The Autonomous Land Vehicle (ALV) Preliminary Road-Following Demonstration

Lowrie, James, Thomas, Mark, Gremban, Keith, Turk, Matthew

James W. Lowrie, Mark Thomas, Keith Gremban, Matthew Turk, "The Autonomous Land Vehicle (ALV) Preliminary Road-Following Demonstration," Proc. SPIE 0579, Intelligent Robots and Computer Vision IV, (11 December 1985); doi: 10.1117/12.950819

SPIE.

Event: 1985 Cambridge Symposium, 1985, Cambridge, United States

The Autonomous land vehicle (ALV) preliminary road-following demonstration

James W. Lowrie, Mark Thomas, Keith Gremban, Matthew Turk

Martin Marietta Denver Aerospace Advanced Automation Technology Section
PO Box 179 Denver CO 80201

The autonomous land vehicle program overview

The ALV project is sponsored by the Defense Advanced Research Project Agency (DARPA) as part of its Strategic Computing Program and contracted through the Army Engineer Topographic Laboratories (ETL) under contract DACA76-84-C-0005. The purpose of the strategic computing program is to advance the state of the art in artificial intelligence, image understanding, and advanced computer architectures and to demonstrate the applicability of these technologies to advanced military systems.[1]

The strategic computing (SC) program is separated into three primary areas--technology base, applications, and infrastructure. The technology base contractors are tasked with pursuing generic long-range high-payoff research in numerous disciplines including image understanding, expert systems, planning and reasoning, symbolic processing architectures, high-speed signal processing systems, and others. The application areas are being funded to transition the technology from the research domain to the military application domain with the intent of demonstrating a series of progressively more complex operational capabilities. Finally, the infrastructure of the SC program provides the framework for both the research community and the application programs. This framework includes information networks, research machines, and system development tools.

The ALV project is one of the SC program's application areas aimed at advancing and demonstrating the state of the art in autonomous navigation and tactical decisionmaking. The project is driven by the series of progressively more difficult demonstrations identified in Table 1. These successive demonstrations were selected because they drive the development of technology in artificial intelligence, image understanding, and advanced computer architectures.

Table 1. ALV demonstration.

Year	Distance	Speed	Capability
May 1985 (Preliminary Road-Following Demonstration)	1 km	5 km/h	The vehicle will traverse a uniform road with smooth curves at a constant speed. During conditions where the vision subsystem is unable to locate the road, the vehicle may follow a prestored map of the track. The vehicle must navigate from visual data over 75% of the distance.
November 1985 (Road-Following Demonstration)	5 km	10 km/h	The vehicle will traverse a nonuniform road with sharp corners. The vehicle speed will vary as a function of vision confidence and road geometry. The vehicle must navigate from visual data 100% of the time. The vehicle must demonstrate an autonomous counter-rotate capability.
1986 (Obstacle Avoidance Demonstration)	20 km	20 km/h	The vehicle will traverse a nonuniform road with numerous intersections. The vehicle must sense and model obstacles placed on the road surface and plan a path to avoid them.
1987 (Crosscountry Demonstration)	10 km	5 km/h	The vehicle must be capable of planning an a priori route through the terrain using a prestored terrain database. The system must then use sensory data to model the local terrain and avoid natural obstacles placed along the route. The position of the vehicle with respect to the route must be monitored and updated as necessary. The vehicle must navigate through rough roadways.

Success of the ALV project depends on careful coordination with the technology base contractors to transfer technology from the research domain to the application domain as rapidly as possible. To simplify the technology transition process, ALV was designed as a flexible testbed that will enable rapid transition from hypothesis to testing. The intent of the testbed is to encourage the technology base contractors to conduct experiments with the vehicle in a realistic environment. The results of these experiments would then lead naturally into design of the demonstration system.

This paper describes the long-range ALV system concept that the project is building toward, the system requirements for road-following, gives an overview of the ALV system as it was configured for the May 1985 demonstration, and contains detailed descriptions of the vision, navigator, pilot, electronic, and vehicle subsystems.

ALV long-range system concept

The progressive system demonstration schedule, along with the requirement to transition capabilities from the strategic computing technology base contractors, makes it essential to define a long-range generic system architecture. It is more beneficial to build each demonstration system within the framework of the long-range system architecture than to discard each demonstration system following its completion. By defining a long-range system architecture and analyzing the long-term requirements, we can project the technology voids that will become the topic for research by the technology base contractors.

Definition of the long-range system architecture has been the topic of a series of working group meetings between various technology base contractors and the ALV project team.[2] The following technology base contractors participated in this definition--University of Maryland, Carnegie Mellon University, SRI International, Advanced Information and Decision Systems, Hughes AI Center, and Honeywell.

Autonomous mobility in a dynamic unstructured environment requires that a system sense its environment, model critical features using the sensed data, reason about the model to determine a mobility path, and control the vehicle along that path. Evaluation of these basic mobility requirements resulted in the definition of the system concept shown in Figure 1. Two additional requirements associated with the objective of the strategic computing program were factored into this configuration. First, the primary emphasis of the program is on perception and reasoning with minimal research being pursued in the areas of control and physical vehicles. Because it is also desirable to rapidly integrate and test numerous concepts on the testbed vehicle, we have defined a "virtual vehicle" consisting of the physical vehicle, the sensors, and the control subsystems. The hardware and software interfaces at this level are well known and experiments that conform to these interfaces can be rapidly integrated and tested.

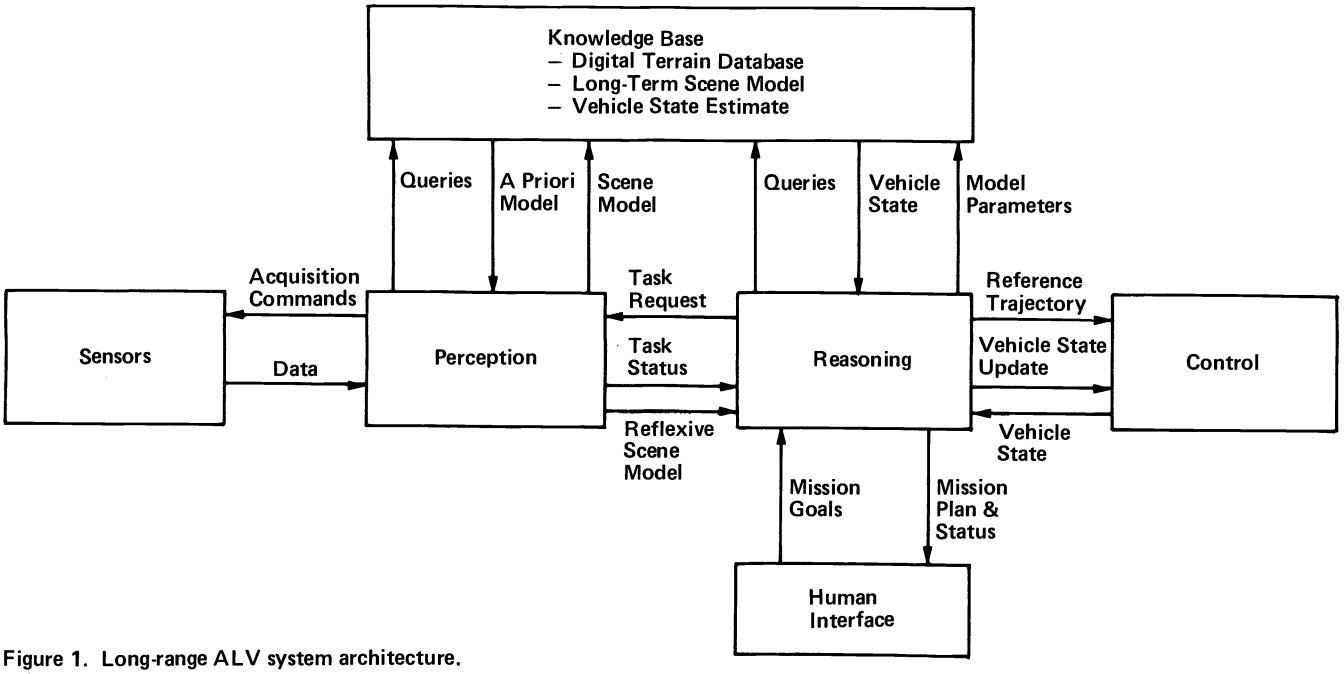


Figure 1. Long-range ALV system architecture.

The human operator will specify the mission goals and constraints that should be factored into decisionmaking through the man/machine interface (MMI). The complexity of these mission goals will increase with each successive demonstration. For May 1983 the goal specification was simply to follow the road for 1 km. In 1987 the goal becomes more complex--travel to point A, perform task B, proceed to point C, . . . Later demonstrations will also include complex tactical situations that must be dealt with.

The reasoning system will interpret these mission goals and decompose them into the operations to be performed by the vision subsystem. As part of this decomposition process, the reasoning subsystem will access a digital terrain database being developed by the Engineer Topographic Laboratories and plan an a priori route through the environment to achieve the mission goals. The perception system is considered to be a resource of the reasoning subsystem. In this capacity the reasoning subsystem will specify goals for the perception system to perform. These goals will include specification of the features of interest, a time allocation for the process, and a focus of attention defining the geometric area to be modeled. The perception subsystem will then decompose this goal into

specific perception tasks. The perception subsystem will have sole control over all sensors and will produce only a high-level symbolic model of the environment for reasoning. Figure 2 illustrates the sensor/perception interface. Following completion of model generation, the perception subsystem will pass the model to the reasoning subsystem along with a status description. The status will indicate if the perception system was able to achieve the goal and if not will describe the potential reasons for failure.

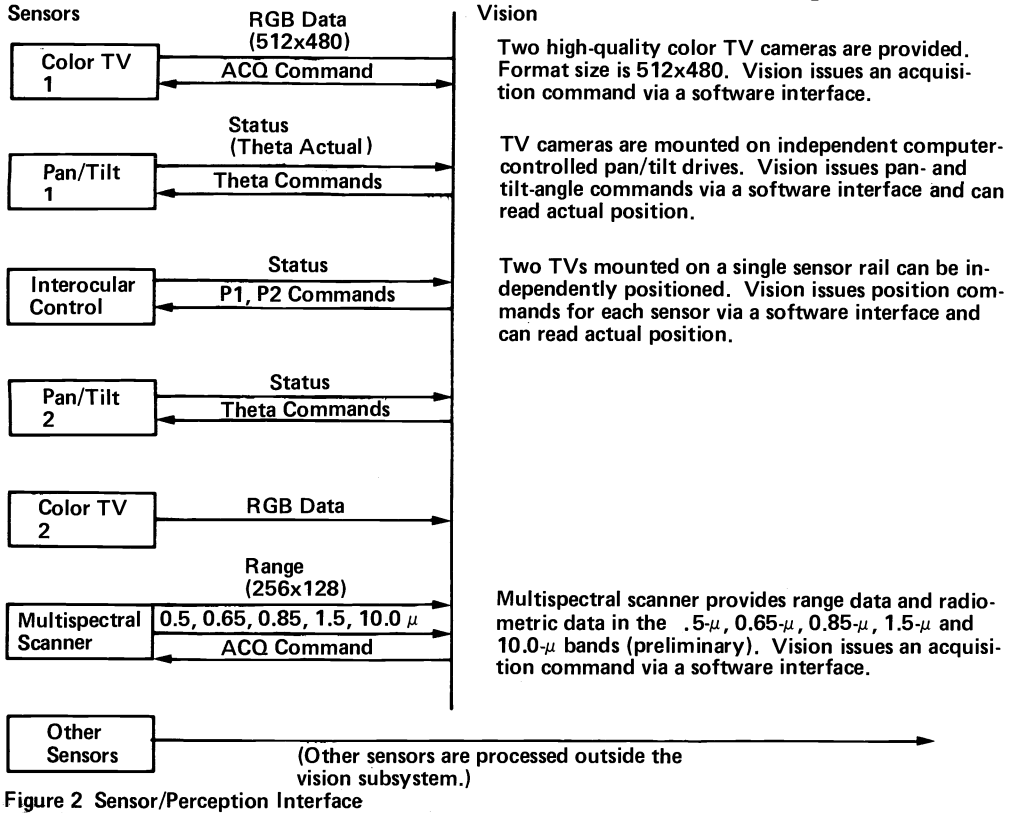


Figure 2 Sensor/Perception Interface

For the 1987 time frame we anticipate there will be two forward-looking high-resolution color TVs mounted on independent pan/tilt mounts with a controllable interocular distance ranging from 1 to 5 ft. Each camera will have an independent 3-channel 8-bit digitizer. A 5-channel multispectral laser scanner being developed by the Environmental Research Institute of Michigan (ERIM) will also be incorporated.

The reasoning subsystem will interpret the perception model and will plan a path for the vehicle to avoid nontraversable regions and localized obstacles. Because of the significant amount of time involved in processing the sensory data to produce a symbolic model, it is not possible in the near future for the vehicle control system to close the high-speed servoloop from visual data. Therefore we have introduced the concept of a reference trajectory whereby the vehicle control system follows a selected path from one model until the next model is generated. Figure 3 illustrates the reasoning control interface portion of the virtual vehicle. The control subsystem will be responsible for three activities. First, it will control the motion of the vehicle along the specified trajectory. Second, the control subsystem will evaluate the specified trajectory and determine such unsafe conditions as sudden high-speed turns. Third, the vehicle state estimate consisting of vehicle position, velocity, heading, pitch, and roll will be maintained within the control subsystem.

The physical vehicle consists of a drive chassis supplied by Standard Manufacturing, a 122-hour auxiliary power unit, and a 60,000-Btu air conditioner. This physical platform is considered to be sufficient to support the electronics for all projected demonstrations and experiments. Figure 4 illustrates the physical vehicle.

System requirements for the May 1985 demonstration

The May 1985 demonstration required the vehicle to autonomously travel on a paved road over a distance of 1 km at a speed of 5 km/hour. The 1-km distance requirement introduced the need for a robust vision subsystem capable of operating on hundreds of successive scenes. The 5-km/hour speed requirement introduced the need for special-purpose computers that could rapidly process imagery. This section summarizes the analyses conducted to define the system requirements for the May 1985 demonstration.

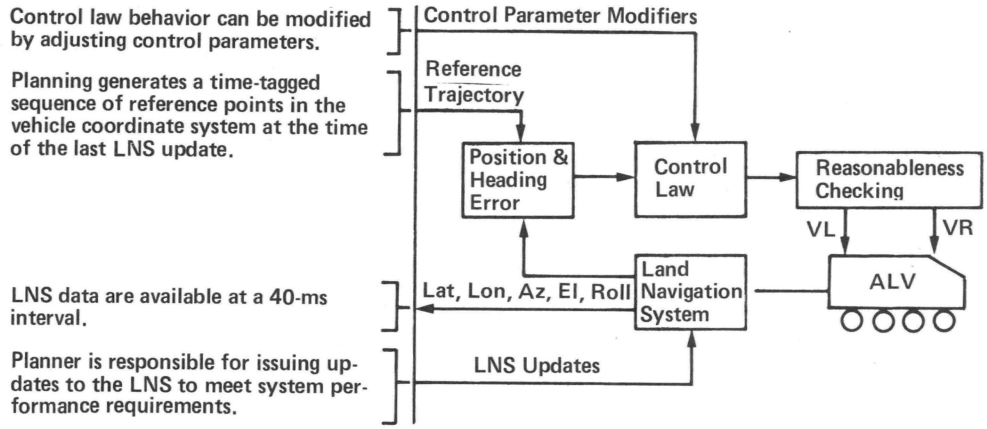


Figure 3. Reasoning control interface.

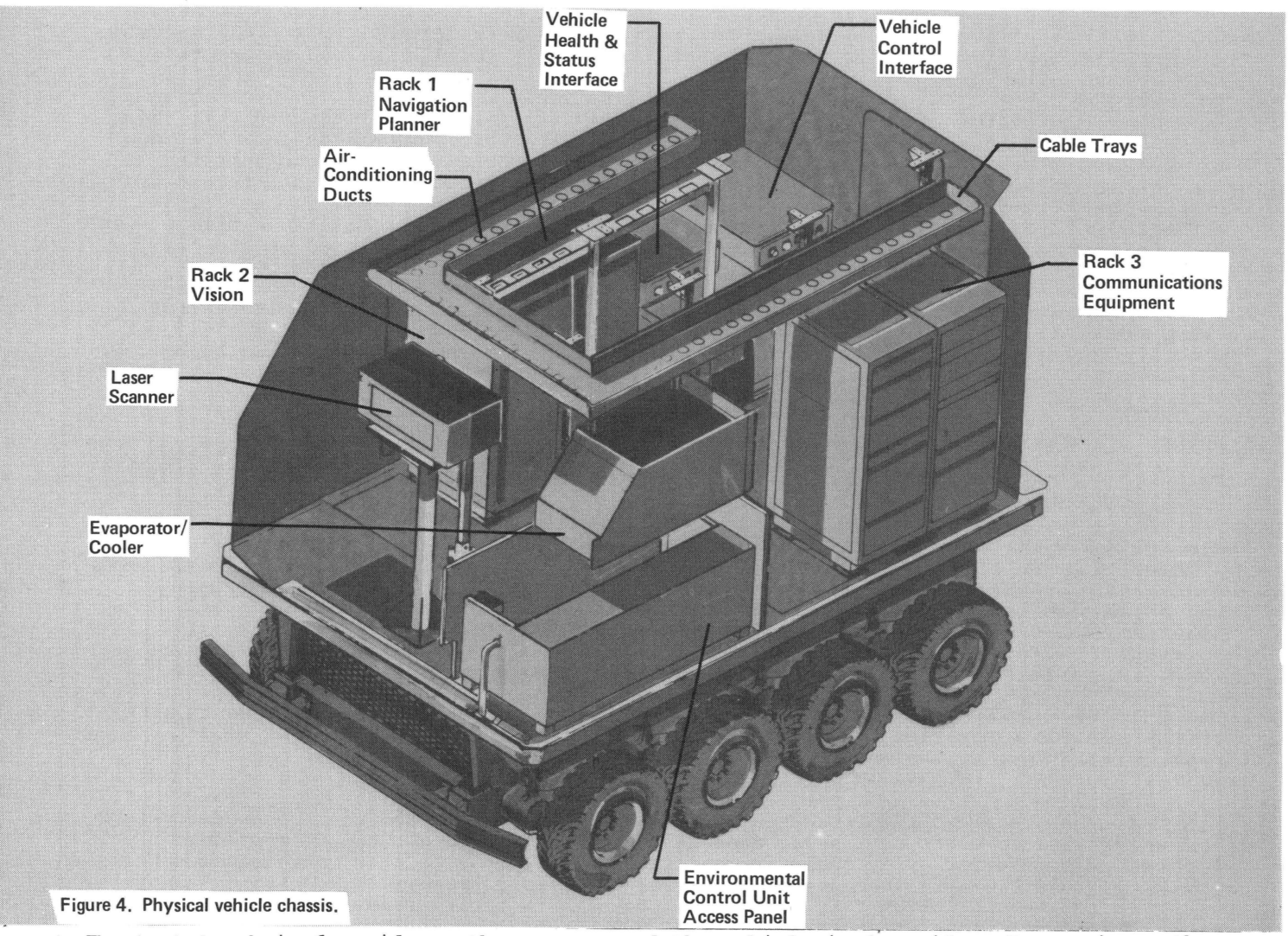
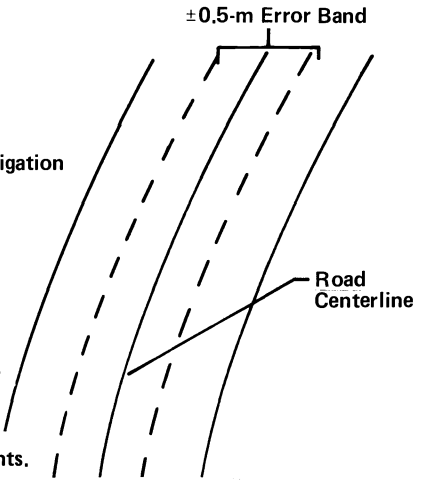


Figure 4. Physical vehicle chassis.

The test track is 6-m wide on the average and the vehicle is 3-m wide. To remain on the road, the vehicle must maintain a lateral error no greater than ±1.5 m. To provide a margin of safety it was decided that the vehicle must travel within ±0.5 m of the road centerline as shown in Figure 5.

As an additional safety factor it was decided that the vehicle could navigate off a prestored map of the road whenever the vision subsystem produced low-confidence models as long as visual information was used more than 75% of the time. To implement this vision override capability the entire test track was surveyed to an accuracy of 0.15 m. When



- Speed — 5 km/h
- Distance — 1 km
- Accuracy — ±0.5 m from Centerline
- Competency — Vision Scene Model Used for Navigation More Than 75% of the Time

Vision Override

- Test track centerline has been surveyed to an accuracy of 0.15 m.
- If the vehicle wanders outside a ±0.5-m error band, the map data will be used to bring the vehicle back to the centerline.
- Vision override is expected to be used in conditions where the vision subsystem cannot locate the road edges.

Figure 5. May demonstration performance requirements.

conditions did not allow the vision subsystem to segment the road in the image or derive the 3-D geometry of the road edges, the map data were used to control the vehicle. Vision override is an artifact of the May 1985 demonstration only and will not be incorporated in future demonstration systems.

To control the vehicle on a continuous basis at a fixed velocity we believed that the vision-based scene model could not be generated at the servoloop update rates. Analysis indicated that the servoloop needed to operate at 40-ms update intervals and that the state-of-the-art vision algorithms were three orders of magnitude slower than that rate. Therefore the concept of a reference trajectory was used (Fig. 6). This concept allows the servoloop to operate at 40 ms while the visual information is updated at a much slower rate. At time T_0 the vehicle acquires the i^{th} image while the control system steers from the $(i-1)^{th}$ trajectory. The i^{th} image is processed up to time T_1 to generate a scene model and corresponding trajectory. At T_1 the vehicle may navigate from the i^{th} trajectory while the $(i+1)^{th}$ image is being processed. The basic constraining factor is that the vehicle must not pass the end of the i^{th} trajectory before the $(i+1)^{th}$ trajectory is ready. From this constraint the following limiting condition can be derived

$$\text{Velocity} * (T_2 - T_0) < \text{Trajectory Length}$$

where

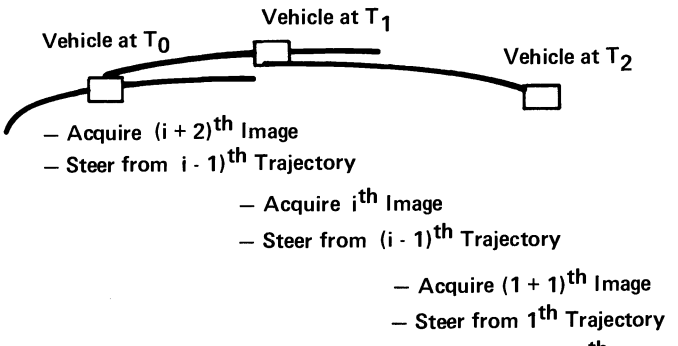
$T_2 = 2T_1$ and T_1 is the processing time associated with generating a reference trajectory.

The maximum trajectory length is driven by the mobility accuracy requirement of ± 0.5 m. The mobility error of the vehicle can be broken into several terms, some of which increase monotonically as a function of the trajectory length. Figure 7 illustrates the error model terms used for this analysis. It should be noted that all of these error models use a small angle approximation. There are four basic terms. The constant control error term includes such effects as numerical errors, actuator noise, gear backlash, and constant biases. Preliminary analysis indicated that this term should be approximately 0.3 m.

The error of the land navigation system (LNS) position outputs increases linearly with distance as 0.002 * D. Terms contributing to this effect include gyro drift, odometer errors, misalignment of the unit with respect to the vehicle drive access, and resolver errors.

The vision subsystem will introduce errors in modeling the environment. This error term is extremely hard to model because it depends on the specific algorithm, the processor being used, the detector array geometry, and the optics. It will also vary as a function of the local conditions in the environment. For analysis purposes we selected an error model based on the assumption that vision will be able to locate road edges to within 5 pixels. This translates into an error time of 0.008 * D.

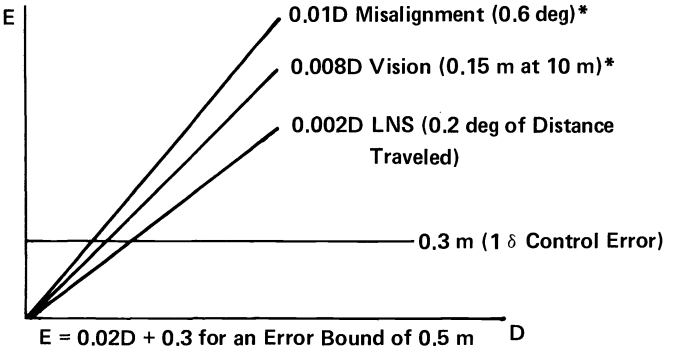
The final error term used for this analysis introduces the effects of misalignment errors between the vision sensors used to build the model of the road and the LNS used to control the vehicle with respect to the model. Analysis of the pan/tilt device and the physical structure of the vehicle indicated that this alignment error will be approximately 0.6 deg, resulting in a vehicle control error of 0.01 * D.



Basic Constraint: The vehicle must not pass the end of the 1^{th} trajectory before the $(1 + 1)^{th}$ trajectory is ready.

Velocity * $(T_2 - T_0)$ < Trajectory Length

Figure 6. Reference trajectory concept.



E = 0.02D + 0.3 for an Error Bound of 0.5 m
D = 10 m

Processing Requirement for One Stage in Pipeline

$T_1 = \frac{0.072}{0.02} s = 3.6 s$

Figure 7. May demonstration-derived requirements—error budget.

Summing the various error terms we find

$$E = 0.02 * D + 0.3 \text{ m}.$$

Given the control error requirement of ± 0.5 m, it is possible to solve for the maximum trajectory length that can be used

$$D = \frac{0.5 - 0.3}{0.02} = 10.0 \text{ m}.$$

From the previously described limiting condition it is possible to solve for the processing time requirement for vision

$$\text{Velocity} * 2T_1 \leq 10 \text{ m}$$
$$T_1 \leq \frac{10 \text{ m}}{2 * 1.4 \text{ m/s}}$$
$$T_1 \leq 3.6 \text{ s}.$$

Therefore the amount of time involved in processing the image to create a reference trajectory must be less than 3.6 s to maintain a vehicle speed of 5 km/h with a control accuracy of ± 0.5 m.

May 1985 perception subsystem

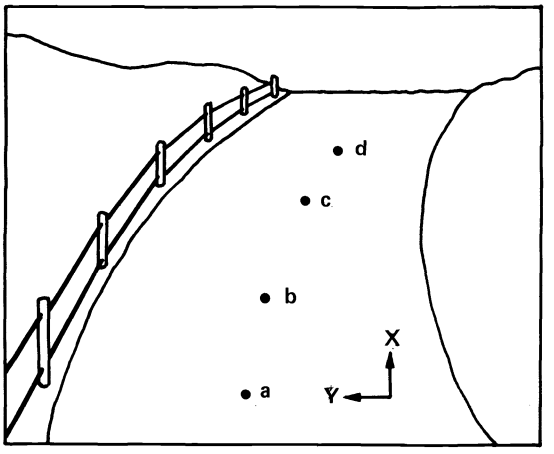
The perception subsystem for the May 1985 demonstration consisted only of visual data processing and was therefore considered to be a vision subsystem. The basic requirement of the vision subsystem is to process the available sensory data and produce a description of a road in front of the vehicle. This description is called a scene model. Scene models must be produced continuously while the vehicle is in motion and must be accurate enough and produced often enough that the vehicle never travels "blind."

The description of the road produced by the vision subsystem is in the form of an array of points representing the road centerline. Centerpoints are ordered from near to far and the unit of measure is meters. These points are given in three dimensions with respect to the vehicle's coordinate system. The vision subsystem is required to output a measure of confidence for each of the centerpoints in the scene model. Figure 8 illustrates a road scene and the corresponding scene model. The vision subsystem resides on a VICOM VDP computer and is in sole control of the video camera, an RCA Hawkeye. The vision subsystem, as shown in Figure 9, is designed to be able to employ multiple road- following algorithms as they became available from the strategic computing technology base contractors. The choice of algorithm is controlled by the vision task sequencer (VITS). A single algorithm was used for the May 1985 demonstration. VITS also mediates all vision subsystem communication and controls the modes of operation--initialization, autonomous, and slave.

The road-following algorithm used in the May 1985 demonstration is executed by two software units, the video data processing unit (VIVD) and the geometry/representation unit (VIGR). VIVD operates on the raw video images to segment the road and to identify points that lie along the road boundary. VIGR takes these points given by image coordinates and uses perspective geometry to compute the locations of road centerpoints in three dimensions. For the May 1985 demonstration, parameters of the vision processing were set to

provide scene models every 2.4 s. Of this time, approximately 2.0 s is spent in segmenting the image and determining the segment boundary, while the rest is spent in computing the 3-D geometry.

VIVD performs the functions of image acquisition, road segmentation, and road boundary extraction. The final result of VIVD processing is a list of pixels representing road boundary points. This list of points is passed to the geometry/representation unit, which performs a 2-D to 3-D transformation and constructs the scene model.



14
02
30
565
} Time Stamp 14:02.30.565

4 } No. of Centerpoints

203
1451
31
} Position Stamp

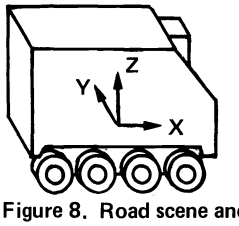
05.01
-- 0.5
0.0
} (X, Y, Z) for Point a

8.35
0.0
0.1
} (X, Y, Z) for Point b

10.09
0.47
0.3
} (X, Y, Z) for Point c

15.21
0.85
0.7
} (X, Y, Z) for Point d

90
85
80
} Confidences in Points a, b, c, d



Scene Model Coordinate System

Figure 8. Road scene and corresponding scene model.

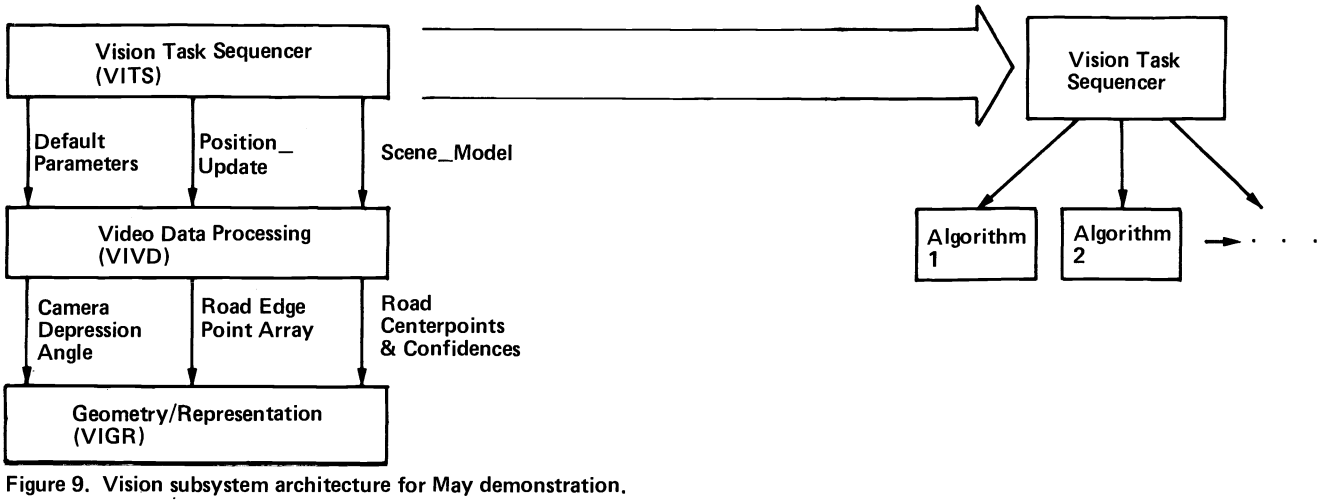


Figure 9. Vision subsystem architecture for May demonstration.

The red (R), green (G), and blue (B) bands produced by a color TV camera are simultaneously digitized and read into VICOM memory. Segmentation of the road occurs in four sequential steps as shown in Figure 10. First, the red, green, and blue images are combined to enhance the road with respect to the surrounding environment. The resulting image is then histogram-equalized to spread the intensity distribution over the entire dynamic range. The image is then repeatedly smoothed to reduce the contrast associated with the surface structure of the road (e.g., exposed aggregates). The last step is to threshold the blurred image to segment the road.

The basic concept behind the segmentation process is that the road is a homogeneous region of relatively constant size. The steps in the process are designed to enhance the difference between road and nonroad pixels, to increase the similarity of road pixels, and to assign known intensity values to road pixels.

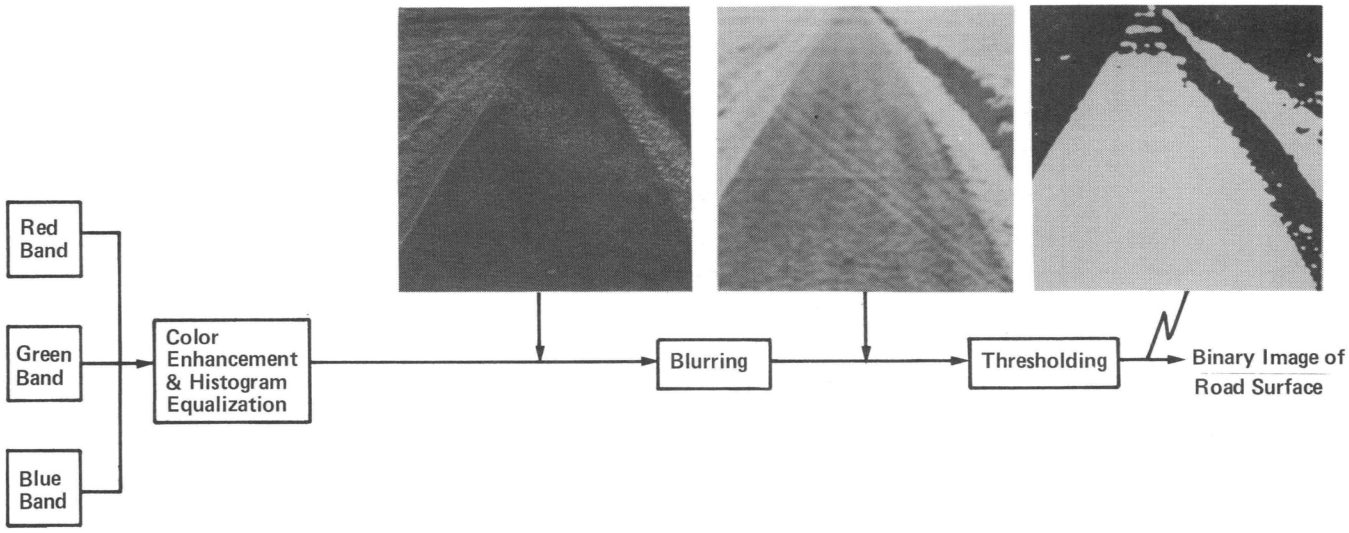


Figure 10. Road segmentation.

The boundary of the segmented road is computed by tracing white pixels that border black pixels. The border pixels are averaged together to smooth the border, and a list of the smoothed border points is generated to be passed to the geometry unit. Figure 11 illustrates the boundary extraction output.

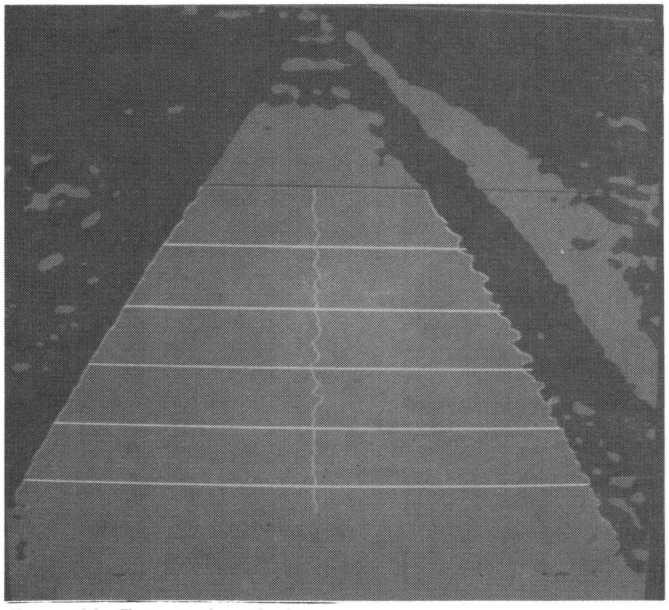


Figure 11. Extracted road edges.

The geometry/representation unit operates on the array of road edge pixels and computes a 3-space location for each pixel. Road centerpoints are computed as well as associated confidence levels and are written into the scene model memory location. The processing steps are discussed below and illustrated in Figure 12. The geometry/representation technique used for the May demonstration was based on the research conducted at the University of Maryland.[3]

The road in the image is modeled by an ordered set of line segments approximating the edges. The first line segments identified are typically 15 to 20 ft in front of the vehicle. Most roads are approximately planar over this distance. Therefore it is assumed that the points of P_1 Q_1 lie in the same ground plane that the vehicle rests on.

The 3-spaced location of the road edges at P_1 and Q_1 can be computed in camera coordinates. This computation is performed using the fact that the intersection of three planes (with linearly independent normal vectors) defines a unique point in space.

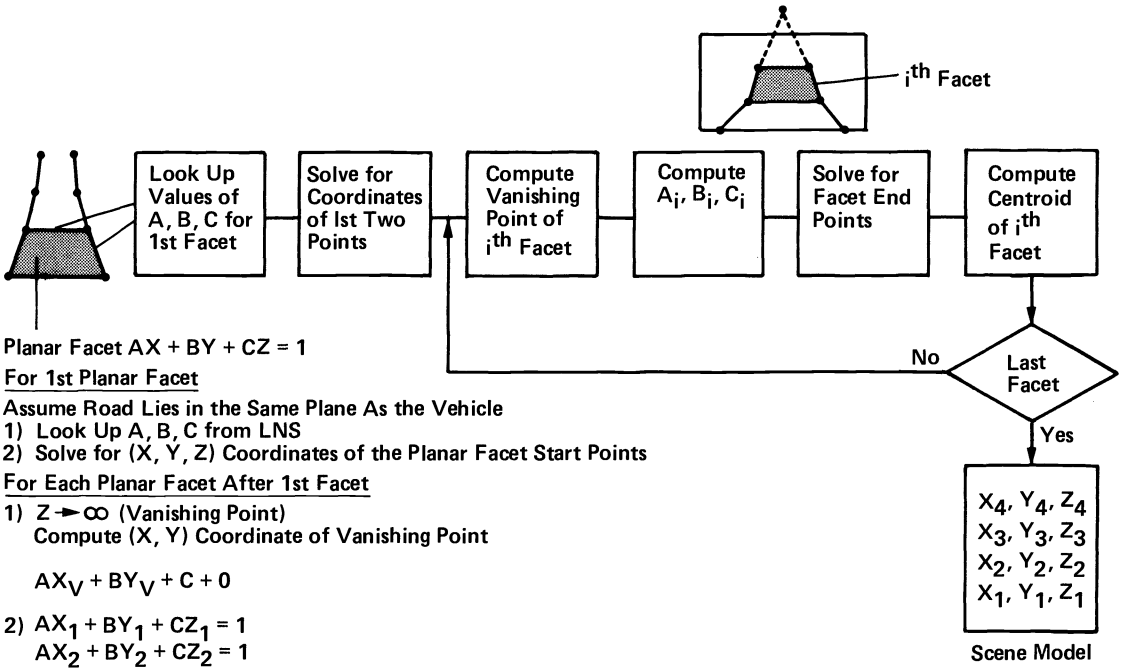


Figure 12. Geometry processing.

We define the equations of three planes that pass through P_1:

The ground plane,

$$\cos(Psi)Y_1 + \sin(Psi)\,Z_1 = height;$$

The horizontal plane through P_1 and the focal point,

$$\cos(-\text{atan}(X_1 - xcen/fx))X_1 + \sin(-\text{atan}(X_1 - Xcen/fx))Z_1 = 0;$$

The vertical plane;

$$\cos(-\text{atan}(Y_1 - ycen/fy))Y_1 + \sin(-\text{atan}(Y_1 - ycen/fy))\,Z_1 = 0;$$

where

Height = camera height above the ground
Psi = camera depression angle
(Xcen, ycen) = pixel location of the center of the image
fx = (image half-width)/(camera horizontal half-angle)
fy = (image half-height)/(camera vertical half-angle)
(X_1, X_1, Z_1) = camera coordinates of the point on the road edge represented by P_1.

The solution to these equations yields the 3-space location of P_1. A similar set of equations can be defined and solved to obtain (U_1, V_1, W_1), the 3-space location of Q_1.

The locations of P_i and Q_i for $i \geq 2$ could be computed the same way if P_i and Q_i were assumed to lie in the same plane as P_1 and P_2. However, the planar assumption generally fails at distances greater than 20 ft.

Alternatively it is assumed that the paired line segments L_i and R_i are coplanar and are parallel in 3-space. If L_1 and R_1 are parallel in 3-space, they converge at a vanishing point in the image. This vanishing point can be easily computed by solving for the intersection of L_1 and R_1 in the image plane, and then correcting for camera geometry. The equation of the plane containing L_1 and R_1 can now be computed.

The equation of the plane is AX + BY + CZ = 1 where A, B, and C are unknown. (X, Y, Z) are the coordinates of a point on the plane.

Dividing through by Z yields

$$A\ X/Z + B\ Y/Z + C = 1/Z.$$

At the vanishing point, $Z \longrightarrow^{\infty} 00$, so $1/Z \longrightarrow 0$. Thus three equations for the plane can be formed

$$A\ (Xvan) + B\ (Yvan) + c = 0$$
$$AX_1 + BY_1 + CZ_1 = 1$$
$$AU_1 + BV_1 + CW_1 = 1.$$

From the equation of the plane it is then possible to solve for the 3-space locations of P_2 and Q_2. The process can then interate for each pair of line segments until all 3-space locations are computed.

A road centerpoint is computed for each pair of line segments. Each pair of segments defines a planar facet in space. The centroid of each facet, found by averaging the coordinates of the corners, is used as a road centerpoint. The confidence for each centerpoint is computed as a function of apparent road width using the equation

$$conf\ [i] = 100 * \left(1 - \frac{known\ width - computed\ width}{known\ width}\right).$$

The final computation performed by VIGR is a check on the "reasonableness" of the centerpoints computed. The equations used to compute 3-space coordinates can easily produce bad values if the edge of the road in the image is not found correctly or if the real edge is jagged. For example, in Figure 13 a pothole in the side of the road results in a jagged edge on one side. As a result, one pair of segments has a vanishing point below the segments themselves. The result, after 2-D to 3-D transformations, is that the road appears to get closer, folding back over the top of the vehicle. A simple check for reasonableness is to test the sign of the difference between successive centerpoints in the Z-direction. If a difference is negative, a "fold" has occurred. The scene model is cleared, and another image is acquired and processed.

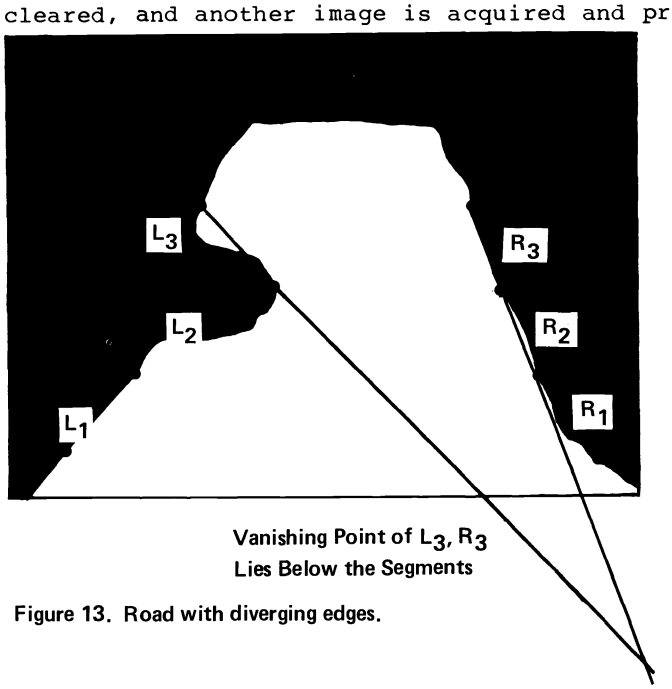


Figure 13. Road with diverging edges.

Reasoning and control subsystems

The basic function of the reasoning subsystem is to generate a detailed reference trajectory prescribing the route of the vehicle using the vision-generated model of the local environment. The control subsystem must generate motor drive commands to steer the vehicle along this prescribed trajectory using position and speed feedback from the onboard land navigation system (LNS) and its associated processor.

Figure 14 graphically depicts the reasoning and control software components and their primary inputs and outputs. The reasoning component consists of two functional units: (1) the navigator, which generates reference trajectories from scene models, and (2) vision override, which produces emulated scene models from *a priori* map data. The pilot

is the only functional unit in the control component. It uses the reference trajectory generated by the navigator and the vehicle position and speed data from the LNS to compute position, speed and heading errors. It evaluates vehicle control signals to drive these errors to zero.

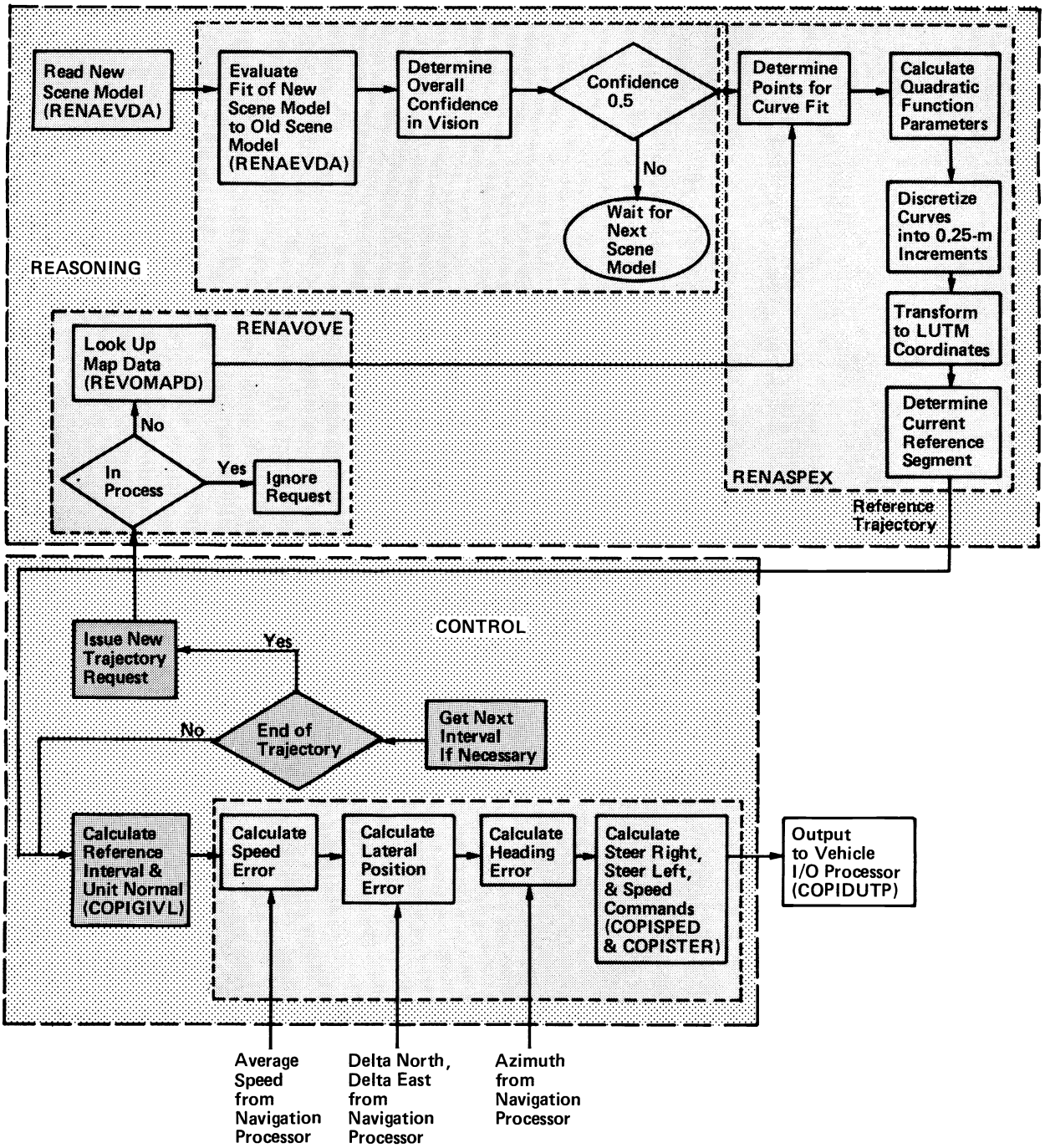


Figure 14. Reasoning and control components.

The pilot software executes in the master 286 processor on a 40-ms interrupt-driven basis coinciding with the timestep between LNS position data updates. The navigator software executes as a background task in the master 286 processor. The navigator acts as an asynchronous task that is activated when a processed scene model becomes available from the vision subsystem, or when the pilot signals that a new trajectory segment is needed. Processing time for the Navigator is about 0.08 s, which takes about 0.3 s to run as a back- ground task.

The reasoning and control subsystems use different coordinates for representing vehicle motion, reference trajectories, and scene models. Because vision acquires an image using

cameras attached to the vehicle, a vehicle-fixed reference frame is optimal. However, the vehicle position is constantly changing and the LNS updates are supplied as delta-northings and delta-eastings, i.e., in terms of an absolute fixed reference. To simplify the pilot analytics, which must be calculated at the real-time update rate, a fixed rectangular coordinate system was selected for representing the vehicle motion and reference trajectories.

Figure 15 illustrates the two coordinate systems used by the reasoning and control subsystems. A fixed rectangular coordinate grid is considered with the origin at the test track starting point, the X-axis pointed east, and the Y-axis directed north. A local coordinate system is fixed to the vehicle with the X-axis directed out the front of the vehicle and the Y-axis out the left side. The vehicle position (X, Y) locates the origin of the local coordinate system with respect to the fixed coordinates and the heading angle is the angle between the fixed Y-axis (north) and the vehicle X-axis. The point transformation for converting a position (X, Y) in the local coordinates to the fixed coordinates (X, Y) is given in terms of these values as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X_v \\ Y_v \end{bmatrix} + \begin{bmatrix} \sin & -\cos \\ \cos & \sin \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} .$$

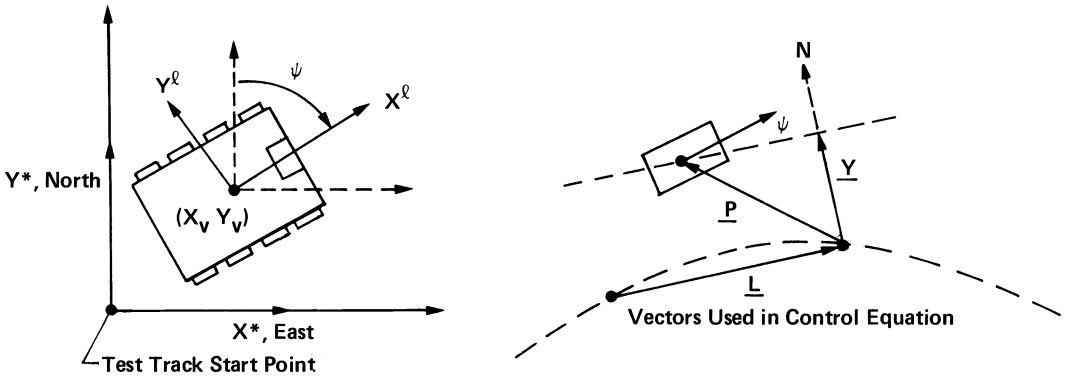


Figure 15. Fixed and local (vehicle) coordinate systems.

Reasoning--The reasoning subsystem operates in either a vision-based mode where the scene model is processed to generate the trajectory or a vision-override mode where map data are used. These two modes of operation are described in the following paragraphs.

Vision data processing--The vision data processing begins with evaluation of the vision scene model to get an overall confidence rating. The overall confidence is based on the confidence supplied with the vision scene model and how well the new scene model points fit the previous fitting curves. The fitting curves that are used to produce smooth trajectories consist of a sequence of three quadratics in terms of the local reference

$$y = a_i x^2 + b_i x + c_i \quad , \; i = 0, 1, 2.$$

To evaluate how well the new data agree with the previous curve, the first few scene model points are converted into the local coordinates and the offset error E_j between these points and the previous fitting curve is computed as (Fig. 16)

$$E_j = Y_j - F(X_j)$$

where $F(X_j)$ is determined from the appropriate quadratic. The offset error E_j is then compensated based on the vision confidence for the point (C_j - a value in the range 0 to 100) giving the compensated error CE_j

$$CE_j = \min \left(1.0, \frac{E_j * C_j}{E_{max}}\right).$$

In this manner, points for which the vision confidence is low will not affect the navigator confidence as significantly as points for which vision confidence is high. In this equation, E_{max} is a scaling factor used to adjust the navigator confidence range. A value of E_{max} = 140 was selected for the demonstration runs. With this value, an offset error of 1.4 m or more gives a compensated error of 1.0 (for C = 100), producing a navigator confidence of zero. The navigator confidence level NC is then computed as

$$NC = 1.0 - \text{avg}(CE_j).$$

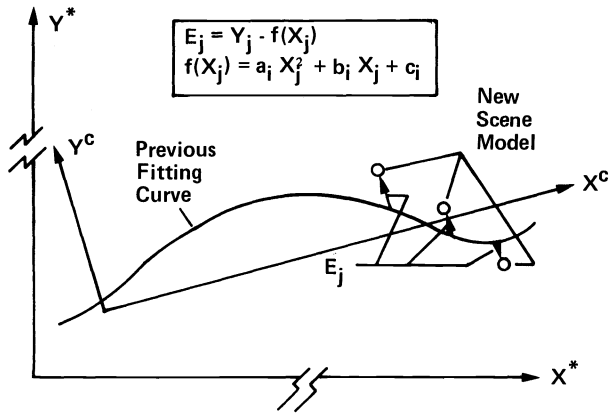


Figure 16. Scene model evaluation.

This value lies in the range from zero to one and is the cumulative representation of how well the new scene points fit the previous trajectory-smoothing curve (1.0 indicates perfect agreement). The total confidence for the scene model is obtained by multiplying this navigator confidence by the average vision confidence

$$TC = NC * avg\ (C_j).$$

If the final confidence rating TC is above a minimum threshold, the scene model is processed. Otherwise, the scene model is discarded and the navigator waits for the next activating semaphore. A threshold of 50 was used for the demonstration and proved acceptable in that only the most questionable scene models were discarded.

Once the scene model has been evaluated and judged fit for processing, the curve data to be used in generating the smooth fitting curves are determined. Because the closest scene model points provided by the vision system are generally about 7 m out from the front of the vehicle, curve data from the previous trajectory must be combined with the current scene model data to generate the reference trajectory directly ahead of the current vehicle position. These old curve data are first transformed into the new local coordinate system as shown in Figure 17. The figure shows the data from two successive scene models. The circled data points are the points that would be used in generating the new trajectory-fitting curves. Quadratic functions were employed for curve fitting because they are the lowest order functions that allow continuity of position and slope to be maintained and because the resulting optimization algorithm leads to a linear set of equations that can be readily solved. The number of quadratics was selected as three to provide a tractable solution with enough parameters to produce a reasonable fit. The fitting curve is defined by

$$a_0(X-X_0)^2 + b_0(X-X_0) + c_0 \ ; \ X_0 \leq X \leq X_1$$

$$Y = f(X) = a_1(X-X_1)^2 + b_1(X-X_1) + c_1 \ ; \ X_1 \leq X \leq X_2$$

$$a_2(X-X_2)^2 + b_2(X-X_2) + c_2 \ ; \ X_2 \leq X \leq X_3.$$

The coefficients a_i, b_i and c_i are computed so the continuity conditions are satisfied and the distances to the curve data points (X(k), Y(k)) are minimized. Coefficients c_0 and b_0 are determined by the initial position and slope constraints.

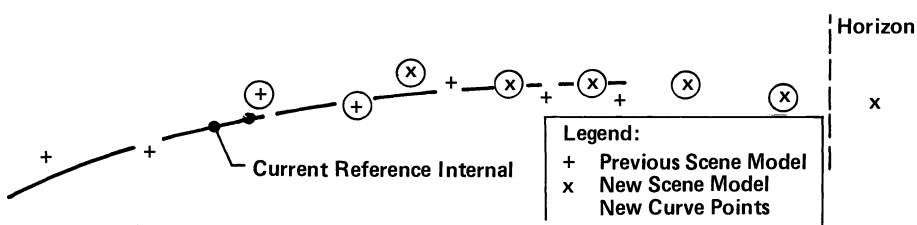


Figure 17. Selection of scene data points for curve-fitting.

The next step in scene processing involves correlating the scene model supplied by vision with the a priori map of the test track. By maintaining this correlation, the trajectory generation can transition smoothly between vision-supplied scene models and map data. Furthermore, because of the LNS errors, the vehicle could be off the road by the end of the 1-km track if it had to rely solely on map data and LNS position values.

For each map point k, a lateral offset L_k between the point and the scene-fitting curve is computed (Fig. 18). The average L of these lateral offsets is found and then converted to a northing and easting update in the fixed reference

$$\begin{bmatrix} \text{update\_east} \\ \text{update\_north} \end{bmatrix} = [T_C] \begin{bmatrix} 0 \\ L \end{bmatrix} .$$

These updates are added to the current position state indicators and to the local curve reference origin (X_C, Y_C) to provide the map correlation as shown in Figure 18.

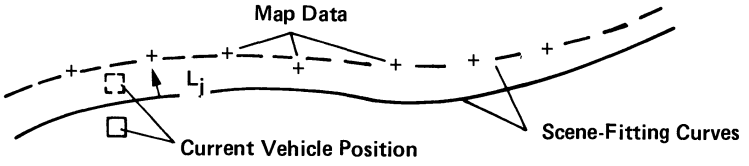


Figure 18. Vision update.

The trajectory-fitting curves are then discretized into short, straightline intervals that the pilot uses as the reference trajectory. Each reference point is gathered first in the local curve reference by evaluating the quadratic fitting function for various values of X. The X-spacing between successive reference points is fixed. Initially, 1-m spacing was used but this produced rough behavior in the curves so the spacing was reduced to 0.25 m for the demonstration runs. In future implementations, the interval spacing will vary depending on the trajectory curvature and the vehicle speed. Once a new reference trajectory is generated, it is recorded and a flag is set to let the pilot software know that a new trajectory is available.

Vision override--The vision override executive generates an emulated scene model for the current vehicle position and transforms this emulated scene model into the curve data usable by the navigator. The emulated scene model is formed by extracting appropriate points from the map database and converting these to a local scene coordinate system. The map database consists of a sequential list of road centerpoints spaced approximately 1 m apart. The set of 15 points are specified in terms of the fixed reference. This produces a set of curve data in local coordinates from which the next reference trajectory is generated.

Pilot software--Figures 14 and 15 illustrate the pilot process flow and vectors used in the pilot computations. The vector $\underline{P}$ is a vector from the destination point of the current reference interval to the current location of the vehicle, and $\underline{L}$ is a vector from the start to the end of the current reference interval. Whenever the scalar product of these vectors is greater than zero ($\underline{P} * \underline{L} \quad 0$), the vehicle has passed the end of the current reference interval and vectors for the next reference interval are determined.

The reference heading vector $\underline{L}$ is first computed. If k is the index of the current interval, it is incremented (k = k + 1) and $\underline{L}$ is computed as

$$L = \begin{bmatrix} L^x \\ L^y \end{bmatrix} = \begin{bmatrix} X(k) - X(k-1) \\ Y(k) - y(k)-1) \end{bmatrix}.$$

All pilot computations are performed in terms of the fixed coordinate reference. A unit normal vector $\underline{N}$ is computed next

$$\underline{N} = \frac{1}{\sqrt{(L^x)^2 + (L^y)^2}} \begin{Bmatrix} -L^y \\ L^x \end{Bmatrix}$$

Finally, the reference heading angle ϕ is evaluated (in the range [0,2P3I])

$$\psi_r = \text{Tan}^{-1} \left(\frac{L^x}{L^y} \right)$$

These variables are then used in the steering and speed control algorithms.

$T = K_0 + Psi + K, Psi + K_2 L + K_3 L$

$Speed = K_5 * [(K_6 * target\ speed) - measured\ speed]$

where K_0, K_1, K_2, K_3, K_4, K_5, K_6, and K_7 are control law constants and T is delta torque about the center of gravity. These control equations were determined from extensive simulation and vehicle testing.[4]

To determine if the vehicle is approaching the end of the current trajectory segment, the interval index k is compared with the total interval count and a flag is set to indicate that a new trajectory segment is needed from the navigator. This flag notifies the navigator to switch to vision override and generate a new trajectory from map data. If the vehicle is at the end of the current trajectory segment, a "stop vehicle" flag is set to gently bring the vehicle to a stop to wait for a new trajectory to become available. If the trip goal has been reached, the "mission complete" flag is set so the software terminates execution.

Conclusions

The first road-following demonstration has been successful and technology development for the second demonstration to be held in November of 1985 is well under way. The technology required for these first two demonstrations required some research and development efforts but did not involve major breakthroughs in the state of the art. To achieve the 1986 and 1987 demonstrations, however, significant technology voids must be overcome. The most significant is the area of image understanding where by 1987 a system must be capable of modeling complex three-dimensional features in an unstructured natural environment. Moreover, the vision system must be reliable enough to process thousands of successive images to continually update this model. Another substantial void is in the area of processing architectures. As the vision algorithms become more robust, the processing requirements will increase dramatically.

Architectures are being developed to address individual elements of the problem. However, some combination of these architectures is required to support the flow of processing from image space through neighborhood operations, global image operations, symbolic processing, and servocontrol. Each of the elements has radically different processing requirements and integrating a single system architecture using processors for all of these elements will be a significant challenge.

Because of the magnitude of these technical voids, it is apparent that no single research organization is going to solve all of the issues. Furthermore, research efforts must be focused on the general long-range technical issues or the program will become shortsighted and will fail to develop the technologies required for the later demonstrations. Therefore success of the annual demonstrations hinges on a continuation of the close cooperation between all elements of the strategic computing program to rapidly transition technology from the research domain to the vehicle. To make this transition effective, the ALV vehicle has been designed as a testbed that will be operational as a research tool in early 1986. Experiments from the research communit will be actively sought at that time to evaluate the state of technology and to prove concepts in the real environment.

References

1. Strategic Computing - New-Generation Computing Technology: Strategic Plan for Its Development and Application to Critical Problems in Defense. Defense Advanced Research Projects Agency, October 1987.

2. Autonomous Land Vehicle - Long-Range Operation Concept Document. Martin Marietta Denver Aerospace September 1985.

3. Visual Navigation System for Autonomous Land Vehicles. A. Wakman, J. Le Moigne, L. Davis, E. Liang, and T. Siddalingaiah: Center for Automation Research, TR 139, July 1985.

4. The Autonomous Land Vehicle, First Annual Report. Martin Marietta Denver Aerospace, September 1985.