FEATURE

Unmanned Ground Vehicle Demo II: Demonstration A

Wendell H. Chun, Martin Marietta Astronautics
Todd M. Jochem, Carnegie Mellon University

The military has an anticipated need for a remotely controlled ground system to perform reconnaissance; surveillance; target acquisition; patrolling; and nuclear, biological, and chemical (NBC) detection. In particular, the U.S. Army Infantry School would like the system to operate in the most dangerous areas of the modem battlefield, open terrain that is highly trafficable. This has led to the premise that the system should be fast. Secondly, discovering the enemy's location is often dangerous with the cost assessed in human lives. From these requirements emerged the Unmanned Ground Vehicle (UGV) programs. Emphasis is on effective robotic technology that has multiservice applications and is unique to unmanned vehicles on the ground [UGVMP, 1993]. The maturity of these robotic technologies will be confirmed in two demonstrations: UGV Demo I held in 1992, and UGV Demo II held in 1996. UGV Demo I focused primarily on teleoperation, while the UGV Demo II is designed to complement the first demonstration by focusing on supervised autonomy.

Figure 1. UGV Demo II scenario

The primary goal of the UGV Demo II program is to demonstrate the utility of advanced UGV systems to conduct tasks that enhance the Department of Defense force structure. This demonstration will combine both an offensive and defensive operation in a militarily relevant situation. For offensive operation, four cooperating UGVs will initiate a movement-to-contact scenario. The vehicles will conduct a screening operation for a manned force using bounding overwatch over semi-arid terrain. Once in overwatch positions, they will use a reconnaissance, surveillance, and target acquisition (RSTA) mission module to observe threats; and to locate, detect, assess, and designate threats for indirect fire. For defensive operation, the vehicles will conduct a retrograde scenario. Four cooperating UGVs will screen a manned force by sequentially occupying preplanned defensive positions to maximize damage to advancing enemy forces. Once the commander determines that the enemy forces have been weakened, the vehicles will move to preplanned locations in the main battle area to help designate remnants of advancing enemy forces.

Key technologies that this program will focus on include: Systems architecture; passive stereo vision for navigation; combined navigation and RSTA functions; autonomous navigation; perception; non-umbilical, non-line-of-sight communications; laser radar (ladar).

PROGRAM BACKGROUND

Principle components of the UGV Demo II scenario are depicted in Figure 1. They include the four vehicles, the multivehicle control unit (MCU), a RSTA target, and the Unmanned Air Vehicle. Each of the four vehicles will have three mobility modes: man-in-the-loop teleoperation, supervised (intermittent, way-point, or computer-aided remote driving), teleoperation, and semi-autonomous operation. The vehicles will be equipped with both onroad and offroad perception and navigation subsystems. They will be able to: create and update digital maps; detect and update obstacles and landmarks; communicate among vehicles, the portable control unit, and the base station; enable coordinated interplay among vehicles with a mobility executive system; and interpret plan scripts to sequence execution among the major mobility modes.

The major navigation emphasis of the program is to robustly drive on arbitrary roads, plan and execute a safe path through the terrain, perceive and avoid obstacles (both on- and off-road), and determine position using landmark recognition. There will be three interim demonstrations (Demonstrations A, B, and C) to demonstrate incremental progress leading up to Demo II. Demonstration A showed basic mobility and teleoperation capabilities on a single vehicle. Demonstration A will be discussed in greater detail in the next section.

In Demonstration B, a single vehicle will be outfitted with a map management subsystem and onroad/offroad perception and navigation subsystems. Map management will include map making, route and path planning, and navigation using annotated maps. The perception system will have capabilities for detecting objects or landmarks specified in the annotated maps. The onroad subsystem will enable high-speed road following of prespecified paths, and medium-speed navigation of simple roadways based on artificial neural network perception of color video imagery in conjunction with obstacle avoidance and object detection and location using active range sensor or stereo camera pair. The offroad subsystem will enable high speed following of prespecified paths with obstacle avoidance and object detection and location from active or passive (stereo) range data, and medium-speed navigation of semi-arid terrain based on terrain analysis from active or passive range data with obstacle avoidance, object detection, and object location. This demonstration is scheduled for July, **1994.**

Demonstration C will be a cooperating vehicle mission. Two vehicles will have enhanced map management, communication, a mobility executive, and mission planning subsystems. Advanced map management provides abstract specification and measurement of the vehicles' global location. Each vehicle will have multivehicle cooperative planning and an advanced, onboard mobility executive that controls the execution of plan scripts. Plan scripts dictate mobility modes, objectives, and actions to be taken at each intermediate goal point. The enhanced perception subsystem will be capable of determining when subgoals have been achieved, measured either abstractly by object detection and location algorithms or quantitatively by location.

Martin Marietta is the system's integrator for the UGV Demo II program. As the integrator, it is Martin Marietta's responsibility to collect and integrate relevant technologies from its cocontractors such as Carnegie Mellon University, Hughes Research Laboratories, Advanced Decision Systems, SRI, Teleos, Jet Propulsion Laboratory, University of Massachusetts, and the University of Michigan. Table 1 has an expanded list of the UGV Demo II cocontractors.

MICOM	Columbia Univ.	Geor Inst. of Tech.
Harry Diamond Labs	SRI International	LGA
LabCom	Jet Propulsion Lab	Carnegie Mellon Univ
CECOM	Teleos Res.	Unv of Massachusetts
NIST	Advanced Dec. Sys.	Univ of Maryland
Topographic Eng. Ctr.	Hughes Res. Lab	Univ of Michigan
NVESD	Univ. of Pennsylvania	Odetics
Cybernet	Univ. of Rochester	Alliant Techn Systems
Hughes STX	Univ. of South. Calif.	Hummel
AM General	MIT Lincoln Lab	STA
Amber Tech.	Hazeltine	Texas Instruments
ERIM	Applied Res. Assoc.	Intel
Army Res. Lab	Lear Astronics	Nichols Res Corp
Colorado State Univ.	Rockwell International	David Sarnoff Res Ctr
Cornell Univ.	AAI	Loral Vought Systems
Cambridge Parallel Proc.	Univ. of Texas-Arlington	Honeywell

Table 1. UGV Demo II Cocontractors

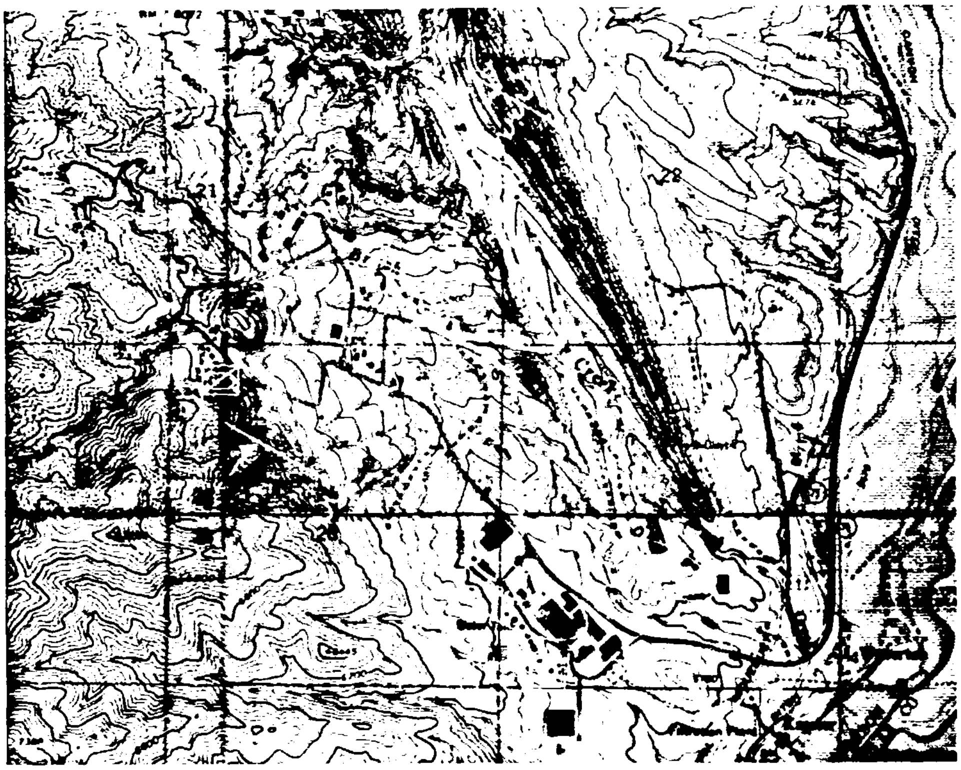

Figure 2. Demo A Military Plan

DEMONSTRATION A

Demonstration A (Demo A) was held at Martin Marietta's Waterton, CO, facility on July 7-8, **1993.** The demonstration took place on parts of the old Autonomous Land Vehicle (ALV) test track [Turk, 1988]. The goal of Demo A was to show basic systems operation and precision navigation on a single vehicle. A map-based planning method was incorporated to ensure its relevancy to the user community. Demo A was manually planned, but Demo B and beyond will feature automatic planning. Starting with a map of the terrain, pertinent features such as paved roads, dirt roads, natural vegetation, trees, landmarks, and possible command sites are indicated on the map. Overlaid on this terrain map was the military plan for the mission: area surveillance and overwatch. The military plan (Figure 2) had an assembly area (starting point) and a final objective (end point). The planned military movement passed through several phase lines that were used as control measures for monitoring the progress of the mission. Significant military implications of this plan are for movement to objective and bounding overwatch. The military plan was used as the starting point to build the robotic plan. The robotic plan was overlaid on both the military plan and the map of the terrain, and connected the vehicle with the soldier at the user interface. The robotic plan included mission planning that controlled the mode changes of the vehicle; i.e., off-road travel to paved-road travel or way-point navigation to dirt-road following. For Demo A, robotic planning consisted of path following using integrated Global Positioning System/Inertial Navigation System (GPS/INS) and neural network road following. This same three-step method will be used for more complex missions.

A layout of the Demo A course is illustrated in Figure 3. It began with the vehicle staged in a dirt area adjacent to a paved mad. Employing the **STRIPE** (Supervised TeleRobotics using Incremental Polygonal Earth geometry) system, the single

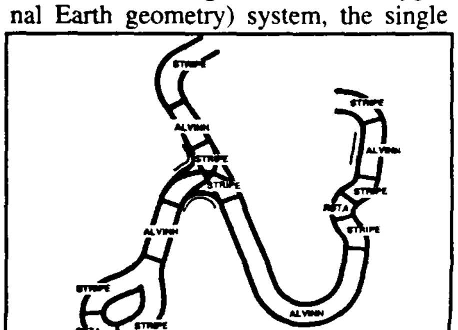

Figure 3. Demo A course

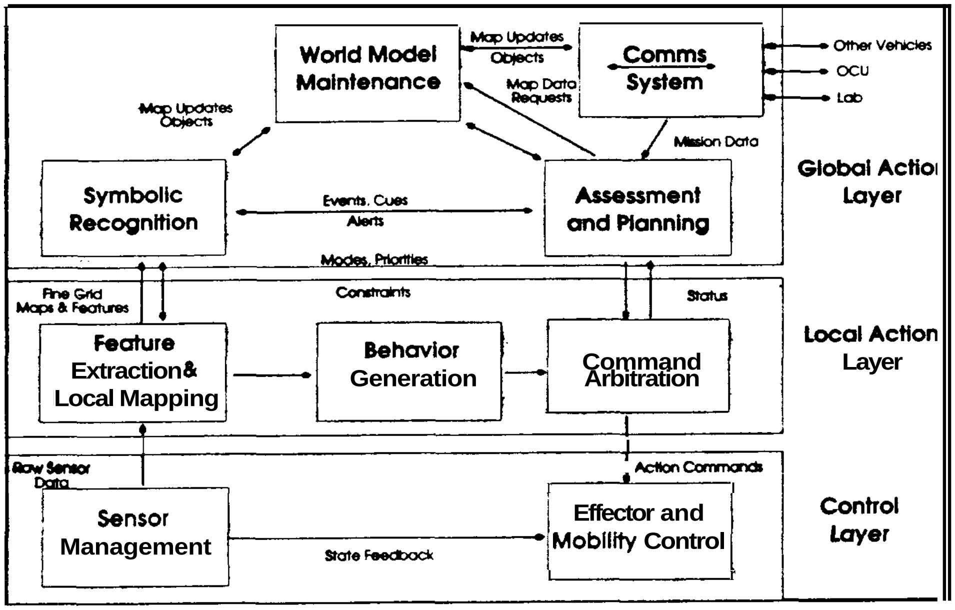

Figure 4. UGV Demo II architecture

vehicle was teleoperated onto the paved road. In the **STRIPE** system, the operator plots the vehicle's chosen trajectory based on a single **2-D** image at the operator console, the **2-D** points are translated to **3-D** points on the vehicle, and the vehicle autonomously follows this path of points. When the vehicle moved a certain distance **or** reached the end of the path, it transmits another image back to the operator workstation and the process repeats itself. Once on the paved road, the vehicle switched to the Autonomous Land Vehicle in a Neural Network (ALVINN) road-following algorithm. **(STRIPE** and ALVINN are discussed in greater detail later in this paper.) Halfway down the first straight section, the vehicle switched its mode back to the **STRIPE** system and the vehicle pulled off the road and parked. At that time a teleoperated RSTA demonstration was performed and images of the adjacent countryside were transmitted back to the operator workstation. At this point, there were no targets in view of the vehicle. **STRIPE** was used to get back on the road and the vehicle navigated down the road using ALVINN. The road made a tight hairpin turn to the right. It straightened out again and the vehicle traveled another kilometer until the paved road intersected a dirt road. The vehicle was teleoperated onto the dirt road using **STRIPE** where ALVINN was used once more. The dirt road made a loop around a small hill. Once close to the hill, the vehicle was switched back to **STRIPE** to navigate the perimeter of the hill to get to its second surveillance position. Halfway around the loop, the vehicle stopped at the overwatch location to conduct another teleoperated RSTA mission. This time a target vehicle was located and a call for fire was initiated. The target was destroyed and the vehicle departed using **STRIPE.** Once on the dirt straightaway, it transitioned back to the ALVINN system. When the dirt road merged back with the paved road, the vehicle changed mode from ALVINN to **STRIPE** and back to ALVINN again. After a short distance, it changed to the **STRIPE** mode and completed its mission in a nearby field where it rendezvoused with a second logistics vehicle to end the demonstration.

Demo A was executed precisely to script and the event was a complete success. The remainder of this paper will focus on the infrastructure of the vehicle and the key technologies used in Demo A.

VEHICLE INFRASTRUCTURE

Architecture – A diagram of the Demo II architecture is shown in Figure **4** and is the product of the UGV Demo II tiger team's effort to capture the essence of the program in a high-level view of the system architecture. **This** architecture is a balance between a class hierarchical structure, which **has** a distinct top-down flavor, and the behaviorist's approach, which has a distinct bottom-up flavor and beginning with feedback loops. This combination of a object-oriented and process-oriented design strategy is best represented by **an** example called the intelligent robot [Wolfe, 1992], **or** the Demo II vehicle in our case. The organization of this architecture is based on the following principles:

- "Behaviors" in our context vary in type from the current popular approach [Brooks, 1986], especially when the system is operating in a purely projective planning and execution style.
- Behaviors reside and operate at the lowest level on a much faster time scale than more "cognitive" processing.
- Behaviors are primarily reactive, while assessment and planning are more projective.
- There are multiple processes operating at various rates within each level.
- "World model" information resides in both upper layers, but there is a distinction of implicit versus explicit representation.

The Demo II architecture uses a three-tiered superstructure: the control layer; the local action layer; and the global action layer. This separation into layers is not intended to be a subsystem level partitioning, but a way of classifying subsystems and providing a method to manage their interfaces. Elements of the layers represent subsystems and elements of the subsystems are modules. Partitioning of the subsystems was done so that they share the same order of complexity. Subsystems sharing significant functionality and/or data were aggregated. Where possible, partitioning was done to minimize communication bandwidth, enhance debugging, and isolate organizational entities. The architecture also provides some insight into the differences between lines of data communication and lines of command and control communication.

The lowest level is the control layer and contains all of the system's low level functions, such **as** closing all the servo loops. Processes in this layer tend to run synchronously in real time. The low-level safety and consistency checks are included in this layer. The middle layer is called the local action layer and contains mainly behavioral and reflexive elements. Most of the data in this layer is ephemeral and this level's "world model" is implicit in the intermediate representations of this layer. Functions in the local action layer operate at a lower rate than the control layer, but still in synchronous real time. This layer responds to mode commands or constraints from the higher layer. Behaviors in this level refer to the generation of some command **or** sequence of commands, which may ultimately **be** committed to action *[Argo,* 1993], depending on the priorities and the current operating mode of the system. We made provisions in this layer for a purely projective planning approach where highly integrated motion and activity plans are sent down from the upper layer to the situation assessor that executes the activity if it satisfies current plans. Trajectory planning and other lower level planning functions can be executed in this layer because of information sharing or other timing constraints.

The highest level is the global action layer and contains the "cognitive" or "reasoning" elements of the system. This layer contains sensor fusion tasks (object recognition, target recognition, landmark recognition, pose refinement, etc.) as well as projective planning tasks (route planning, path planning, activity planning, etc.). Data at this level is persistent; i.e., the explicitly represented world model as opposed to the implicit world model of the local action layer. It also contains any reasoning about the model, such as multivehicle (coordinated) control, situation assessment, mission monitoring and plan monitoring. The data are also of coarser grain and more symbolic in nature. The functions in this layer operate slower than the lower two layers and tend to be event-driven, or periodic processes.

HARDWARE

The Demo A vehicle is based on the AM General M998-series high-mobility multipurpose wheeled vehicle (HMMWV), a light, highly mobile, diesel-powered, 4-wheel-drive tactical vehicle that uses a 1-1/4-ton chassis. It has a curb weight of 2359 kg (5,200 lb) and a payload capacity of 1134 kg (2,500 lb.). The suspension of our vehicle has been enhanced with upgraded wheels and variable-weight rear springs to increase payload capacity to 1468 kg (3,236 lb.). Steering, throttle, service brake, parking brake, and transmission functions of the vehicle are remotely controlled by push-pull cables attached to ball screws, except for steering, which is actuated through a dc motor/grooved-power belt and pulleys arrangement. Power for the vehicle comes from two 7.2-kW generators in the rear passenger footwells. The generator sits flush with the bed of the HMMWV to preserve the payload area for the electronics. Five 52 x 19-in. electrical racks are situated in two rows in the payload area behind the roll bar. One row has three racks facing toward the back and the other row faces forward, leaving an operationally clear area in the center. The ball screws are on a plate at the base of the racks adjacent to the roll bar. One rack is dedicated to the controls electronics and the vehicle control system. A second rack houses the packet radio communications system, the GPS, and the image recording equipment. An entire third rack is reserved to hold special-purpose, image-processing electronics such as a DAP or image understanding architecture (IUA). The fourth rack has a VME card cage and the computers. The final rack contains video equipment and an additional monitor.

Sensory input is from cameras located over the cab of the vehicle for navigation, or on the top of the shell for RSTA. Both the ALVINN camera and the RSTA camera can be gimbaled around on pan/tilt mechanisms. The imaging bus and frame grabbers are Datacube Maxbus hardware that interfaces to a VME bus [Gothard, 19921. To record and play back camera images, a commercial S-VHS format analog recorder is available to analyze vehicle navigational performance or pretrain the ALVINN networks. On the VME bus is a SUN Sparc workstation used as a server arid two Sparc 2s to operate the navigation software such as ALVINN- The Sparcs are connected through Ethernet to the packet radio and the vehicle control system. There is a second VME bus in the vehicle control system that houses a couple of Sparc control boards, receives inputs from the Global Navigation System (GNS), and the reflexive memory. Vehicle control commands are sent from the control boards to the digital servo electronics and to the motor amplifiers that actuate various motors. Analog position signals are returned through the analog acquisition board and back to the controller boards to close the control loop.

The entire payload area is enclosed in a custom aluminum shell to seal the electronics from the environment. A commercial 50,000 Btu/hr. air conditioning system with its own diesel engine keeps the electronics at the proper operating temperature. The GNS for Demo A [Gothard, 19931 consists of a low-cost fluxgate magnetometer, two inclinometers, four odometers, a turn rate sensor, and GPS code. GPS is required to determine the position of the vehicle for the STRIPE portion of the demonstration. Hazeltine packet radios are standard military hardware that constrains the teleoperation capability of our vehicle: 16 kb/sec data rate and a 10 km, near-line-of-sight operational range per link. The front passenger seat of the HMMWV is a software test station for running and debugging soft-

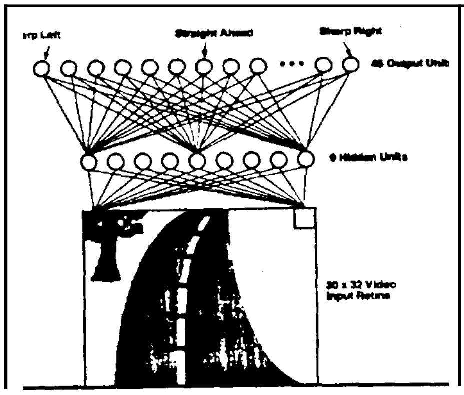

Figure 5. ALVINN architecture

ware. This software test station has a Sparc computer terminal and a set of video monitors.

SOFTWARE

The software system is allocated into three primary functions: mission planning; mission execution; and mission assessment. Mission planning takes information from external systems and generates a specific mission plan (either team mission or vehicle mission) and map overlay data for the vehicles to accomplish. Mission execution takes a validated vehicle mission plan and performs the activities needed to meet the mission objective. The functions that support the mission execution include vehicle preparation, activity sequencing, task generation, task execution, and task monitoring. Finally, the mission assessment includes all functions to determine if the mission results meet the mission objective.

The vehicle subsystem software configuration includes:

1) ***Perception*** – Converts raw sensor signals into positional information, maps, feature lists, and imagery for symbolic recognition, planning and assessment, and behavior generation. Perception is divided into three computer software components (sensor preprocessing, feature extraction and local mapping, and symbolic recognition). Sensor preprocessing accepts data from navigation and other mobility sensors and performs routine pixel- or line-based processing to produce highquality imagery and signals. Feature extraction and local mapping is tasked to accept data from sensor preprocessing and perform feature extraction and image transformations to produce Cartesian elevation maps, feature lists, and fully processed images to be used by the ALVINN software. Symbolic recognition's primary responsibility is to recognize landmarks and alarm points that are annotated in the world model.

2) ***Reasoning*** – Is responsible for executing the mission plan by issuing appropriate activities and tasks that the vehicle is required to perform. Reasoning is composed of four software components (behavior generation, command arbitration, assessment and planning, and world model maintenance). When given a set of conditions, modes, and some limited knowledge of the surroundings, behavior generation creates a sequence of candidate commands producing the desired system actions. Command arbitration determines which of the competing behaviors is activated, based on a priority management scheme that is responsive to constraints and weights from assessment and planning. Assessment and planning gives information on the state of the world and the state of the vehicle, and determines the actions to be performed. World model maintenance maintains information about the external environment in a variety of representations as needed for efficient operations, including other vehicles' position, the vehicle's own relationship to the environment, and the vehicle's internal environment.

3) ***Control*** – Is responsible for executing the time-intensive task of vehicle control and status as well as control of navigation and RSTA sensors and their associated pointing mechanisms. Control includes four software components (handle message, control state, manage the operator workstation **(OWS),** and monitor vehicle location). Message handling receives commands and requests from multiple sources and sends commands to provide actuation or control goals for the vehicle. Control state reads in configuration data, and initializes control software and performs calibration sequences for actuators and sensors. Manage **OWS** uses software for command generation, data monitoring, and operator interface functions. Monitor vehicle location monitors the sensor preprocessing to obtain vehicle position and orientation updates.

4) ***Utilities*** – Provide common services that allow coordinated access to limited computational and communication resources on the vehicle. Utilities are composed of five software components (communications, data management, process management, system services, and test and replay). Communications provide a mechanism for interprocess and intervehicle communications. Data management executes database services, communications logging, and data logging. Process management is responsible for initializing processes, initializing communications and broadcast, synchronizing time at start-up, monitoring processes, and assigning process priorities. System services provide standardized operating system utility services. Test and replay provides the capability to simulate a run-time session from a previous execution or run. For Demo II, we are using Carnegie Mellon University's TCX-Task Communications, an interprocess communications package whose goal is to provide routines for controlling the sending and receiving of information and maintaining channels of communications between cooperating processes.

Cocontractor produced applications software was integrated into the demonstrations, and furnished as government-furnished property. This software, in addition to Martin Marietta-developed software, is initially placed in the software development library for development, test, integration, and release.

KEY DEMO A TECHNOLOGIES

Road Following Using ALVINN – In vision-based autonomous driving, the objective is to steer a robot vehicle based on input from an onboard video camera. The vehicle is equipped with motors on the steering wheel, brake, and accelerator pedal, enabling computer control of the vehicle's trajectory. Road following is a typical autonomous driving task that has been addressed using artificial neural networks. For this task, the input consists of images from the video camera, and the output is a steering command that will keep the vehicle on the road.

The connectionist model for autonomous road following used in the ALVINN system [Pomerleau, 19921 is the feed forward multi-layer perceptron shown in Figure **5.** The input layer consists of a single 30x32 unit "retina" onto which a video image is projected. Each of the **960** input units is fully connected to the four-unit hidden layer, which is in turn fully connected to the output layer. The 50-unit output layer is a linear representation of the currently appropriate steering direction. The center-most output unit represents the "travel straight ahead' condition, while units to the left and right of center represent successively sharper left and right turns. To drive the Demo A vehicle, an image from the video camera is reduced and pre-processed to 30x32 pixels and projected onto the input layer. After propagating activation through the network, the output layer's activation profile is translated into a vehicle steering

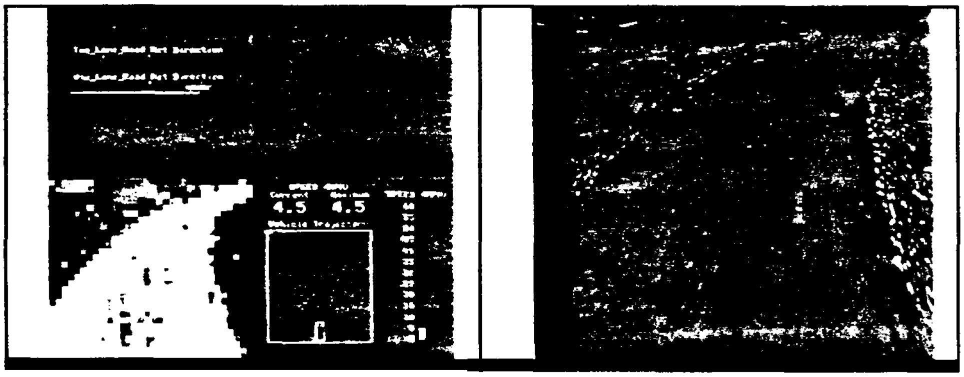

Figure 6. Diagram of weights for a typical road scene

command. Instead of simply selecting the output with the highest activation level, as is typically done in classification tasks like character recognition and speech recognition, the steering direction dictated by the network is taken to be the center of mass of the "hill" of activation surrounding the output unit with the highest activation level (Figure *6).* Using the center of mass of activation instead of the most active output unit to determine the direction to steer permits finer steering corrections, improving ALVINNs driving accuracy.

The most interesting and novel aspect of the ALVINN system is the method used to train it. In this technique, called training "on-the-fly," the network is taught to imitate the driving reactions of a person. As a person drives, the network is trained with back-propagation using the latest video image as input and the person's steering direction as the desired output. To allow generalization to new situations, variety is added to the training set by shifting and rotating the original camera image in software to make it appear that the vehicle is situated differently relative to the road ahead. The correct steering direction for each of these transformed images is created by altering the person's steering direction for the original image to account for the altered vehicle placement. For example, if the person was steering straight ahead, and the image was transformed to make it appear the vehicle is off to the right side of the road, the correct steering direction for this new image would be to steer toward the left to bring the vehicle back to the road center. Adding these transformed patterns to the training set teaches the network to recover from driving mistakes, without requiring the human trainer to explicitly stray from the road center and then return.

Running on two Sparc workstations onboard the Demo A vehicle, training on-the-fly required about two minutes during which the driver drove over about a 1/4- to **1/2-mile** stretch of training road. During this training phase, the network typically is presented with approximately 50 real images, each of which is transformed 20 times to create a training set of 1000 images. Once it has teamed, the network can accurately traverse the length of the road used for training, and generalize to drive along parts of the road not encountered during training. Also, because determining the steering direction from the input image merely involves a forward sweep through the network, the system is able to process 10 images per second, and drive at up to ***55*** mph. This is more than five times as fast as any non-connectionist system using comparable hardware [Kluge, 1990][Crisman, 1990]. The flexibility provided by the neural network has allowed ALVINN to learn to drive under a variety of weather and lighting conditions. Individual networks have been trained to drive on single-lane dirt and paved roads, two-lane suburban and city streets, and multi-lane divided highways. On the highway, ALVINN has driven over 90 miles without human intervention.

There are a number of factors that contribute to the success of ALVI". The first is the fact that the training set is augmented with additional patterns by shifting and rotating each live video image. If the network is trained exclusively on live camera images, it quickly makes a steering mistake when driving autonomously, from which it **has** not learned to recover, and drives off the road. Perhaps **a** more fundamental characteristic of the task that makes autonomous driving easier than **a** task such as speech and character recognition is the fact that spatial invariance is not required in this domain. In fact, it is the positions of features like lane markings in the input image that determine the correct steering response. Because spatial invariance is difficult to achieve using artificial neural networks, tasks that do not require this property are easier to learn. A third factor contributing to ALVINN's success **has** been the care taken in task decomposition. Instead **of** training a single network to drive in all situations, separate networks are training **for** different road types. For example, different networks are trained to drive on city **streets** and divided highways.

Teleoperation with STRIPE – STRIPE is a method of semi-autonomous teleoperation that allows a vehicle to traverse accurately while communicating with the operator on a very low bandwidth link [Kay, 19931. When driving on hilly or irregular terrain, it is difficult to navigate a path from a **2-D** image. The problem is that it is not easy to map a 2-D image to the **3-D** world. Typical mapping approaches to this problem require either physically measuring or inferring the **3-D** geometry. Previous methods have proven to be difficult and time inaccurate. STRIPE overcomes these problems by using an on-line approach to determine the path's geometry. Instead of knowing the shape of a particular path before attempting to drive it, the shape of the path is computed accurately as the vehicle drives.

In this method, we must interpret **2-D** images using **3-D** geometry. To do this, some additional information is required, be it stereo images or a 2-D image with additional constraints about the world. The **"flat Earth"** assumption is a simple way to constrain the problem. It assumes all points in the world lie in the same plane. Instead of modeling the world as a single plane, **STRIPE** models the world as a collection of polygons. As the vehicle moves, new polygons are continuously added to its internal world model. Each new polygon is derived from sensing the orientation of the patches of the ground beneath the vehicle as it moves. The path in front of the vehicle becomes increasingly accurate as the world model is incrementally refined. Points far ahead of the vehicle are rough approximations, but the incremental polygonal representation will almost always give adequate 3-D projections for the next few points used for the steering command. It is usually safe to assume that the next little bit of road immediately in front **of** the vehicle lies in the same approximate ground plane of the vehicle itself.

In the **STRIPE** scenario, a single image is transmitted from the vehicle to the operator. The operator uses a mouse to pick **a** sequence of points in the image that the vehicle should follow (Figure 7). This sequence of **2-D** points is transmitted back to the vehicle. These **2-D** points, together with the incremental polygonal Earth technique, transform the **2-D** image points to **3-D** world points along the desired path in real time on the vehicle.

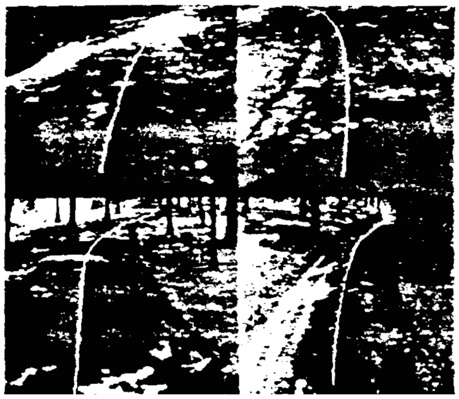

Figure 7. Examples of STRIPE sequence ***of*** *points*

After the vehicle moves a certain distance or reaches its last point, another image is transmitted to the operator and the process is repeated. To **speed** up the process, an image can be transmitted in parallel with the vehicle following a path once the first path has been defined.

STRIPE keeps track of certain coordinate transformationsto do the incremental projection of **a** path created at a previous position onto the next ground plane. The transformation to make this system work is between the old camera's coordinate system and the current vehicle's coordinate system. The position information required is relative data taken from **our INS** system and the **GPS** system, **STRIPE** was used in Demo A to get onto the road to start the demonstration, ***to*** position itself for the RSTA mission, and to transition between the various road-following modes.

MISSION DEMONSTRATION WITH TELEOPERATED RSTA

In the Demo II scenario, the vehicle will position itself to do a overwatch where it will employ the RSTA mission module to observe threats and target such threats for indirect fire. In Demo A, the operator can control the RSTA sensors and view images through teleoperation at the **OWS.** More specifically, he can control the pan and tilt, zoom, and focus of the RSTA camera and control the image capture and communication functions at the workstation. The teleoperated control of the pan, tilt, and zoom in Demo A shows relevance to Demo II in which these functions will be autonomous. The goals for Demo II are (1) the system automatically detects targets using the forward-looking infrared (FLIR) sensor and identifies it using the ladar **sensor;** and, **(2)** the system automatically controls the sensor using "sensor-guided search." When target detection ***oc***curs during sensor-guided search, the RSTA module automatically bore sights the target (i.e., trains the narrow field of view ladar and camera on the target) ***so*** that the target can be identified.

For Demo A, the vehicle was equipped with a single color camera on a pan/tilt mechanism to perform the limited RSTA mission. The operator at the **OWS** has the following capabilities:

1) Direct, teleoperated control of the pan and tilt, zoom, and focus functions of the RSTA camera;

2) Adjust pan and tilt, and **zoom** controls without sending commands to the vehicle; the commands are sent once the desired adjustments are made;

3) Situational awareness tools (in the form of **a** map) so the operator can control and interpret where the sensor is pointing with respect to meaningful terrain features. The situational awareness tool graphically displays the sensor footprint with respect to terrain for both current and potential sensor poses;

4) To capture and transmit an image or sub-image from the vehicle to the **OWS.**

5) To store a few of the most recent images in a queue at the **OWS;**

6) To display images from either the queue or archive;

7) To display live video from the RSTA camera at the OWS and overlay graphics on the video indicating the operator's current selection for the sub-image.

All of these capabilities were used in the RSTA portion of the demonstration. The typical scenario for the RSTA mission is to first activate the RSTA module by turning on the RSTA control screens at the **OWS** when the vehicle is stationary. The operator has the choice of controlling the pan, tilt, **zoom** functions of the RSTA camera manually with the aid of the maps in the situational awareness tool. He can use the maps to inspect current and past sensor footprints or he can ensure optimal area coverage by viewing the map/sensor **footprint** display. In addition **to** the situational awareness tool, the operator inspects the RSTA sensor output through the live video imagery at the **OWS.** The live video allows the operator to search for targets in the scene, simulating the search function for Demo II.

CONCLUSION

Demonstration **A** was held at Martin Marietta's Waterton, CO, facility and it was a success. The vehicle demonstration was executed precisely to script over two days for a crowd of approximately 200 visitors. It was set in **a** militarily relevant scenario, including **a** military briefing to **start** the day. Demo A fulfilled its goal of demonstrating basic mobility and teleoperation capabilities for a single vehicle. The framework was laid with the architecture, hardware, and software subsystems to be able to perform Demonstration **B** in 1994. Currently, the large UGV Demo II team of cocontractors is busy working on their respective technologies in preparation for next year's demonstration.

ACKNOWLEDGMENTS

This work was sponsored by the Advanced Research Projects Agency (ARPA) under the contract Surrogate Semiautonomous Vehicle (SSV), contract number DAAH01-**92-C-R**101, monitored by the U.S. Army Missile Command. We would like to thank Mike Toscano, Office of the Secretary of Defense; Erik Mettalla, formerly of ARPA and now with the University of Texas, Arlington; and Chuck Shoemaker, currently at ARPA, for their support of this program. We would also like to thank Richard Luhrs and Steve Munkeby, former and current program managers of the **SSV** program; Chuck Thorpe and Dean Pomerleau, Carnegie Mellon University; and the rest of the UGV Demo II team members for their continual support and encouragement.@

References

[Argo, 1993] Peter *Argo.* "Simulation of Arbiter and Actuator Properties and Discussion of their Effects Upon Vchiclc Trajectory Tracking " SPIE. Vol 2058. Mobile Robots VIII, Boston. MA, Nov. 1993.

[Brooks, 1986] R. A. Brooks: "A Robust Layered Control System For A Mobile Robot " IEEE Journal of Robotics and Automation , Vol RA- 10, No. 1, March 1986

[Crisman, 1990] J. D. Crisman and C. E. Thorpe: "Color Vision for R a d Following." Vision and Navigation: The CMU Navlab, Charles Thorpe (Ed.). Kluwer Academic Publishers. Boston. 1990.

[Gothard, 1992] B. Gothard, R. Spomer, J. Medina, and W. H Chun: "Design Issues with the SSV Platform" SPIE. Vol. 1831. Mobik Robots VII, Boston. MA. Nov 1992

[Gothard, 1993] B. Gothard, R. Etersky, and R. Ewing "Lessons Learned *on* a Low-Cost Global Navigation System for the Surrogate Semi-Autonomous Vehicle " SPIE, Vol 2058. Mobile Robots VIII. Boston. MA. Nov. 1993

[Kay, 1993] J. S. Kay, and C E Thorp. "STRIPE Supervised TeleRobotics Using Incremental Polygonal Earth Geometry." Intelligent Autonomous Systems 3 (IAS-3), Groen, Hirose, Thorpe (Eds.), IOS Press.

[Kluge, 1990] K. Kluge and C. E. Thorp "Explicit Models for Robot Rod Following " Vision and Navigation· The CMU Navlab, Charles Thorpe (Ed), Kluwer Academic Publishers, Boston. MA. 1990.

[Pomerleau, 1992] D. A. Pomerleau: Neural Network Perception fa Mobile Robot Guidance. Ph D. dissertation, Carnegie Mellon Univmiry. February 1992

[Turk, 1988] M A. Turk, D *G.* Morgenthaler, K. D. Gremban, and M . Marra: "VITS-A Vision Sysum for Autonomous Land Vehicle Navigation." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 3. May 1988

[UGVMP, 19931 Unmanned Ground Vchiclc Master Plan. Department of Defense, August 1993.

[Wolfe, 19921 W .J. Wolfe and W H. Chun "Robot Architectures and Design Paradigms " SPIE, Vol 1831, Mobile Robots VII, Boston, MA. Nov. 1992.