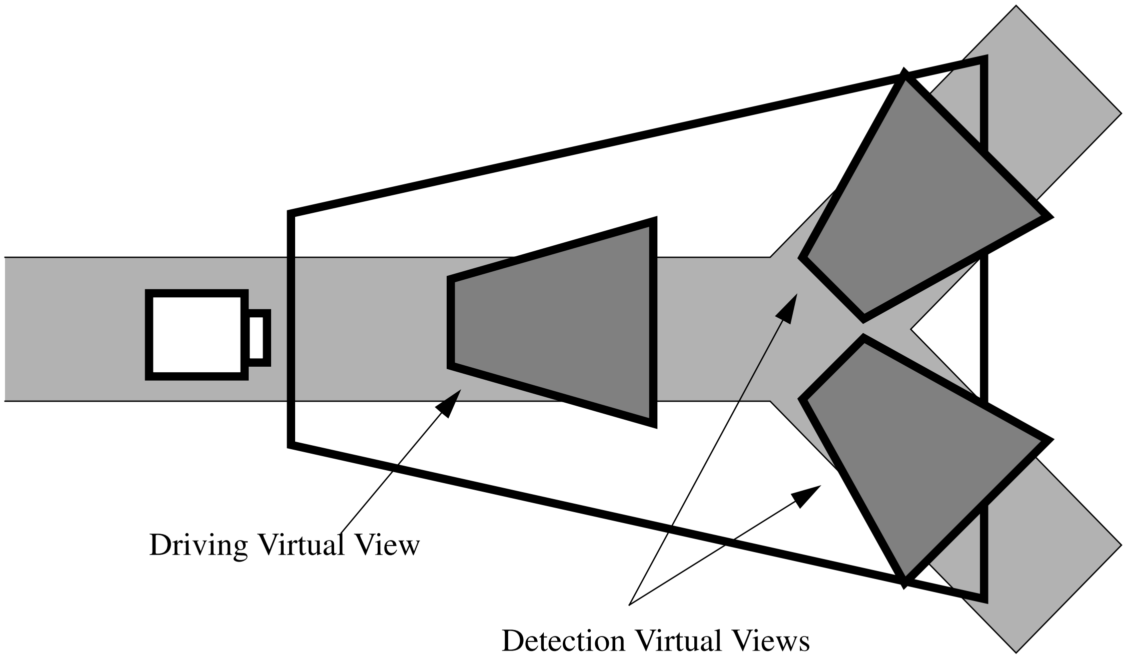


Figure 44 "Y" intersection geometry.

position of the road or branch using information from the virtual camera views, the effect of errors which the previous traversal algorithms do not compensated for, like the 2 meter bias mentioned in Section 4.5.2, and non-linear road or branch segments, can be minimized.

4.5.5 Initial "Y" Intersection Detection Experiments

In this experiment, the goal was to drive along a single lane road, search for and detect a "Y" intersection, and drive onto one fork or the other. Figure 44 depicts this scenario. The central point of this experiment was to determine if intersections could be detected by extending the work done for detecting single roads. This experiment was more difficult than the previous road detection experiments for two reasons. First, it required that the system keep the vehicle on the road and at the same time look for the intersection branches. Second, it required that the system find two road branches rather than just one. Another factor adding to the difficulty of the scenario is that the intersection lies at the crest of a small hill - each of the road segments which meet at the intersection is inclined. This means that the flat world assumption is violated.

Figure 45 Camera on pan-tilt mount.

The road which the vehicle is travelling upon as well as each of the road branches are of the same type. Virtual views were created 9 meters in front of the vehicle. The view which was used to search for the left fork was angled 25 degrees to the left of straight ahead. The one used to search for the right fork was 20 degrees right of straight. Because of the geometry of the situation, the IRRE threshold value, which both virtual images were required to exceed, was lowered to 0.70. The experiment was conducted several times, with the results from each being similar to those of the single road case presented earlier. The system was able to drive the vehicle at low speeds (5 m.p.h.) and detect each of the road branches. Although not as pronounced as in the single road detection case presented earlier, the system still had problems navigating onto either branch.

4.6 Intersection Detection and Traversal Experiments on the Navlab 5

After taking into consideration the lessons learned in the initial experimental set, the traversal techniques were enhanced to overcome the two major problem areas: a fixed camera location and violations of the assumptions about the geometry of the road branch. The fixed camera problem, which had previously limited the effective field of view of the system, was overcome by simply placing the camera on a pan-tilt mount located on the roof of the vehicle. See Figure 45. The geometric constraint violation problem, which previously caused poor traversal performance, was resolved by using image-derived information to continually update the estimate of the branch location and orientation. Adding these capabilities allowed the system to reliably detect and navigate two additional intersections. The first was a “Y” intersection in a park near campus while the other was a “T” junction from a driveway onto a rural road in suburban Pittsburgh. While not exhaustive, these two locations allowed the system to be safely and easily tested on a variety of intersection geometries.

4.6.1 Detection with Known Geometry and Unknown Location

This scenario mimics the “Y” detection experiment in Section 4.5.5 in that the goal was to detect a road junction while the vehicle was in motion. Two modifications to the detection algorithm were made from the previous scenario to facilitate this goal. First, the detection of only a single branch of the upcoming intersection was sufficient for the intersection to be considered detected. The reason that only a single branch of the intersection could be searched for relates to the computing requirements necessary for reliably controlling the vehicle versus the available processing power on the Navlab 5. Basically, the Navlab 5 computer is not as powerful as the one used in the “Y” intersection detection experiments on the Navlab 2 described in Section 4.5.5. Searching for all intersection branches would have likely led to either unacceptable driving performance or to missing the intersection completely because of low search frequency.

The second difference was that the camera being used was located on a pan-tilt mount, instead of in a fixed position and orientation. This meant that the camera could be positioned so that the road, as well as a substantial portion of the anticipated road branch location, could be imaged. In the original “Y” experiments, the camera was fixed, limiting the number of branch locations and the amount of actual and virtual camera view overlap which was possible.

Initially in this scenario, ALVINN controlled the vehicle in normal lane keeping mode. See Image 1 of Figure 46. While driving, the system received a message from a TLDM indicating that an intersection was approaching. Although the exact location of the branch was not known, its orientation with respect to the current road segment was given. Using this information, the system created the appropriate virtual camera view, called the Detection View, which would properly image the branch when it appeared. For the experiments described here, the road branch being

Image 1

Image 2

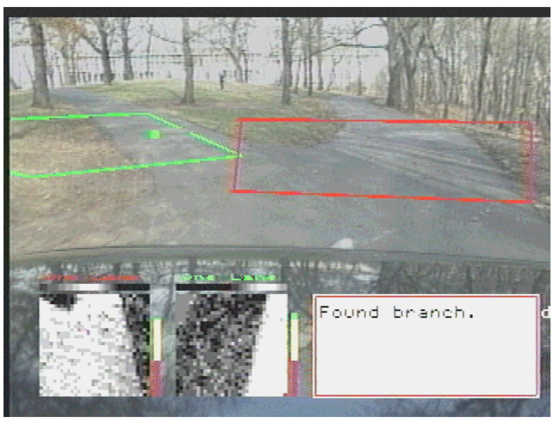


Image 3

Figure 46 Intersection branch detection from a moving vehicle.

searched for was oriented approximately 40 degrees left of straight ahead. In addition to being angled 40 degrees, the Detection View was typically located 7 meters in front of the vehicle. This distance was selected so that the branch would be detected enough in advance to perform the traversal maneuver but close enough so that violations in the flat world assumption did not become significant. After creating the Detection View, the system determined if the current pan location of the actual camera was sufficient to image both views completely. If not, which was typically the case, the system automatically panned the camera so that the largest portion of the Detection View was in the field of view of the actual camera, while maintaining the entire Driving View in the actual camera's field of view. See Image 2 of Figure 46. Note that after the actual camera has been panned, it is no longer in the same orientation as when the ALVINN network was trained. But because the virtual camera is at a fixed location with respect to the vehicle and is independent of the actual camera location, the images it created allowed ALVINN to continue driving reliably.

New images from the Detection View were created approximately 4 times per second and passed to ALVINN's neural network for processing. The intersection branch was considered detected when the IRRE confidence value of the network, in response to a Detection View image, became greater than a predetermined threshold value. See Image 3 of Figure 46. The threshold value was typically set to 0.75. When detection occurred, the system indicated this to the safety driver who stopped the vehicle. At this point, the system began to localize the intersection branch and navigate through the intersection. This process is described in detail in Section 4.6.3.

4.6.2 Detection with Unknown Geometry and Known Location

This intersection detection scenario is the opposite of the previous. In this case, the location of the intersection is known, but the geometry of the intersection is not. Specifically, ALVINN has knowledge about where the center of the intersection is located with respect to the vehicle, but does not know the orientation of any of the intersection's branches. The goal is to find each branch and pass this information to a TLDM which can request that one of the branches be taken using the traversal method described in Section 4.6.3**,** or that the information be stored for later use.

The detection process begins with the vehicle located a known distance from the intersection center. The detection algorithm uses a radial search technique to create virtual camera views which image different hypothesis branch locations. Virtual views are created a set distance from the intersection center at varying orientations. Figure 47 shows the radial search technique and the hypothesis road branch locations used for the intersection experiments described here. It also shows the virtual camera views used to search for each branch. For the intersections used in this experiment, the angular change between hypothesis views was 45 degrees, while the vehicle's distance from the intersection center ranged between 7 and 10 meters. Images taken at each of these hypothesis road branch locations are shown for the "Y" intersection in Figure 48 and for the "T" intersection in Figure 49.The sixth image in each figure shows which hypothesis intersection branches the system believes are likely to be actual branches as determined by the simple detection method described in the next paragraph.

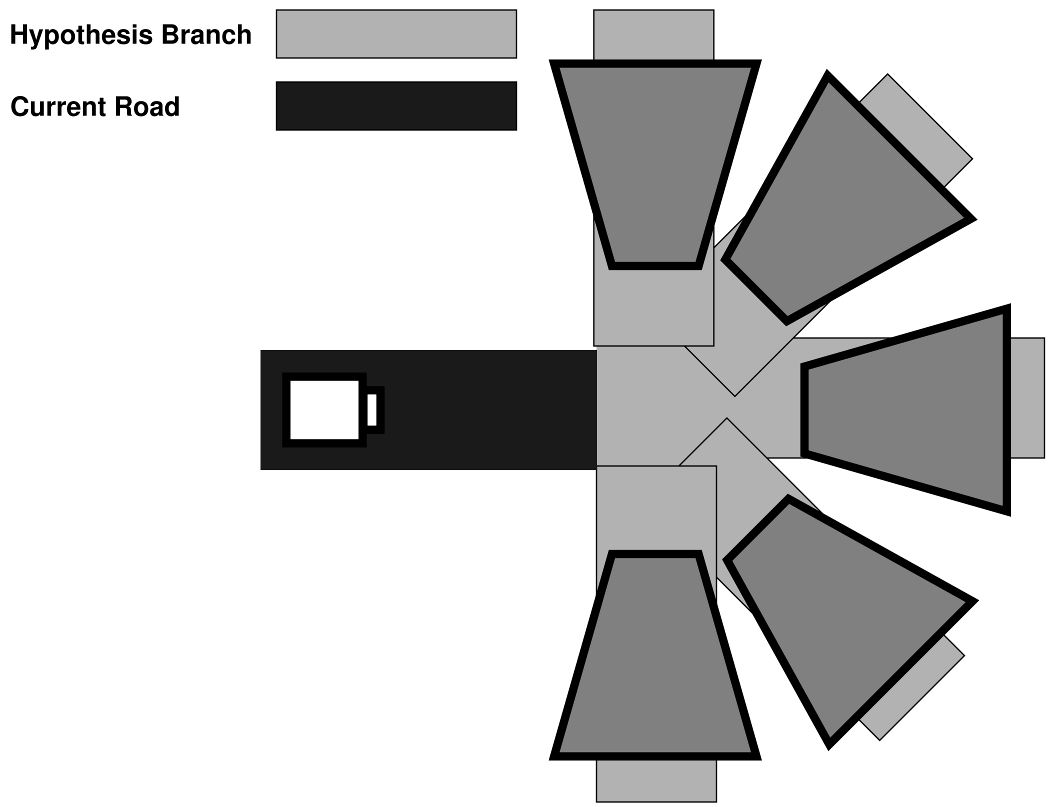


Figure 47 Hypothesis camera views.

As before, the basis for signaling detection is a high IRRE value. If the hypothesis view images an actual road branch, the corresponding IRRE confidence metric will be high, and the orientation of the branch being examined can be saved for further processing. Figure 50 shows enlarged versions of the preprocessed ALVINN image from each of the first five images of Figure 48. (Figure 51 shows the same information derived from Figure 49.) In these figures, along with the preprocessed image is the IRRE confidence value which ALVINN's network produced when shown the image. For preprocessed images created from virtual cameras which did not image actual road branches, the IRRE value is very low, but for the images which were of actual road branches, the confidence value is significantly higher. From this examination, it is evident that by thresholding based on the IRRE value, hypothesis views which image actual road branches can be disciminated from those that do not.

Note that the road branch does not need to be aligned perfectly with the hypothesis view for a high IRRE value to be created, and detection to be indicated. See Image 2 of Figure 50. As before,

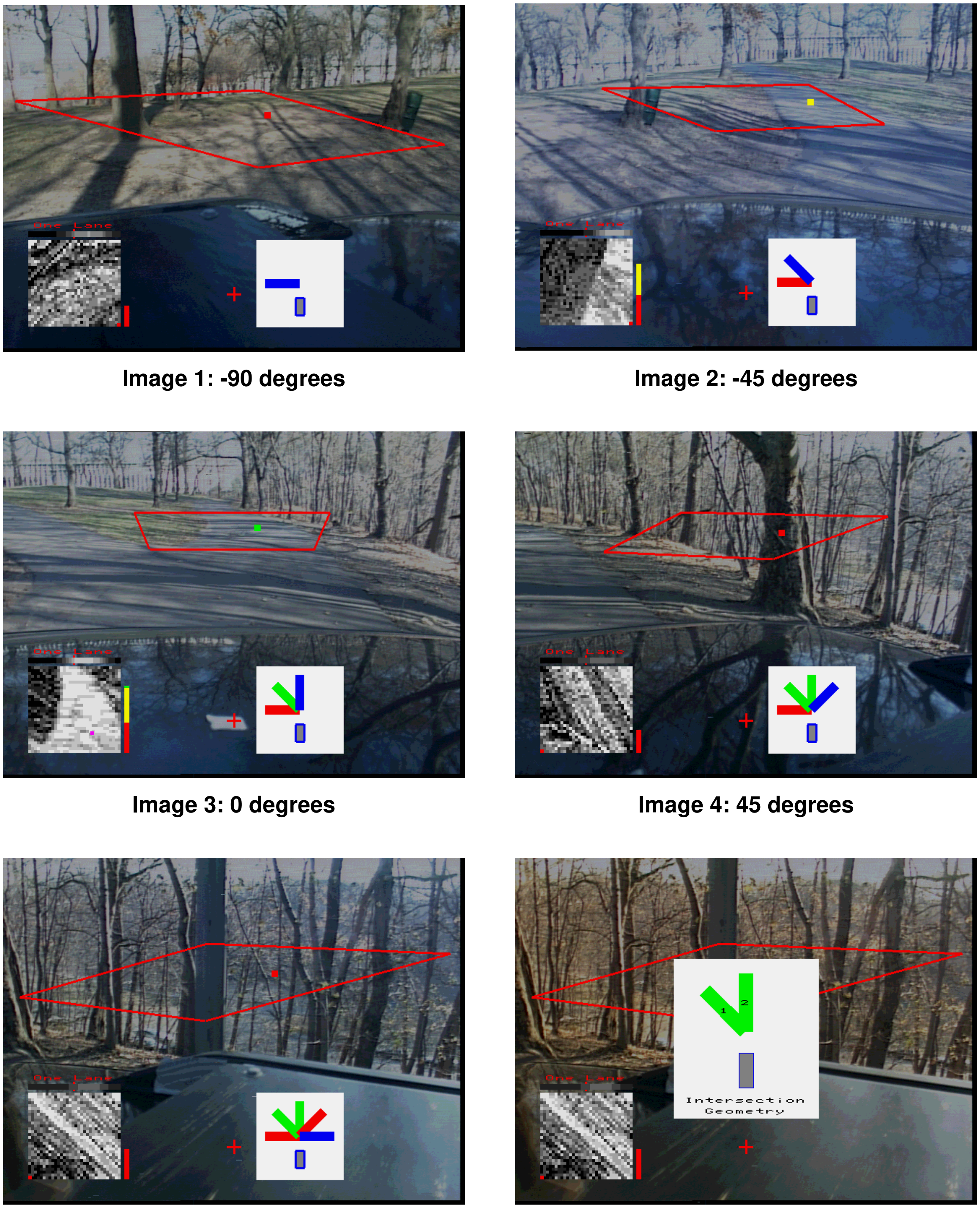


Image 1: -90 degrees

Image 2: -45 degrees

Image 3: 0 degrees

Image 4: 45 degrees

Image 5: 90 degrees

Image 6: Detected Branches

Figure 48 Searching for "Y" intersection branches.

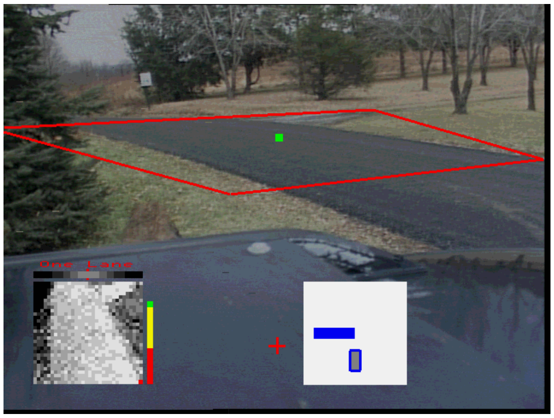


Image 1: -90 degrees

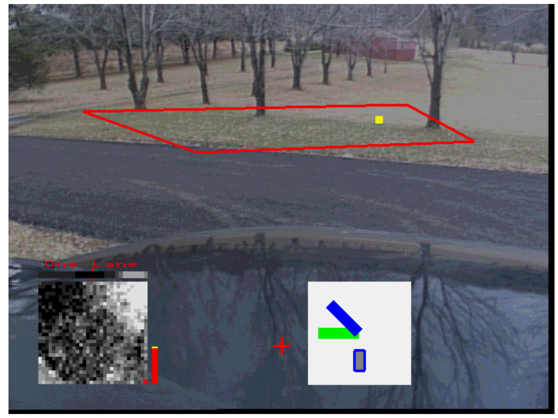


Image 2: -45 degrees

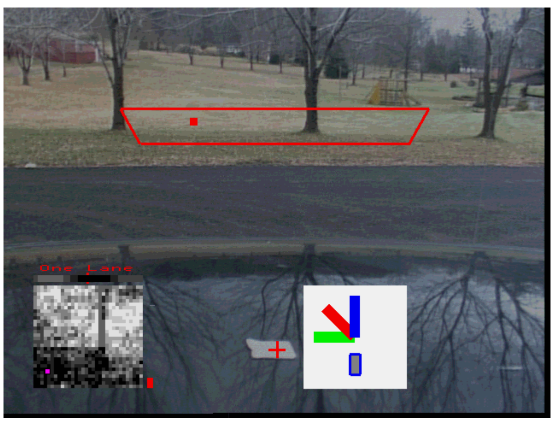


Image 3: 0 degrees

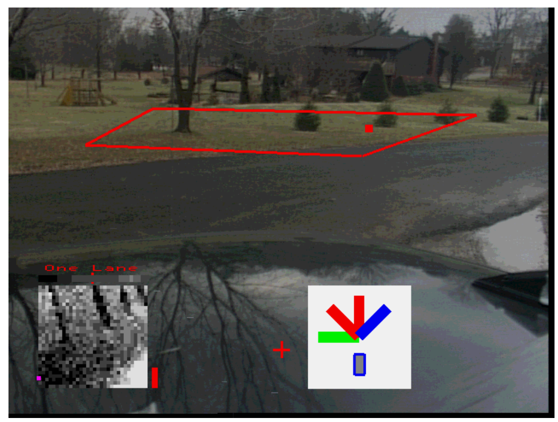


Image 4: 45 degrees

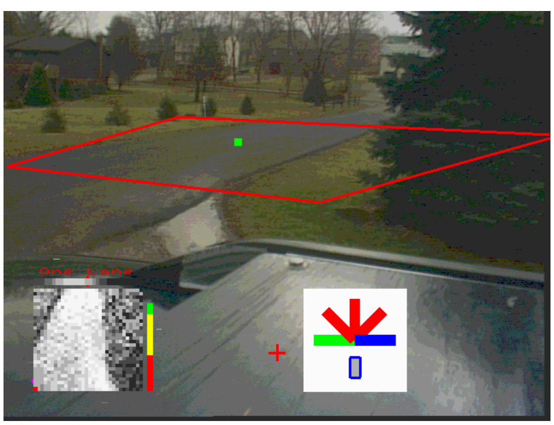


Image 5: 90 degrees

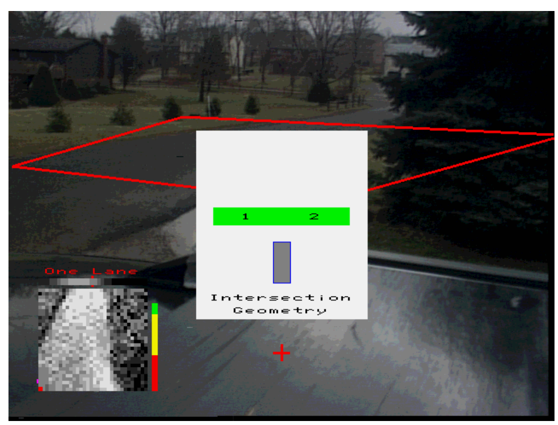


Image 6: Detected Branches

Figure 49 Searching for “T” intersection branches.

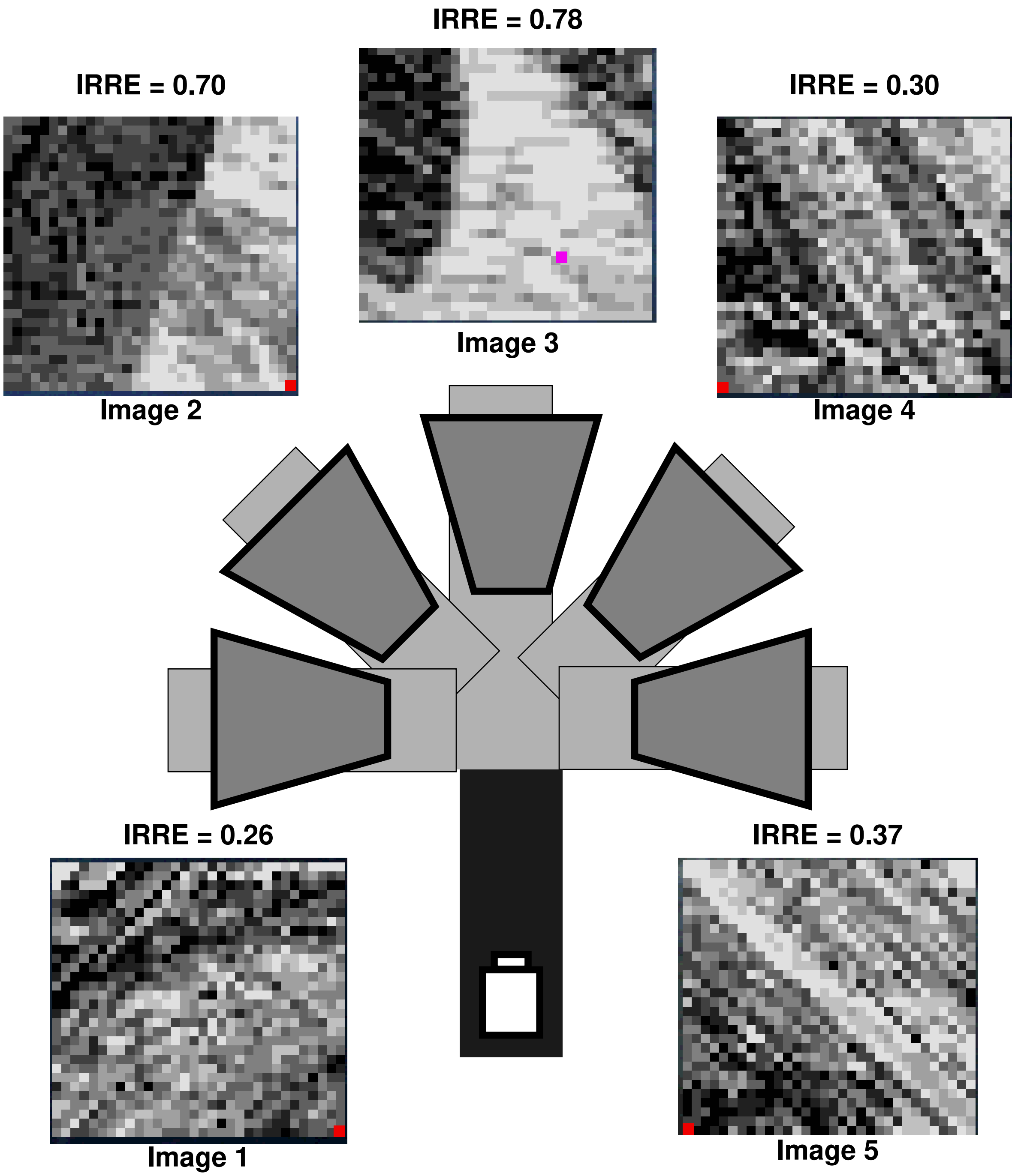


Figure 50 Images and IRRE values for the “Y” intersection.

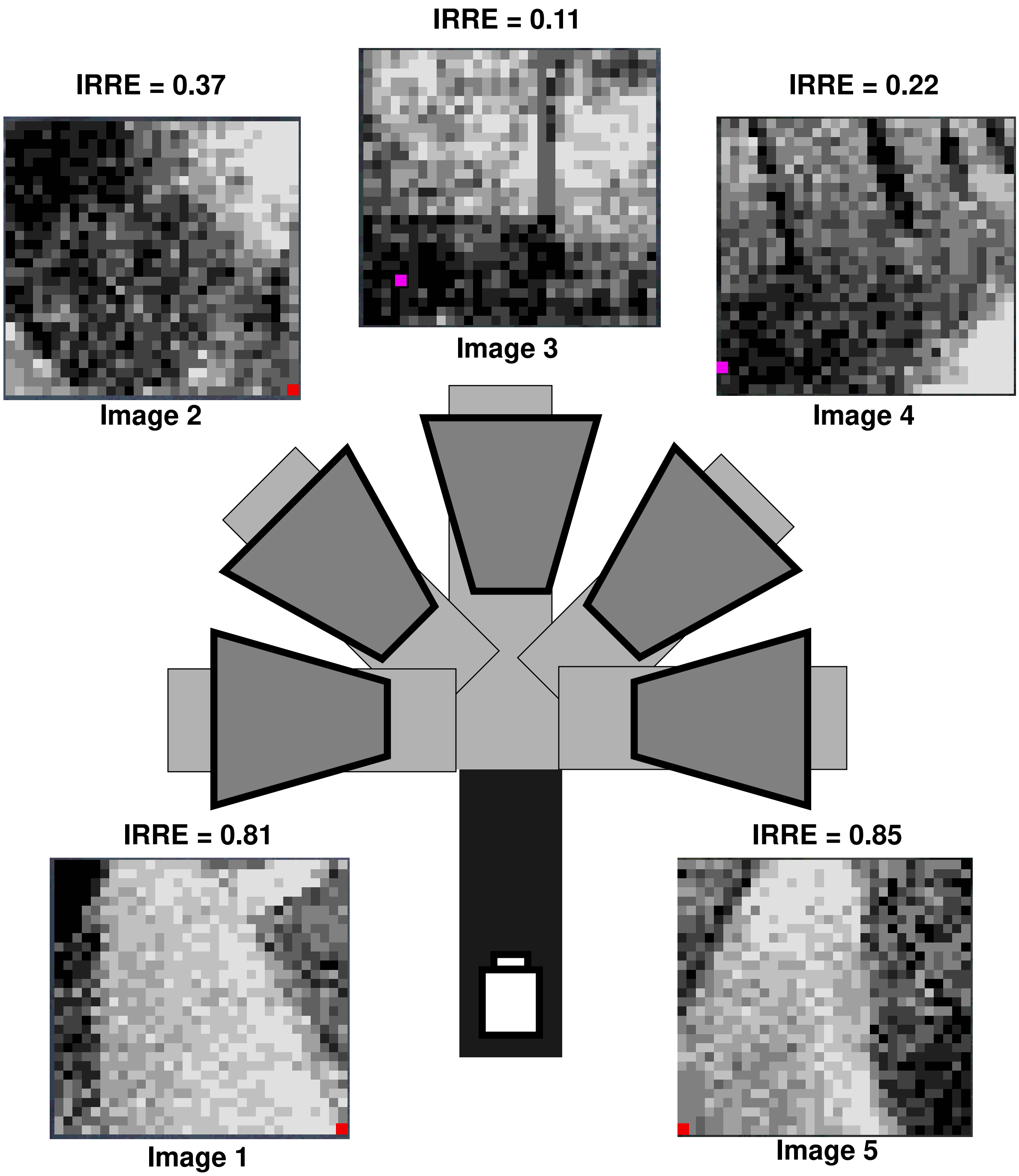


Figure 51 Images and IRRE values for the “T” intersection.

this is due to ALVINN's ability to respond correctly to images in which the vehicle looks as if it is misaligned with the road.

After the road branches have been detected, the system waits for a TLDM, in this case the safety driver, to indicate which of the found road branches should be taken. This is shown in Image 6 of Figure 48 and Figure 49. For the "Y" intersection, the system has detected the left fork as well as the intersection branch which represents the continuation of the current road, while for the "T" intersection, the left and right branches have been found. After the selection has been made, the system localizes the selected intersection branch and traverses the intersection using the techniques described in the next section.

4.6.3 Intersection Localization and Traversal using Active Camera Control

In the earlier experiments on the Navlab 2 using a fixed camera, traversal onto the intersection branch after detection had occurred was the major problem. All of the traversal methods tried in that experiment basically made assumptions about the intersection geometry. Using those assumptions, the vehicle drove blind, using dead reckoning, through the intersection and hopefully onto the road branch. Those techniques were brittle - if the assumptions did not match the real world, or if errors built up in the vehicle's estimate of position, there was no way to recover. The intersection traversal algorithm presented here lessens these problems by using active actual camera control and image-based vehicle localization. By using a moveable camera, the target intersection branch can always be kept in view so that the image-based vehicle localization techniques can continually refine and update the vehicle's position with respect to the destination intersection branch.

Before intersection traversal can take place, the road branch must be localized to a greater degree of accuracy than was done for detection. Like in many cases presented earlier in this dissertation, this is necessary because of ALVINN's ability to correctly respond to images in which the vehicle appears misaligned with the road. Because of this, the exact location and orientation of the road branch with respect to the vehicle is not known. Because lateral translation and orientation errors in virtual view alignment to the road cannot be determined from a single road image and its associated output, the view alignment process must take place in two steps.

4.6.3.1 Preliminary Lateral Offset Adjustment

After the intersection has been detected, the first step in localizing it further is to use the output displacement of the network to update the position of the virtual camera imaging the road branch. This is done by moving the view laterally, perpendicular to the hypothesis branch direction, for a distance equal to the output displacement of the network. After the view has been moved, an image is created from this new location and passed through ALVINN's network, producing another output displacement which is again used to adjust the view. This process is continued until the output displacement of the network changes sign, meaning that the current and last view have "bracketed" the view location which will produce zero output displacement. In this last step, the final displacement from straight ahead is very small.

Images 1 though 4 of Figure 52 show the progression of view locations during this portion of the localization algorithm. The input image and associated output is shown in the lower left portion of the image. Note how the output displacement from straight ahead, shown above the preprocessed image, decreases as the virtual view become better aligned with the road branch. After the correct location has been bracketed, the localization algorithm moves into the branch orientation determination phase.

4.6.3.2 Alignment to the Intersection Branch

Although the first phase of road branch localization causes the output displacement of the network to become nearly zero, the orientation of the view with respect to the intersection (and the road branch itself) cannot be assumed to be correct. Figure 53 illustrates this concept. In this figure, the preprocessed image along with ALVINN's output displacement created using each image, when the vehicle is in both the left and right configurations with respect to the road is shown. Note that in each case the displacement is near zero although the vehicle is only properly aligned with the road in the right example. This occurs because ALVINN is trained to determine the position of the road center the lookahead distance in front of the vehicle. In both cases shown in Figure 53, the center of the road is in the middle of the image at the lookahead distance, so ALVINN outputs a zero displacement for each, despite the differences in road orientation in each image. (This characteristic of ALVINN is also the cause of some of the problems associated with

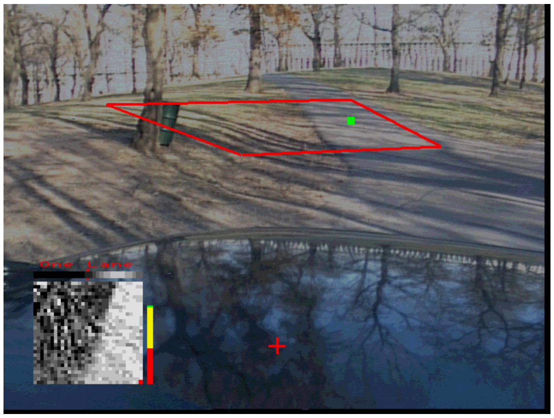


Image 1

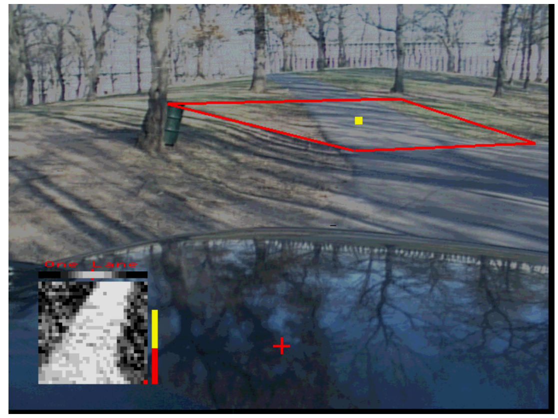


Image 2

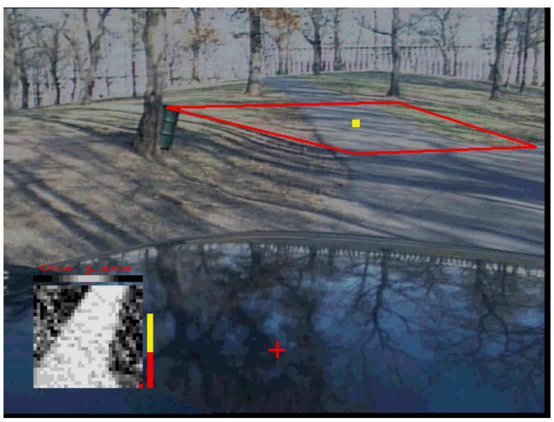


Image 3

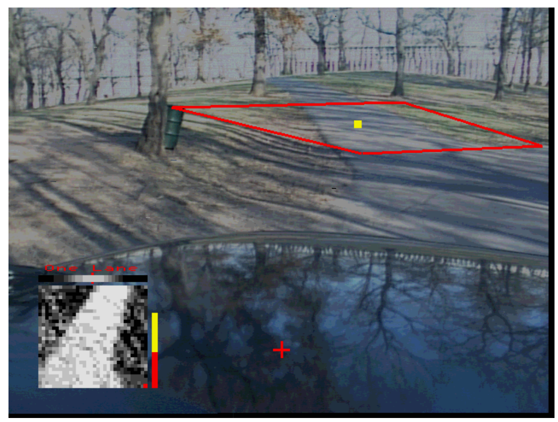


Image 4

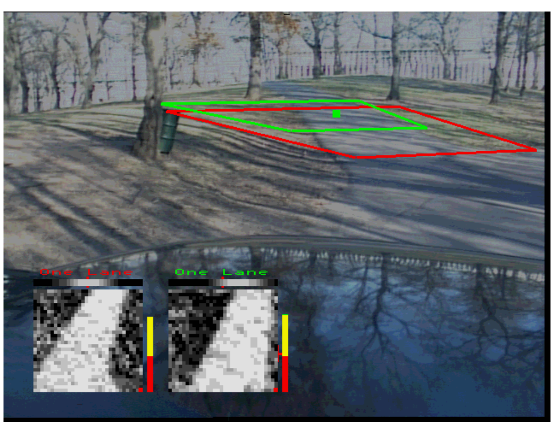


Image 5

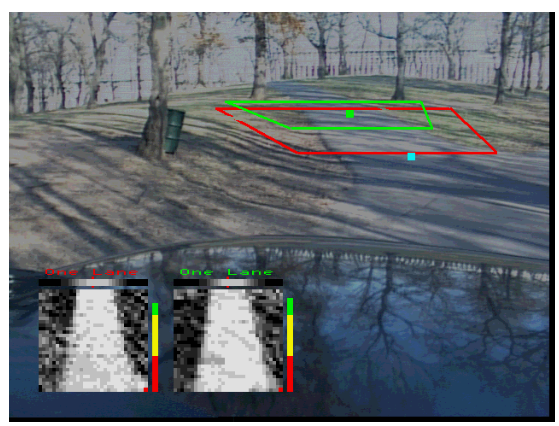


Image 6

Figure 52 Lateral and angular branch localization.

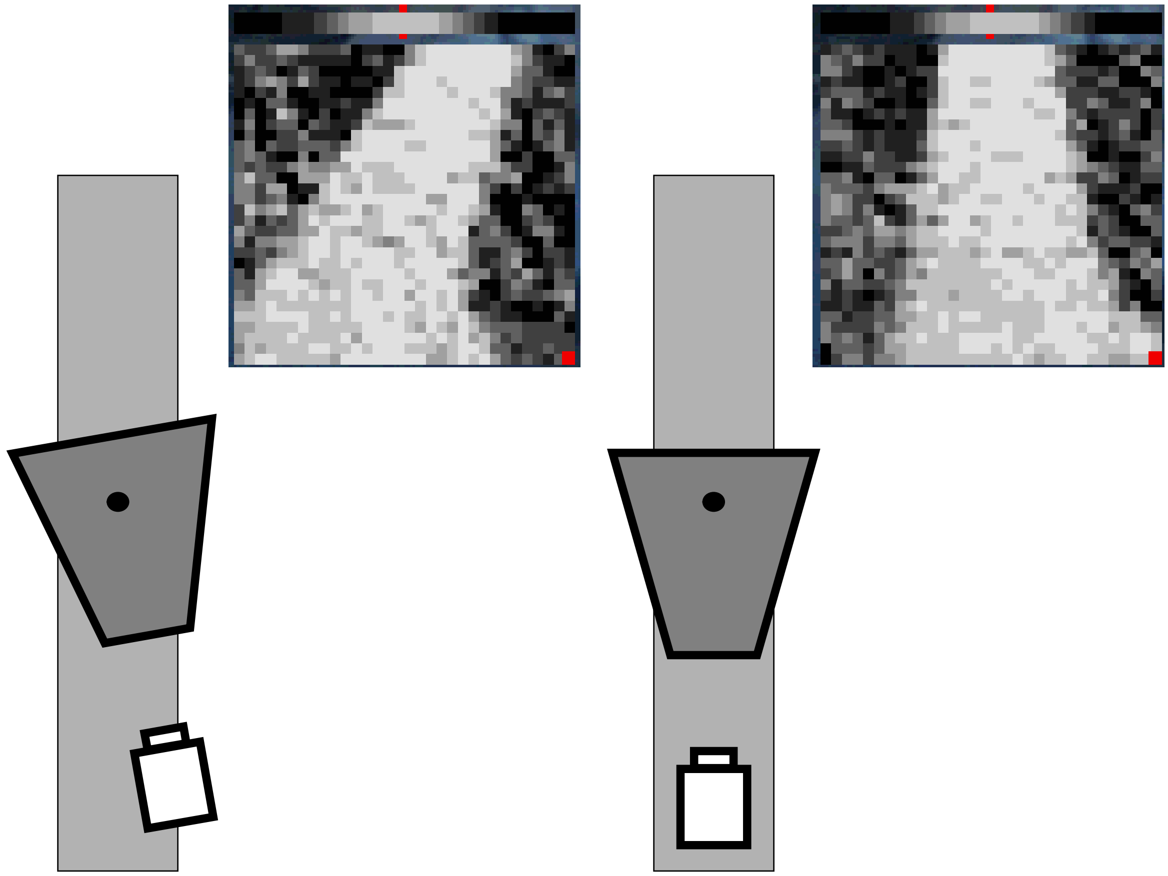


Figure 53 Possible alignments with zero output displacement.

the intersection traversal algorithms presented earlier. They assumed that the road direction was at the same orientation as the virtual camera view which was used to detect it.)

In order to accurately determine the intersection branch orientation, a second view is required. This view, called the Projection View, is typically created between 3 and 5 meters in the direction of the current estimated road branch orientation. Figure 54 shows the actual road scene (Image 5 of Figure 52) along with a diagram of the original and Projection View arrangement.

If the original view is properly aligned with the intersection branch and accurately reflects the branch's location and orientation, creating an image using the Projection View and passing it through ALVINN's network should yield an output displacement close to zero. This is because the previous alignment step reduced the lateral offset of the original view to near zero, in effect centering the original view over the longitudinal axis of the road branch. If the original view is at the correct orientation, projecting 5 meters along the branch orientation should also create a view

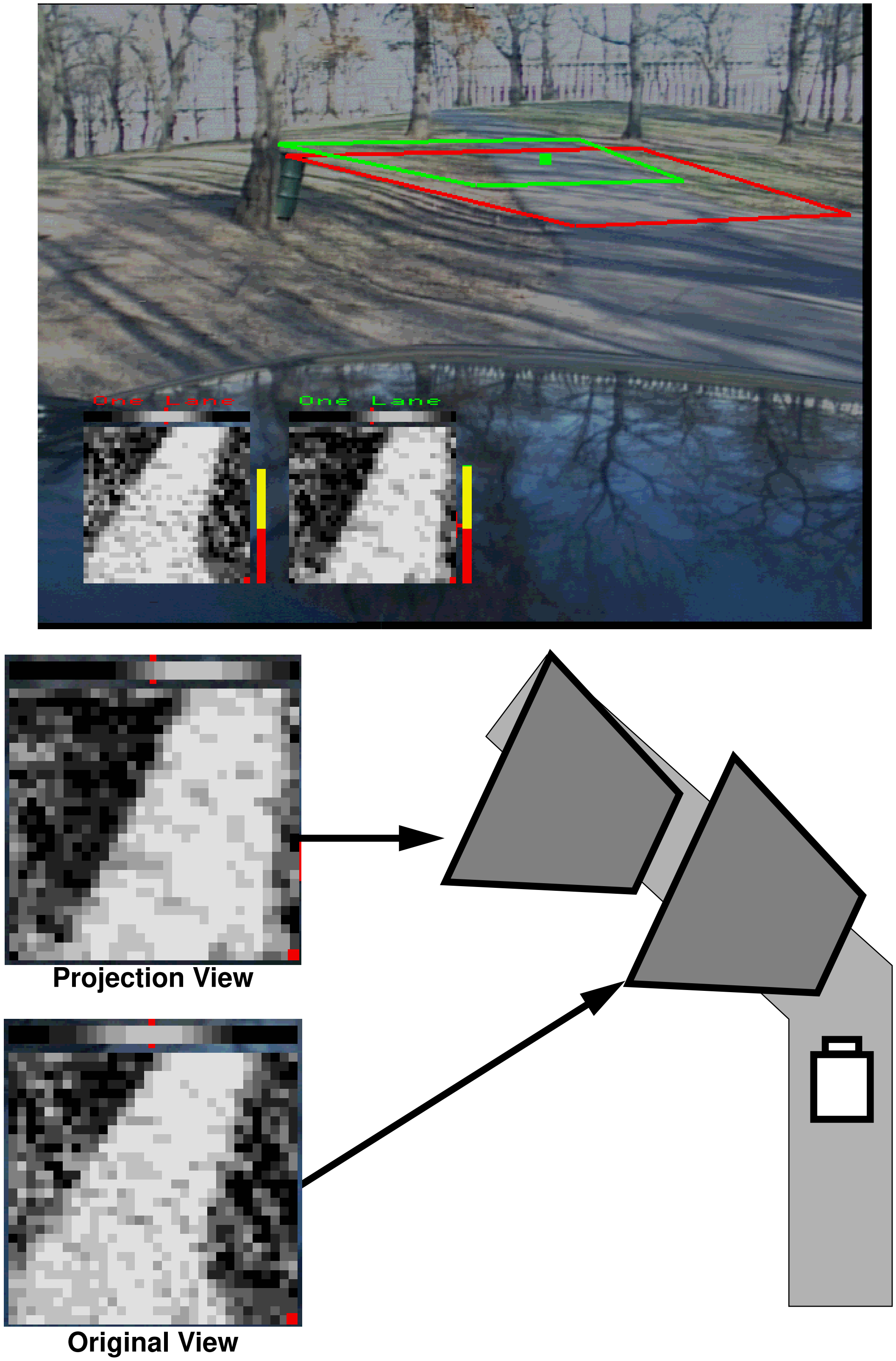


Figure 54 Orientation localization.

which is centered over the longitudinal axis of the intersection branch. Images created by the Projection View at this new position should also produce zero output displacement. As is shown in Figure 54, this is not the case. Although the original view has zero displacement, indicated by the centered gaussian hump of activation over the preprocessed image, it was not aligned correctly with the intersection branch. Because the original view was misaligned, the Projection View is also misaligned which results in a non-zero output displacement from the image created by it. In Figure 54, the gaussian hump indicating the network output displacement created from the Projection View image is shifted right to reflect this misalignment.

The output displacement difference from zero that the Projection View image causes is a measure of the misalignment in orientation between the original view and the intersection branch. By using the output displacement from the Projection View, the projection distance, and the location of the original and Projection Views, the amount of this angular misalignment can be computed. See Figure 55. After rotating the original view to its correct orientation, its lateral position must also be corrected. This is necessary because the rotation correction was done about the original view's location and not about a point on the longitudinal axis of the intersection branch. This problem is depicted in Figure 56. From the same information used to compute the orientation error, the lateral offset error, which is a result of the orientation error correction, can also be computed. This is also shown in Figure 55. After both error values have been computed, they can be used to update the original view location and orientation so that it more closely matches the intersection branch geometry. Image 6 of Figure 52 shows the original and Projection View after the entire localization procedure has been completed. The original view and Projection View are both producing output displacements near zero, indicating that they are aligned with the longitudinal axis of the intersection branch.

4.6.3.3 Traversal

Once the position and orientation of the target branch has been determined, the next step is to traverse the intersection. There are two issues which must be considered and resolved in order for intersection traversal to be successful: tracking the branch as the vehicle moves through the intersection and the vehicle control algorithm.

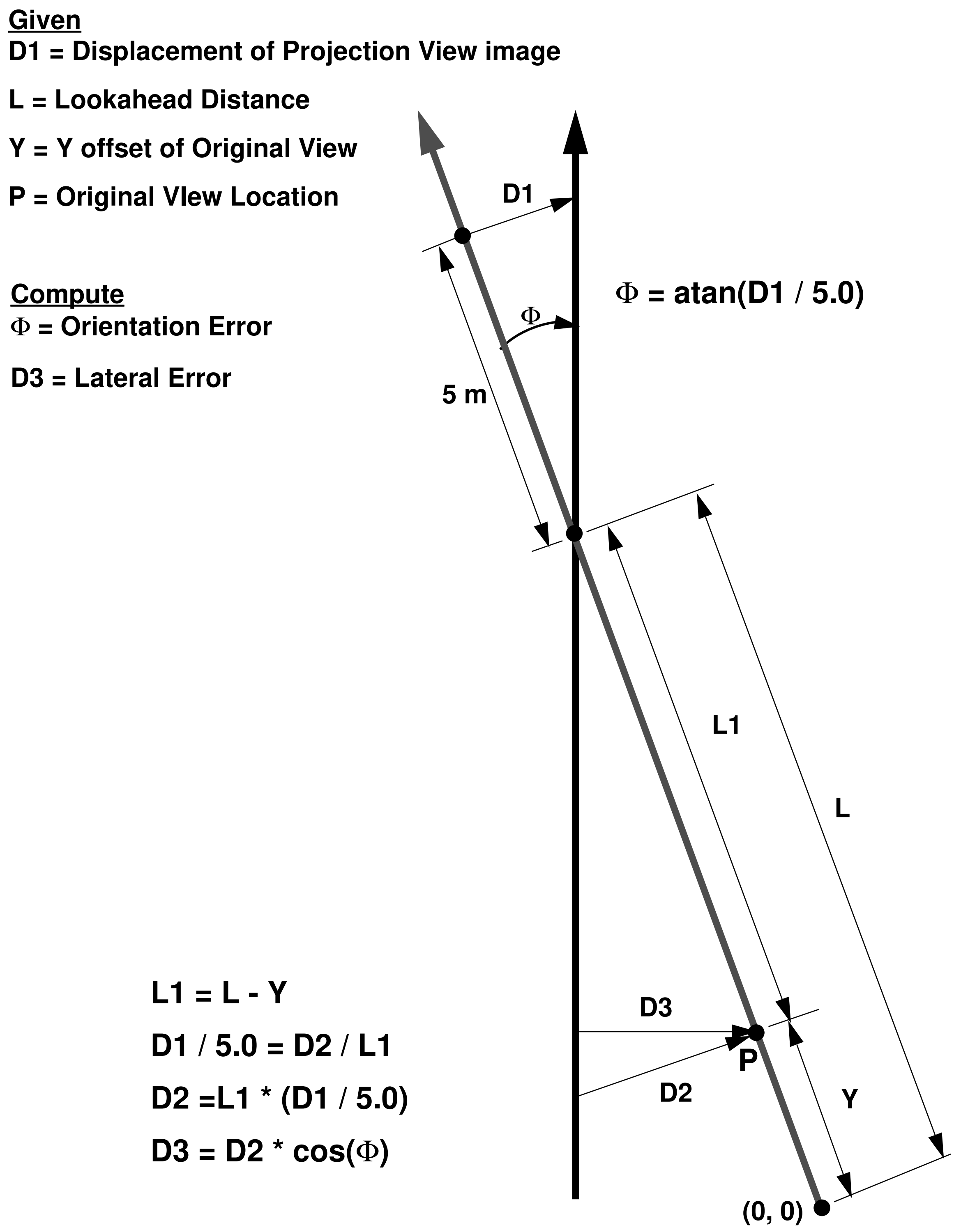


Figure 55 Orientation and lateral offset error correction.

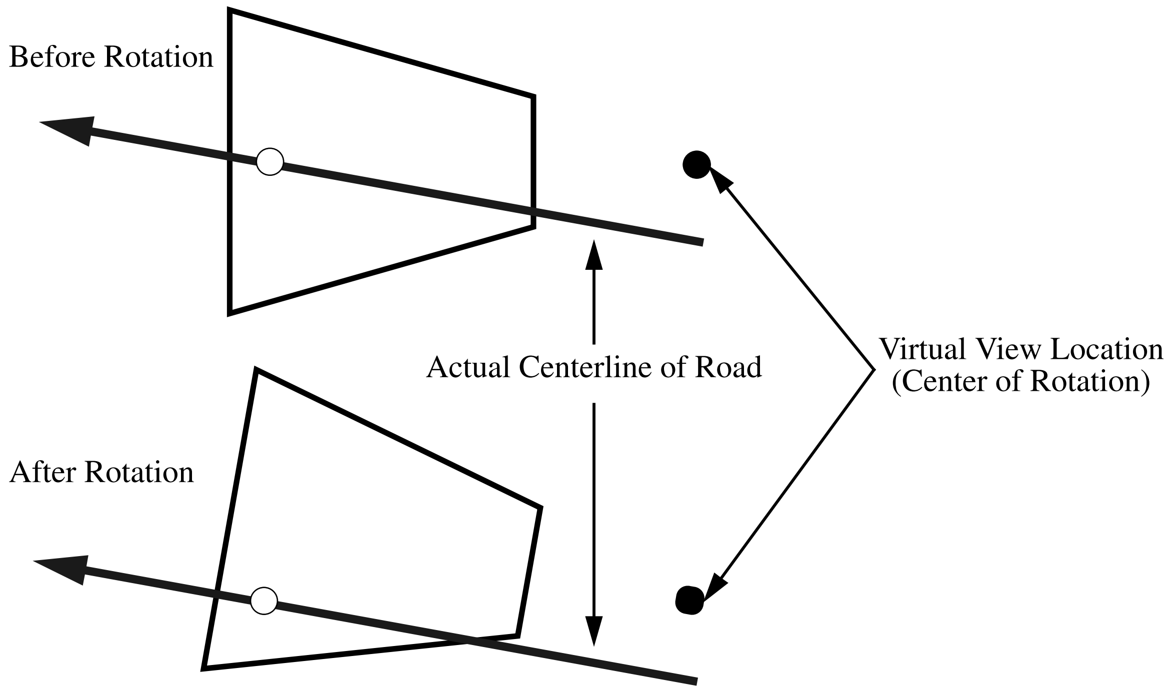


Figure 56 Lateral misalignment after rotation.

The branch tracking problem was solved by adapting the branch localization algorithm presented in the previous section. During traversal, the system continually updated the location and orientation of the original virtual camera view by repeating the alignment procedure presented in the previous section. When the original view was about to move out of the field of view of the actual camera, the system automatically detected this and panned the actual camera appropriately. The ability of the system to correctly orient and localize the intersection branch during traversal is shown for the “Y” intersection in Figure 57 and Figure 58. Images from a traversal of the “T” intersection are shown in Figure 59 and Figure 60. Note that a camera pan occurs before Image 4 in Figure 57 and before Images 3-7 and Images 9 and 11 in Figure 59 and Figure 60.

Creating an acceptable vehicle control algorithm for navigating intersections was one of the most difficult tasks in this work. The majority of the methods tried caused the vehicle to either severely cut corners, overshoot, or generally become misaligned with the road. A contributing factor to these problems was the lack of accurate geometric information about the intersection branch as the vehicle turned.

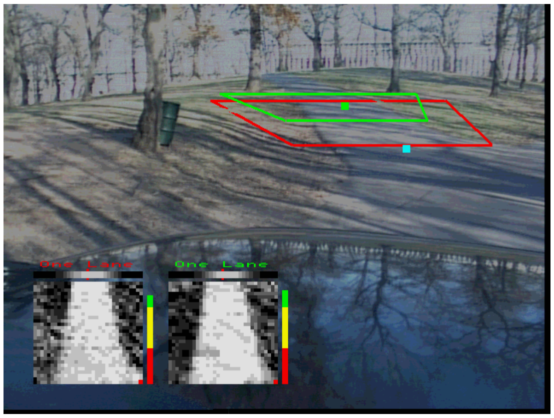


Image 1

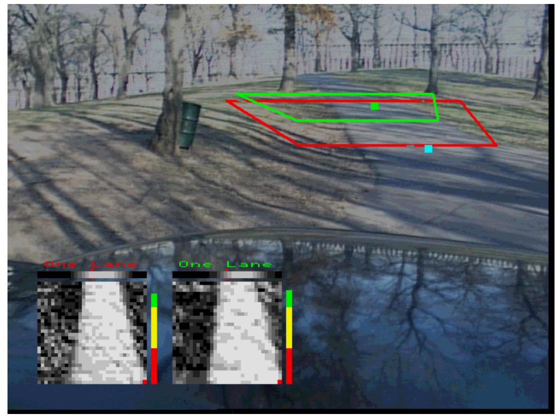


Image 2

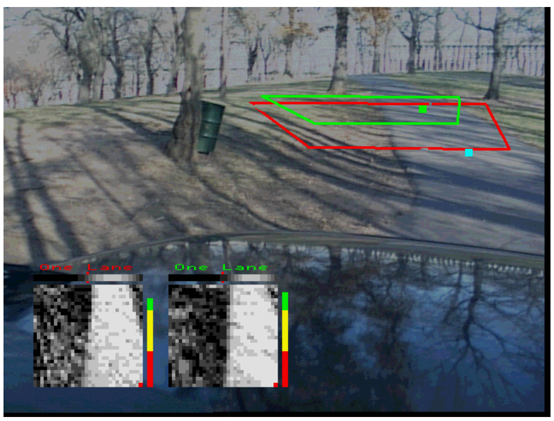


Image 3

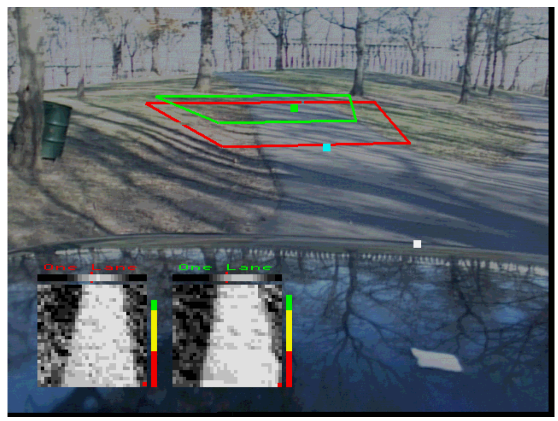


Image 4

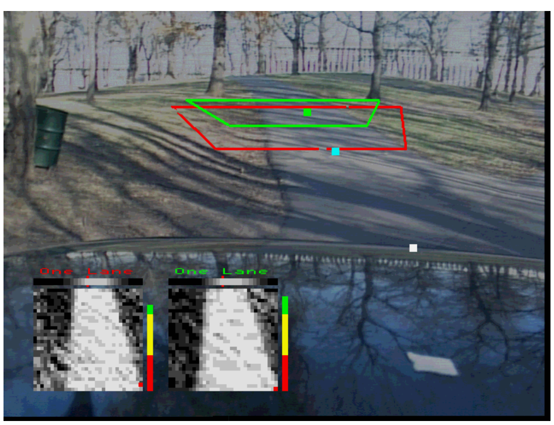


Image 5

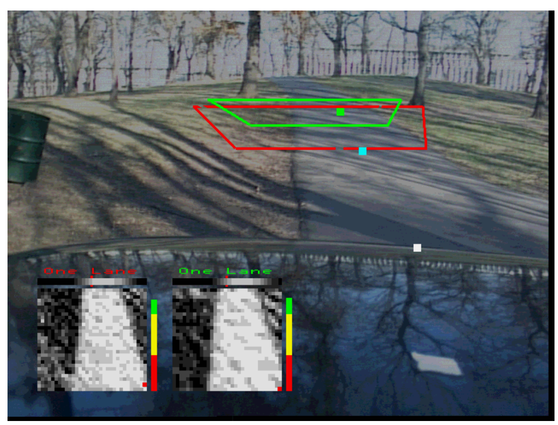


Image 6

Figure 57 Initial “Y” intersection traversal images.

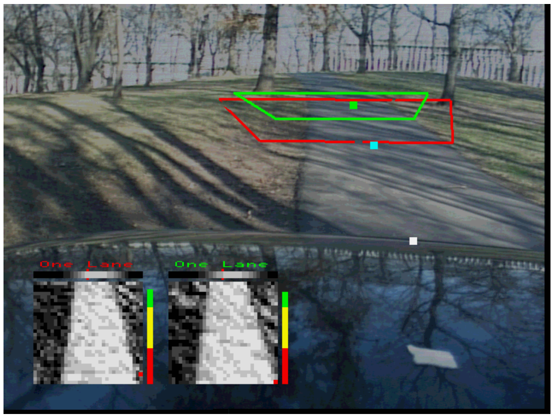


Image 7

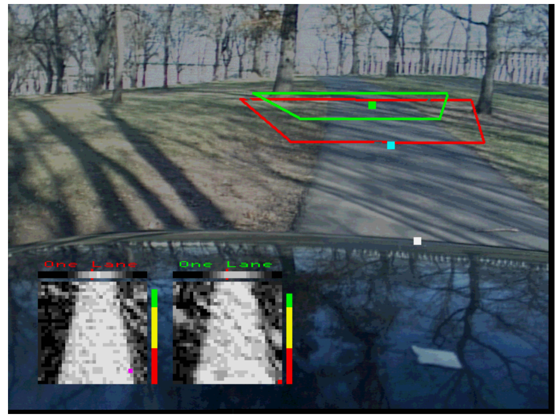


Image 8

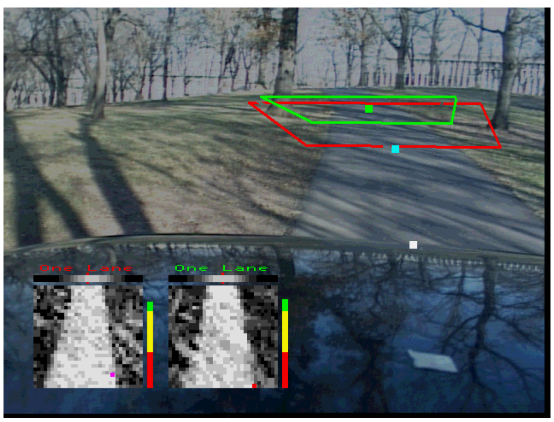


Image 9

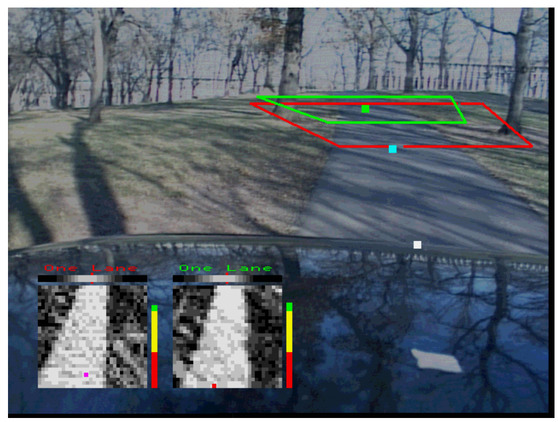


Image 10

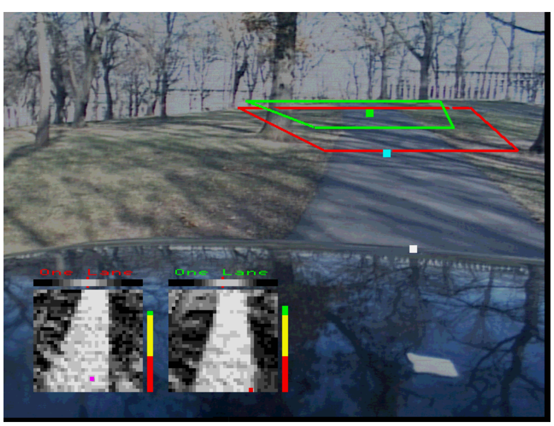


Image 11

Figure 58 Subsequent "Y" intersection traversal images.

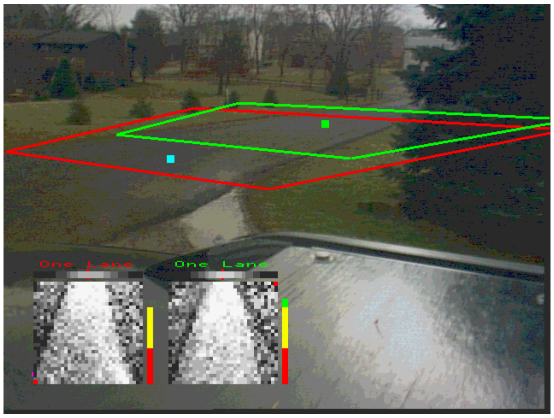


Image 1

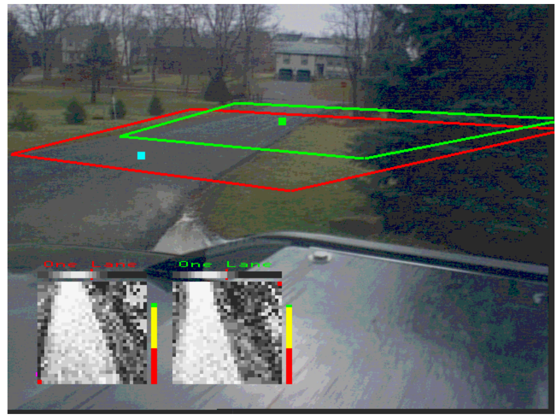


Image 2

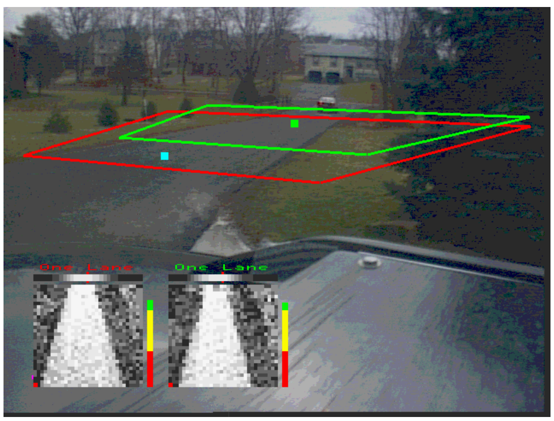


Image 3

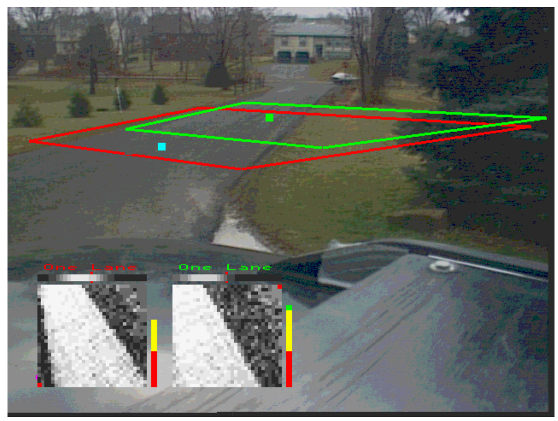


Image 4

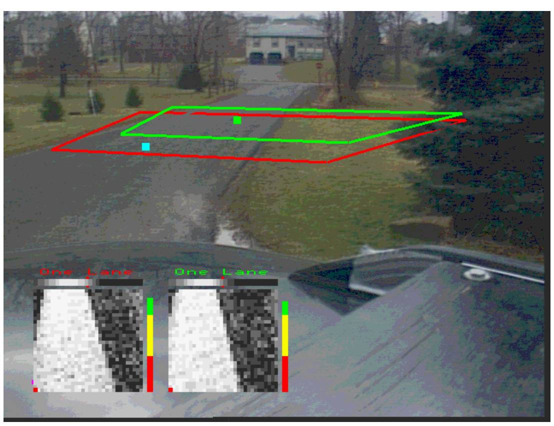


Image 5

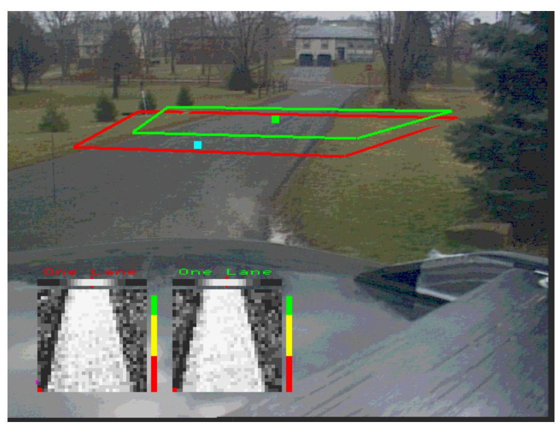


Image 6

Figure 59 Initial “T” intersection traversal images.

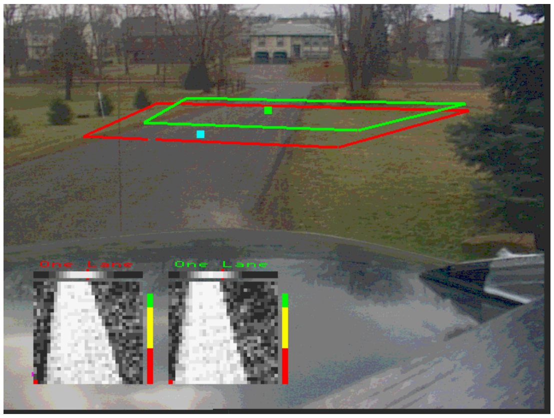


Image 7

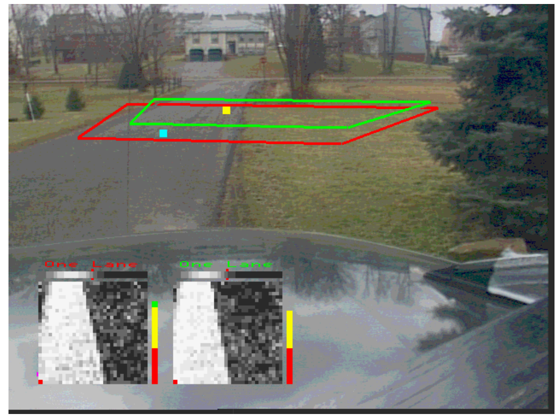


Image 8

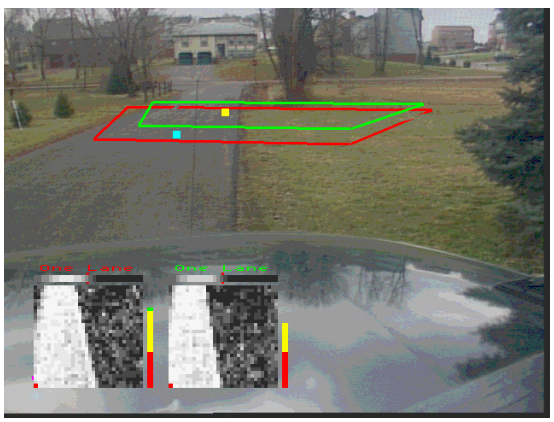


Image 9

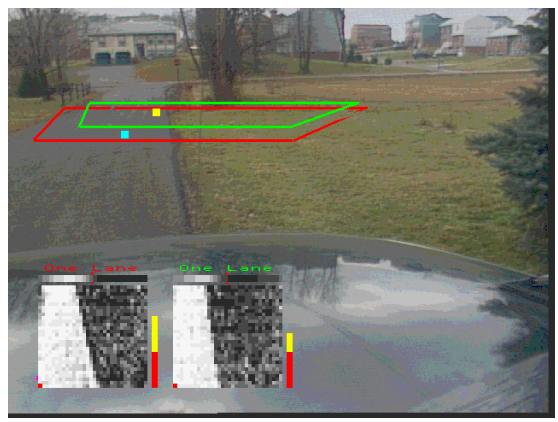


Image 10

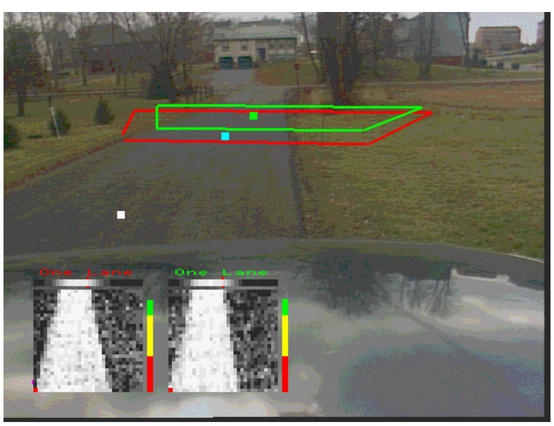


Image 11

Figure 60 Subsequent “T” intersection traversal images.

As shown in the traversal figures, this problem was alleviated by tracking the intersection branch using a combination of traditional and virtual active vision techniques. But given this, many of the vehicle control algorithms still had difficulty matching the vehicle heading to the road orientation. Based on this observation, the vehicle control algorithm shown in Figure 61 was developed. This algorithm takes into account the branch orientation as determined by the localization algorithm.

The vehicle control algorithm finds tangent points on two lines representing the vehicle's current heading and the intersection branch orientation. The first tangent point, P1, is defined to be the current vehicle position while the second point, P2, is on the intersection branch axis. The distance along the branch that P2 is located, measured from the intersection center point, C, is defined to be equal to the distance from P1 to C. After being computed, this distance is held fixed throughout the intersection traversal. C is computed before traversal begins by finding the intersection point of the branch axis and the line representing the vehicle heading. As mentioned earlier, the orientation of the branch axis is not fixed at the hypothesis view orientation, but rather is the refined branch orientation derived during the branch localization phase.

This construction insures that a circle can be found which will intersect P1 and P2 tangentially. The radius of the circle that intersects these tangent points is the arc that the vehicle uses to drive through the intersection. For each new image, the tangent point, P2, and arc to drive are recalculated based on the new location of the intersection branch so that any errors in vehicle control, positioning, pan angle, or non linear branch geometries are taken into account.

4.7 Results and Discussion

As mentioned earlier, branch detection is the most robust part of the system. Using simple thresholding as the disciminating technique, the system was able to successfully detect each intersection branch in 33 of the 35 cases on the two "Y" and "T" intersections in the second experimental set. These trials were distributed about evenly over the moving vehicle "Y," the stationary "Y" and "T" detection scenarios. In no cases did the system detect a branch which was not present.

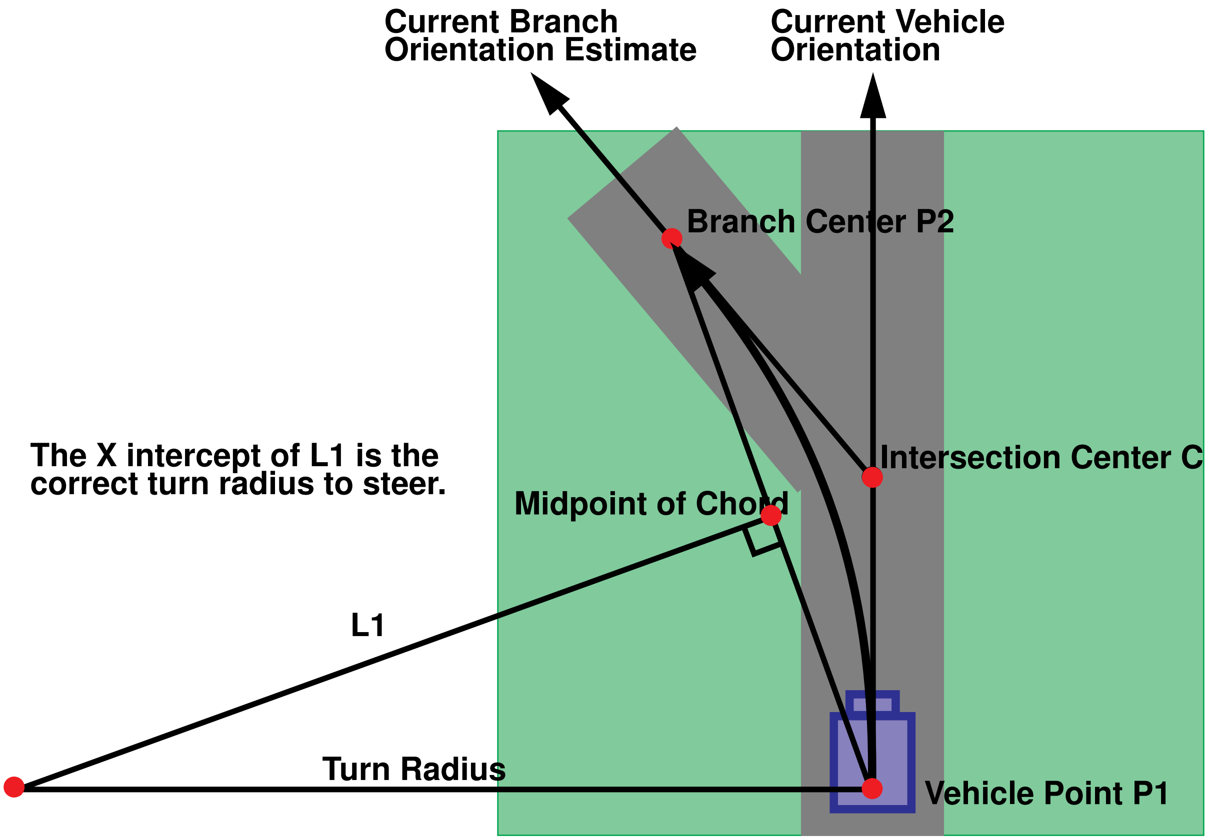


Figure 61 Traversal turn radius determination.

Both failure cases were on the "T" intersection. In one case, the system successfully detected one of the two branches. The other branch's confidence value fell just below the threshold but was still much higher than any of the other three hypothesis locations. In the other failure case, neither branch was detected. For this case, of the two real branches that should have been detected, one did have a noticeably higher IRRE value, but it was still below the threshold. The other branch's IRRE value was not significantly different than any of the other branches. In this case the problem can be attributed to a change in the ambient lighting, from overcast skies to sunshine, to which the camera was not able to properly adjust. In any case though, this indicates that although detection is robust, it is not foolproof and redundant branch verification procedures are necessary.

In all 34 scenarios which the system detected at least one branch, it was able to properly move the vehicle onto the branch and continue driving. The system was able to drive the vehicle onto the left and right branches of the "T" intersection as well as navigate onto the 45 degree branch of the "Y" intersection. Although the control algorithm during traversal is very simple, having a moving camera and tracking the road branch throughout the maneuver allowed it to work robustly over the experimental domain on which it was tested.

It is reasonable to assume that the detection method will work for any road branch type which the base neural network can learn to drive on. If this assumption is true, this system will have an advantage over other road and intersection detection systems which require the researcher to program in new detection methods when new road types are encountered.

The experiments took place in real, but fairly constrained environment. A robust road and intersection detection system must be able operate in more challenging environments - on typical city streets, with other cars, and with more extensive interaction with higher level knowledge. In addition to these factors, the system must be able to account for violations of the flat earth assumption.

Chapter 5 Other Systems

5.1 Introduction

The Virtual Active Vision tools which I have developed and used with the ALVINN lane keeping system are designed to be independent of the lane keeper, only requiring that the system use images to drive. Because of this, other researchers have been able to directly use these tools, as well as the concepts that they embody, to develop and enhance two other vision based lane keeping systems. The first system, called ROBIN, uses virtual cameras to achieve better lane keeping performance than it was previously capable of attaining. ROBIN uses virtual cameras to create images which have a more appropriate feature distribution than those taken directly from the video camera. Unlike ROBIN, which used the same implementation of virtual cameras that ALVINN uses, the RALPH driving system uses adaptations of three Virtual Active Vision techniques to achieve state-of-the-art lane keeping performance. A more detailed description of each system and the role that Virtual Active Vision tools play in increasing their performance follows.

5.2 ROBIN

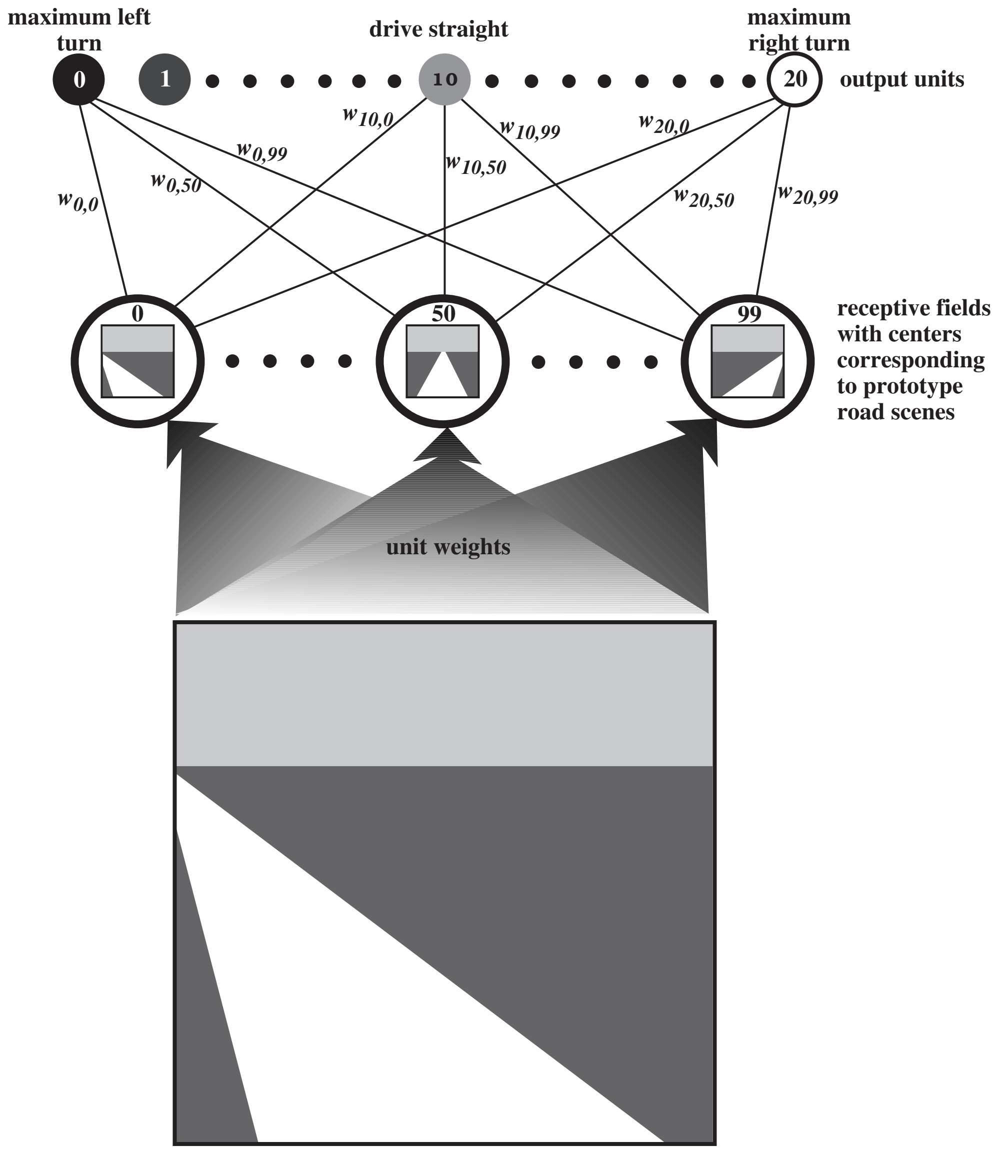
ROBIN is a neural network autonomous lane keeping system first developed at the University of Maryland and then extended at Lockheed Martin Corporation (LMC) in Denver, Colorado. ROBIN is based on the radial basis function network architecture and learns how to drive by capturing prototypical driving scenes for a discrete set of steering directions during a training phase.

In autonomous driving mode ROBIN generates a degree-of-match between the input image and each prototypical driving scene captured during training. It then uses the steering directions associated with the prototypical scenes, weighted by the degree of match between the prototypical scene and the current scene, to control the vehicle. Figure 62 shows a diagram of the ROBIN network architecture.

In order to differentiate between different steering situations, there must be a detectable difference between the corresponding driving scenes for those situations. The more detectable the scene difference, the more reliable ROBIN's determination of the appropriate steering direction will be when driving in autonomous mode. As would be expected, features in the road scene provide the key cues for scene differentiation. ROBIN primarily exploits the contrast between the on-road and off-road portions of the image, since these are typically the most prominent features in road scenes. Other features that are evident in typical driving scenes are road lines and oil spots, but due to reduction in resolution from the camera image (640x480) to the low resolution input image (32x30), they are not used very well by ROBIN [44].

The maximum variation of road scenes occurs when the number of pixels representing off-road features approximately equals the number of pixels representing the on-road features, leading to an off-road to on-road pixel ratio of one. On LMC's testbed vehicles, the camera is mounted on a pan/tilt platform so that it can be mechanically moved to optimize the off-road to on-road pixel ratio for a particular road-scene. The optimization of the off-road to on-road pixel ratio is limited, however, to the mechanical range of the camera platform, and often the desired pixel ratio of one cannot be achieved. In order to better maintain a ratio of one, virtual cameras have been added to the ROBIN system. Because virtual cameras can be positioned anywhere in space, it is possible to find a location where the off-road to on-road pixel ratio is approximately one.

Before virtual cameras were added to ROBIN, its ability to control LMC's testbed vehicle at moderate speeds was unsatisfactory. When driving down a straight stretch of road, the vehicle, a converted U.S. Army HMMWV, would exhibit oscillatory behavior, gradually swerving from road edge to road edge. This behavior was due to the pixel ratio being only about 1 off-road pixel for every 8 on-road pixels. In order to equalize the number of off-road and on-road pixels, a vir-



desired weight relations after training

$$w_{0,0} \approx w_{10,50} \approx w_{20,99}$$
$$w_{0,0} > w_{0,50} > w_{0,99}$$
$$w_{10,50} > w_{10,0} > w_{10,99}$$
$$w_{20,99} > w_{20,0} > w_{20,50}$$

Figure 62 ROBIN network architecture.

tual camera was used to resample the input image from a location that was more appropriate. (The actual camera was mounted on the right side of the vehicle, about a foot above the hood. From this viewpoint, the vertical field of view of the lens caused the camera to image above the horizon.) When tested in this new configuration, ROBIN’s oscillatory driving behavior was eliminated and the system was be able to drive LMC's testbed vehicle on four lane divided highways at speeds up to 45 m.p.h.[45]

5.3 RALPH

RALPH (Rapidly Adapting Lateral Position Handler) is a vision based, adaptive lane keeping system that has driven our testbed vehicle over 7000 autonomous miles in the last 9 months. This includes a 2850 mile transcontinental trip from Washington D.C. to San Diego, CA, during which RALPH autonomously steered the Navlab 5 98.2% of the way. RALPH decomposes the problem of steering the vehicle into three steps, sampling the image, determining the road curvature, and determining the lateral offset of the vehicle relative to the lane center. The results of the last two steps are combined into a steering command which can either be used to control the steering wheel of our testbed vehicle or provide an estimate of the proper direction to drive, if the system is operating in a warning capacity [42].

The RALPH algorithm is designed to look for parallel, linear features on the road. These features can be solid or dashed lines, the road/shoulder boundary, tire tracks left by a preceding vehicle, or even the oil spot down the center of the lane. When imaged from a camera mounted behind the windshield, these features are not parallel, but rather converge because of perspective distortion. Because the algorithm expects the features to be parallel, the image must be resampled. The resampling is done so that the portion of the road within 1 meter on either side of the lane and extending from about 20 to 70 meters in front of the vehicles is extracted. This area is enclosed by the light colored "imaging trapezoid" shown in Figure 63 and corresponds to the area which would be imaged by a virtual camera located about 45 meters in front of the vehicle which was looking straight down at the roadway. The pixels within the imaging trapezoid are selectively sampled so as to create a low resolution (30x32 pixel) image in which important features such as lane markings, which converged towards to top of the original image, appear parallel. Although the implementation of this functionality is different than those used for the experiments described in the previous chapters, the effect is the same - the image processing task is made easier through the use of Virtual Active Vision techniques and more robust performance is possible.

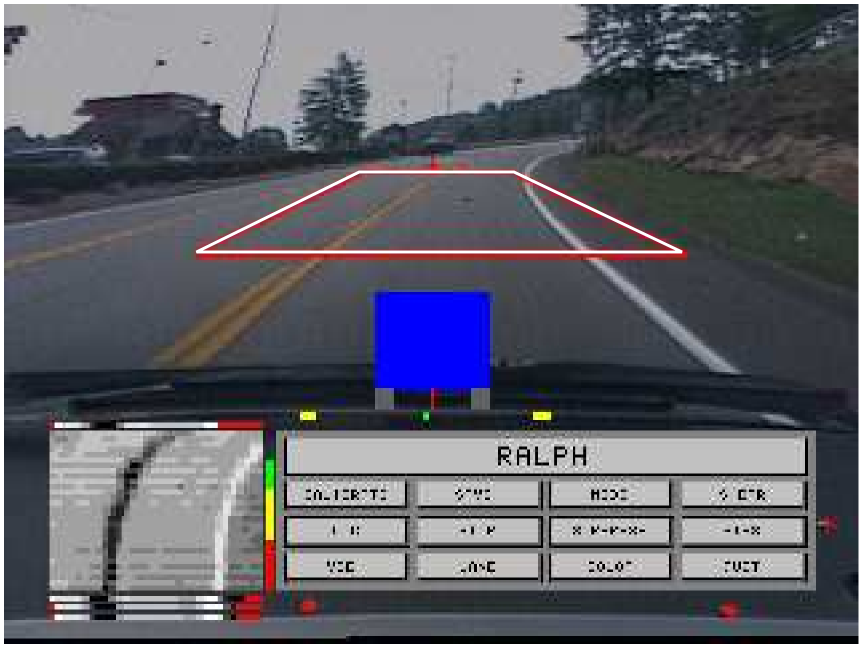


Figure 63 Ralph display.

Another instance where the RALPH system uses concepts much like those developed in this thesis is for focusing attention on the areas of the scene which contain the most important information for accomplishing the task at hand. One example of this is adjusting the location of the imaging trapezoid based on the current vehicle speed. As the vehicle speed increases, RALPH moves the imaging trapezoid further out, away from the vehicle, so that the upcoming road curvature can be determined in time for the system to react to it. This movement of the imaging trapezoid serves the same function as the virtual camera views used for lane transition and intersection detection - it allows the underlying lane keeping system to be used without modification, to accomplish a task that could typically require fundamental changes in the lane keeping algorithm. A second example of moving the imaging trapezoid to increase driving performance occurs when the system detects that the horizon, or other irrelevant feature, is in the distal portion of the imaged area. Because these features do not correspond to those in the rest of the imaged area, driving performance can suffer. To alleviate this problem, the imaging trapezoid is moved closer to the vehicle, eliminating the spurious feature from the imaged area.

Finally, the RALPH system uses a similar concept to the lane transition methods described in Chapter 3 to transition from one lane to another. It differs in that RALPH is able to offset the vehicle from the center of the driving lane without laterally moving the imaging trapezoid. This is possible because unlike ALVINN, RALPH separately determines the vehicle's lateral displacement from the center line of the road and the upcoming road curvature. Using this knowledge RALPH is able to move the vehicle toward the centerline of the road by incrementing the target lateral offsets. Once it is at the centerline, it shifts its hypothesis of where the road features are located in the opposite direction of the lane transition. This action produces the same effect that lateral movement of virtual camera view has when used with ALVINN. In RALPH's case, when the system matches the newly shifted features to the road scene, the system will be able to finish the transition into the destination lane.

5.4 Discussion

Like ALVINN, ROBIN and RALPH are vision based lane keeping systems. Their operational domain is one in which geometric relationships are very important and useful. Each system implements and uses Virtual Active Vision tools in different ways. In the ROBIN system, the implementation of virtual cameras described in this dissertation is used only to increase the performance of an existing lane keeping system to an acceptable level. In contrast, a virtual camera-like imaging trapezoid is at the core of the RALPH system. All subsequent processing depends on the transformation that this view applies to the image. Without it, all later steps are invalid. In addition to being very dependent on a virtual camera-like sensor, RALPH uses many active view control techniques to improve lane keeping robustness. It does this by adjusting the viewing location so that it images only road features. In effect, RALPH is doing on-the-fly optimal camera placement. The ALVINN system uses virtual cameras in ways both similar to and different from ROBIN and RALPH. Like ROBIN, virtual cameras have been added to an existing lane keeping system. But they are not used to simply increase lane keeping performance. In ALVINN, active view control techniques are used to build tactical driving capabilities. But in RALPH, similar techniques are used to increase the robustness of the base lane keeping system.

By examining how ROBIN, RALPH, and ALVINN implement and use virtual cameras, domains where virtual cameras are likely to be useful can be identified and the typical progression of deployment can be better understood. Domain areas which include tasks where simple geometric transformations of the viewpoint can substantially simplify the processing are ones which are especially amenable to the application of Virtual Active Vision tools. Initially these tools will be added onto existing systems to help improve performance. For example, take a simple vision based parts identifier. The system developer could implement a system to detect parts only when they were located in a certain position and orientation on a conveyer belt. By supplementing this basic system with many virtual cameras, it would be able to identify parts which were in any position and orientation on the belt. This type of use is similar to ROBIN. The next stage of deployment of virtual cameras is represented by ALVINN and RALPH. In both of these system, virtual cameras do more than simply improve rudimentary performance. They are the foundation on which new capabilities are built. Their scope of the capabilities differ, with RALPH focussing on novel view placement procedures to eliminate spurious driving features, while ALVINN uses virtual cameras to add tactical driving capabilities. An example of this level of deployment using the parts identifier case is tracking parts and triggering some action based on where the parts were located on the belt. For example, a part could be tracked until a character recognition program was able to read its serial number using a virtual camera stabilized image.

Another example of a real vision system which is in the first stage of deployment is the face detection system being developed at Carnegie Mellon [47]. This system is designed to find faces in an input scene regardless of their location. In this system a scale dependent neural network is trained to discriminate between images which contain faces and those that do not. During testing, portions of the image are resized so that if they contain a face, it will be at the correct scale for the neural network to accurately predict it is presence. This functionality maps directly to the parts identifier example - a simple system is enhanced using a virtual camera-like tool.

It would be easy to say that all vision tasks could benefit from Virtual Active Vision tools, but at some point the exact nature of these tools must be characterized. The dividing line between Virtual Active Vision tools and simple image subsampling lies in the ability of true virtual cameras to simplify the processing of underlying systems by representing the world in a common, consistent manner. In addition to simplifying processing, true Virtual Active Vision tools enable basic systems to exhibit more advanced functionality through the use of active view control. As a result of controlling the areas in the scene which are imaged, advanced capabilities can be achieved. The

Virtual Active Vision tools present in the ALVINN and RALPH system clearly fit this definition but it is questionable whether the ROBIN and face detection systems really use Virtual Active Vision tools, or just advanced subsampling techniques. Until active view control is added to these systems, the true power of Virtual Active Vision tools has not been exploited.

Chapter 6 The Testbed Vehicle

6.1 Introduction

Research into self driving vehicles and driver monitoring systems has reached the point where long duration and distance field testing has become feasible. Unfortunately, vehicle and computer systems which provide the functionality to accomplish these tests have previously been too expensive or inconvenient. This chapter describes a simple, yet powerful, platform designed to work on any passenger vehicle. The platform, called PANS (Portable Advanced Navigation Support), was used for the majority of the experiments presented in this thesis. In total, it has supported over 9000 miles of autonomous driving over the past 9 months, including the "No Hands Across America" transcontinental trip in which the RALPH [42] road following system successfully drove 98.2% of the distance from Washington D.C. to San Diego, CA - 2797 out of 2849 miles. See Chapter 5 for a partial description of the RALPH lane keeping system.

6.2 Motivation

The price/performance ratio of computing hardware has dropped significantly in the past decade while our understanding of the autonomous navigation problem has increased at a similar pace. Together, these factors have had a positive effect on the size, profile and performance of Carnegie Mellon University's mobile robots. In 1986, a Chevy panel van was converted into the Navlab 1. This vehicle had 5 racks of computing equipment including a Warp supercomputer, but it wasn't until the late 80's that software systems could drive the Navlab 1 at its top speed of 20 m.p.h. In 1990 the Navlab 2, a converted U.S. Army HMMWV, was built. This vehicle has three Sparc 10 computers, which are used for high level data processing, along with two 68000-based computers used for low level control. On this vehicle, our software systems can drive over rough terrain, avoiding obstacles, at speeds up to 6 m.p.h. and on-road at 70 m.p.h. Both of these vehicles use steering wheel and drive shaft encoders and an expensive inertial navigation system for position estimation.

Our newest vehicle, the Navlab 5, is a 1990 Pontiac Trans Sport donated to us by Delco Electronics. See Figure 64. This vehicle is used exclusively for on-road navigation experiments including autonomous lane keeping, lateral roadway departure warning and support, and curve warning. These task-specific systems run on the PANS platform (Portable Advanced Navigation Support). The platform provides a computing base and I/O modalities for system developers as well as low level services like position estimation, steering wheel control, and safety monitoring. The PANS platform is powered from the vehicle's cigarette lighter and is completely portable.

All high level processing, including position estimation and vehicle control, is done on a Sparc LX class portable workstation equipped with a color video digitizer. The only additional processor is an HC11 microcontroller that implements functions like low level steering motor control and safety monitoring.

Position estimation is done on the PANS platform using input from 2 sensors - a differential equipped GPS and a fiber optic rate gyro. When available, a steering wheel position encoder is also used. Local (x, y, heading) and global (latitude and longitude) position along with vehicle velocity, distance traveled, and turn radius, are supplied to application programs using an inter-process communications mechanism.

6.3 PANS Overview

The goal when designing the PANS platform was to develop a robust yet simple system which could provide better on-road performance than the current Navlab 2 at a substantially lower cost. The Navlab 2, a converted US Army HMMWV, is a good platform for off-road navigation, where extra ruggedness is necessary and short (less than 10 miles) missions are the norm. It is not well

Figure 64 The Navlab 5, a 1990 Pontiac Trans Sport.

suited for on-road driving research because of its size, complexity, and temperamental operational nature. Also, on-road driving systems have progressed to the point were experimental runs in the hundreds or even thousands of miles have become practical.

PANS was designed to address these issues. It uses simple, well engineered commercially available components, that were integrated in a straightforward manner. And because it is designed to be used in a unaltered passenger vehicle, it has no special power or cooling requirements. Also, the future users of the system were involved from the beginning in the design, fabrication, and operationalization of all PANS components. This effort led to a highly usable and maintainable platform.

All high level application computing is done on a Sparc LX class portable computer manufactured by RDI Computer Corporation. (Figure 65 shows all of the PANS hardware.) Key components of this computer are a 50MHz MicroSparc CPU, 32 MB's of RAM, 970 MB's of hard disk space, and a 1024x768 active matrix LCD display. (For comparison, this processor is about equivalent to a 486DX2/66 using Spec ratings as a guide.) The laptop contains an optional Peripheral Expansion Unit which is equipped with two SBUS slots and space for additional hard disk drives. The two SBUS slots contain a Datacell color video digitizer and a Performance Computer Company quad serial port expansion unit. The laptop runs SunOS 4.1.x.

The digitizer input is connected to a Sony DXC-151A color camera. The camera is outfitted with a Pelco TV8 ES-1 auto iris, manual focus lens. This camera/lens combination has proven to be exceptional in providing high quality images even in harsh conditions like heavy shadows in bright sunlight and at night, using only the vehicle headlights for illumination. This camera provides RGB as well as NTSC video output. The camera can be mounted in three different positions, depending on the software system that is in use. When using PANS to test forward looking lateral vehicle control and driver monitoring algorithms, it is mounted on the inside rear view mirror mounting bracket. For the downward looking lateral lane position system it is mounted on a special suction cup plate, which is attached to the side window of the vehicle. For intersection detection and traversal experiments, the camera is mounted on a Directed Perception pan tilt unit that is located on the roof of the vehicle. This unit can pan the camera approximately 160 degrees in either direction and tilt it from -30 degrees to 45 degrees.

The output from the digitizer, which is usually just a playthrough of the incoming video signal from the camera along with overlay graphics, is connected to a Sony FDL-X600 color LCD monitor. The display is mounted on the dashboard, directly in front of the forward passenger seat.

A key component for both our local and global positioning algorithm is a Trimble SVeeSix - CM2, differential ready GPS system. This unit's specifications are typical for entry level 6 channel GPS receivers: 25 meter position and 0.1 meter/second velocity accuracy without SA. The positional accuracy figure improves to 2 - 5 meters when operating in differential mode. These numbers have been experimentally verified to be correct. The GPS unit is interfaced to the portable computer using a serial line.

Differential corrections are supplied by a Navstar base station unit, mounted at a known location on a tower on top of our vehicle storage area. This unit supplies standard RTCM-104 differential corrections using Motorola Cellect modems over a cellular phone link to the SVeeSix. This distribution mechanism has proven to be robust in areas of poor cellular coverage and over extremely long baselines.

The second component which is integral to the PANS positioning system is an Andrew Corporation AUTOGYRO® with digital output. This fiber optic gyroscope provides updates to the position estimation system running on the portable computer at 10 Hz using a 9600 baud serial line.

Figure 65 PANS components inside the Navlab 5.

The unit can measure rotation rates between 0.02 degrees/second and 100 degrees/second. In addition to rotation rate, device temperature is provided over the serial link. This allows for compensation of the unit's bias drift to the 18 degree/hour level. (0.005 degree/second).

Low level vehicle control and safety monitoring are accomplished using an HC11 microcontroller. The HC11 uses a serial line connection to receive commands from and send information to the vehicle control and position estimation (VCPE) module running on the laptop. The primary function of the HC11 is to servo the steering wheel and provide turn radius information to the VCPE module. The HC11 is equipped with a quadrature decoder board and a digital to analog converter. The quadrature decoder board provides the current steering wheel position as given by the steering wheel encoder. A PID control algorithm running on the HC11 uses this information along with the target steering position supplied by the VCPE module to compute an appropriate steering motor torque, which is passed to the motor amplifier using the D/A board. In addition to controlling the steering wheel, the HC11 provides an interface to the vehicle's cruise control module, turn signals and brake lights. The cruise control interface allows the VCPE to do rudimentary speed control.

The second function of the HC11 is to monitor system safety at a low level. (High level safety measures are implemented in the VCPE module.) There are six mechanisms for doing this. The first is monitoring the lateral acceleration of the vehicle. If the VCPE module detects too high of an acceleration value, it commands the HC11 to disengage the steering motor. The second is through monitoring commands from the VCPE module. This module is the HC11's link to the rest of the system. If for any reason, the HC11 stops receiving commands from the VCPE module, it disengages the steering wheel by removing power to the steering motor. The third safety mechanism associated with the HC11 is the user engage/kill switch. The switch, which is typically mounted on the dashboard of the vehicle, allows the user to initiate automatic steering control and to quickly stop it if the situation warrants. Again, this is done by removing power to the steering motor. The fourth safety mechanism that the HC11 provides is a heartbeat signal, which goes to a separate, custom monitoring board. If the heartbeat signal is ever absent, the monitoring board can independently cut power to the steering motor. The fifth safety mechanism is the steering motor itself. It is intentionally underpowered, and can provide a peak torque of only 2 ft-lbs. This torque is sufficient to servo the wheel to a desired position, but is small enough to be easily overcome by the safety driver in case he must take over. The final safety mechanism is steering wheel position error monitoring. This safety mechanism is provided so that the system does not continue to fight the safety driver if intervention is necessary. This mechanism is implemented by removing power to the steering motor if the steering wheel has not moved toward its commanded position within a short period of time. In all cases, if power is cut to the steering motor, the user must actively reset the safety system before autonomous steering control can resume.

The PANS platform requires very little power. It uses about 140 watts, most of which are required by the portable computer. The power breakdown is as follows: computer 90 watts, camera 12 watts, LCD display 9 watts, fiber optic gyroscope 7.5 watts, GPS 1.4 watts, other 10 watts. Because of these minimal requirements, the system is operated using a commercially available Radio Shack DC/AC inverter which is connected to the vehicle's cigarette lighter. When the steering motor is used during autonomous control experiments, an additional 72 watts (maximum) of power is needed. The motor is also powered from vehicle's electrical system, but a separate connector is used to avoid overloading the cigarette lighter circuitry.

The final piece of hardware is the steering wheel motor and encoder. Although not strictly part of PANS, it is required for autonomous lane keeping experiments. The motor is from a retired robot arm and is equipped with a Hewlett-Packard optical quadrature encoder. It drives the steering wheel using a chain and sprocket mechanism, and is mounted under the dash on a modified steering column support bracket. The motor has been sized so that it provides adequate torque for highway driving but still allows easy operator takeover - about as difficult as driving with reduced power steering.

6.4 Position Estimation

A design goal of the PANS platform was to accurately estimate vehicle state parameters without physically attaching sensors to the vehicle, maximizing portability from one vehicle to another. This is achieved by avoiding contact sensors such as potentiometers or position encoders whenever possible, and instead relying on non-contact sensors, including a GPS and gyroscope.

Part of the PANS platform is the Vehicle Control and Position Estimation (VCPE) module. In addition to providing vehicle control and safety services, this modules provides global and local

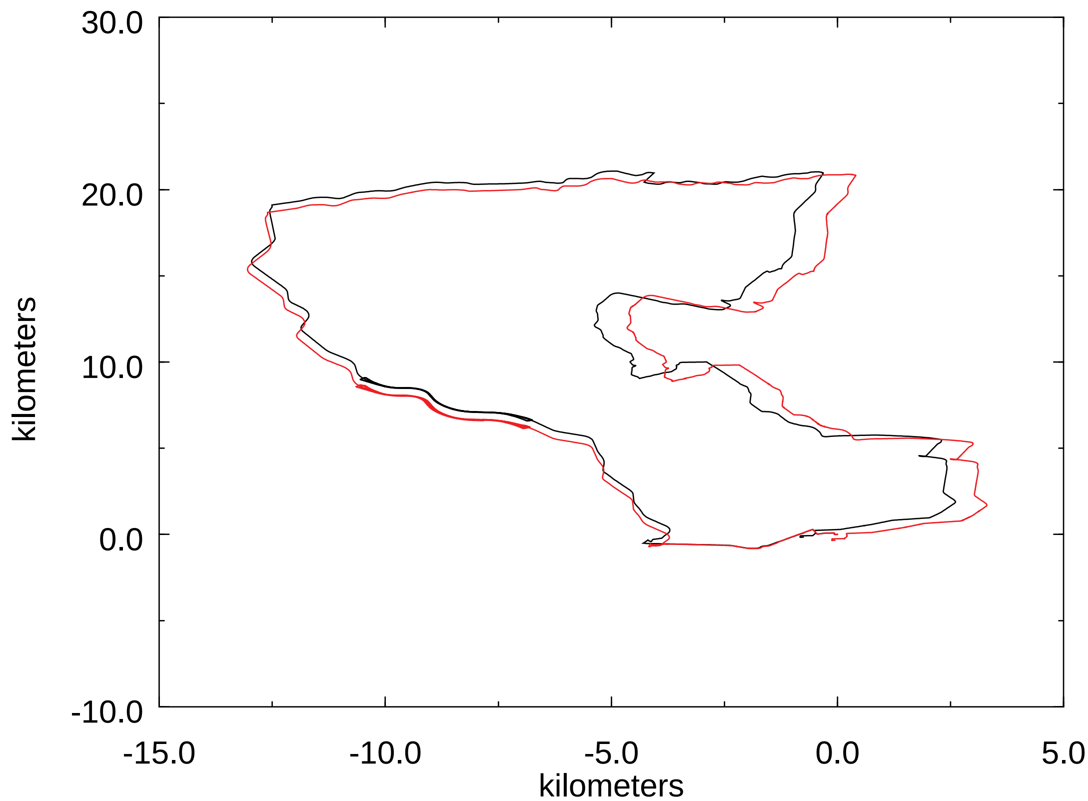


Figure 66 Two runs on the 100 km course.

position estimates to high level applications. Position estimates are updated at 20 Hz using the latest available sensor data.

6.4.1 Global and Local Positioning

Global position is provided using information from the GPS in either latitude/longitude/altitude or UTM coordinates. The VCPE module automatically detects when the GPS is operating in 2D or 3D mode, as well as when differential corrections are available, and provides this status information, along with global position data, to client applications. Because of the low update rate of the GPS, linear extrapolation is done between new GPS readings so that more accurate global position estimates can be attained. Using the GPS in differential mode, vehicle position can be determined to within 5 meters.

The VCPE module also provides a local estimate of 2D position. This position is updated more quickly than the global GPS position (see below) and does not contain the large position jumps that are possible when using GPS data. The origin of the local coordinate frame is the location where the vehicle was positioned when the VCPE module was started. The coordinate frame is arranged so that north, as provided by the GPS, is the positive Y axis. The positive X axis is defined to be due east (90 degrees clockwise from north.) In addition to X and Y position, the VCPE module provides estimates of heading, turn radius (rate of change of heading), vehicle velocity and total distance traveled. The following paragraphs detail how each of these values is calculated.

6.4.2 X Y Position

New X Y position estimates are calculated using velocity and heading information from the GPS along with turn rate information from the gyroscope or steering encoder. Specifically, new X Y positions are computed by projecting along the current vehicle turn radius. The projection begins at the old vehicle position (x, y, heading) and continues along the turn radius for a distance determined by the current vehicle velocity and the time since the last update. Although simple, this method is robust in many different scenarios including both stop-and-go city driving as well as high speed highway driving. The X Y position accuracy of the VCPE is consistently below 0.8% of distance traveled and has exhibited even better performance during trials on closed test tracks.In one experiment to determine the accuracy of local position estimation, the vehicle was driven four times around a closed, 12 km, test track at Transportation Research Center (TRC) in Columbus, Ohio. During the experiment, the vehicle was traveling at velocities between 35 and 40 meters/second. At the end of this experimented, the accumulated error was less than 40 meters. This figure is less than 0.1% of distance traveled.

Two runs on a more challenging 100 km course that included downtown city streets, interstate highways and rural roads are shown in Figure 66. This course included several stoplights and U turns, which lead to vehicle velocities between 0 to 25 meters/second. This figure clearly shows position differences accumulating between the two runs as the distance traveled increases. The overall local position error was still quite small. On one of the runs, the error was about 0.35% of distance traveled while on the other it was about 0.77%. A close-up of the start/end points of the two runs is shown in Figure 67.

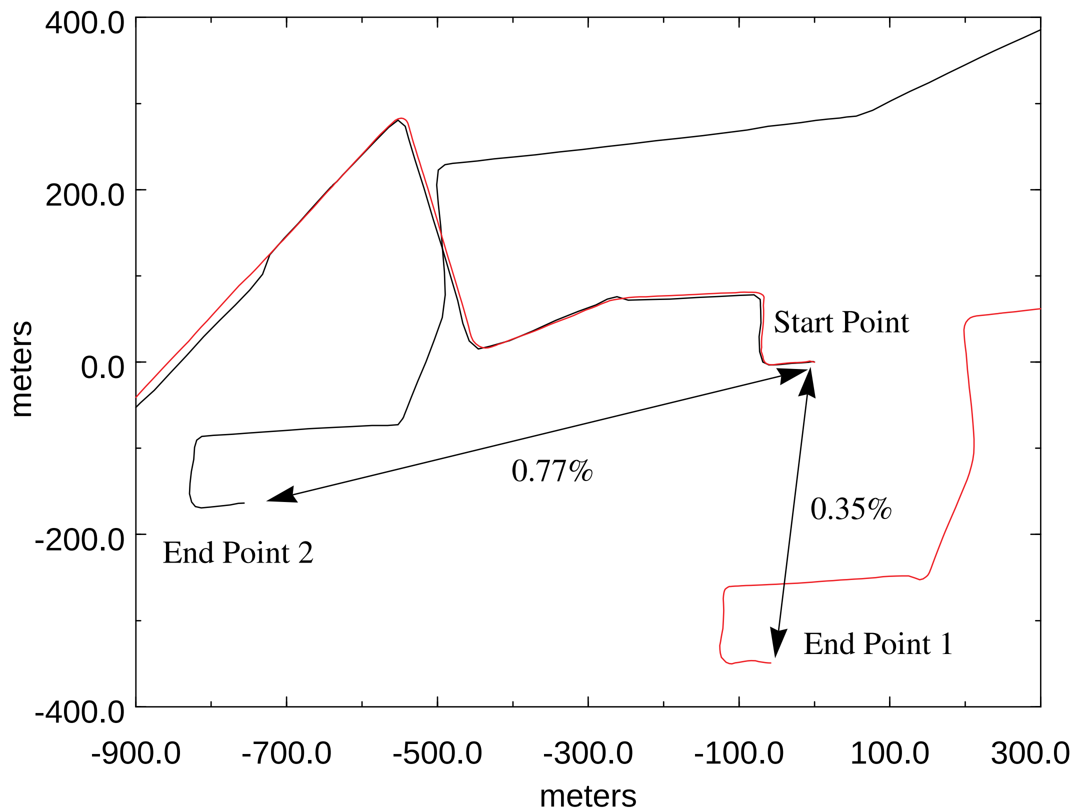


Figure 67 Start/end points for the two 100 km runs.

6.4.3 Heading

Heading is determined using information from the GPS system. The GPS provides an estimate of heading once per second. Between GPS readings, heading is updated using turn rate information from the fiber optic gyroscope or the steering wheel encoder. When new GPS heading data becomes available, it over-writes the current heading.

6.4.4 Turn Radius

The vehicle turn radius is usually derived from two sources - the fiber optic gyroscope and the steering wheel encoder. Both instruments can be used independently of each other, but normally, the gyroscope is used to calibrate the steering wheel encoder. (The steering wheel encoder can also be manually calibrated.)

Calibration is accomplished by computing the turn radius using rate of change of heading information from the gyroscope along with the current vehicle speed. The formulation is shown in the following equation.

$$radius = \frac{180 \bullet \underline{velocity}}{\pi \bullet \overline{heading}}$$

radius is in meters
$\underline{velocity}$ is in meters/second
$\overline{heading}$ is in degrees/second

The VCPE module compares the gyroscope-based and steering wheel encoder turn radius measures, and slowly adapts the encoder calibration parameters so that the two sensors match. We have found that in order to insure accuracy using this approach, the vehicle speed must be greater than 10 meters/second. Using this technique, the current vehicle curvature can be estimated with an accuracy of 0.000333 meters^{-1}.

The gyroscope can also be used stand-alone to determine the turn radius when a steering wheel encoder is not available. While the 10 Hz output of the gyroscope is not sufficient for closed loop control, this level of accuracy and update frequency is more than sufficient for monitoring the driver's steering command in a lane departure warning system.

Finally, if neither gyroscope or steering wheel encoder are available, turn radius is computed using the vehicle speed and differentiating the GPS supplied heading information. Because updates only occur about once per second in this mode, it is used only to estimate the current vehicle position - not for controlling the vehicle or as a measure used for driver warning.

6.4.5 Velocity

Vehicle velocity is mainly acquired using the GPS. The GPS specifications state a velocity accuracy of 0.1 meter/second. Although not verified to this level, we have empirically determined that it is accurate to about 0.5 meters/second. A secondary source of vehicle velocity is from the speed pulse count that is used by the vehicle's cruise control module. This signal is currently only used when driving at low speeds (where GPS doesn't register velocity consistently) or when the GPS signal is not available. We have found this measure to be less accurate than the GPS, but it is available at lower speeds and updated more frequently.

Chapter 7 Conclusions

7.1 Summary

In this dissertation, I have developed both the Virtual Active Vision tools and a suite of techniques based on them, which enable autonomous vehicles to execute tactical level driving tasks. I have shown that these techniques, in some cases coupled with active camera control, can reliably control the vehicle while it executes tasks such as lane transition, exit and entrance ramp detection and traversal, obstacle avoidance maneuvers, and intersection detection and navigation. I have used the ALVINN lane keeping system as a base on which to build these techniques and have shown that Virtual Active Vision tools extend its capabilities to new domains. The techniques were all tested and validated in real world driving scenarios.

7.2 Contributions

In developing tools for executing tactical level driving tasks, I have investigated and advanced several areas of vision based, on-road mobile robot navigation. These areas range from the development of a theoretical, unified framework for executing many tactical level driving tasks, to the development of algorithms based on this framework which allow vision based lane keeping system to exhibit state-of-the-art tactical driving skills. These areas of contributions, along with others, are described in the following paragraphs.

This dissertation presents a novel method for analyzing and executing tactical level driving tasks using vision based lane keeping systems. I have developed a framework in which vision based lane keeping systems can be used to execute tactical level driving tasks. Others have succeeded in developing systems explicitly designed to execute a particular tactical task, like finding intersections or determining if other driving lanes are present. But because of their narrow scope, extension of these systems to other driving tasks is difficult. The tools I have developed provide a unified structure for executing a wide range of tasks, using existing low level, vision based control algorithms as their foundation. This framework provides a canonical way of viewing tactical level driving tasks and I have demonstrated that this framework is powerful and flexible enough to allow robust driving performance in real world scenarios. In addition, this framework does not limit the driving performance of the underlying lane keeping system. This framework has been principally demonstrated on two task areas - lane transition and intersection navigation. Within these areas, several sub-tasks were examined and techniques to execute them were developed. In addition to working in simulation, these techniques were validated on real roadways using our testbed vehicle. In each case, the framework provided by Virtual Active Vision tools made possible competent execution of the target driving task.

At a more experimental level, I have demonstrated new algorithms for vision based vehicle control in challenging, real work driving situations. For lane transition, new algorithms were developed which allow vision based execution of five lateral control functions which are required for typical highway driving. In addition, I have demonstrated algorithms which can robustly detect and traverse intersections using only visual cues. The intersection detection algorithm uses the intersection itself, rather than secondary or absent features, to determine if a road junction is present. The traversal algorithm requires only image-derived information to move the vehicle through the intersection and on to the desired road branch. Traditional traversal techniques like dead reckoning are not required. Both the detection and traversal algorithms are based on the ability of the system to actively control the camera position so that the portions of the image required for the task are in view. By integrating active camera control, advanced performance was achievable.

In addition to being useful when combined with high-level, tactical decision modules, I have shown that Virtual Active Vision tools can be used successfully with bottom up view control approaches. One example of this idea was presented in discussion on intersection branch detection and localization in Chapter 4. Information derived from the image was used to move the view, both laterally and by rotation, so that it became properly aligned with the road. This technique required no high level knowledge - only information from the current road scene. A second example, presented in Chapter 5, is the RALPH system's ability to move the view closer to the vehicle when it detects that the distal portion of the image contains irrelevant information. Once again, this action is a result of some characteristic in the image, not a directive from a high level module. Both of these examples illustrate an important characteristic of bottom-up view placement approaches - they increase the performance and autonomy of the system because they allow it to view the world in which it is operating in a way that is more meaningful and understandable.

Next, I have shown that by working within the Virtual Active Vision framework, development of new driving capabilities is accelerated. An illustration of this concept is presented in Chapter 3. In this chapter, several techniques relating to lane transition were presented. The first of these to be investigated was the basic case of moving the vehicle from one lane to another. After this was accomplished, the development of the other related capabilities, like exit lane detection and traversal, was straightforward. It merely involved adapting the lane transition techniques to the new problem, adding some functionality as called for by the task, or removing that which was no longer necessary.

I have also demonstrated that a reactive lane keeping system can be effectively used to execute tasks that require guidance from other, more knowledgable sources. Specifically, I have shown that the ALVINN neural network driving system, when enhanced with Virtual Active Vision tools, can be used as the core of a system which can perform many tactical driving tasks. This result directly addresses and at least partially refutes the belief that purely reactive systems like ALVINN are not suitable to perform tasks that require some model or a priori knowledge of the world. ALVINN's ability to execute these tasks, without loss of flexibility or performance, is a direct result of the transparent model that Virtual Active Vision tools place on the system.

Finally, Virtual Active Vision tools and the concepts that they instantiate have proven to be usable by other vision based lane keeping systems. Virtual Active Vision tools enhance these systems' driving performance and increase their functionality, just as they have done for ALVINN.

7.3 Future Work

Although the methods I have developed are designed to be used by Tactical Level Decision Modules, the implementation of the interface between the TLDM and the lane keeping system has not been the focus of this research. For autonomous driving research to continue to progress and move to the next level of competence, TLDM's must be more tightly integrated, using tools like those presented here, with the low level modules responsible for vehicle control. A good example of this concept has been developed and tested in the ARPA Unmanned Ground Vehicle (UGV) program, where the goal is to integrate many low level functionalities into a system which can execute militarily relevant tasks. Other researchers are currently investigating the applicability of the decision making technology used for the UGV program to high level vehicle guidance and control during tactical on-road driving tasks. Their research is from the point of view of the high level module, and it is anticipated that it will complement the work described in this dissertation [13].

A related and more defined area of necessary future work is the integration of geometric terrain information. For highway driving, the flat world assumption is sufficient for good performance at most tasks, but for city driving tasks like intersection detection and navigation, the need for terrain information becomes apparent. By supplementing Virtual Active Vision tools with terrain information, the creation of virtual view images would be more accurate, leading to better performance. Current work is underway by other researchers to develop terrain sensing systems and detailed digital maps that can provide the kind of information that is required. It should be noted that any terrain information would be helpful, ranging from the approximate slope of a large portion of the image to centimeter accurate range maps.

As is shown in the intersection branch detection and RALPH view control algorithms, bottom-up approaches for positioning virtual camera views hold much promise for extending the range of tactical tasks which can be executed as well as for increasing the performance and autonomy of the underlying lane keeping system. From these early results, the value of examining methods which use image features to focus attention on relevant scene areas is evident. Research is already underway, using a tool called a saliency map, to determine what image features are relevant [1]. A saliency map can robustly indicate which regions of the image are meaningful, where meaningful is defined by the task being executed. By combining the results of this work with the already developed virtual camera view control algorithms, more capable systems can be developed.

Another area of future work is the application of the tools that I have developed to alternative and next generation lane keeping systems. This may be done directly using the Virtual Active Vision techniques described in this thesis, or by adapting the underlying principles to suit the form of the new systems. This evolution may continue to simply be using the software libraries that implement the Virtual Active Vision tools that I have developed or may expand to hardware implementations of a few key tools. As reported in Chapter 5, this type of work is already underway. The difficultly in proceeding along this line of research is not deciding whether Virtual Active Vision techniques will provide benefit to other lane keeping systems, but rather lies in assessing the cost/benefit of using the tools directly, or just the concepts that they represent. A careful analysis of a particular lane keeping system's capabilities along with its functional requirements is needed to determine what aspects of Virtual Active Vision will be of most benefit to the system and how they should be implemented. It is expected that by integrating Virtual Active Vision concepts from the start, like what was done for RALPH, the base lane keeping system will become more versatile.

A similar area to that discussed in the last paragraph is the application of Virtual Active Vision tools to domains other than navigation. Countless researchers use focus of attention and active vision techniques, similar to those presented in this dissertation, to simplify the image processing task. These methods could be coarsely characterized as Virtual Active Vision, but because they are usually inherent to the system and specialized to a single task, the application of the methods to other similar tasks is difficult. The power of Virtual Active Vision tools comes from their ability to extend the performance of simple vision based modules. They do this by representing the input image in a manner that is familiar to the base system. Additionally, when the system lacks and needs geometric grounding, Virtual Active Vision tools can provide it. These characteristics point to Virtual Active Vision tools' potential usefulness in two broad application areas. The first is for tasks which can be simplified by geometrically transforming the input image into a canonical form. Visually indentifying several different randomly placed parts on a moving conveyor belt is an example of this. The identification system could be designed to differentiate parts only when they are in a single position and orientation in the image. Virtual Active Vision tools could be used to transform the image of all potential parts, which may be at different locations and orientations on the belt, into a pose suitable for processing.

The second application area is artificial neural network based image processing. Because neural networks typically learn to associate an input image pattern with a particular output, the correct presentation of the image pattern is very important. Furthermore, Virtual Active Vision tools extend a network's generalization capabilities without requiring additional, potentially detrimental, training. Essentially, Virtual Active Vision tool allow the network to learn the simplest task. They can be used to transform possibly confusing images into ones which can be correctly interpreted by the network. Related work in face detection using neural networks and techniques like those just described has already been reported and is quite promising [47].

A final area of future work is in the application of Virtual Active Vision tools to additional driving tasks. One such task is vision based obstacle detection. Virtual Active Vision techniques similar to those used for detecting the presence or absence of other driving lanes may be applicable to this domain. One idea is to use virtual cameras to image only the scene areas which could contain relevant obstacles. It is unclear whether Virtual Active Vision tools could provide a great benefit for this type of task, because obstacle detection systems already typically work in the language of geometry.

As is likely evident, the particular tasks that were examined and the underlying vision system which was used in this dissertation were uniquely suited for Virtual Active Vision tools. This is understandable because the concept of Virtual Active Vision was developed based on the requirements of these tasks and the capabilities and limitations of the vision system. In general, the tools that I have developed as part of this thesis research can be used most directly with, and make the largest contribution to, systems which operate in a confusing environment and need abundant geometric information to perform successfully, yet have difficulty acquiring or incorporating that same information. This description exactly characterizes the execution of tactical driving tasks by a vision-based, adaptive driving system. As a result, the ALVINN lane keeping system, supplemented with Virtual Active Vision tools, was able to perform many tactical level driving tasks, with little modification.

Chapter 8 References

[1] Baluja, S. and Pomerleau, D.A. "Using the Representation in a Neural Network's Hidden Layer for Task-Specific Focus of Attention," C. Mellish (ed.) *The International Joint Conference on Artificial Intelligence 1995*, Montreal, Canada. pp133-139.

[2] Behringer, R., Holt, V. Dickmanns, D. "Road and Relative Ego-State Recognition," *1992 IEEE Symposium on Intelligent Vehicles,* June 29 - July 1, 1992, Detroit, Michigan, USA.

[3] Bohrer, S., Zielke, T. and Freiburg, V. "An Integrated Obstacle Detection Framework for Intelligent Cruise Control on Motorways," *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[4] Broggi, Alberto. "A Massively Parallel Approach to Real-Time Vision-Based Road Markings Detection," *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[5] Chen, X., Dagless, E., Zhang, S. and Thomas, B. "A Real-Time Plan-View Method for Following Bending Roads," *1993 IEEE Symposium on Intelligent Vehicles*, July 14-16, 1993, Tokyo, Japan, pp. 219 - 224.

[6] Crisman, J. D. *Color Vision for the Detection of Unstructured Roads and Intersections.* Ph.D. dissertation, Carnegie Mellon University, May, 1990.

[7] DeMenthon, D. and Davis, L. "Reconstruction of a Road by Local Image Matches and Global 3D Optimization," *1990 IEEE International Conference on Robotics and Automation,* Cincinnati, OH, USA, May 13-18, 1990.

[8] Dichmanns, E.D. and Zapp, A. "Autonomous high speed road vehicle guidance by computer vision." *Proceedings of the 10th World Congress on Automatic Control*, Vol. 4, Munich, West Germany, 1987.

[9] Dickmanns, E.D. "Active Bifocal Vision," *7th International Conference on Image Analy-*

sis and Processing, Monopoli, Italy, September 20-22, 1993.

[10] Dickmanns, E. D., Behringer, R., Dickmanns D., Hildebrandt, T., Maurer, M., Thomanek, F., and Schielen, J., "The seeing passenger car 'VaMoRs-P,'" *1994 IEEE Symposium on Intelligent Vehicles*, pp. 68-73.

[11] Dickmanns, E.D. "Performance Improvements for Autonomous Road Vehicles," *Intelligent Autonomous Systems-4,* March, 1995, Karlsruhe, Germany.

[12] Enkelman, W., Nirschl, G., Gengenbach, V., Kruger, W., Rossle, S., Tolle, W. "Realization of a Driver's Warning Assistant for Intersections," *1993 IEEE Symposium on Intelligent Vehicles*, July 14-16, 1993, Tokyo, Japan, pp. 72-77.

[13] Gowdy, Jay. "SAUSAGES: Between Planning and Action," CMU Technical Report CMU-RI-TR-94-32, June 1994.

[14] Graefe, Volker "Vision for Intelligent Road Vehicles," *1993 IEEE Symposium on Intelligent Vehicles*, July 14-16, 1993, Tokyo, Japan, pp. 135-140.

[15] Grupen, R., Connolly, C. Souccar, K., and Burleson, W. "Toward a Path Co-Processor for Automated Vehicle Control," *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[16] Haga, T., Sasakawa, K., and Kuroda, S. "The Detection of Lane Boundary Markings Using the Modified Spoke Filter," *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[17] Harada, H. "Stability Criteria and Evaluation of Steering Maneuver in *Driver-Vehicle System*," JSME Int. J. C, Dyn. Control Robot. Des. Manuf. (Japan), vol.37, no.1, March 1994, pp. 115-22.

[18] Hatipoglu, C., Ozguner, U., and Unyelioglu, K. "On Optimal Design of a Lane Change Controller," *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[19] Hess, R.A. and Modjtahedzadeh, A. "A Preview Control Model of Driver Steering Behavior," *1989 IEEE International Conference on Systems, Man and Cybernetics,* Cambridge, MA, USA, November 14-17, 1989.

[20] Jochem, T.M. Pomerleau, D.A., Thorpe, C.E. "MANIAC: A Next Generation Neurally Based Autonomous Road Follower," *Intelligent Autonomous Systems-3,* February 1993, Pittsburgh, PA, USA.

[21] Jochem, T.M. and Baluja, S. "Massively Parallel, Adaptive, Color Image Processing for Autonomous Road Following," in *Massively Parallel Artificial Intelligence*, Kitano and Hendler (ed), AAAI Press, 1994.

[22] Jochem, T. M. "Using Virtual Active Vision Tools to Improve Autonomous Driving Tasks," CMU Technical Report CMU-RI-TR-94-39, October, 1994.

[23] Jochem, T.M., Pomerleau, D.A., and Thorpe, C.E. “Vision Guided Lane Transition,” *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[24] Jochem, T. M., Pomerleau, D.A., Kumar, B. and Armstrong, J. “PANS: A Portable Navigation Platform,” *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[25] Kenue, S.K. (1989) “Lanelok: Detection of lane boundaries and vehicle tracking using image-processing techniques,” *SPIE Conference on Aerospace Sensing, Mobile Robots IV,* Nov. 1989.

[26] Kim, K.I., Oh, S.Y., Kim, S.W., Jeong, H., Han, J.H., Lee, C.N., Kim, B.S. and Kim, C.S. “An Autonomous Land Vehicle PRV II: Progress and Performance Enhancement,” *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[27] Kluge, K. *YARF: An Open Ended Framework for Robot Road Following*. Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, February 1993.

[28] Kluge, K and Thorpe C. “Intersection Detection in the YARF Road Following System,” *Intelligent Autonomous Systems 3,* February 1993, Pittsburgh, PA, USA.

[29] Kluge, K., and Lakshmanan, S. “A Deformable Template Approach to Lane Detection,” *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA.

[30] Kushner, T. and Puri, S. “Progress in Road Intersection Detection for Autonomous Vehicle Navigation,” SPIE Vol. 852, Mobile Robots II, 1987, pp. 19 - 24.

[31] Lotufo, R., Dagless, E., Milford, D., Thomas, B., “Road Edge Extraction Using a Plan-view Image Transformation,” *4th Alvey Vision Conference.*

[32] Meng, M. and Kak, A., “Mobile Robot Navigation Using Neural Networks and Nonmetrical Environmental Models,” *IEEE Control Systems,* October 1993, pp. 30-39.

[33] Muller, N. and Baten, S. “Image Processing Based Navigation with an Autonomous Car,” *Intelligent Autonomous Systems-4,* March, 1995, Karlsruhe, Germany.

[34] Mysliwetz, B. and Dickmanns, E. “A Vision System With Active Gaze Control for Real-Time Interpretation of Well Structured Dynamic Scenes,” Intelligent Autonomous Systems, December 8-11, 1986, Amsterdam, Netherlands.

[35] Nashman, M. and Schneiderman, H. “Real-Time Visual Processing for Autonomous Driving,” *1993 Symposium on Intelligent Vehicle,* July 14-16 1993, Tokyo, Japan, pp. 373-378.

[36] Nelson, W. “Continuous-Curvature Paths for Autonomous Vehicles,” *1989 IEEE International Conference on Robotics and Automation,* Scottsdale, AZ, USA, May 14-19, 1989.

[37] Pomerleau, D.A. “Efficient Training of Artificial Neural Networks for Autonomous Navi-

gation," *Neural Computation 3:1*, Terrence Sejnowski (Ed).

[38] Pomerleau, D.A., "Progress in Neural Network-based Vision for Autonomous Robot Driving," *1992 IEEE Symposium on Intelligent Vehicles,* June 29 - July 1, 1992, Detroit, Michigan, USA.

[39] Pomerleau, D.A., Gowdy, J., and Thorpe, C.E. "Combining Artificial Neural Networks and Symbolic Processing for Autonomous Robot Guidance," *1992 IEEE Symposium on Intelligent Vehicles,* June 29 - July 1, 1992, Detroit, Michigan, USA.

[40] Pomerleau, D.A. *Neural Network Perception for Mobile Robot Guidance*. Ph.D. dissertation, Carnegie Mellon University, February, 1992.

[41] Pomerleau, D.A. and Touretzky, D.S. "Analysis of Feature Detectors learned by a Neural Network Driving System," *Intelligent Autonomous Systems-3,* February 1993, Pittsburgh, PA, USA.

[42] Pomerleau, D. A. "RALPH: Rapidly Adapting Lateral Position Handler," *1995 IEEE Symposium on Intelligent Vehicles*, September 25-26, 1995, Detroit, Michigan, USA

[43] Richter, S. and Wetzel, D. "Adaptive Change of Strategies for Road Recognition and Road Tracking," *Intelligent Autonomous Systems-4,* March, 1995, Karlsruhe, Germany.

[44] Rosenblum, Mark. "The Use of Neural Networks for Machine Vision Applications," University of Maryland Department of Computer Science Master's Thesis, May 1994.

[45] Rosenblum, Mark. Email messages, September 12 and September 29, 1995.

[46] Rossle, S., Kruger, V., and Gengenbach. "Real-Time Vision-Based Intersection Detection for a Driver's Warning Assistant," *1993 Symposium on Intelligent Vehicle,* July 14-16 1993, Tokyo, Japan, pp. 340-344.

[47] Rowley, H., Baluja, S., Kanade, T. "Human Face Detection in Visual Scenes," To Appear in: *Neural Information Processing Systems 8.*

[48] Siegle, G., Geisler, J., Laubenstein, F., Hagel, H. and Struck, G. "Autonomous Driving on a Road Network," *1992 IEEE Symposium on Intelligent Vehicles,* June 29 - July 1, 1992, Detroit, Michigan, USA.

[49] Smith, D.E. and Starkey, J.M. "Effects of Model Complexity on the Performance of Automated Vehicle Steering Controllers: Controller Development and Evaluation," Vehicle Systems Dynamics (Netherlands), vol.23, no.8, November 1994, pp. 627-45.

[50] Solder, U. and Graefe, V. "Object Detection in Real Time," SPIE Mobile Robots V, 1990.

[51] Struck, G., Geisler, F., Laubenstein, H., Nagel, H., Siegle, G. "Interaction Between Digital Road Map System and Trinocular Autonomous Driving," *1993 IEEE Symposium on Intelligent Vehicles*, July 14-16, 1993, Tokyo, Japan, pp. 461- 466.

[52] Sukthankar, R., Pomerleau, D., Thorpe, C. "Panacea: An Active Sensor Controller for the

ALVINN Autonomous Driving System," *1993 International Symposium on Robotics Research.*

[53] Taniguchi, M., Hosaka, A., Kurami, K., and Takei, A. "The Development of Autonomously Controlled Vehicle, PVS," *1991 Vehicle Navigation and Information Systems Conference,* October 20-23 , 1991, Dearborn, MI, USA, pp. 1137-1141.

[54] Turk, M., Morgenthaler, D., Gremban, K., and Marra, M. "VITS - A Vision System for Autonomous Land Vehicle Navigation," *IEEE Transactions in Pattern Analysis and Machine Intelligence,* Volume 10, Number 3, May 1988.

[55] Ulmer, Berthold "VITA - An Autonomous Road Vehicle (ARV) for Collision Avoidance in Traffic," *1992 IEEE Symposium on Intelligent Vehicles,* June 29 - July 1, 1992, Detroit, Michigan, USA.

[56] Weshofen, K. P. "Real-time Road Scene Classifaction Based on a Multiple-lane Tracker," *International Conference on Industrial Electronics*, San Diego, CA, USA, pp 746-751.

[57] Weber, J., Koller, D., Luong, Q.T., and Malik, J. "An integrated stereo based approach to automatic vehicle guidance," *International Conference on Computer Vision,* June, 1995, Boston, MA, USA.

[58] Wallace, R., Matsuzaki, K., Goto, Y., Crisman, J., Webb, J., and Kanade, T. "Progress in Robot Road Following," *1986 IEEE Conference on Robotics and Automation,* pp. 1615-1621.

[59] Waxman, A., LeMoigne, J., Srinivasan, B.. "Visual Navigation of Roadways," *1985 IEEE Conference on Robotics and Automation,* St. Louis, MO., USA, March 1985, pp. 862-867.

[60] Waxman, A., LeMoigne, J., Davis, Larry., Srinivasan, B., Kushner, T., Liang, E., Siddalingaiah, T. "A Visual Navigation System for Autonomous Land Vehicles," *IEEE Journal of Robotics and Automation,* Volume RA-3, number 2, April 1987.