Sentient Autonomous Vehicle using Advanced Neural net Technology

J.B.Siddharth Jonathan, Arvind Chandrasekhar and T.Srinivasan,
Department of Computer Science and Engg,
Sri Venkateswara College of Engineering, Sriperumbudur, India.
jonathansiddharth@yahoo.co.in, arvindcac@hotmail.com, tsrini@svce.ac.in .

Abstract— Over the past decade, the field of automated intelligent transport systems has been the focus of rigorous research. This paper proposes Sentient Autonomous Vehicle using Advanced Neural net Technology (SAVANT), an automated transport system with significant advantages over previous attempts in this field. The system uses a multi-layer feed-forward neural network with back propagation learning. In addition, the design of SAVANT involves the convergence of a plethora of technologies like a Global Positioning System (GPS), a Geographic Information System (GIS), and laser ranging. SAVANT can guide a mobile agent through a hostile and unfamiliar domain after being trained by a human user with domain expertise. One of the many areas in which SAVANT scores against the competition is that the system is completely domain independent and incurs substantially less processor overhead. SAVANT thus provides more functionality even though it requires considerably less input as compared to other attempts in this field. This reduction in the size of the input vector translates into more efficient and faster processing. Another of SAVANT's hallmark features is its ability to negotiate turns and implement lane-changing maneuvers with a view to overtaking obstacles. It does this by employing a novel technique, Selective Net Masking. A simulation of SAVANT's neural network was performed on a variety of network topologies, and the best network selected.

***Index Terms*—Back propagation learning, Frontal and Side Impact Collision Vectors, Selective Net Masking**

I. INTRODUCTION

Man has long dreamed of designing machines that are capable of operating themselves, one of the basic milestones to develop truly intelligent systems that can intuitively learn how to perform specific operations and execute them to perfection. One field in which several forays have been made to this end is that of automated transport systems.

The primary test of any intelligent navigation system is its ability to negotiate a standard road or highway environment successfully by identifying curves in the road and obstacles in its path and take appropriate action in each case. Changing lanes or overtaking slower vehicles is also a desirable feature.

A large number of the systems proposed to date [1]-[4] use a video camera feed as input. In [1]-[3], this input is fed to a base neural network. The video feed incurs massive processor overhead, thereby increasing the complexity and size of the network and reducing the efficiency of the system as a whole.

The design of a system that rivaled [1] but did not use a neural network was presented in [4]. However, all these systems are constrained by the domain in which they operate. Another issue plaguing these systems is that they are not lighting independent, as they operate based on a camera feed.
B. Freisleben et al presented, in [5], an interesting system that relied on strategically placed sensors instead of a video feed. We use a similar concept in our approach.

This paper proposes SAVANT, a highly advanced system with considerable advantages over previous attempts in the field of intelligent transport systems. SAVANT uses a *multi-layer feed forward neural network* with *back propagation learning*. In addition, the design of SAVANT involves the fusion of a variety of technologies like a Global Positioning System (GPS), a Geographic Information System (GIS) and laser ranging. SAVANT can guide a mobile agent through a hostile and unfamiliar domain after being trained by a human user with domain expertise. This system incurs a lot less processor overhead compared to the other approaches discussed. SAVANT thus provides more functionality even though the input vector is far smaller than in earlier systems. This reduction in the size of the input vector translates into more efficient and faster processing. Another salient feature of SAVANT is its ability to change lanes and pass a slower vehicle or a stationary obstacle ahead of it.

The rest of the paper is organized as follows. Section II describes the previous work in this field. The proposed system and its design is presented in Section III. All algorithms relevant to the operation of SAVANT are dealt with in Section IV. In Section V, the performance analysis of SAVANT with a number of network topologies is addressed and the best topology of those tested is subjected to more rigorous analysis. Finally, Section VI summarizes the paper and gives insight into possible further development and future directions of ongoing work.

II. RELATED WORKS

Each of these following attempts has had its share of success accompanied by specific drawbacks. Here, we consider four existing approaches:

A. *ALVINN* and *MANIAC*

ALVINN (Autonomous Land Vehicle In a Neural Network) and MANIAC (Multiple ALVINN Networks In Autonomous Control), developed at Carnegie Mellon University to control the NavLab vehicles there, are outfitted with computer-controlled steering, acceleration and braking. Sensors include color stereo video, scanning laser range finders, radar and inertial navigation. Both systems require training by a human driver for about five minutes (followed by application of the back propagation algorithm for ten minutes) before they are ready to drive. The signal from the vehicle's video camera is preprocessed to yield an array of pixel values that are connected to a 30x32 grid of input units in the neural network.

ALVINN computes a function that maps from a single video image of the road to a steering direction. The output is a layer of 50 units, each corresponding to a steering direction. ALVINN's performance is impressive in the cases where it has been trained. It is unable to drive on a road type for which it has not been trained, and is also not very robust with respect to changes in lighting conditions or the presence of other vehicles.

MANIAC (Multiple ALVINN Networks In Autonomous Control) is composed of several pre-trained ALVINN networks, each trained for a single road type that is expected to be encountered during driving. The superstructure combines data from each of the ALVINN networks and does not simply select the best one. The output from the ALVINN networks can be taken from either their output or hidden units.

B. *ELVIS*

ELVIS is a road-following system based on ALVINN using the same input and output, but without using a neural network.

Like ALVINN, ELVIS observes the road through a video camera and observes human steering response through encoders mounted on the steering column. ELVIS learns the eigenvectors of the image and steering training set via principal component analysis. These eigenvectors roughly correspond to the primary features of the image set and their correlations to steering. Road-following is then performed by projecting new images onto the previously calculated eigenspace.

While ELVIS principal component analysis minimizes the total error in the image reconstruction and steering vector, ALVINN directly minimizes the steering output alone.Although the speed of training of the two systems is roughly equal, ALVINN is faster at run-time because it requires far fewer calculations. The primary advantage of ELVIS is its simplicity and lack of pre-defined structure.

C. *Sensor networks*

One of the primary disadvantages of a system using a video camera feed is the enormously high processing power required. The performance can be improved by using a sensor network that depends on a number of non-video sensors such as range finders on the vehicle to sense the environment around the vehicle and take appropriate action. SAVANT is essentially a sensor network, such as that described in [5].

III. THE DESIGN OF THE PROPOSED SYSTEM

SAVANT brings together a plethora of technologies, namely a Global Positioning System (GPS), a Geographic Information System (GIS), a laser transceiver bank and a multilayer feed forward neural network.

A. *GPS*

The GPS in SAVANT obtains the coordinates of the location of the vehicle. This information is relayed to the GIS in order to obtain information necessary for the operation of SAVANT.

B. *GIS*

The characteristic of a GIS that is most relevant to our discussion is the fact that one can 'point' at a location, object, or area on the screen and retrieve recorded information about it from off-screen files. With specific reference to SAVANT, the data from the GIS that are relevant are (i) road locations/maps, (ii) road angles, (iii) road widths, and (iv)

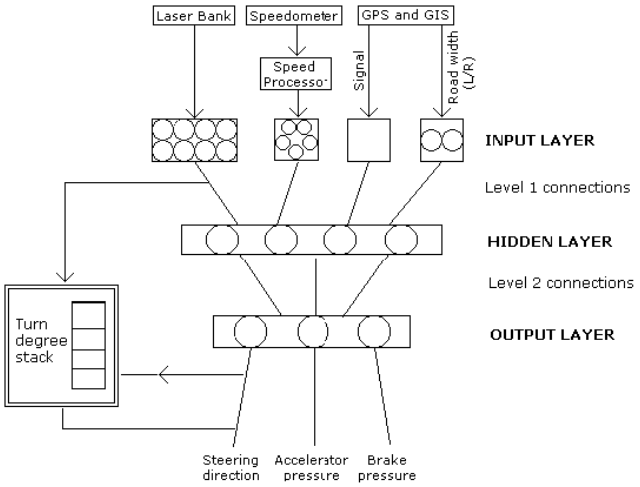

Fig. 1. General organization of SAVANT.

signals.

. Highly dynamic information such as the color of the signal at the intersection ahead can be obtained by maintaining a connection to the local traffic police records.

All other data is readily available from standard GIS. Thus, based on the current location of the vehicle, all necessary details can be obtained.

In brief, we see that the GPS is instrumental in pinpointing the precise location of the vehicle, while the GIS provides real time information based on this location.

C. *Laser transceiver banks*

The transceiver bank consists of arrays of laser beam emitters and detectors. In a standard configuration, light is sent from the source by an emitter. When the beam reaches an object, it is reflected back to the source. A detector at the

source receives the reflected beam and indicates that an object is present. Based on the time between emission and detection T, and the speed of the laser beam S, the distance D of the object from the transceiver bank is determined.

$$D = S \times (T / 2) \quad (1)$$

The system possesses two banks of transceivers. The frontal transceiver bank provides a *Frontal Impact Collision Vector (FICV)* which holds the information necessary to detect and avoid an obstacle in front of SAVANT. The transceiver banks at the sides of the vehicle provide a *Side Impact Collision Vector (SICV)* which holds the information necessary to detect the presence of obstacles that are parallel to SAVANT in the adjacent lanes. The SICV is used during the process of overtaking by changing lanes. The neural network uses the collision vector information to take appropriate action (such as applying the brakes).

D. Multilayer Feed Forward Neural Network

The network under study here has two layers and one hidden layer. Its input vector is comprised of

- FICV
- Current speed
- Traffic signal information
- Road width (left and right)

The output vector is comprised of

- Steering angle
- Accelerator pressure
- Brake pressure

Based on input, appropriate output values are generated by the neural network on the basis of the weights, thereby guiding the vehicle along the right direction at the appropriate speed.

IV. ALGORITHMS

1) SAVANT algorithm

```
function SAVANT()
define:
   network, a multilayer network
   input,   input vector
   output, output vector

begin
DataAcquisition()
network ← NeuralNetworkLearning()
repeat
   input ← values from the input sensors
   output ← RunNetwork(network, input)
   if (RoadAngle = 0) and (output.SteeringAngle ≠ 0)
         output ← Overtaking(output)
   else
         output ← Turn(RoadAngle, output)
   end if
   Implement the output vector
indefinitely
end
```

This is the primary driver function of SAVANT. First, the training set sample data is acquired (DataAcquisition). This is followed by the neural network learning phase where the link weights are updated based on the input-output sample sets (NeuralNetworkLearning). After this comes the real-time application of SAVANT. The system continuously reads the input from the various sensors, derives the appropriate outputs by running the neural network and implements these outputs.

1) DATA ACQUISITION algorithm

```
function DataAcquisition()
begin
repeat
   Read input from
         • FICV
         • Speedometer reading
         • Traffic signal information
         • GIS (Road widths)
   Read the human driver's response regarding
         • Steering angle
         • Accelerator pressure
         • Brake pressure
   Formulate the training set entry which comprises of the
   input vector and the Driver response.
   Add the training set entry to the training set table.
until end of training run
return
```

This is the initial period where a human driver trains the vehicle. In effect, the system just observes the human's actions under different input conditions.

2) *NEURAL NET REAL TIME LEARNING* algorithm

```
function NeuralNetworkLearning( ) returns a network
define:
   network, a multilayer feed forward Neural Network with
   randomly assigned weights
   sample a structure with content
      Input,      input vector
      Output,     output vector
      α, the learning rate

begin
repeat
      for each sample in training set
      begin
         sample.Input ← input vector
         sample.Output ← output vector
         network ← BackPropUpdate(network, sample, α)
      end
```

until *network* converges
return *network*

The (input, output) pairs obtained from the human driver during the data acquisition phase are now applied to the network to set it up. The network link weights are updated by means of Back Propagation.

3) BACK PROPAGATION

function BackPropUpdate (*network*, *example*, α) returns a network
inputs:
network, a multilayer network
example , a structure containing vectors *input* and *output*
α, the learning rate
begin
$O \leftarrow$ RunNetwork(*network*, *example.input*)
$Err \leftarrow T - O$
$W_{ji} \leftarrow W_{ji} + \alpha \times a_j \times Err_i \times g'(in_i)$
for each subsequent layer in *network* do
begin
$\Delta_j \leftarrow g'(in_j) \sum_i W_{ji} \Delta_i$
$W_{kj} \leftarrow W_{kj} + \alpha \times I_k \times \Delta_j$
end
return *network*

Back Propagation is an effective method of dividing the contribution of each weight to the error. We try to minimize the error between each target output and the output actually computed by the network. Here, Err represents the difference between the output O and target T which is the desired output. I_i indicates the input to unit i, W_{ji} the weight on the link from unit j to unit i, α the constant learning rate and a_j the activation value of the unit j. $g'(in_i)$ refers to the derivative of the activation function of the weighted sum of inputs to the unit i. Δ_i is the product of Err_i and $g'(in_i)$.

4) TURNING algorithm

function Turn(θ, *output*) returns *output*
inputs:
θ, The turning angle in degrees. A value in the range of 0 to 180 degrees implies a right turn and a value from 180 to 360 degrees implies a left turn.
output, output vector

begin
flag $\leftarrow 1$
if θ is identified to be a right turn
flag $\leftarrow 0$
if flag = 1 magnitude $\leftarrow 360 - \theta$
else magnitude $\leftarrow \theta$
limit $\leftarrow$ ceiling(magnitude/10)
iterator $\leftarrow 0$
while iterator < limit
begin
if (flag = 0)
output.SteeringAngle = $10°$
else
output.SteeringAngle = $-10°$
iterator $\leftarrow$ iterator + 1
Implement the output vector
end
limit $\leftarrow$ magnitudeMOD10
iterator $\leftarrow 0$
while iterator < (limit – 1)
begin
if (flag = 0)
output.SteeringAngle = $1°$
else
output.SteeringAngle = $-1°$
iterator $\leftarrow$ iterator+1
implement the output vector
end
if (flag=0) *output.SteeringAngle* = $1°$
else *output.SteeringAngle* = $-1°$
return *output*

The GIS input for road angle is monitored external to the neural net at all times. If the road angle is not zero, the neural net output for steering angle is subjected to *Selective Net Masking,* an implementation technique by which specific nodes of the neural network are overridden while the rest of the neural network operates as normal. Here, the steering angle output is completely determined by the turning system and overrides the existing value calculated by the neural network. The system negotiates turns first in steps of $10°$ and then in steps of $1°$ to achieve the desired turn angle accurately.

5) OVERTAKING algorithm

function Overtaking(*output*)
input:
output, output vector
define:
network, a multilayer network
input, input vector
flag, a binary variable indicating an ongoing incomplete overtaking maneuver
SICV, inputs from the side impact laser bank

begin
if(SICV $\neq$ 0)

```
begin
      output ← RunNetwork(network, input)
return output
end if
push output.Steering direction onto stack
flag ← 1
repeat
   implement the output vector
   output ← RunNetwork (network, input)
   if (input.FICV = 0)
   begin
      if (Stack is not empty)
         output.Steering direction ← (-1×top of stack)
         Pop the topmost element off of the Stack
      else
         flag ← 0
   end if
   if (output.Steering direction ≠ 0) and (input.FICV ≠ 0)
      push output.Steering direction onto stack
until flag=0
return output
```

When the system finds that the neural net output for steering angle and the road angle are both non-zero, it realizes that an opportunity to overtake exists. It first checks its SICV to see if there are any obstructions to its side. If not, the system turns in the required direction by 10 degrees. It sets a flag to indicate that it has not straightened yet and pushes the steering angle on to the stack. It does this so that once the car has turned by the appropriate amount to overtake the other car by changing lanes, it can straighten itself by popping the contents of the stack one by one and implementing the complement of the steering direction. The flag is reset once the stack is empty, and the system then resumes its usual modus operandi.

V. Performance Analysis

We compared three scenarios with varying neural network topologies as shown in Table I. These topologies differ only in the number of nodes in the hidden layer.

TABLE I
NUMBER OF NODES PER LAYER IN EACH SCENARIO

Scenario	Input layer	Hidden layer	Output layer
1	16	3	3
2	16	4	3
3	16	5	3

We first compare the three neural net topologies on the basis of the number of epochs they take to stabilize and the final error at which they stabilize. The back propagated error as a function of the number of epochs of training in each of the scenarios is presented in Fig. 2 and 3.

On analyzing the graphs, we found that the network represented by Scenario 2 stabilized the fastest. We then plotted the percentage of standard test set entries this network responded correctly to as a function of the size of the training set used to train the network. The results are shown in Fig. 4.

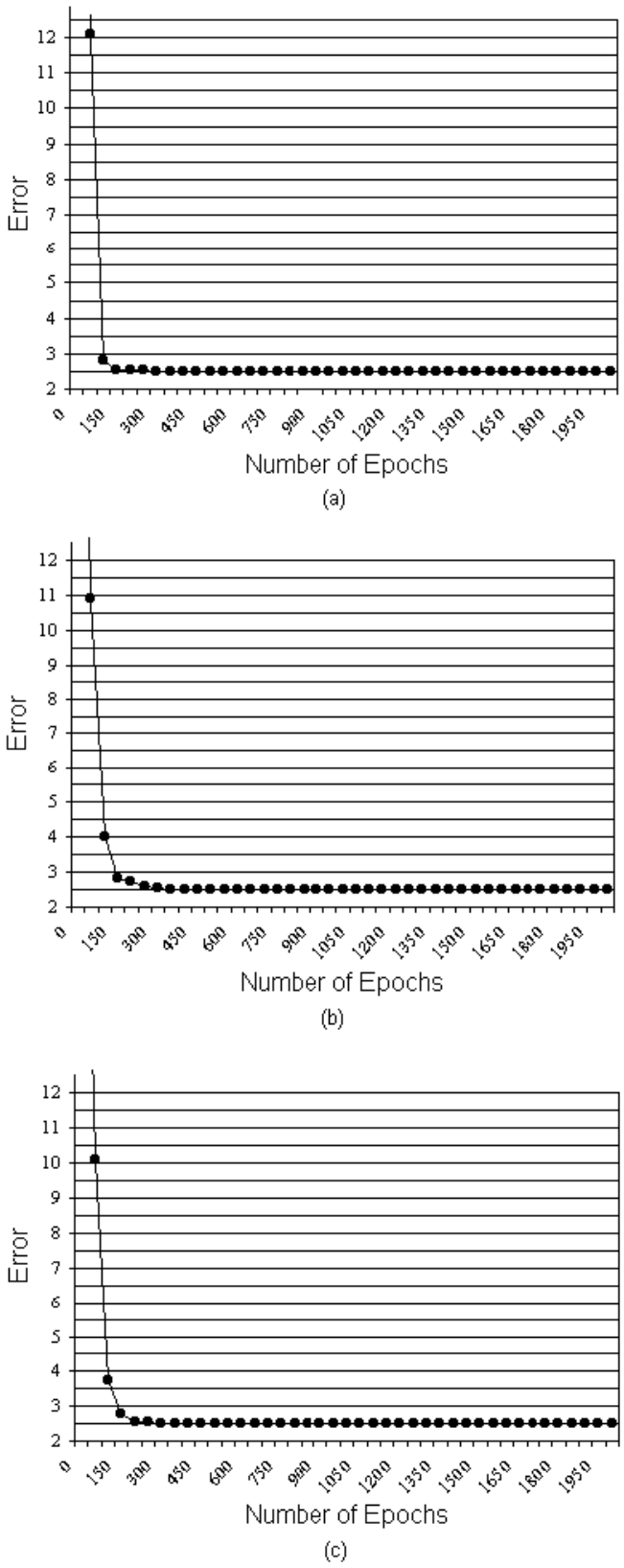

Fig. 2. Back propagated error as a function of the number of epochs of training in the initial stages of training in (a) Scenario 1, (b) Scenario 2, and (c) Scenario 3.

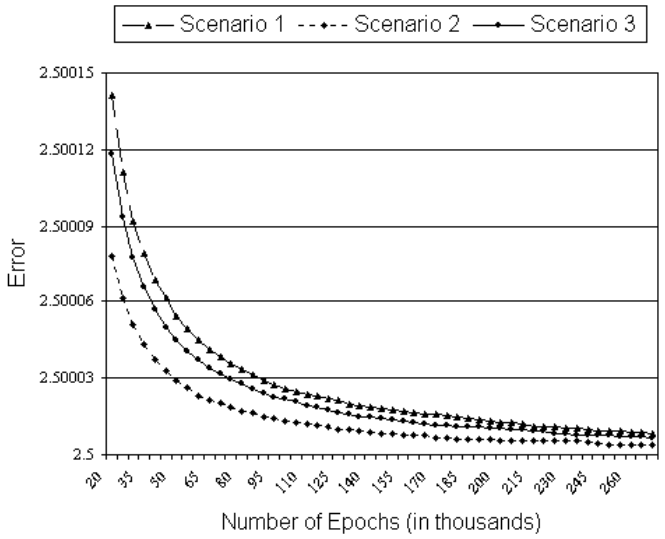

Fig. 3. The back propagated error as a function of the number of epochs of training in each of the scenarios.

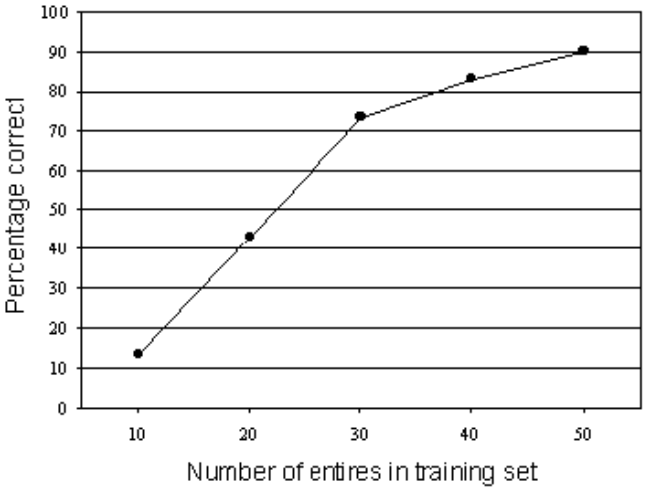

Fig. 6. The percentage of the test set values for which the network responds correctly as a function of the number of entries in the training set.

VI. Concluding remarks

We present a possible alternative approach to the above. Here, the traffic signal status is not part of the input set. Instead, a Real Time Operating System is used to override the neural network when a red signal is detected. When the GIS indicates a red signal, control is temporarily wrested from the neural network. The RTOS executes an appropriate interrupt handler the situation (say, by slowing down). When the interrupt service routine completes execution, control is transferred back to the neural network.

Selective Net Masking can be used to override specific nodes of the neural network like the brake pressure node. This enables the neural network to retain over all control, while still allowing the RTOS to handle red traffic signals by taking appropriate action.

Another possible direction of further work is to develop aids to support such intelligent systems, along the lines of the methods discussed in [11].

SAVANT marks a milestone in the evolution of intelligent transport systems. Its numerous advantages like domain independence, ease of training, low processor overhead, reduction in size of the input vector and the ability to handle complex operations like overtaking and turning make it truly revolutionary.

While there are, as always, certain avenues for improvement or modification, SAVANT lays the groundwork on which several simple yet efficient intelligent transportation systems can be based in the future.

The technology is very exciting and promising. It is only a matter of time before the automobile giants wake up to the neural net revolution and employ systems similar to SAVANT to manufacture cars that are truly autonomous in every sense of the word.

References

[1] C. E. Thorpe, M. Hebert, T. Kanade and S. Shafer, "Toward Autonomous Driving: The CMU Navlab," IEEE Expert, V 6 #4, August 1991.

[2] D. A. Pomerleau, Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers, 1993, ISBN 0-7923-9373-2.

[3] T. M. Jochem, D. A. Pomerleau and C. E. Thorpe, "MANIAC: A Next Generation Neurally Based Autonomous Road Follower," Proceedings of the International Conference on Intelligent Autonomous Systems, 1993.

[4] J. Hancock and C. Thorpe, "ELVIS: Eigenvectors for Land Vehicle Image System," Proc. of IEEE/RSJ International Conf. on Intelligent Robots and Systems, Pittsburgh, PA, 1995, pp. 35-40.

[5] B. Freisleben and T. Kunkelmann, "Combining Fuzzy Logic and Neural Networks to Control an Autonomous Vehicle," Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, IEEE Press, 1993, pp. 321-326.

[6] R. Aufrere, J. Gowdy, C. Mertz, C. Thorpe, C. Wang, and T. Yata, "Perception for collision avoidance and autonomous driving," Mechatronics, Vol. 13, No. 10, December, 2003, pp. 1149-1161.

[7] M. Bertozzi and A. Broggi, "GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection," IEEE Transactions on Image Processing, 1998.

[8] J. Hancock, M. Hebert, and C. Thorpe, "Laser Intensity-Based Obstacle Detection", Proceedings 1998 IEEE/RSJ International Conference On Intelligent Robotic Systems (IROS '98), Vol. 3, October, 1998, pp. 1541 - 1546.

[9] C. Wang, C. Thorpe and A. Suppe, "Ladar-Based Detection and Tracking of Moving Objects from a Ground Vehicle at High Speeds," IEEE Intelligent Vehicles Symposium (IV2003), June, 2003.

[10] C. Thorpe, R. Aufrere, J.D. Carlson, D. Duggins, T.W. Fong, J. Gowdy, J. Kozar, R. MacLachlan, C. McCabe, C. Mertz, A. Suppe, C. Wang and T. Yata, "Safe Robot Driving," Proceedings of the International Conference on Machine Automation (ICMA 2002), September, 2002.

[11] P. Griffiths, D. Langer, J. A. Misener, M. Seigel and C. Thorpe, "Sensor Friendly Roadway and Vehicle Systems," IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, May 2001 .