Initial Results in Vision Based Road and Intersection Detection and Traversal

Todd M. Jochem

CMU-RI-TR-95-21

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

April 1995

This research was partly sponsored by DARPA, under contracts "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-R059, monitored by TACOM) as well as a DARPA Research Assistantship in Parallel Processing administered by the Institute for Advanced Computer Studies, University of Maryland.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. government.

Initial Results in Vision Based Road and Intersection Detection and Traversal

Todd M. Jochem

Dean A. Pomerleau

Charles E. Thorpe

The Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213

Abstract

The use of artificial neural networks in the domain of autonomous driving has produced promising results. ALVINN has shown that a neural system can drive a vehicle reliably and safely on many different types of roads, ranging from paved paths to interstate highways[9]. The next step in the evolution of autonomous driving systems is to intelligently handle road junctions. In this paper, we present an addition to the basic ALVINN driving system which makes autonomous detection of roads and traversal of simple intersections possible. The addition is based on geometrically modelling the world, accurately imaging interesting parts of the scene using this model, and monitoring ALVINN's response to the created image.

1. Introduction

Much progress has been made toward solving the autonomous lane-keeping problem. Systems 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 to determine where lane markings are expected[2][5][6], while others are based on artificial neural networks which learn the salient features required for driving on a particular road type[3][4][9]. In general, many aspects of this problem are no longer research issues, but rather engineering tasks.

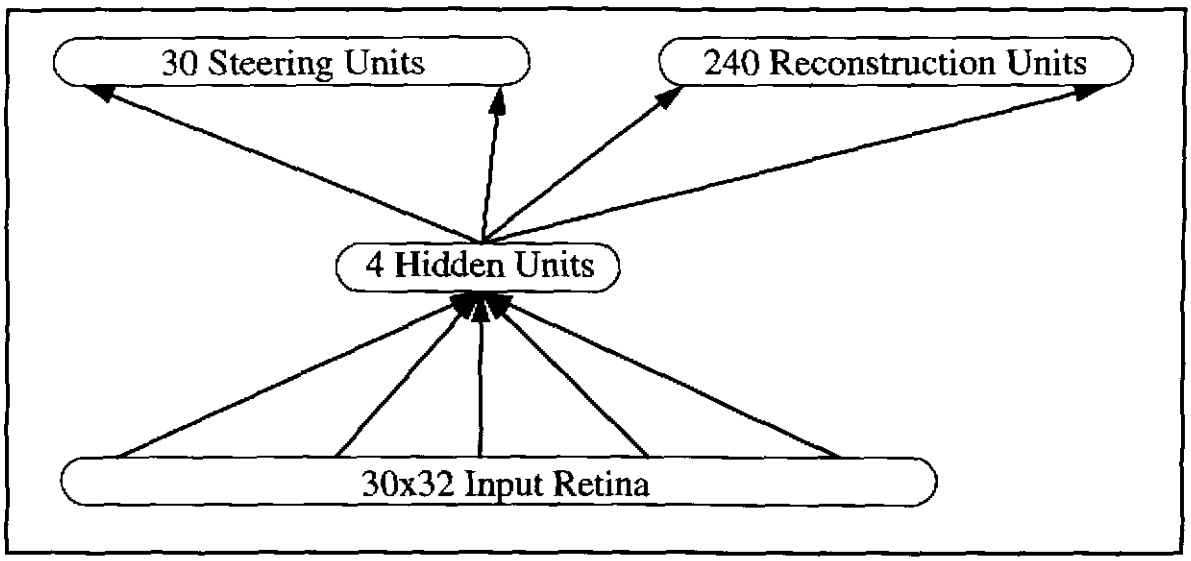

Figure 1: ALVINN network architecture.

The current challenge for vision-based navigation researchers is to create systems that maintain the performance of the already developed lane-keeping systems, and add the ability to do higher level driving tasks. These tasks include actions such as lane changing, localization, and intersection detection and navigation. This papers examines the task of road and intersection detection and navigation.

ALVINN (Autonomous Land Vehicle In A Neural Network)[9] is the neural network based lane-keeping system upon which the work presented in this paper is based. 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, (and 240 additional reconstruction units) ALVINN can quickly learn, using back-propagation, the correct mapping from input image to output road location. See Figure 1. This steering direction can then be used to control our testbed vehicle, a converted U.S. Army HMMWV called the Navlab 2. On this vehicle, ALVINN has driven at speeds up to 55 m.p.h. for 90 continues miles.

The extended system, one which is capable of detecting roads and intersection, is called ALVINN

VC (VC for Virtual Camera). ALVINN VC uses the robust road detection and confidence measurement capability of the core ALVINN system along with an artificial imaging sensor to reliably detect road segments which occur at locations other than immediately in front of the vehicle (and the camera.)

The imaging sensor that ALVINN VC uses is called a **virtual camera** and is described in detail in section 2. Virtual cameras are the fundamental tool upon which the techniques presented in this paper are based. They provide a mechanism for determining the appropriateness of vehicle actions. Finally, they do not compromise the robust driving performance of the original ALVINN system.

In order to more tightly integrate virtual cameras into a high performance driving system, one significant change to the basic ALVINN system was needed. This change was moving away from a system that produces a steering arc to one which produces 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.

2. The Virtual Camera

A virtual camera is simply an imaging sensor which can be placed at any location and orientation in the world reference frame. It creates artificial 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, called virtual images, 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 most road following scenarios, it accurately represents the world near the vehicle. Virtual camera views from many orientations have

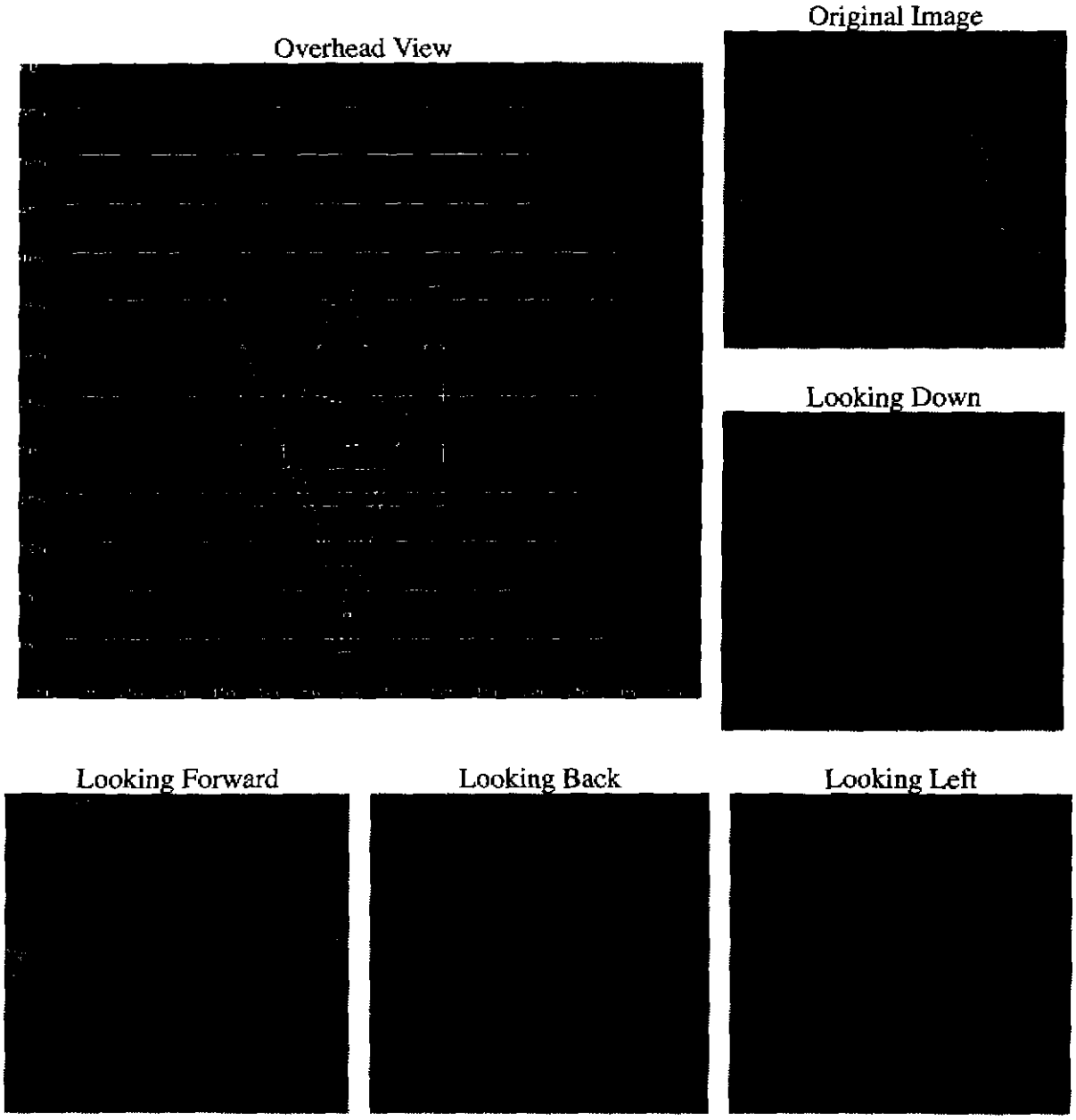

Figure 2: Typical virtual camera scenes.

been created using images from three different actual cameras. The images produced by these views have proven to be both accurate and usable by ALVINN VC to successfully navigate on all road types which the original ALVINN system performed. Figure 2 shows some typical virtual camera scenes.

An interesting issue that is a general theme of this paper is showing that virtual cameras are well suited for merging neural systems with symbolic ones. Virtual cameras impose a geometric model on the neural system. In our case, the model is not a feature in an image, but rather a canonical image viewpoint which ALVINN VC can interpret. To ALVINN VC, the virtual camera is a sensing device. It is ALVINN VC's only link to the world in which it operates. ALVINN VC doesn't care where the virtual camera is located, only that it is producing images which are similar to those one which it was trained and can thus be used to locate the road. This interpretation may seem to trivialize ALVINN VC's functionality, but in reality, finding the road is what ALVINN VC is designed to do best. The virtual camera insures that the system gets images which will let it do its job to the best of its ability. The details of creating appropriate virtual camera locations and interpreting the resulting output are left to other, higher level modules. So in essence, the virtual camera imposes a geometric model on ALVINN VC without it knowing, or even caring, about it. The model allows the system to exhibit goal directed, intelligent behavior without compromising system performance.

3. Detection Philosophy

There are three principles upon which road and intersection detection and 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. Road junctions should be traversed by actively tracking the road or intersection branch.

These principles and their relationship to ALVINN VC, as well as other road and intersection detection systems, are examined in greater detail in the following sections.

3.1. 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 robot's world now, rather than when the model was built. The pixels from the current image should be used to determine if a road or 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 entirely miss the desired road or intersection. But in a data directed scheme, the system can use images of its current surroundings to locate the desired road or intersection.

A data driven method does not exclude using some kind of model to assist it in determining where a road or intersection is likely to occur, though. In the example above, a coarse model 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 can begin searching for it. In this way, the model allows the system to focus it resources (computing power) on the most important task - staying on the road - until they are needed for some other task (finding the intersection). Another example is that a model could provide information about what type of intersection or road is approaching, and thus aid the system in determining which area of the input image contains information required for detection. The model only guides, rather than executes, the act of detection.

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 is not really be occurring - the robot would just moni-

tor it position and execute 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.

3.2. Detection Signaling

Positive detection should be signaled by the *presence* of a road or intersection branch as opposed to the absence of some other feature. This means that positive detection is signaled when the system 'sees' a road or 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 based system, this principle translates to using the capabilities of the system, rather than its shortcomings, to derive useful information from the scene.

3.3. Active Search

When a person turns onto a road or traverses an intersection, they don't 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 road or intersection they want to traverse, they continue to monitor the road 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 be used in computer based road and intersection detection.

A person is able to monitor the road by turning their head so that the road they are moving onto is

continually in view. For computer 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 envisioned. In this joint method, the desired placement of the virtual cameras would guide the movement of the actual camera.

3.4. ALVINN VC

ALVINN VC uses a priori knowledge that specifies where and when appropriate virtual cameras should be created. The cameras are created relative to the vehicle and the creation does not coincide with when the intersection is actually located, but rather somewhere before it occurs. Instead of "The intersection is expected now," the system deals with information like "Start looking for a single lane road." The camera's location, and the network associated 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 image it in a way that is meaningful to the system's neural networks. (The image is transformed so that it is in the same orientation as the images which were used to train the network.) By continually monitoring the network's confidence for each virtual camera, the system can determine when the road or intersection is present.

ALVINN VC currently adheres to the first two principles of road junction detection and traversal

mentioned earlier. Also, methods are under development that will allow ALVINN VC to actively track roads using a combination of active camera control and intelligently placed virtual cameras.

4. Other Systems

Several other groups have built systems to study the road and intersection detection problem. 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 a road or intersection is present. Few use active camera control. The following sections present a sampling of these systems.

4.1. 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 the Navlab II on city streets and 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, as 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[7].

4.2. 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 systems neural network. When the system noticed a sharp increase in this error metric, indirectly caused by the

absence of the important driving features, it signaled to the coarse mapping module that an intersection was present. At this point, the output vector of the neural network was examined for a bimodality. If one was found, the reply from the coarse 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 means that the intersection and intersection branches must fall within the field of view of the camera and that the orientation of the branches must be within the neural networks range of response[9].

4.3. 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 this system used to detect intersection 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 intersection 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[1].

4.4. 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 value. Although this system looks for positive indicators of intersections, it is very reliant on models which are based not on the geometry of the intersection, such as its branches and their orientation, but rather on markings painted in the roadway or in the center of the intersection[10].

5. Experimental Results

A series of experiments was conducted to assess the usefulness of virtual cameras for autonomously detecting roads and intersections. All of the experiments described were performed on the Navlab 2. The experimental site 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. ALVINN VC 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 first experiment was very simple and designed to test the basic ability of virtual cameras to create images which were usable by the system. For this, a single virtual view was used to detect an upcoming road and to navigate onto it.

The second experiment was more challenging - virtual cameras were used to not only keep the vehicle on the road, but to also detect an upcoming 'Y' intersection. After the intersection was detected, the system used higher level information to choose the appropriate fork to follow.

5.1. Road Detection Experiment

This experiment was designed to test the system's robustness for detecting and navigating onto single roads. In this experiment, the system had information that the vehicle was approaching a road perpendicularly. Its job was to detect the road, drive the vehicle onto it, and then continue

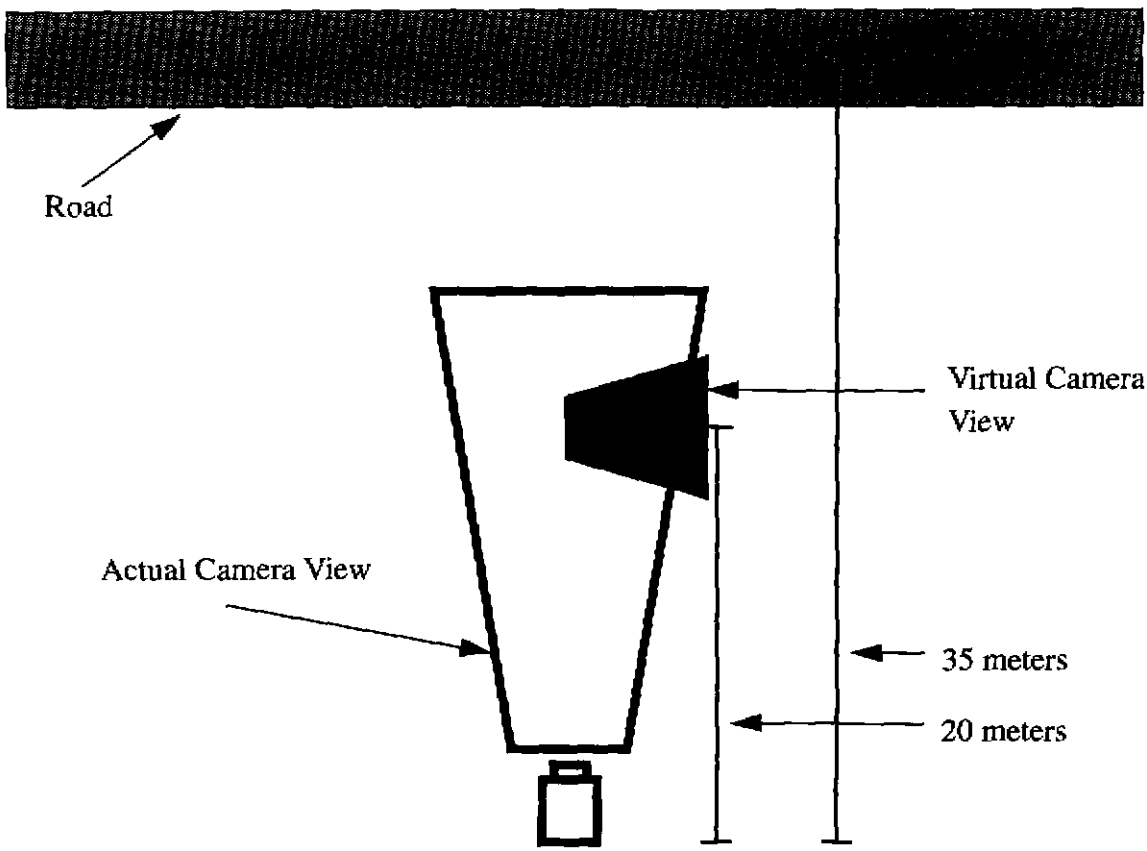

Figure 3: Road Detection Scenario.

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 corresponds 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 of the road which was to be detected, and aligned perpendicularly to it. A virtual view that was rotated -90 degrees from the direction of vehicle travel was created. This view was placed 20 meters in front of the vehicle to allow enough time for turning onto the road. See Figure 3. 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 the experiment are presented in more detail in the following sections.

5.2. 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 is described in greater detail in Section 5.2.1. and 5.2.2. 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 VC reported that it had located the road.

5.2.1. 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[9].

5.2.2. 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, the system'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 4 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 small image superimposed in the lower right corner is the image created by the virtual camera. 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.

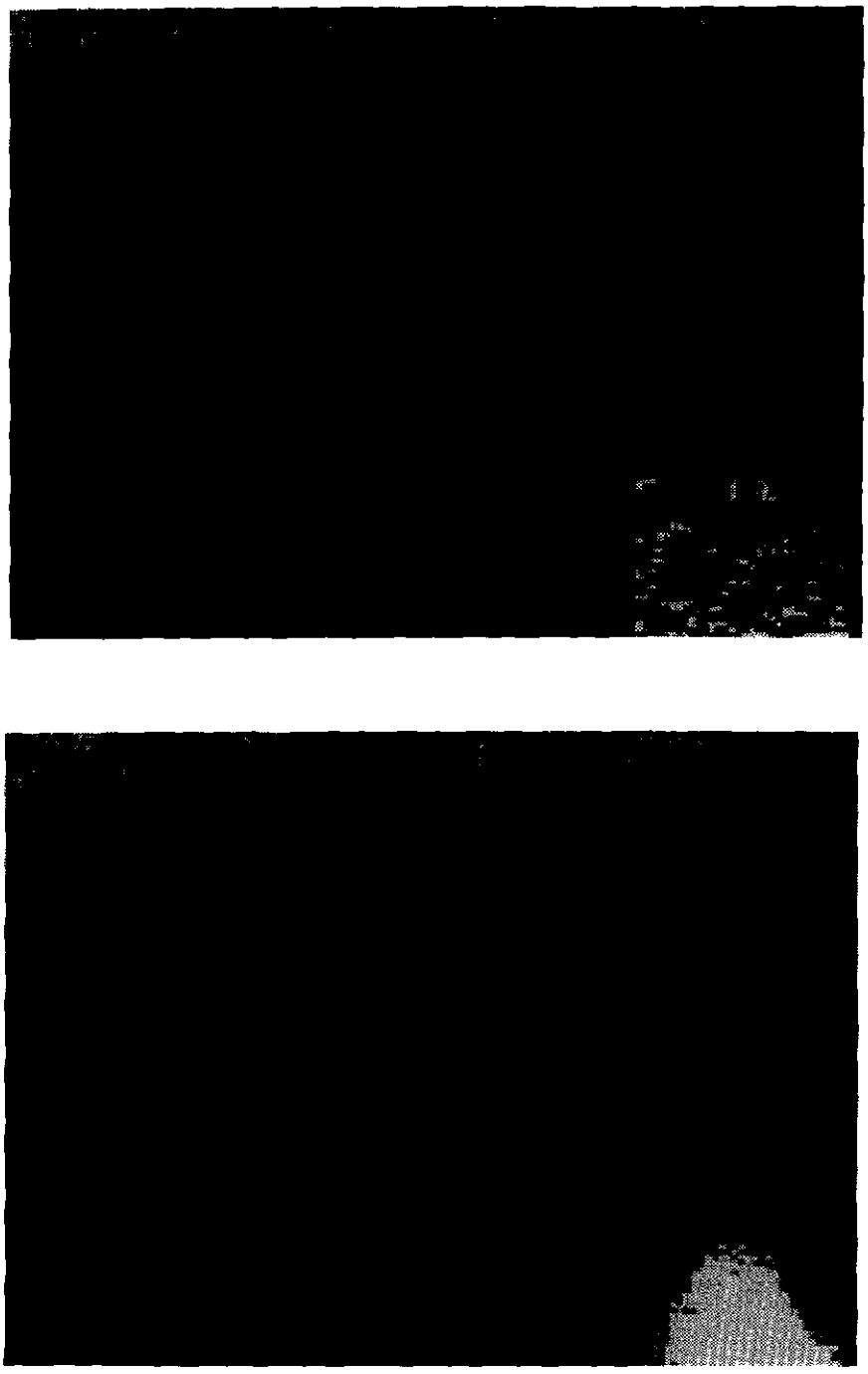

Figure 4: The top image shows the scene in front of vehicle as viewed by the actual camera just before the virtual view, outlined in red, images the road. The small image in the lower right is the preprocessed virtual camera view that the system's network uses. The bottom image shows the same scene a 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.

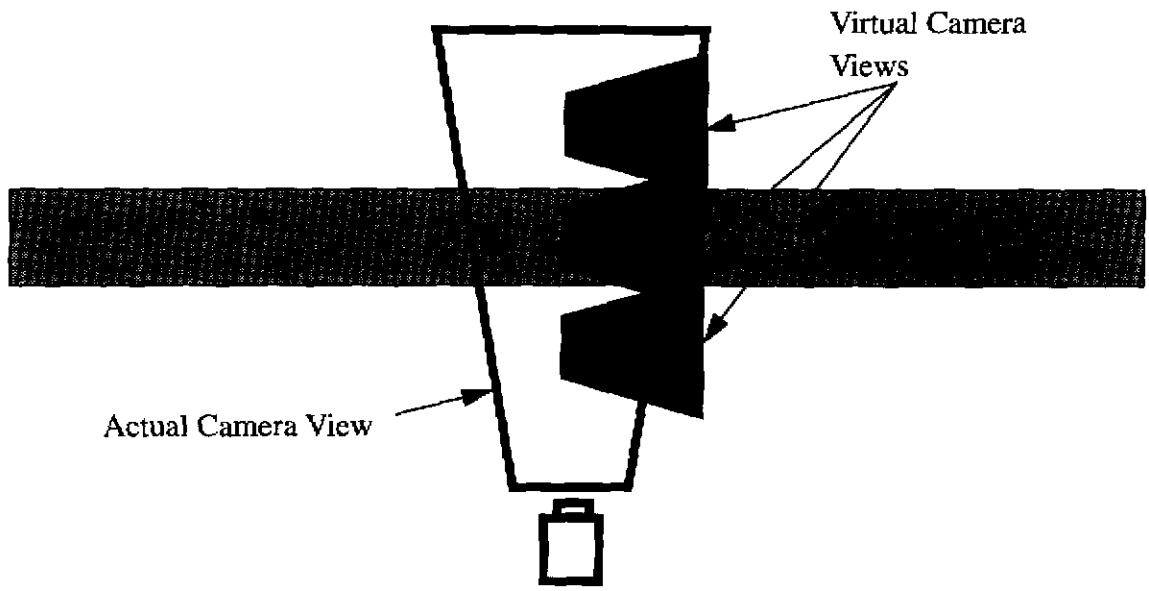

Figure 5: Virtual Camera Spacing.

To 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 5 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 to 29 meters.

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 result of this experiment are shown in Figure 6. It shows the IRRE response with respect to the position of the virtual view for each

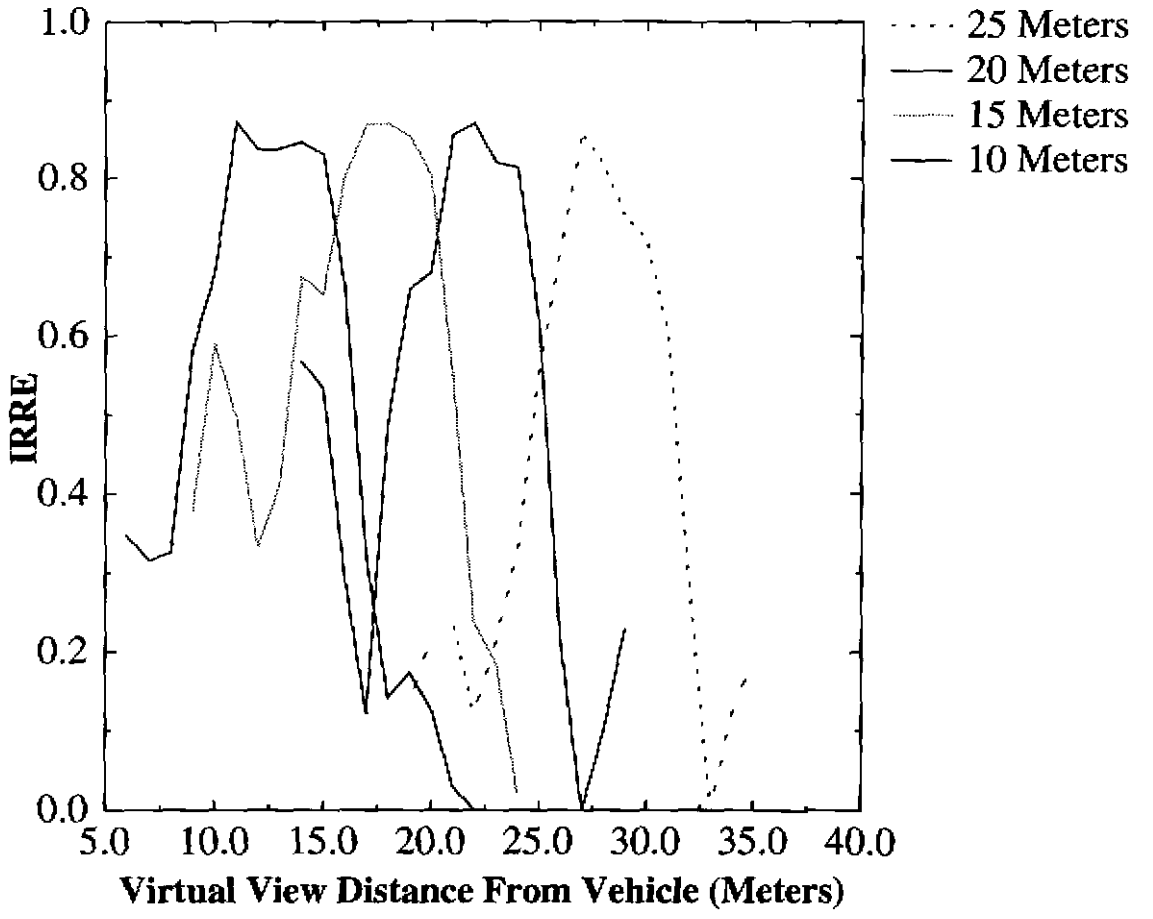

Figure 6: IRRE response as a function of virtual camera distance from vehicle using images taken at different distances from the road. For comparison, when the system is driving on a familiar road, the IRRE response is typically between 0.70 and 0.95.

actual image. 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 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 7. 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 right corner is very similar to the top image in Figure 4. 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 increase correspondingly. The trend continues until the virtual view is centered over the road, as shown in

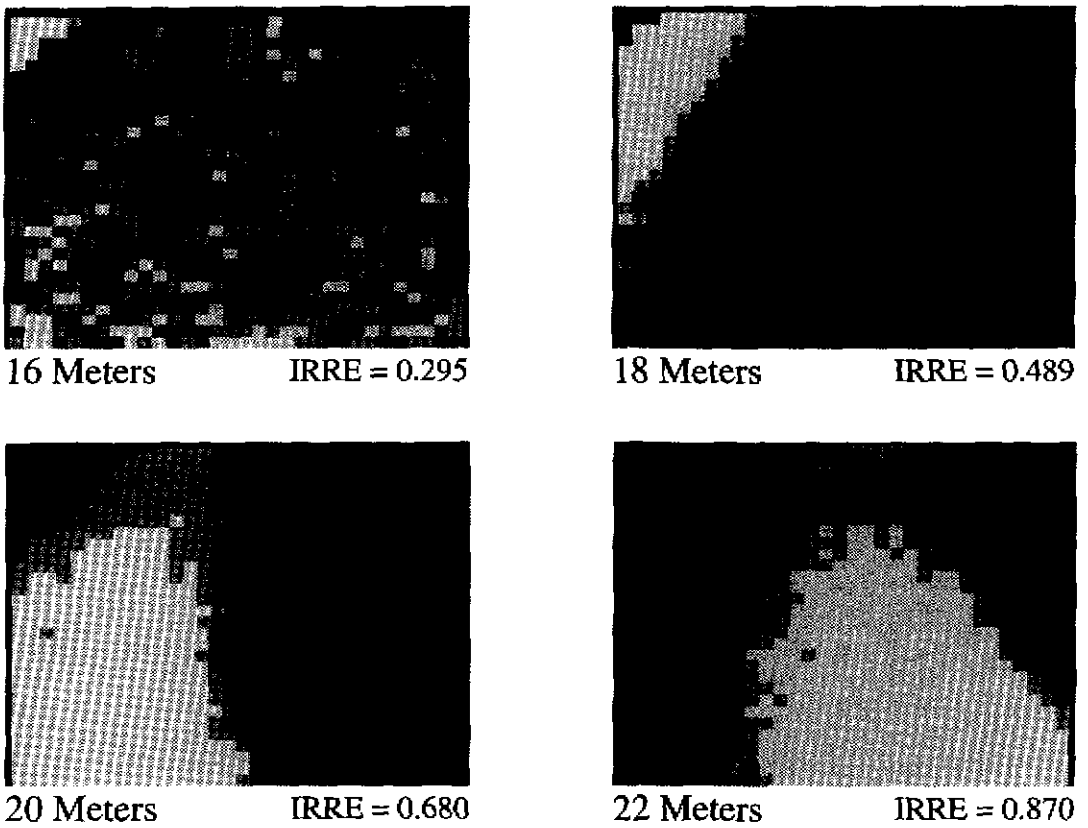

Figure 7: The IRRE confidence value increase, as expected, when the virtual view images larger portions of the road.

the lower right image. At this location, the IRRE value is at its peak.

Each of the IRRE response curves shown in Figure 6 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 6 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 6 into this coordinate frame is shown

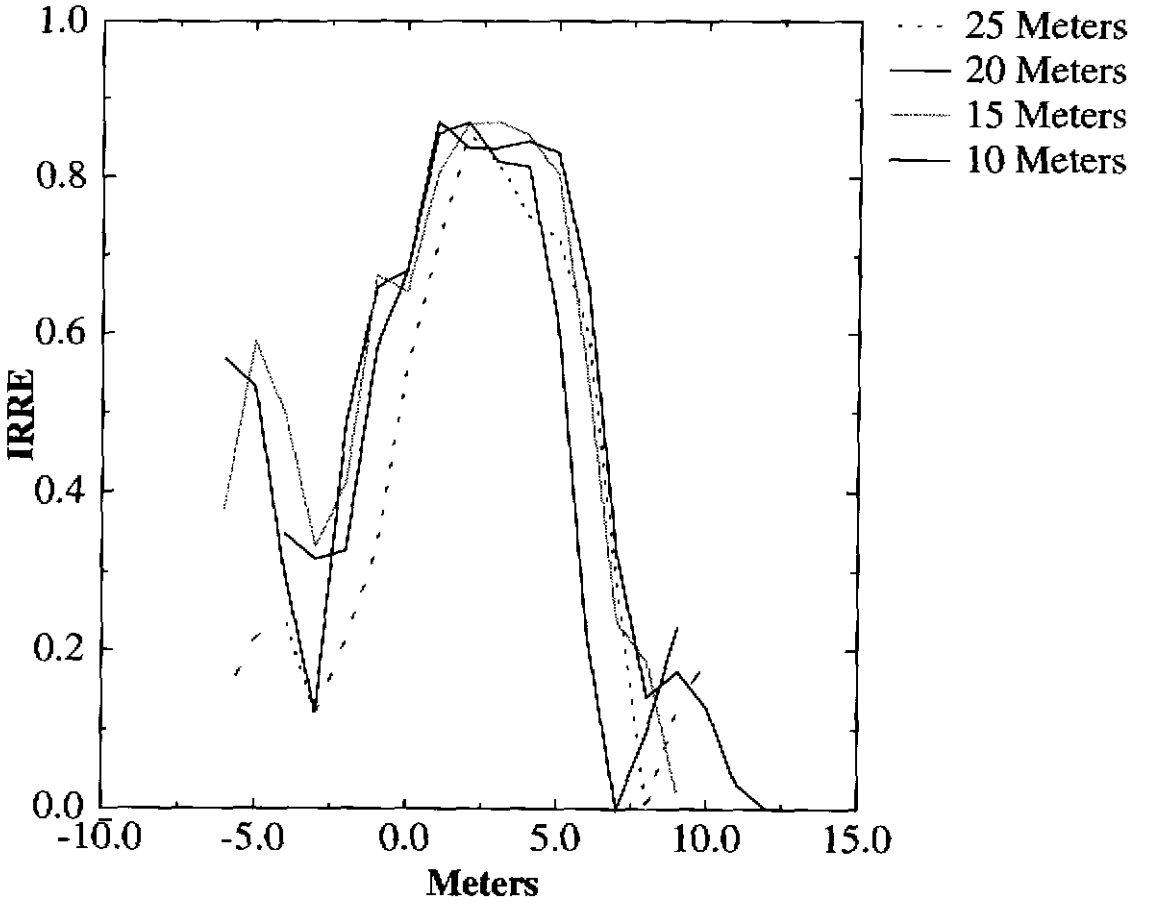

Figure 8: IRRE response as a function of virtual camera distance from the center of the road. Negative number are closer to the road and positive number are further away.

in Figure 8. This graph shows that detection accuracy, at least when approaching the road perpendicularly, does not degrade as distance from the road increases.

5.2.3. 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 the system'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[8]. (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 5.3.2.

5.3. 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. For each algorithm, the case of a 90 degree intersection angle is discussed.

5.3.1. 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 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 ori-

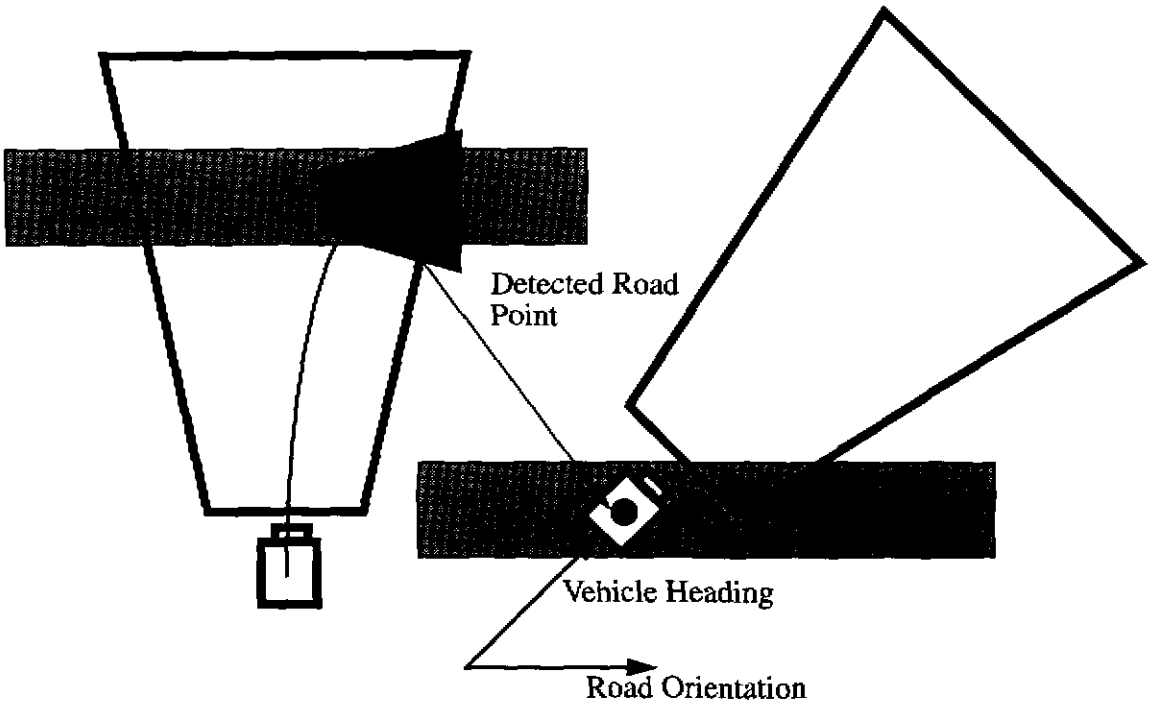

Figure 9: Vehicle/Road Misalignment.

entation. See Figure 9. This mismatch was a function of the angle at which the vehicle approached the road. Small angles led to minor misalignment, while larger one 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 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, the road position computed using the virtual view does not provide enough offset from the original direction of travel to allow the necessary turn to be executed. The virtual view, and, in turn, the computed road position, is constrained by the actual camera - a large portion of the virtual view must be within the actual camera's field of view in order for real-

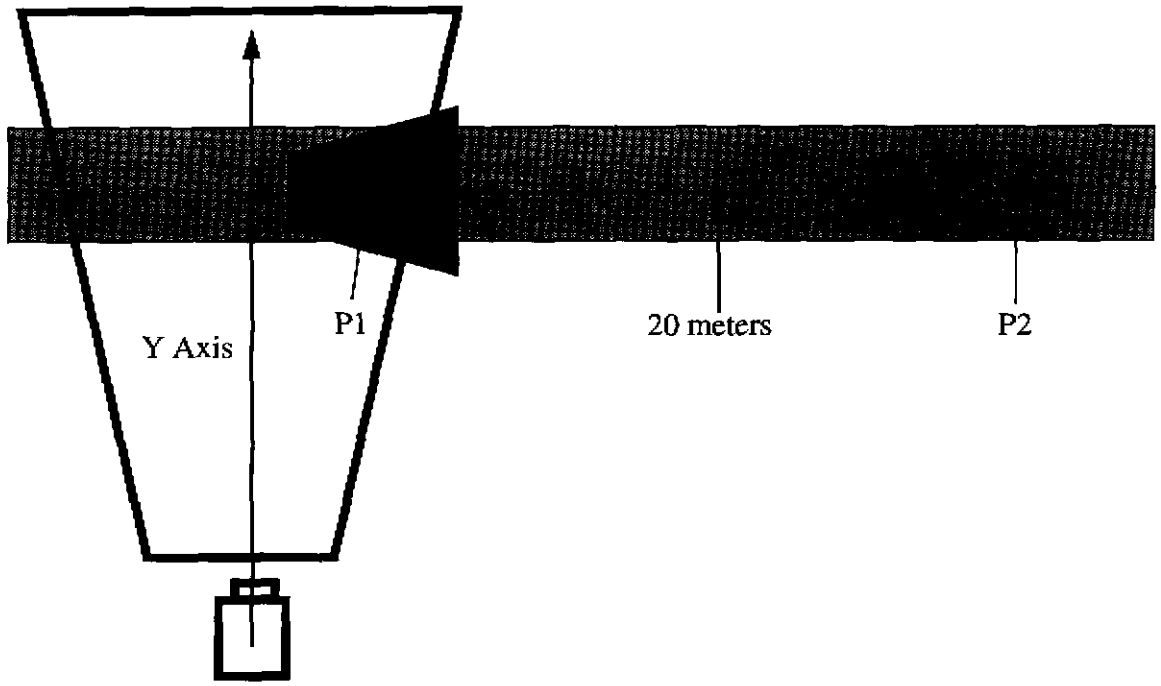

Figure 10: P2 Point Projection Diagram.

istic images to be created. This result suggests that another point on the road, further along it in the desired direction of travel, is needed.

5.3.2. 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). 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 10.) P2 is assumed to be on the road. The projection distance was typically 20 meters. See Figure 10.

Whereas the first technique computed a single 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 increase 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 point. 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 represent 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 argument 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 reason 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 large, 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 VC's neural network is trained on

images which are created as if the vehicle was shifted and/or rotated from it 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.

5.3.3. Network Output Based Projection and Active Camera Control

Although moderate success was achieved using the previous two methods, system performance was far from optimal. Two additional road alignment methods have been developed but not sufficiently tested to warrant reporting results in this paper. The first involves using the system's steering output to increase the accuracy of the projected point. In this method, the vehicle location that would be required to give rise to the virtual view being used to detect the road is computed. Because the network is trained to produce the offset from straight ahead at some lookahead distance and because the 'virtual' vehicle location is known, it is possible to more accurately estimate the local orientation of the road. Figure 11 illustrates this procedure.

The second method involves applying the third principle of detection stated in section 3. 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. In this method, the system continually tracks the road using knowledge from previous images about where it was located. This method uses changing virtual camera views to transform the important areas of the scene into ones the system can understand. By continually updating the position of the road in this manner, errors which occur and cannot presently been compensated for, like the 2 meter bias mentioned in section 5.2.2., can be minimized. This type of system is the goal of this

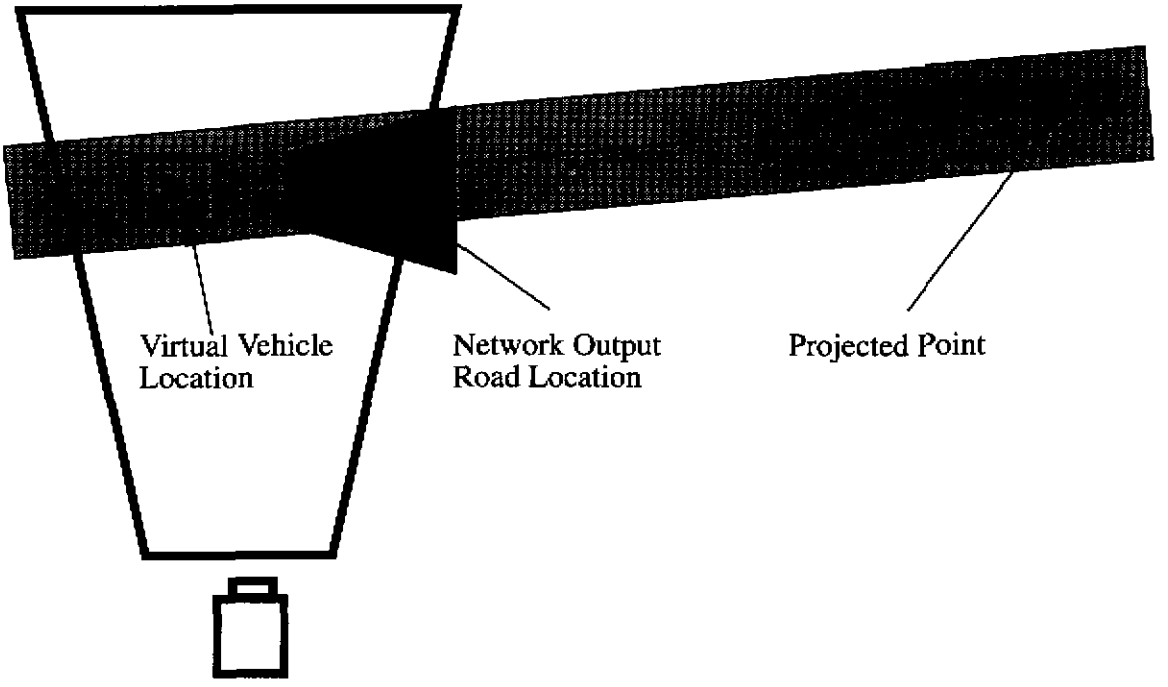

Figure 11: Network Output Based Point Projection Diagram.

research.

5.4. 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. See Figure 12. 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 then 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 are inclined. This means that the flat world assumption is violated.

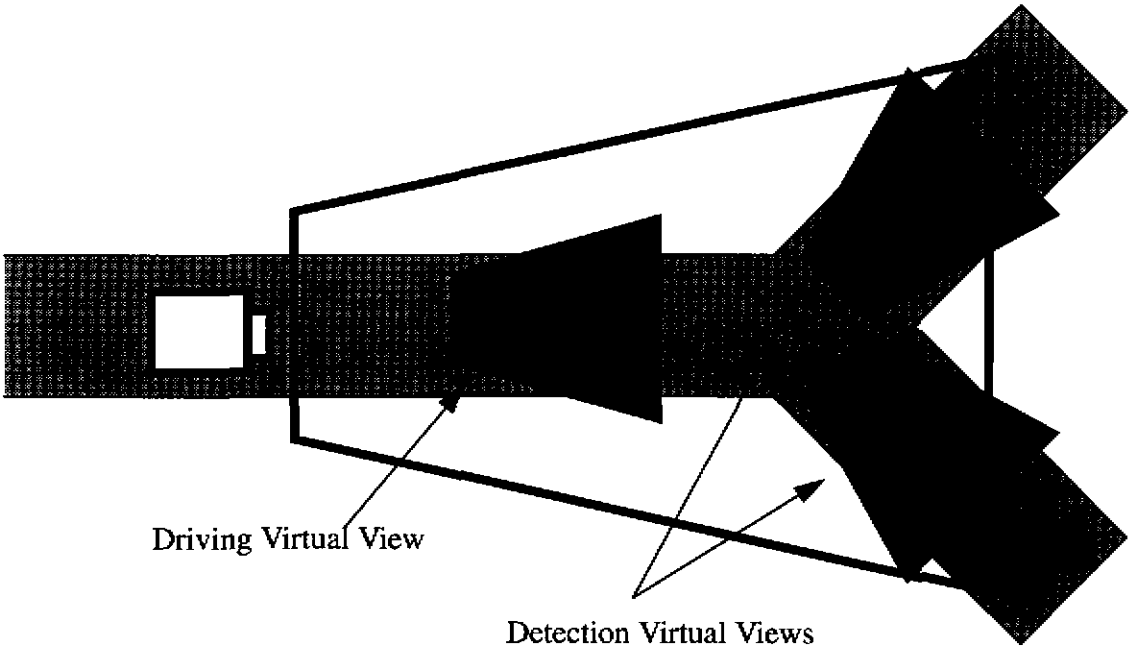

Figure 12: "Y" Intersection Geometry.

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 for was 20 degree right of straight. Because of the geometry of the situation, the IRRE threshold value, which both virtual images were required to produce, on a single actual image 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. 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 fin the single road detection case presented earlier, the system still had problems navigating onto either branch.

6. Conclusions and Future Work

Clearly, there is much work left to be done to robustly detect all roads and intersections. This paper presents a vision based approach which uses the core of a robust neural network road follower to accurately detect single lane, unlined roads. It is reasonable to assume that the detection method is directly extendable to any road type which the base neural network can learn to drive on. If this assumption is, in fact, found to be 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.

Currently, the weak link of the system is its ability to navigate road junctions once they are found. Although the active camera control methods mentioned have not been tested in real world scenarios, they will likely form the basis for forthcoming research.

Finally, the results presented were from a 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. These areas are also actively being pursued.

7. Acknowledgments

This research was partly sponsored by DARPA, under contracts "Perception for Outdoor Navigation" (contract number DACA76-89-C-0014, monitored by the US Army Topographic Engineering Center) and "Unmanned Ground Vehicle System" (contract number DAAE07-90-C-R059, monitored by TACOM) as well as a DARPA Research Assistantship in Parallel Processing administered by the Institute for Advanced Computer Studies, University of Maryland.

8. References

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

[2] 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.

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

[4] Jochem, T. 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.

[5] 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.

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

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

[8] Pomerleau, D.A. "Efficient Training of Artificial Neural Networks for Autonomous Navigation," *Neural Computation 3:1,* Terrence Sejnowski (Ed).

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

[10] Rossle, S., Kruger, V., and Gengenbach. "Real-Time Vision-Based Intersection Detection for a Driver's Warning Assistant," *Intelligent Vehicle '93,* July 1993, Tokyo, Japan.