An Expert Vision System for Autonomous Land Vehicle Road Following

Sven J. Dickinson
Larry S. Davis

Center for Automation Research
University of Maryland
College Park, MD 20742

1 Abstract

A production system model of problem solving is applied to the design of a vision system by which an autonomous land vehicle (ALV) navigates roads. The ALV vision task consists of hypothesizing objects in a scene model and verifying these hypotheses using the vehicle's sensors. Object hypothesis generation is based on the local navigation task, an a priori road map, and the contents of the scene model. Verification of an object hypothesis involves directing the sensors toward the expected location of the object, collecting evidence in support of the object, and reasoning about the evidence. Constructing the scene model consists of building a semantic network of object frames exhibiting component, spatial, and inheritance relationships. The control structure is provided by a set of communicating production systems implementing a structured blackboard; each production system contains rules for defining the attributes of a particular class of object frame. The combination of production-system and object-oriented programming techniques results in a flexible control structure able to accommodate new object classes, reasoning strategies, vehicle sensors, and image analysis techniques.

2 Introduction

The development of an Autonomous Land Vehicle (ALV) involves the development of computer vision techniques by which a vehicle can autonomously navigate itself through the environment. Although the goals for the ALV are broad, including both on- and off-road navigation, the work presented here is primarily concerned with the road following task. From images acquired from a camera, the ALV vision system constructs a model of the environment; this scene model contains the objects visually identified by the ALV. Based on this collection of objects, the vehicle plans a course and moves through the environment.

For the road following task, the scene model contains either objects that represent the road or objects from which the location of the road can be deduced. Obviously, the direct detection of a patch of road would be most useful; however, in the event that the ALV vision system cannot directly identify the road, the detection of other objects may suggest the location of the road. For example, telephone poles and ditches often run parallel to the road; their presence may thus provide clues as to its location and direction. In certain cases, major landmarks contained in a road map such as buildings may be used to infer the road location; however, such information is more useful in registering the vehicle to some absolute location.

Construction of the scene model is complex. The selection of which object to track depends on the navigation goals of the ALV, the history of object tracking, the contents of the scene model, and information from the road map. Verifying the existence of an object requires directing vehicle sensors towards the object, fusing data from different sensors, and selecting algorithms for image analysis. Methods for performing all these tasks are continually evolving as the road following task becomes better understood. New objects must be tracked by the ALV, new sensors are available to track objects, and new image processing techniques are identified for sensor image feature extraction. The successful evolution of an ALV vision system hinges on the ability of its control structure and knowledge representation schemes to accommodate these changes.

We propose the design of a system for constructing an ALV scene model, offering a flexible control structure able to accommodate new strategies for object tracking, sensor selection, and feature extraction. The goal of the system is to provide a flexible tool for the development of ALV road following software. The design is based on concepts described in [Hanson & Riseman], but offers a unique implementation based on a set of communicating production systems.

3 System Overview

The task of building a scene model for the ALV consists of two major subtasks:

1. deciding what object to look for and where to look for it;
2. verifying that the object exists in the world.

These two functions are performed by the Scene Model Planner (the Planner) and Scene Model Verifier (the Verifier), respectively; together, they form the Scene Model Builder (the Builder). The data flow diagram for the Builder is presented in Figure 1. The Planner, in addition to interpreting and updating the scene model, is aware of the local navigation task and initiates queries to the a priori road map. The Verifier controls the movement of the sensors and acquires the sensor image data. In a hypothesize-and-test paradigm, the Planner sends object hypotheses to the Verifier, while the Verifier returns verified objects to the Planner. The dataflow of the Builder proceeds as follows. The Planner first determines the scene model requirements of the local navigation task; for example, following a straight road requires that the left and right road boundaries be contained in the scene model. Next, the Planner looks at the road map and the partial scene model and decides what objects may be useful in locating the road; for example, it might decide that a road patch, a ditch, or even a row of telephone poles is sufficient to define a road boundary. The Planner then decides the type and expected location of the object to be tracked, hypothesizes the object, and passes the hypothesis to the Verifier.

The Verifier attempts to verify the hypothesis by directing the vehicle's image sensors towards the expected location of the object. The object is then located in the sensor images and its image location is mapped to a 3-D location based on a fixed point of reference. The confidence with which the object is found becomes a measure of its verifiability. Once the confidence is defined, the hypothesis is returned to the Planner for inspection. If the object is deemed sufficiently verified, it is added to the scene model. Otherwise, the Planner determines the next course of action; for example, the object may be hypothesized in a different location, or a new object hypothesized.

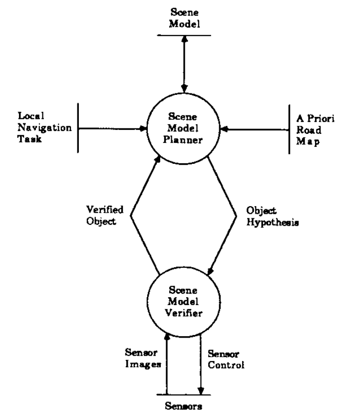

Figure 1: The Scene Model Builder Dataflow

4 Modeling World Objects

Objects in the ALV scene model exhibit the following relationships:

- *Component Relationships.* For example, an intersection is made up of four connecting roads, stop lights, etc. These component objects, in turn, can be decomposed into their component objects, e.g. a road may be defined as a pair of left and right segments, each edge representing the border between road and shoulder, or shoulder and background. Primitive objects such as lines and surfaces extracted from an image cannot be decomposed.

- *Spatial Relationships.* For example, telephone poles are often located near the road and run parallel to the road.

- *Property Inheritance.* For example, a three-dimensional line segment may be defined by a pair of endpoints. The edge separating the road surface from the shoulder is a specialization of a three-dimensional line segment; thus, in addition to its properties specific to a road edge, it inherits the endpoint properties of the three-dimensional line segment.

To accommodate these relationships, frames have been chosen to model objects [Minsky]. The following sections describe the object frames defined in the system; the frame attributes are discussed in more detail in [Dickinson & Davis].

A planar ribbon is defined as a pair of facing and parallel three-dimensional line segments. A road patch is a specialization of a planar ribbon, whose three-dimensional line segments represent the left and right features of the road. Thus, a road patch frame inherits the attributes of a planar ribbon. Road patches are oriented and may be connected together to form a piece of road; the front of one road patch may be connected to the back of another. The orientation of a road patch is based on the assigned orientation of the initial road patch; typically, the back end of the initial road patch is closest to the ALV, while the front end is furthest from the ALV. The left and right features, i.e. three-dimensional segments, are oriented looking from the back to the front of the road patch. Figure 2 depicts the vehicle with respect to a series of road patches.

A world segment is defined as a three-dimensional line segment. A road patch segment is a specialization of a world segment representing a road feature, i.e. the boundary between the road surface and the shoulder surface or the boundary between the shoulder surface and the vegetation or background. Thus a road patch segment frame inherits the attributes of a world segment. Road patch segments are oriented and may be connected together to form a continuous linear feature; the front of one road patch segment may be connected to the back of another. The orientation of a road patch segment is based on the orientation of its parent road patch.

A camera segment is defined as a two-dimensional line segment extracted from a camera image. A road patch camera segment is a specialization of a camera segment representing the two-dimensional projection of a three-dimensional road feature. Thus a road patch camera segment frame inherits the attributes of a camera segment frame. Road patch camera segments are oriented and may be connected together to form a continuous two-dimensional linear feature; the front of one road patch camera segment may be connected to the back of another. The orientation of a road patch camera segment is based on the orientation of its parent road patch segment.

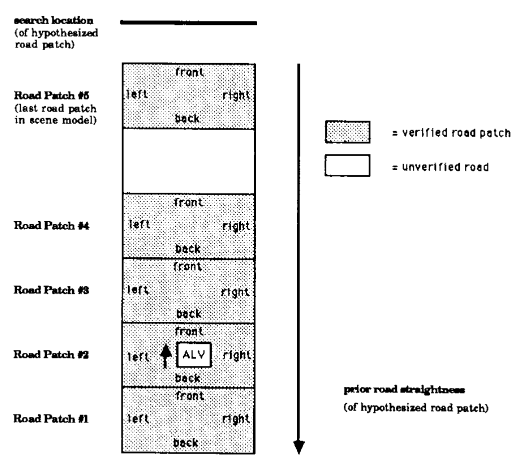

Figure 2: A Series of Road Patches

5 The Scene Model Planner

The scene model is central to the ALV vision system. It is accessed by the Planner in determining what object to search for, and by the ALV navigator when plotting a course through the scene model objects. Based on the local navigation task, the a priori map, and the scene model, the Planner decides what type of object to track and verify. To simplify the initial implementation of the Planner, we have assumed a constant local navigation task of following the road ad infinitum, and an a priori road map which contains the approximate locations of intersecting roads along with an approximate road width. Hence, the production system consistently selects a road patch for verification by instantiating a road patch frame whose attributes are undefined.

The next task of the Planner is to choose the search location of the road patch hypothesis. The production system first queries the scene model for the directional history of the road. If the direction has varied erratically, then the Planner's confidence as to the location of the road patch is low. Imposing the constraint that the hypothesized road patch must be connected to the last road patch in the scene model improves the likelihood of verifying the road patch hypothesis. Hence, the Planner, implemented by a production system, defines the search strategy as "connected"; the search location is defined to be the leading edge of the connected road patch. If the road has been found to be recently straight for, say, at least 10 meters, then the Planner assumes that the road beyond the scene model is also straight. In this case the search strategy is defined to be "disconnected" and the search location is extrapolated a distance of 10 meters from the end of the scene model. If successful, the scene model can be built more rapidly in this fashion, ultimately resulting in higher vehicle speeds.

The Planner is now ready to send the road patch hypothesis to the Verifier, where evidence is gathered in support of it. Once complete, the Verifier returns the hypothesis to the Planner; all the attributes in the road patch frame are now defined. If the evidence is deemed acceptable by the Planner, it will add the verified object to the scene model. However, if the evidence is considered unacceptable, several options are available to the Planner:

1. hypothesize the object at a different location;
2. hypothesize a different object;
3. retain the verified components of the unverified object.

Currently, only option 1 is implemented and proceeds as follows. If the unverified road patch hypothesis is disconnected, i.e. the Planner ventured out beyond the end of the scene model to hypothesize the road patch, the hypothesis is abandoned and a connected road patch is hypothesized back at the end of the scene model. If the unverified road patch hypothesis is connected, the Planner aborts the road following task. Figure 3 summarizes the actions taken by the Planner.

6 The Scene Model Verifier

The role of the Verifier is to receive an object hypothesis from the Planner, collect evidence in support of the object, and return the verified object to the Planner. More specifically, when the Verifier receives an object hypothesis in the form of a sparsely defined frame, it proceeds to fill in the empty attributes; if the object

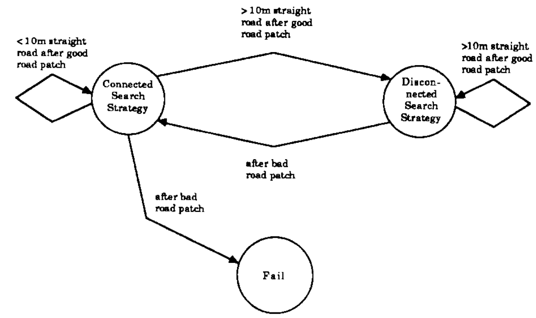

Figure 3: The Scene Model Planner States

has component parts, e.g. a road patch segment, the Verifier must create and define these frames. To accomplish this task, a separate blackboard has been assigned to each class of object.

When an object hypothesis is posted on the blackboard corresponding to its class, knowledge sources are activated to fill the empty attributes of the hypothesis. As in the case of the Planner, each blackboard is implemented as a frame, providing a set of attributes and inheriting the capabilities of a production system. The attributes provide links to other blackboard frames and system modules, e.g. vehicle pilot and image processor. The production system rules control the activation of knowledge sources, i.e. when the left-hand side of a rule matches the contents of the factual database, the right-hand side activates a knowledge source. Ties are resolved by the conflict resolution strategy.

Blackboard frames, like object frames, possess both component and inheritance relationships; spatial relationships are undefined for blackboard frames. For example, the road patch blackboard has an attribute pointing to the road patch segment blackboard; although there are two component road patch segments for each road patch, there is only one road patch segment blackboard on which every instance of a road patch segment object is posted. When an object blackboard is instantiated, it may inherit the attributes and rules of other object blackboards.

When the Planner hypothesizes an object, the Verifier invokes a top-down approach to verify the object. In Figure 4, this approach has been applied to the verification of a road patch hypothesis. Once the Planner creates the road patch hypothesis, it posts it on the road patch blackboard (a specialization of a planar ribbon blackboard). The rules belonging to the road patch blackboard, acting as daemons, invoke knowledge sources to define the attributes of the now sparsely defined road patch hypothesis. The rule antecedents ensure that the attributes are defined in a specific order. When rules fire to define the left road patch segment component, the activated knowledge source create a road patch segment object hypothesis, and posts it on the road patch segment blackboard. At this point, control is transferred to the road patch segment blackboard while the road patch blackboard is put to sleep.

Responding to a new object hypothesis on their blackboard, the rules belonging to the road patch segment blackboard proceed to define the attributes of the road patch segment object. When rules fire to define the camera segment attribute, the activated knowledge source creates a road patch camera segment, defines a subset of its attributes, and posts it on the road patch segment blackboard. Control is transferred to the road patch camera segment blackboard.

The rules at the road patch camera segment blackboard define an image search area and an image processing line detector, and proceed to extract a segment from the image. Once the road patch camera segment hypothesis is completely defined (including a confidence measure), control is passed back up the hierarchy. The process is repeated for the right hand segment.

This system of communicating blackboards offers many advantages to the system builder. Each modular blackboard controls the definition of a single object class; as new classes are created, new blackboards are defined. If the definition of a class is changed, i.e. attributes are added or deleted, new sets of rules are added or deleted. Since rules map to single attributes, the alteration of one set of rules will have little or no impact on rules corresponding to other attributes. Within each blackboard, the inherent advantages of a rule-based system are clear. Rule-based activation of knowledge sources provides a data-driven, flexible control structure, while English-like rules provide readability and support maintainability.

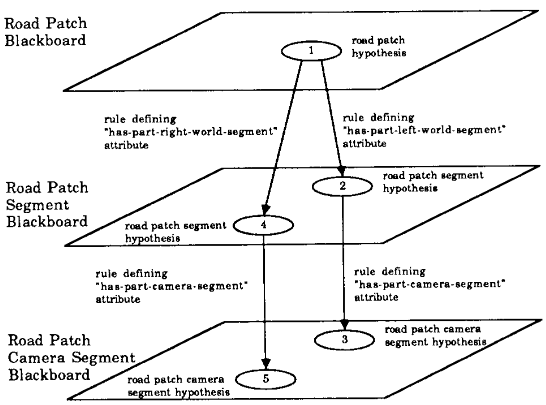

Figure 4: Top-Down Hypothesis Verification

7 Experimental Results

In this section we demonstrate the system on two road sequences; the road images were taken from the Martin Marietta ALV test track in Denver, CO. The current implementation runs in the Maryland Franz Lisp environment [Allen et al.], under UNIX[1] 4.3BSD on a VAX[2] 11/785. As described earlier, all system modules are frames implemented using the Maryland Franz Flavors package [Wood]; the production system frames inherited by the Planner and Verifier blackboards are implemented using YAPS [Allen]. YAPS is an antecedent-driven production system similar to OPS5 [Forgy], but offering more flexibility. Functions bound to the frames are implemented in Lisp; C routines are called from the Lisp environment for numerically-intensive processing. All image display functions are provided by a Vicom image processor.

The first sequence is shown in Figure 5 and demonstrates the construction of a scene model containing a curved road. At the bottom of the image, the initial search windows are placed according to the a priori points on the side of the road; the search windows are indicated by the rectangular boxes which contain the extracted line segments. From then on, the road patch connected search strategy is repeatedly invoked to verify successive connected road patches. Following the insertion of the eighth road patch into the scene model, over 10 meters of straight road have been accumulated. In this case, the disconnected search strategy is invoked resulting in a search location 10 meters beyond the end of the scene model. However, because the vehicle could not predict the upcoming curve in the road, the predicted search location is off the road, ultimately yielding a road patch with very poor total confidence (due to lack of parallelism and poor width). The Planner aborts the disconnected search strategy and invokes the connected search strategy from the previously verified road patch in the scene model. As a result, the curve is successfully navigated, as shown in Figure 6.

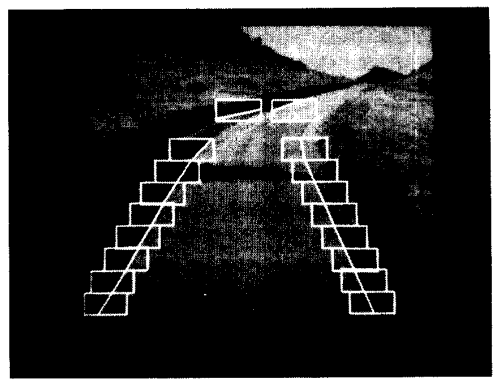

Figure 5: Tracking A Curved Road - Frame 1

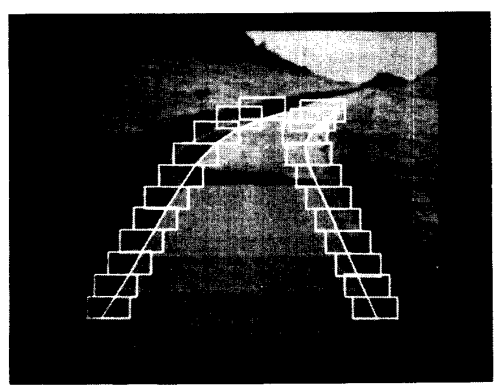

Figure 6: Tracking A Curved Road - Frame 2

[1]UNIX is a trademark of Bell Laboratories.

[2]VAX is a trademark of Digital Equipment Corporation.

8 System Evolution

In Section 7 we demonstrated two search strategies invoked by the Planner to verify a road patch. As these strategies are rather simplistic, we expect that they will evolve with time, and have designed the control structure to accommodate such evolution. In this section, we demonstrate the flexibility of the control structure by exploring the effects of altering the search strategy used by the Planner.

The current search strategy dictates that the Planner abort the disconnected search strategy in the event of an unverifiable road patch hypothesis. Rather than returning to the connected search strategy, we might try to relax our projected search location and re-hypothesize the road patch halfway between the unverified road patch and the last verified road patch in the scene model. We replace our old rule with the following rule:

```
(defp disconnected-to-halfway-retry
     (hypothesized object -object)
     (goal (process road patch hypothesis -object))
 test (eq (<- -object 'search-strategy) 'disconnected)
     (cond ((not (null (<- -object 'total-confidence)))
            (< (<- -object 'total-confidence)
               MIN-ROAD-PATCH-CONFIDENCE)))
 -->
     (<- -object 'set-search-location
        (<- -object 'set-search-location (halfway
           (<- (<- (<- *yaps-db* 'scene-model)
                ':retrieve-most-recent-road-patch)
               'search-location)
           (<- -object 'search-location)))))
```

In this case, the *search location* is calculated by a function finding the midpoint between the last road patch in the scene model, and the unverified road patch. Applying this strategy to the unverifiable road patch in Figure 5, we obtain the results depicted in Figure 7.

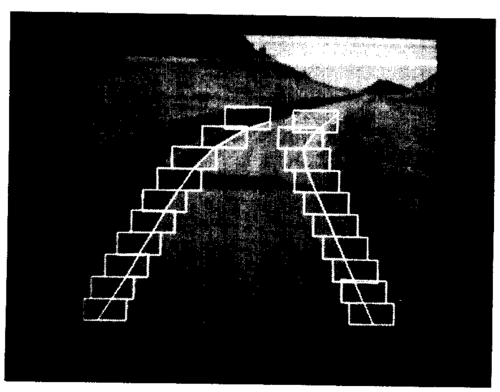

Figure 7: Dynamically Decreasing Road Projection

9 Related Work

The decomposition of an object both by component and by level of abstraction, and the construction of hierarchical frame networks, bear close resemblance to techniques used in the VISIONS system [Hanson & Riseman]. In that system, the long term memory (LTM) contains a priori visual knowledge of the world, while the short term memory (STM) represents the interpretation of the scene. Both the LTM and STM are structured as a hierarchy of levels of representation defining the levels of object abstraction. The control strategy first decides which partial model (frame network) to focus on, expands (hypothesizes) a node, and finally verifies the node. Although originally defined for outdoor house scenes, this work has been extended to the road following task [Arkin et al.].

[Lawton et al.] describe a system resembling the VISIONS system. The short term memory acts as a dynamic scratchpad for the vision system, containing object hypotheses, incoming imagery, and the results of feature extraction. When hypotheses accumulate sufficient evidence, they are moved to the long term memory, which includes a priori terrain representations. The control structure provides both top-down and bottom-up hypothesis instantiation over the network hierarchies. Although hypothesis instantiation in the above systems is both top-down and bottom-up, the entire sensor image is processed to initialize the short term memory with image features; local processing in our system is based on [Le Moigne et al.].

[Smith & Strat] describe an information manager that is the core of a sensor-based autonomous system. A centralized knowledge database is proposed, accessible to a community of independent asynchronous processes. The representation scheme organizes data tokens in both an octree and a semantic network thus supporting both spatial and semantic queries. The independent asynchronous processes can be activated by daemons embedded in the database or by procedure call.

10 Conclusions

The system described in this report provides a flexible architecture for constructing an ALV scene model. The representation of objects as networks of frames offers a natural grouping of knowledge; the multiple layers of abstraction facilitate the addition of new sensor features in support of existing world objects. Construction of the frame network is provided by a set of modular blackboards providing top-down instantiation of the frames in the network. Each blackboard, implemented by a production system, is an "expert" in defining a particular class of frame; the English-like rules governing the invocation of knowledge sources are easy to understand, and narrow the gap between control specification and implementation. From a system maintenance standpoint, all object frames, blackboards, production system tools, and object oriented programming tools are off-the-shelf; these facilities are documented, tested, and readily accessible. The implementation languages supporting the system cover the needs of the programmer; YAPS offers high-level encoding of control strategy, Lisp provides symbolic manipulation, C speeds up numerical processing, and Flavors facilitates inter-object communication.

The system is currently being expanded to support new planning and verification strategies. The Planner is being supplemented with strategies for road following in the event that a connected road patch cannot be verified. This includes proceeding past an unverified road patch provided that it contains a verified component, and invoking an exhaustive search for road patches in a given area; the latter strategy will be accomplished using bottom-up verification in which road patch camera segments posted at lower levels generate instances of road patches at upper levels. This integration of top-down and bottom-up verification will remove some of the burden placed on the Planner of accurately predicting the location of an object.

11 Acknowledgements

Many people deserve thanks for their contribution to this work. In particular, the authors wish to thank Jacqueline Le Moigne and Rand Waltzman for insightful discussions at the early stages of the project. David Harwood and Sergio Antoy offered helpful suggestions during the design phase, while Todd Kushner and T.K. Siddalingaiah provided hardware support in the implementation. The authors would also like to thank Suzanne Stevenson and Azriel Rosenfeld for their excellent comments on earlier drafts of this paper.

References

[Allen et al.] Allen, Elizabeth M., Randall H. Trigg, and Richard J. Wood, "The Maryland Artificial Intelligence Group Franz Lisp Environment, Variation 2", Technical Report TR-1226, Department of Computer Science, University of Maryland, November 1983.

[Allen] Allen, Elizabeth M., "YAPS: Yet Another Production System", Technical Report TR-1146, Department of Computer Science, University of Maryland, December 1983.

[Arkin et al.] Arkin, Ronald C., Edward M. Riseman, and Allen R. Hanson, "AuRA: An Architecture for Vision-Based Robot Navigation", Proceedings: 1987 DARPA Image Understanding Workshop, pp. 417-431, Los Angeles, CA, February 1987.

[Dickinson & Davis] Dickinson, Sven J. and Larry S. Davis, "An Expert Vision System for Autonomous Land Vehicle Road Following", CAR-TR-330, University of Maryland, October 1987.

[Duda & Hart] Duda, Richard O. and Peter E. Hart, Technical Report CAR-TR-330, Center for Automation Research, University of Maryland, October 1987. "Pattern Classification and Scene Analysis", pp. 393-396, John Wiley & Sons, Inc., 1973.

[Forgy] Forgy, C. L., "OPS5 User's Manual", Technical Report CMU-CS-81-135, Department of Computer Science, Carnegie-Mellon University, July 1981.

[Hanson & Riseman] Hanson, Allen R. and Edward M. Riseman, "VISIONS: A Computer System for Interpreting Scenes", in Hanson, Allen R. and Edward M. Riseman, (eds.), "Computer Vision Systems", Academic Press, Inc., 1978.

[Lawton et al.] Lawton, Daryl T., Tod S. Levitt, Christopher C. McConnell, Philip C. Nelson, and Jay Glicksman, "Environmental Modeling and Recognition for an Autonomous Land Vehicle", Proceedings: 1987 DARPA Image Understanding Workshop, pp. 107-121, Los Angeles, CA, February 1987.

[Le Moigne et al.] Le Moigne, Jacqueline, Allen M. Waxman, Babu Srinivasan, and Matti Pietikainen, "Image Processing for Visual Navigation of Roadways", Technical Report CAR-TR-138, Center for Automation Research, University of Maryland, July 1985.

[Minsky] Minsky, Marvin, "A Framework for Representing Knowledge", in Patrick Henry Winston (ed.), "The Psychology of Computer Vision", McGraw-Hill, Inc., 1975.

[Smith & Strat] Smith, Grahame B., and Thomas M. Strat, "Information Management in a Sensor-based Autonomous System", Proceedings: 1987 DARPA Image Understanding Workshop, pp. 170-177, Los Angeles, CA, February 1987.

[Srinivasan] Srinivasan, Babu, "Image Processing Algorithms for Navigating Roadways", MS Thesis, University of Maryland, 1984.

[Wood] Wood, Richard J., "Franz Flavors: An Implementation of Abstract Data Types in an Applicative Language", Technical Report TR-1174, Department of Computer Science, University of Maryland, June 1982.