# AVCS Research at Carnegie Mellon University

Dean Pomerleau, Charles Thorpe, Dirk Langer, Julio K. Rosenblatt and Rahul Sukthankar

Robotics Institute

Carnegie Mellon University

Pittsburgh PA USA

## Abstract:

For the last 10 years, Carnegie Mellon University has been building increasingly competent systems for autonomous driving. Our approach has been to develop smart vehicles, capable of driving in natural outdoor environments without intervehicle communication or infrastructure modifications. Our computer-controlled vehicles now drive themselves at speeds up to 55 mph and for distances of over 90 miles on public roads without human intervention. They are capable of driving both during the day and night, on a wide variety of road types. They can sense and avoid obstacles, and even automatically parallel park. These technologies have been develop as part of ARPA's Unmanned Ground Vehicle (UGV) program, with the goal of reducing the need for human presense in hazardous situations such as battlefield surveillance missions. These advances can also reduce the risk to civilian drivers as part of advanced vehicle control systems. The techniques we have developed are suitable both for AHS applications where the vehicle is controlled automatically, and in driver warning systems where the role of the AVCS system to monitor the environment and suggest actions for the human driver. This paper presents some of the capabilities of our systems, and the processing techniques that underlie them. These techniques include: artificial neural networks for road following, model-based image processing for convoy following, smart obstacle maps based on sonar, ladar and microwave sensor processing and integrated control systems.

## Vehicles

In the Navlab project we are building systems for autonomously driving in the unstructured outdoor environment. Our vehicles combine sensing, sensor interpretation, planning, control, and testbed vehicles to create integrated navigation systems. The Navlab (Figure 1 left) and Navlab II (Figure 1 right) provide convenient testbeds for autonomous navigation, and for data collection on a movable outdoor platform. Features of the vehicles include:

- Computer-controlled steering and speed control, on a modified van (Navlab) and HMMWV (Navlab II).
- State-of-the art workstations, plus rack space and conditioned power for additional processors or electronics.
- Room for onboard researchers to observe the computer displays and vehicle performance.

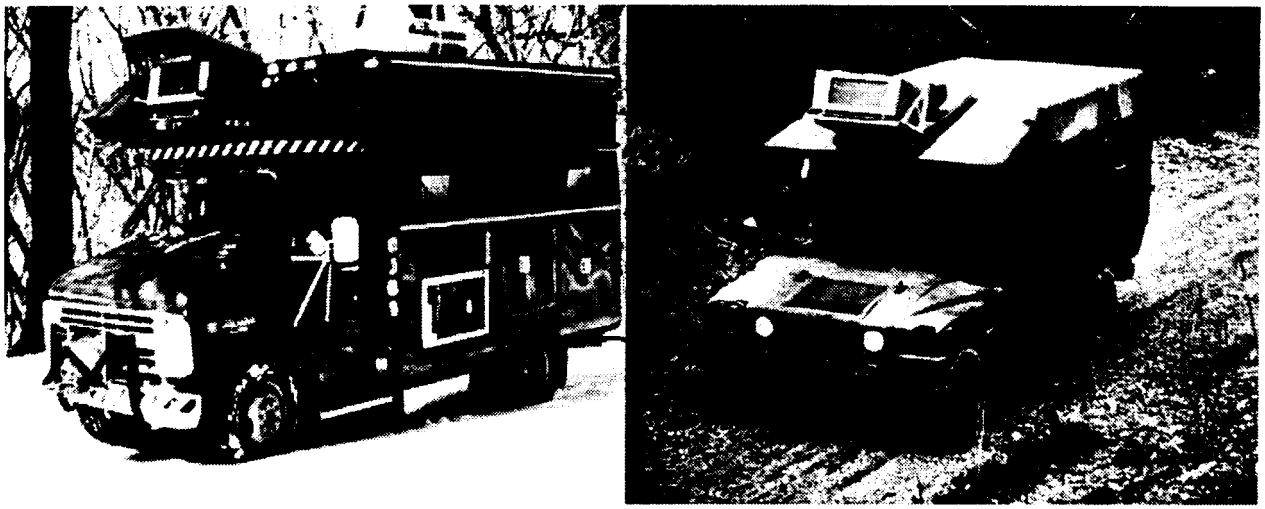

**Figure 1: Navlab I (left) and Navlab II (right)**

- Sensors, including color video, scanning laser rangefinders, FLIR, stereo video, sonars, radar, inertial navigation, and a stabilized sensor platform.

We have recently begun to reconfigure the Navlab II in order to improve overall system performance and efficiency, and to improve the long term maintainability of the vehicle. The reconfigured vehicle will have Sparc 10 computers for general-purpose computing; Sparc boards running VxWorks for real-time functions; a new lightweight color video camera and pan/tilt mount; and a second-generation FLIR. Upgrading to newer computers, and redesigning our cooling system, will allow us to shrink the air conditioning and power needs, and thus to significantly reduce vehicle weight. As the electronics shrink, we are moving closer to the goal of a small, power-efficient, unobtrusive electronics package, which could be designed into a passenger car. For more details about the CMU testbed vehicles, see (8).

## Road Following: ALVINN

Lateral position estimation and control is an important AVCS capability. The lateral positioning system we have developed is a simulated neural network called ALVINN (Autonomous Land Vehicle In a Neural Network). ALVINN's architecture consists of a single hidden layer back-propagation network. The input layer of the network is a 30x32 unit two dimensional "retina" which receives input from the Navlab's video camera. Each input unit is fully connected to a layer of five hidden units which are in turn fully connected to a layer of 30 output units (See Figure 2). The output layer is a linear representation of the direction the vehicle should travel in order to keep the vehicle on the road. The centermost output unit represents the "travel straight ahead" condition, while units to the left and right of center represent successively sharper left and right

turns.

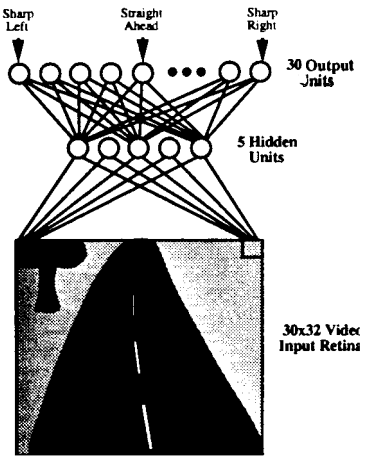


**Figure 2: ALVINN Network Architecture**

To drive the Navlab, a video image from the onboard camera is injected into the input layer. Activation is passed forward through the network and a steering command is read off the output layer. The most active output unit determines the direction in which to steer the vehicle.

Instead of programming ALVI▾▾ to steer, ALVI▾▾ learns to drive by observing the behavior of the human driver. ALVI▾▾ is shown video images from the onboard camera as a person drives, and told it should output the steering direction in which the person is currently steering. The back-propagation algorithm alters the strengths of connections between the units so that the network produces the appropriate steering response when presented with a video image of the road ahead of the vehicle. After about three minutes of watching a person drive, ALVI▾▾ is able to take over and continue driving on its own.

Because it is able to learn what image features are important for particular driving situations, ALVINN has been successfully trained to drive in a wide variety of situations including single lane dirt roads, single lane paved bike paths, two lane suburban neighborhood streets, and lined two lane highways. *Also,* since the processing performed to determine the steering direction is simple, ALVINN is able to process over 10 images per second and drive at up to **55** mph. In its most successful run to date. ALVI▾▾ has driven for over 90 miles without human intervention.

ALVINN has proven itself **as** an effect autonomous driving system. As part of our AVCS research we are currently focusing on using ALVI▾▾ as a run-off-road warning device. Instead of actively controlling the vehicle, ALVI▾▾ will monitor the steering behavior of the human driver and alert him if he appears to be drifting off the road. Only if it appears that a crash is unavoidable without immediate intervention will the system take active measures to steer the vehicle back into

its lane. Since ALVINN's neural network can quickly adapt to new situations by simply observing the steering behavior of the driver, it should provide a flexible framework upon which to a build lateral position monitoring and control system for an intelligent cruise control. For a more detail description of ALVINN and its capacities, see (2).

## Car Following: RACCOON

ALVI ▾▾ has proven its ability to estimate lateral position and steer the Navlab vehicle based on the appearance of the road. But what if the road features missing or difficult to see? Such is often the case when driving at night, particularly in the presence of other vehicles which may obscure the road's markings. RACCOON (Real-time Autonomous Car Chaser Operating Optimally at Night) is a system we have designed to cope with these difficult situations and complement the ALVINN system (3). RACCOON visually tracks the vehicle ahead, maintaining a safe headway and following the path of the lead vehicle. RACCOON has successfully followed lead vehicles on winding roads at night in light traffic at **32** km/h.

The input to this system consists of a sequence of images from a color camera, digitized at 15 Hz. RACCOON examines a region of interest in each image surrounding the expected location of the lead vehicle. Pixels in this area are thresholded for absolute brightness and redness (to eliminate spurious reflections and headlights). Since taillights vary tremendously from car to car, and also over time (as brake lights and turn signals are illuminated), a detailed model of taillight appearance is rejected in favor of a simple bounding box which surrounds all the red lights on the back of the lead vehicle.

The position and size of this box can be used to extract the relative position of the lead vehicle with respect to the camera. The horizontal position of the bounding box can be used to calculate the lateral displacement of the lead vehicle, but only if the distance to the lead vehicle is known. Since the road cannot be assumed to be flat, the vertical position of the bounding box is not a good indicator of the distance to the car ahead. Since brake lights can change the vertical size of the bounding box, this measurement is also error prone. In contrast, the horizontal extent of the bounding box is much more stable. The only major factor determining the horizontal size of the box is the distance to the lead vehicle. The effects of foreshortening due to lead vehicle yaw are small enough to be ignored for typical driving situations. The equations for converting these simple image measurements into relative position are straightforward and computationally efficient, allowing us to process images very quickly. Although nighttime scenes are ideal, this algorithm also works during the day if the lead vehicle illuminates its taillights. If desired bright decals or infra-red light sources can be substituted for taillights without modification to the algorithm.

Given the position of the lead vehicle, the straightforward approach to car following is to steer

the autonomous vehicle so that it heads towards the taillights of the lead vehicle. Speed can be controlled so that the robot vehicle remains a constant distance behind the lead car. This naive implementation may produce satisfactory results on straight roads when both vehicles are moving at the same speed; however it fails in any realistic scenario since lead vehicles change speed and make tums to follow winding roads, and steering towards taillights results in comer cutting --- possibly causing an accident as the computer controlled vehicle drifts into oncoming traffic or off the road entirely.

RACCOON solves these problems by creating an intermediate map structure which records the lead vehicle's trajectory. The path is represented by points in a global reference frame, and the computer controlled vehicle is steered from point to point. The autonomous vehicle follows this trail while keeping the lead vehicle's taillights in sight. Since every point on the trail is guaranteed to be on the road, the robot vehicle navigates around comers and obstacles rather than through them. A second important advantage is that the autonomous vehicle is not constrained to follow at a constant distance, but may instead follow at its own pace. By changing the problem from "car following" to "path tracking", the system is able to drive competently in real situations.

RACCOON is implemented as a module which allows easy integration with existing autonomous driving systems. In particular, it can complement a road follower like ALVI" in situations where ALVI" gets confused. Other applications for RACCOON include convoy following and intelligent cruise control.

## Smart Obstacle Maps: GANESHA

RACCOONS ability to track vehicles and maintain a safe separation distance is a valuable capability for an AVCS system. However many objects in the environment are not easily located and tracked using monocular video images of the environment. GANESHA (Grid based Approach for Navigation by Evidence Storage and Histogram Analysis) uses the other sensor modalities available on the Navlab vehicles, including sonars, ladar, millimeter wave radar, and trinocular stereo, to map obstacles around the vehicle (4). Each sensor measurement is used to update a local obstacle map, stored as a grid. The vehicle position is kept at a fixed point in the map. As the vehicle moves, objects in the map are moved from cell to cell. Once an object falls outside the map boundary it is discarded and the information is lost. Using just a local map has the advantage that error accumulation owing to dead reckoning is kept small, since only relative movements are considered. At present the area covered by the local map is 16.4 m x 70.2 m. Each grid cell has a resolution of 0.4 m along the x-axis. Hence there are 41 cells along x. Along the y-axis three different resolutions are used:

1. 0.4 m between -22.4 m and 10.2 m
2. 2.0 m between 10.2 m and 30.2 m

3.4.0 m between 30.2 m and 70.2 m

with respect to vehicle position. This results in 101 cells along y. Each cell has a set of parameters or annotations associated with it, which are described below:

- Object Type: when the object was last seen, and by which sensor.
- Position: The x-y position of the object, used to get finer resolution than a single cell.
- History: The number of times an object was detected in a particular cell.

The resolution of the grid is fairly coarse and hence the position parameter is kept to avoid gross error accumulation when objects are transformed in the map. Only one object is kept per grid cell. Thus measurement uncertainty is part of the grid cell representation and any object detected within an area covered by a particular cell is taken to belong to the same object. New objects detected by the sensors are added to the map after the positions of all previous objects in the map are updated. The map parameter History is used to evaluate the confidence that a particular cell is occupied by an object. A higher value of History indicates a higher confidence.

Our most ambitious use of **GANESHA** used the map for driving parallel to a row of parked cars, avoiding obstacles, and eventually finding a parking space and autonomously parallel parking (see Figure 3).

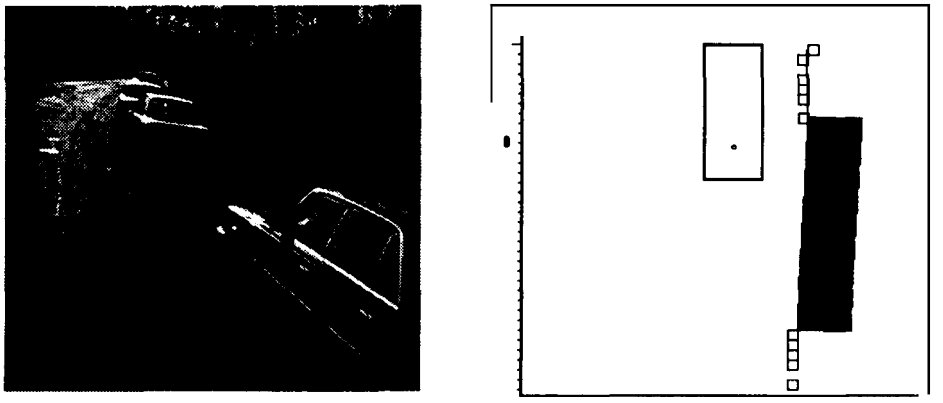

**Figure 3: Park scene (left) and local map generated by Genesha (right). The shaded region represents the gap detected between cars, within which Genesha will park the Navlab.**

## System Integration

While each of the components described so far solves part of the **AVCS** problem, to fully exploit their capabilities requires an integrating framework. The Distributed Architecture for Mobile Navigation (DAMN) is a behavior-based architecture for mobile robot driving (5). In contrast to more traditional centralized AI planners that build a world model and plan an optimal path through it, a behavior-based architecture consists of specialized task-achieving modules that operate independently and are responsible for only a very narrow portion of vehicle control, thus avoiding the need for sensor fusion. A distributed architecture has several advantages over a cen-

tralized one, including greater reactivity, flexibility, and robustness. Figure **4** shows the organization of the DAMN system in which individual behaviors such as road following (ALVI[illegible]) and obstacle avoidance (GANESHA) send steering or speed commands to the arbitration module which combines these inputs into a single steering direction and speed command. Within the framework of DAMN, behaviors provide the task-specific knowledge for controlling the vehicle. Each behavior runs completely independently and asynchronously, providing votes to its appropriate arbiter, each at its own rate and according to its own time constraints. The arbiter periodically combines all the latest commands from each behavior and issues a command to the vehicle controller.

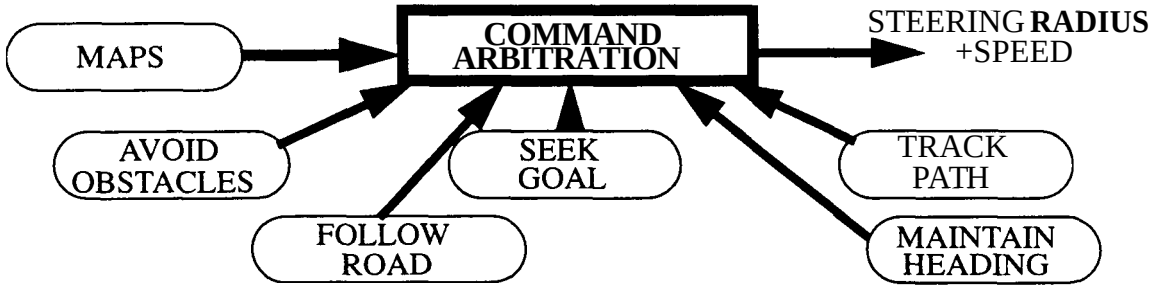


**Figure 4: DAMN System Organization**

Each behavior votes for or against each of a set of possible vehicle actions. An arbiter then performs *command fusion* to select the most appropriate action. Vehicle commands such as steering turn radius are discretized into a fixed set of possible alternatives, and each behavior then votes for or against each command option, with varying weights reflecting the relative priority of the behaviors. The arbiter then computes a weighted sum of the votes, and the command choice with the highest value is selected and issued to the vehicle controller.

## Discussion and Conclusions

Vehicle-based perception techniques hold great promise for advanced vehicle control systems. A major benefit of vehicle-centered AVCS techniques is that they require little if any modifications to the roadway infrastructure, and therefore can be adopted incrementally. But no single perception method can perform all the tasks required of a truly advanced AVCS system. We have taken a modular approach, in which specialized perception modules solve parts of the AVCS problem. We have developed modules for lateral position control, headway maintainance, and obstacle detection/avoidance. In addition, we have created an integrating framework, called DAMN, which allows us to combine these capabilities into competent driving systems. These integrated systems have been demonstrated in situations ranging from an unstructured cross-country navigation to high speed

freeway driving.

The original goal of our work was to develop unmanned vehicles capable of operating in hazardous environments for the department of defense. In this role, we have successfully transferred our technology to both university and commercial partners, including: Hughes, Martin Marietta, RedZone Robotics, ADS, Univerisity of Massachusetts, JPL and others. With encouragement from ARPA, we are now investigating dual-use applications of our results, particularly in the area of advanced vehicle control systems for NHS. To facilitate this effort, we recently secured a contract from the department of transportation to investigate the potential of our technology for preventing run-off-road collisions. This civilian sponsorship will enable us to refine and quantitatively evaluate our systems for NHS applications.

## Acknowledgments

This work was partially sponsored by ARPA, under contracts titled "Perception for Outdoor Autonomous Navigation", monitored by the Topographic Engineering Laboratories, and titled "Unmanned Ground Vehicle System", monitored by the Tank Automotive Command; and partially sponsored by the National Science Foundation, under a contract titled "Annotated Maps for Autonomous Underwater Vehicles". The authors thank their colleagues on the Navlab project: Omead Amidi, Haralabos Athanassiou, Barry Brumitt, R. Coulter, Chris Fedor, Jim Frazier, Jay Gowdy, Martial Hebert, Jennie Kay, Takeo Kanade, Al Kelly, Jim Moody, George Mueller, Bill Ross, Anthony Stentz, Rahul Sukthankar, Red Whittaker, and Todd Williamson.

## Bibliography

1 C. Thorpe, editor. ***Vision and Navigation: the Camegie Mellon Navlab*** Kluwer Academic Publishers, 1990.

2 D. Pomerleau. ***Neural Network Perception for Mobile Robot Guidance.*** Ph.D. dissertation. Carnegie-Mellon Technical Report CMU-CS-92-115, also published by Kluwer Academic Publishers, 1993.

3 R. Sukthankar. RACCOON: A Real-time Autonomous Car Chaser Operating Optimally at Night. In ***Proc. Intelligent Vehicles '93.*** 1993.

4 D. Langer and C. Thorpe. Sonar based Outdoor Vehicle Navigation and Collision Avoidance. In ***Proc. IROS '92.*** 1992.

5 J.K. Rosenblatt and D.W. Payton. A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control. in ***Proc. of the IEEE/INNS International Joint Conference on Neural Networks,*** Washington DC, June 1989.

6 C. Thorpe, O. Amidi, J. Gowdy, M. Hebert, and D. Pomerleau. Integrating Position Measurement and Image Understanding for Autonomous Vehicle Navigation. Proc. *Workshop on High Precision Navigation,* Springer-Verlag Publisher, 1991.

7 C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Toward Autonomous Driving: The CMU Navlab. Part I: Perception. and Part II: System and Architecture. *IEEE Expert,* V 6 # 4 August 1991.

8 C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Vision and Navigation for the Carnegie-Mellon Navlab. *IEEE PAMI*, V. 10 # 3, 1988.