SCARF: A Color Vision System that Tracks Roads and Intersections

Jill D. Crisman, *Member, IEEE,* and Charles E. Thorpe, *Member, IEEE*

***Abstract*—SCARF is a color vision system that recognizes difficult roads and intersections. It has been integrated into several navigation systems that drive a robot vehicle, the Navlab, on a variety of roads in many different weather conditions. SCARF recognizes roads that have degraded surfaces and edges with no lane markings in difficult shadow conditions. It also recognizes intersections with or without predictions from the navigation system. This is the first system that detects intersections in images without *a priori* knowledge of the intersection shape and location. SCARF uses Bayesian classification, a standard pattern recognition technique, to determine a road-surface likelihood for each pixel in a reduced color image. It then evaluates a number of road and intersection candidates by matching an ideal road-surface likelihood image with the results from the Bayesian classification. The best matching candidate is passed to a path-planning system that navigates the robot vehicle on the road or intersection. This paper describes the SCARF system in detail, presents results on a variety of images, and discusses the Navlab test runs using SCARF.**

I. Introduction

SCARF is a color vision system that detects roads and intersections for an intelligent mobile robot. To navigate a mobile robot, a navigation system consists of perception systems to sense the environment, path-planning systems that decide on a pathway, and vehicle control systems that actuate the motion of the robot. For road navigation, at least one perception system must sense the location of the roads, and the path-planning system must generate paths that keep the robot in the proper lane while avoiding any obstacles. This work focuses on the detection of roads and intersections in color images for robot navigation systems.

Current road-detection systems perceive certain types of roads under a limited conditions. Many systems detect highway lane markings but cannot perceive rural roads that do not have painted lines. Some systems rely on detecting the road edges but will often fail if these edges are degraded and broken. Some systems rely on having a map to describe the shape of the road and to predict when and where new roads will appear. Many road-detection systems concentrate on finding single roads in images and ignore intersections. This specialization is due to the real-time nature of the task, often trading more general capabilities for vehicle speed. In general, road-detection systems are specialized for specific scenarios.

This research focuses on detecting the most difficult road scenario without giving up the capability of perceiving less complex situations. In particular, we are able to navigate a real robot vehicle on unstructured roads and intersections that may have:

- no lane or edge lines painted on the road surface,
- degraded road edges,
- road surface scars,
- strong shadow conditions, and
- no map information.

These roads lack a clearly defined feature (such as road edge lines), which can be used for their detection, making these roads the most difficult to detect.

Specifically, this paper describes our unstructured road-detection system, Supervised Classification Applied to Road Following (SCARF). We first highlight, in Section II, the capabilities and limitations of various road-detection systems that can detect intersections. In Section III, we present the implementation details of SCARF. We show, in Section IV, results of this algorithm running on several unstructured road and intersection sequences. Finally, in Section V, we discuss this system and its contributions and limitations.

II. Overview of Intersection Detection Systems

There are many different road-detection algorithms, of which only a few can also detect intersections. Many road-detection methods rely on the presence of specific structured features such as road edge lines or lane markings. These include the General Motors Lanelok System [10], [11], the VaMoRs System [5], [6], [16], [26], the Yet Another Road Follower (YARF) System [12], and the Intelligent Car System [22]. These systems usually have very fast processing speeds and are very well suited for navigation on structured roads. However, since these systems rely on certain features of the road, they are not directly applicable to unstructured roads.

A unique unstructured road navigation system, Autonomous Land Vehicle In a Neural Network (ALVINN) [18], does not explicitly detect the roads in the camera images. ALVINN is a neural network that produces steering angles when shown images of roads. With this input/output model, ALVINN does not recognize intersections nor produce a confidence measure of its performance.

Other structured road-detection systems explicily detect either road edges or surfaces. One approach locates the road edges in an image by examining gradient images [15], [25].

Manuscript received June 17, 1991; revised March 6, 1992. This work is supported by DARPA under Road Following Contracts DACA 76-89-C-0014 and DACA 76-85-C-0003. Portions of this paper were presented at SPIE Mobile Robots V, Cambridge, MA, November 1990.

J. D. Crisman is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115.

Charles E. Thorpe is with the Robotics Insititute, Carnegie Mellon University, Pittsburgh, PA 15213.

IEEE Log Number 9205080.

These systems can process the images quickly but can often fail if the road edges are degraded, partially occluded, or if there are shadows present. Other systems identify road surfaces using a histogram and threshold approach [9], [13], [23], [24]. These systems are robust on certain unstructured roads but can run into difficulty in shadows or degraded road-surface conditions. The Shadow Boxing System [23] is a two-dimensional classification approach that is much more successful at handling shadow conditions, but it experiences difficulties if the road surface is degraded or partially covered by leaves.

Three unstructured road systems can also detect intersections: the Carnegie Mellon Sidewalk System [9], the FMC System [14], and the University of Maryland System [25]. The Sidewalk system uses a histogram and threshold approach to label pixels in the image as road and off-road. The road pixels are collected into polygons. The edges of the polygon are matched against road edges in a map to determine the location of the intersection. The FMC System first locates the candidate road edges in gradient images and uses constraints to determine the road location. It then smooths over breaks in the road edges to allow for lost edges. This system can sometimes detect edges of intersection branches in the segmentation, but the algorithm assumes that all breaks in road edges are a result of broken road edges so the branches were ignored when fitting the model. The University of Maryland System extracts road boundary points, hypothesizes which points correspond to road edges in a map, and then searches for the best match of the points to the intersection model. In each of these systems, a map is used to predict the location and shape of the intersection.

These three systems all detect intersections, but they rely on clean intersection images and a good prediction of the intersection shape. The intersection predictions often come from a map stored inside the navigation system. SCARF does not rely on map information or *a priori* shape information. Instead, it looks for all possible intersections that appear in the image based on what the system has seen in previous images.

III. SCARF

SCARF serves as a road and intersection detection module for a mobile-robot navigation system. A color camera is mounted on the top of our test vehicle, the Navlab [7], and is tilted toward the ground so that approximately half of the area of the image views the road. After SCARF finds the roads in the image, the center line of the roads are backprojected onto the ground plane. The road locations (and possibly obstacle locations from another preception module) are sent to a planning algorithm. The position and velocity of the vehicle is detected with optical encoders and with an inertial navigation system. Using this estimated motion, the new locations of roads and intersections can be predicted from prior predictions as SCARF digitizes a new image. This prediction is input to SCARF and used to locate sample road and off-road pixels. In most of our test runs, obstacle detection was not used, and a pure-pursuit path-planning algorithm directed the vehicle to stay on the detected roads. To simplify the system, the location

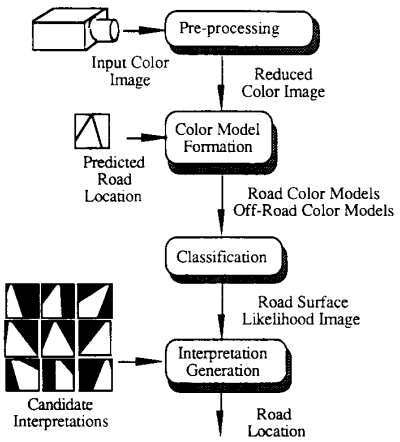

Fig. 1. SCARF block diagram.

of the previous road or intersection in the image was used as a prediction for the next image. The initial road location is selected by the user or detected by an unsupervised system called the UNSCARF algorithm [3].

SCARF consists of two main functions: road-surface detection and interpretation generation. The road-surface detection takes the input image and a prediction of the location of the road and produces a road-surface likelihood image where each pixel contains the likelihood that it belongs to the road surface. The interpretation generation module matches a set of road and intersection models to the road-surface likelihood image to determine the most likely interpretation. This interpretation, backprojected to the ground plane, and its confidence are then used by the navigation system to guide the vehicle. Each of the steps of the SCARF algorithm are discussed in the following sections, concluding with results of SCARF processing a number of road-image sequences.

A. Road-Surface Detection

A block diagram of road-surface detection is shown in Fig. 1. The preprocessing stage filters the input images to reduce their size and the noise in the images. The color model formulation module then uses the previous road or intersection descriptions projected onto the image to determine a set of Gaussian models for both the road and off-road colors. SCARF then compares the pixels in the reduced size images with the road and off-road samples to determine the pixel's likelihood of being part of the road surface, thus forming a road likelihood image. The interpretation generation (discussed in the following section) uses ideal models of road-surface likelihood images to select the most likely road or intersection in the image.

1) Preprocessing: This module takes the full-resolution input color image and produces a low-resolution color image. This reduces the amount of data for faster processing of the input image. The goal is to maintain information about the original colors of the image while eliminating some of the noise inherent in the imaging process. The reduction replaces a block of pixels in the original image with one pixel in the reduced image.

There are several methods for using the multiple pixel values in the high-resolution image to determine the low-resolution pixel value. Subsampling, averaging, and median filtering [1] have all been tried in SCARF. Subsampling picks one of the multiple pixels to be the low-resolution pixel value. This method is the fastest, but it is highly susceptible to noise, since a noise pixel can be selected. Another technique is to average the block of pixels to determine the resulting pixel value. This is almost as fast and is effective at reducing noise in the image. However, this also has the effect of blurring the data in the image. The final technique, median filtering, selects the median value of the multiple pixels to be the pixel in the low-resolution image. This is the most computationally expensive, but it is effective at reducing the image noise without blurring the data. All of these techniques have worked with SCARF, but we have empirically found that the best results are when the image reduction is done by a combination of subsampling and averaging. First the image is averaged over a 2 × 2 window and the resulting averaged image is then subsampled. By choosing our averaging window to be small compared to the distance between subsamples, the blurring effect of the averaging is reduced, and the resulting computation is kept relatively small.

Typically, the input color images are 480 × 512 pixels or 240 × 256 pixels of 3-band color data. This is reduced to 60 × 64 pixels or 30 × 32 pixels. This reduces the image data size by a factor of 16, 64, or 256.

2) Color Model Formulation: This module derives models for both road and off-road colors. Road and off-road are both represented by multiple Gaussian color models thereby allowing multiple colors to represent the road surface. These multiple models are formed by first determining sample regions in the image. Using the pixels from the road and off-road sample regions, SCARF clusters the samples into sets of similarly colored pixels. Then a Gaussian model is fit to each set of road and off-road sample sets. These models are then output to the classification module.

Regions of road samples and off-road samples are first selected. The sample region of road pixels is selected to lie well within the predicted location of the previously found road in the image. Only pixels lying on the ground plane (i.e., below the horizon) are considering during our region formation. Initially, all of the pixels that lie on the predicted road are assigned to the road sample region and all others are assigned to the off-road sample region. Next, all pixels that lie within a specified horizontal distance from the predicted road edges are removed from the road and off-road sample regions. This provides a margin for errors to allow for inaccurate knowledge of vehicle motion and inexact fitting of shape models. The pixels remaining in the road region form a road sample region $\{\boldsymbol{x}\}_{\text{road}}$, and the pixels remaining in the off-road regions are collected into an off-road sample region $\{\boldsymbol{x}\}_{\text{off-road}}$.

Next, the pixels in the sample regions are separated into sets having similar colors using a standard nearest mean clustering method [8]. We found that four color classes for each road and off-road gave us empirically the best results. All sample road-region pixels are first arbitrarily assigned to one of the four road color classes. The mean value of each class is then computed. Next, each sample pixel is reclassified into the class whose mean is closest to the sample pixel value. The "compute means" and "reclassify" loop is repeated until none of the pixels change their class. Typically this iteration converges rapidly in a few steps, so we compute a fixed number of iterations. An identical procedure is performed on the off-road samples to obtain off-road sample sets.

The road and off-road sample sets are then used to compute multiple-class Gaussian color models. A road class label w_r is assigned to each road sample set. Similarly, an off-road label w_o is assigned to each off-road sample set. Each class is modeled by the mean color $\boldsymbol{m}_i$ of a sample set, a covariance matrix $\boldsymbol{C}_i$ representing how the individual colors elements are interrelated, and the number of samples $\boldsymbol{N}_i$ used to form the model. This results in a set of road classes represented by a statistical model, and a set of off-road classes represented by their statistical model

$$\begin{aligned}\text{road} &\Leftarrow \{w_r | w_r \Leftarrow (\boldsymbol{m}_r, \boldsymbol{C}_r, N_r)\}\\ \text{off-road} &\Leftarrow \{w_o | w_o \Leftarrow (\boldsymbol{m}_o, \boldsymbol{C}_o, N_o)\}.\end{aligned}$$

The models are computed using standard statistical equations on the sample sets

$$\boldsymbol{m}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in w_i} \boldsymbol{x}_i$$

$$\boldsymbol{C}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in w_i} \boldsymbol{x}_i \boldsymbol{x}_i^T - \boldsymbol{m}_i \boldsymbol{m}_i^T.$$

3) Classification: This module takes the reduced color images and the color models and computes the probability that each pixel $\boldsymbol{x}$ in the image is a road-surface pixel, based on how well the color of the pixel matches the color model. Each pixel is assigned the value $P(\text{road}|\boldsymbol{x})$, the probability of road for the pixel value $\boldsymbol{x}$.

For each pixel in the reduced color image $\boldsymbol{x} = [\text{R G B}]^T$, we compute $P(\text{road}|\boldsymbol{x})$ using Bayes rule

$$P(\text{road}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\text{road})P(\text{road})}{P(\boldsymbol{x})}$$

where $P(\boldsymbol{x})$ is calculated by

$$P(\boldsymbol{x}) = P(\boldsymbol{x}|\text{road})\, P(\text{road}) + P(\boldsymbol{x}|\text{off-road})\, P(\text{off-road}).$$

The *a priori* probabilities $P(\text{road})$ and $P(\text{off-road})$ are computed as the percentage of samples used to form the road and off-road color models, respectively. These are effectively the expected area of the road and off-road regions. Because of the clustering formation of the classes, the road classes (or road color models) w_r are assumed to be nearly disjoint. Therefore, the likelihood $P(\boldsymbol{x}|\text{road})$ is found as the maximum of each of the road's color class probabilities

$$P(\boldsymbol{x}|\text{road}) = \max_{w_r} \{P(w_r) P(\boldsymbol{x}|w_r)\}$$

where $P(w_r)$ is the percentage of road sample pixels used to compute the color model for w_r. Since each of the individual road colors is represented by a Gaussian function, the

likelihood can be characterized by its mean vector $\boldsymbol{m}_r$ and its covariance matrix $\boldsymbol{C}_r$, so

$$P(\boldsymbol{x}|\text{road}) = \max_{w_r} \left\{ (2\pi)^{-3/2} |\boldsymbol{C}_r|^{-1/2} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_r)^T \cdot \boldsymbol{C}_r^{-1}(\boldsymbol{x}-\boldsymbol{m}_r) \right] \right\}.$$

We can find the maximum of this function by computing the maximum of its natural logarithm, which saves the computation of the exponential function for each road class. Therefore,

$$\ln(P(\boldsymbol{x}|\text{road})) = \max_{w_r} \left\{ L_r - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_r)^T \boldsymbol{C}_r^{-1}(\boldsymbol{x}-\boldsymbol{m}_r) \right\}.$$

where

$$L_r = -\frac{1}{2}\ln[(2p)^3|\boldsymbol{C}_r|]$$

is a constant that is calculated only once per image for each road class.

Using this procedure, we label each pixel $\boldsymbol{x}$ with the likelihood that it is a road pixel based on how well it matches a multi-class Gaussian model for road and off-road colors.

4) Results and Discussion: The SCARF likelihood image modeling is compared with other approaches in Fig. 2. The top row of the figure shows a few different unstructured road images. The second row displays the magnitude of the Sobel edge operator [1] on the above images. The Sobel magnitude images give an indication of the edge information available in the images and show the difficulty of using road-edge-based techniques on these unstructured roads. In the left image, the edge response is very noisy in the leaves. In the middle image, the edge response is strong on the right side of the road but weaker in the shadow. In the third shadowed image, shadow edges are much stronger than the road edges. The third row of images are the results of a histogram-and-threshold approach [1] using hand-selected thresholds. The left image shows that only fragments of the road surface are apparent, and in the shadowed images the histogram and threshold approach does better at separating shadows from sunny regions rather than road from off-road regions. The final row of images in Fig. 2 shows the road-surface likelihood images for SCARF. The brighter the pixel is shown, the more likely it is to be road. SCARF has a much better responses to all of the example images than do other approaches.

This classification is effective since it used multiple color classes for both road and off-road, and since it adapts these color models for each input image. Four color classes were chosen since this number seemed to give the best results. With four road classes, we can allow for two different pavement types as well as shadowed and nonshadowed conditions in one scene. We have noticed, however, in a simple scene where there are fewer than four road colors (or off-road colors), some of the color models will drop out in the color model formulation when the number of pixels assigned to the class becomes too few to fit a Gaussian distribution. As we increase the number of classes allowed, we also increase the amount of computation required to form the Gaussian models and to classify the image. We found that four classes was the smallest number of classes needed to model the scenes we tested. Future work could include an analysis of the number of classes required to more accurately represent the road scene.

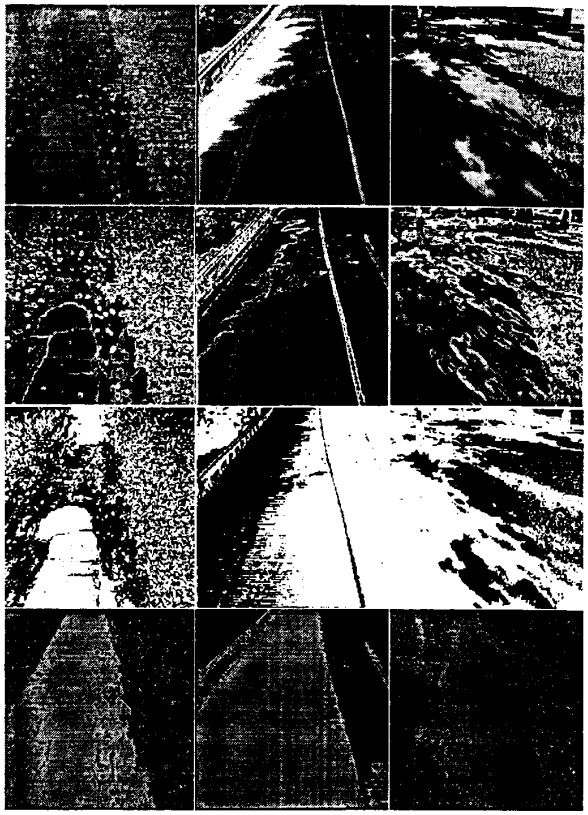

Fig. 2. Comparison of various road detection approaches. The top row shows three unstructured road image examples. The column of images under the road examples show processing on the example images. The second row shows the magnitude of the Sobel edge operator where the brighter pixels represent stronger gradient magnitude. The third row shows the result of a thresholding where a human selects the threshold value. The bottom row shows the results of SCARF road surface identification. The intensity of the pixel value corresponds to the likelihood of it being on the road surface.

B. Road Model Matching

In this module, SCARF selects, from a candidate set of road and intersection interpretations, the road or intersection that best matches the road-surface likelihood image. This is done using a matched filtering technique [2]. A binary image (or mask) is created for each candidate road and intersection that models how the candidate interpretation would ideally appear in the image. The candidate interpretation is evaluated by how well its mask compares with the road surface likelihood image from the classification module of SCARF. The candidate whose mask best matches the likelihood image is selected as the interpretation.

We first describe the road and intersection models. Next, we discuss how these models are used to generate the candidate masks. We then describe in detail the candidate-matching algorithm. Finally, we present our heuristic for limiting the number of candidates that we test.

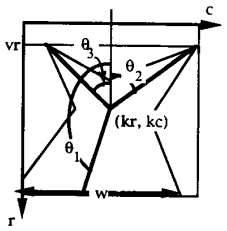

Fig. 3. SCARF intersection modeling.

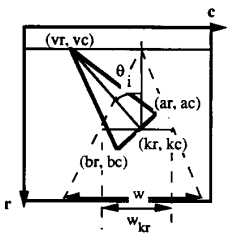

Fig. 4. Branch edge parameters.

1) Road and Intersection Modeling: The intersection model for SCARF contains the following values:

$$\text{intersection model: } (B, kr, kc, \{\theta_b\}, w, vr)$$

as shown in Fig. 3. There are B branches in the intersection which all meet at the common point, or kernel location, (kr, kc) in the image. The center lines of the branches are represented by the angle at which the branch leaves the intersection with respect the a vertical line. Therefore, the intersection model keeps track of the number of roads in the intersection B, the kernel position of the intersection (kr, kc), and a set of angles $\{\theta_b\}$, one for the center line for each of the adjoining roads. A width parameter w is used to describe the constant horizontal road width in the image at the last row br of the image, and the vr parameter specifies a constant vanishing row in the image.

We assume that roads:

- are locally straight,
- have constant width, and
- are lying on a ground plane.

The ground plane assumption fixes the horizontal location of the vanishing point in the image vr. The constant road width sets w to be constant. Although these assumptions limit the number of road possibilities that we model, we have found that by sampling the road frequently these approximations can still identify roads in the image well enough to navigate the robot through hills, valleys, and winding curved roads. These assumptions limit the dimensionality of the model and therefore provide a better match than higher order models in noisy images.

This model has a variable number of parameters depending on the number of intersection branches. If $B = 1$ and the kernel location is assumed to be the vanishing point in the image, this represents a straight road and the model can be reduced to:

$$\text{straight road } = \{1, vr, vc, \theta, w, vr\}.$$

If there are two branches, i.e., $B = 2$, then the intersection model represents a curving road in the image by a piecewise approximation to the road shap:

$$\text{curved road } = \{2, kr, kc, \theta_1, \theta_2, w, vr\}.$$

2) Candidate Generation: The candidate interpretation masks are idealized binary road images containing 1 if the pixel lies on the road surface of the candidate and 0 otherwise. This mask is used to match with the surface information generated by SCARF as described in in the next section.

To start the mask generation, the candidate interpretation mask is filled with 0; then for each branch of the input intersection model, the branch road edges and a cross section of the branch are computed. The pixels lying within the branch road edges and the branch cross section are then given the value 1 in the candidate interpretation mask. After all of the branches have been filled, the resulting mask will have a 1 at every pixel location on the road of the interpretation; otherwise the locations will have the value 0.

The end points (ar, ac) and (br, bc) of the cross section of a branch, shown in Fig. 4, are given by the following equations: if $0° < \theta_1 < 180°$:

$$ar = kr + \beta w_{kr} \sin\theta_i \qquad ac = kc - w_{kr} \cos\theta_i$$
$$br = kr - \alpha w_{kr} \sin\theta_i \qquad bc = kc + w_{kr} \cos\theta_i$$

otherwise:

$$ar = kr + \alpha w_{kr} \sin\theta_i \qquad ac = kc - w_{kr} \cos\theta_i$$
$$br = kr - \beta w_{kr} \sin\theta_i \qquad bc = kc + w_{kr} \cos\theta_i$$

where w_{kr} is the horizontal width of a vertical road at the kernel location and is computed by

$$w_{kr} = \frac{(kr - vr)}{(br - vr)} w.$$

The parameters α and β are used to approximate the perspective projection of distances in the image plane. These parameters scale the horizontal width of the road to approximate the vertical road widths in the image. This approximation was chosen since the calibration of many test sequences were unknown. This approximation has been successful at representing intersections in the test sequences as well as intersections on vehicle tests. These scalings are assumed to be constants for every row.

The orientations of the branch road edges are determined by the vanishing point of the center line of the branch. Since all lines parallel to the center in the ground plane will intersect at the vanishing point (due to prospective projection), the branch edge lines will intersect at the vanishing point. The vanishing point (vr, vc) is determined from the branch center line. The branch edges are determined from the vanishing point and the cross section end points.

3) Candidate Evaluation: If the new intersection branch is valid, or if there is only one branch in the intersection, then a correlation between the candidate interpretation mask and the road-surface likelihood image determines the discrepancy

value 1 of the candidate interpretation. If the candidate interpretation mask is $q(r,c)$ and the input likelihood image is $p(r,c)$, then the correlation value is computed by the equation

$$\gamma = \frac{1}{RC}\sum_{r=1}^{R}\sum_{c=1}^{C}|p(r,c) - q(r,c)|.$$

This measures the difference between the candidate interpretation mask and the likelihood image. The candidate interpretation mask has the value of 1 inside the road surface of the candidate interpretation, and a value of 0 otherwise. This is an ideal road-surface likelihood image for the candidate interpretation. The correlation compares the difference, pixel by pixel, between the input road-surface likelihood image and an ideal likelihood image for each interpretation. Since the maximum difference between the images is equal to the number of pixels in the image, and the correlation value is normalized by the number of pixels in the image, a confidence value ρ can be computed from $\rho = 1.0 - \gamma$. The confidence ρ is a value between 0 and 1.

C. Limiting the Intersection Search Space

An exhaustive search of all road and intersection candidates is computationally very expensive. To limit the number of candidates, SCARF interpretation generation evaluates candidates in a sequence. First, the best single road is found in the road-surface likelihood image. By evaluating the confidence ρ_1 of this interpretation, SCARF determines if the main road is believed to exist in the image. If the confidence falls below a threshold T, no main road is found. If it exceeds the threshold, the first branch is found that, when added to the main road, forms the best y-shaped or λ-shaped intersection. (A y-shaped intersection consists of a main road with a branch extending from the road where a λ-shaped intersection has a branch coming onto a main road.) This confidence ρ_2 is compared with the confidence of the single road interpretation of the image ρ_1. If the confidence of the single road is larger, the process exists with a straight-road interpretation of the image. If ρ_2 is larger than ρ_1, the intersection of the center lines of the main road and the main branch determine the kernel location (kr, kc) of the intersection in the image. In the branch matching, an interpretation space for all possible branches extending from the kernel location is computed. The intersection construction then finds the best possible set of branches to form the intersection interpretation and the confidence ρ of the intersection reported by the system.

Each matching phase of the interpretation generation module evaluates possible intersection interpretations using the same evaluation procedure. This candidate evaluation is described in the next section. Each interpretation matching phase is then described, and the intersection model construction is presented.

1) Main Road Matching: This matching phase finds the best single road interpretation of the road-surface likelihood image. A single road in the image can be represented by a single branch intersection, i.e.,

$$\text{intersection model of main road} = \{1, kr, kc, \theta, vr, w\}.$$

For matching, we parameterize this road with the end points of the center line, where the main road crosses the base row of the image and crosses the vanishing row of the image:

$$\text{main road} = \{vr, vc, br, bc, w\}.$$

This representation is easily converted into the intersection model by selecting the kernel location to lie at the base row of the image, i.e., $(kr, kc) = (br, bc)$ and $\theta = \tan^{-1}[(kc - vc)/(kr - vr)]$. Therefore, the road can be parameterized with just two parameters and assuming a constant road width w:

$$\text{main road} = \{vc, bc\}.$$

The candidate generation forms an intersection model for each (vc, bc) pair and generates a candidate interpretation mask. The candidate mask is compared with the road-surface likelihood image. The main road with the highest confidence, ρ_1, is selected as the main-road interpretation.

2) Finding the First Branch: If a main road is found, we can constrain the search for the best main branch. The center line of the main branch must intersect the center line of the road. Therefore, for any position along the main-road center line, we can check for main branches extending from the center line at a variety of angles.

The model for a main-branch interpretation is a three-road intersection where the angles of the first two branches are specified by the main road:

$$\text{main branch interpretation} = (3, kr, kc, \theta_1, \theta_1 + 180°, \theta_2, vr, W)$$

where θ_1 is determined from the previous main-road search. Since the intersection branch must intersect the main road center line, the parameter kc can be computed for each value of kr. We also assume that the branch will have the same width as the main road w. Therefore, the problem of finding a two-branch intersection has two parameters:

$$\text{main branch} = \{kr, \theta_2\}$$

since we are looking for only branches that extend from a predetermined main-road line.

Again the candidate generation builds an interpretation mask for possible intersections formed from the main road and a candidate branch. These intersections appear y-shaped or λ-shaped. Each branch is first tested to be sure that it does not excessively overlap the main road. The candidate intersections with nonoverlapping branches are then assigned a confidence by the candidate evaluation module using the interpretation mask formed from the main road and the branch. The interpretation with the highest confidence ρ_2 is selected as the best interpretation. If this interpretation has a higher confidence than the main road interpretation, the kernel location of the intersection is used to find the additional branches of the intersection.

3) Finding Additional Branches: If a main branch is found by the previous step, the system searches for multi-branch intersections. The locations of the main road and the first branch determine the kernel position of the intersection. The

intersection is built from a set of branches that are added individually to the intersection model.

The model of an intersection is:

$$\text{intersection} = (B, kr, kc, \{\theta_b\}, vr, w).$$

The main road and the first branch have constrained kr and kc. W and vr are assumed to be constants. Now we have to determine the number of intersection branches B and the angles $\{\theta_b\}$ of these branches.

The confidence of each single-branch interpretation extending from the kernel location is collected in a one-dimensional branch-interpretation space. The confidence peaks in this space are detected and ordered by highest confidence to form a list of the possible intersection branches. The branch with the highest confidence is taken from the list to form the first branch of the intersection. The next best branch is added to the first branch to form a candidate intersection. If this interpretation produces a better confidence from the candidate evaluation module, then it is added to the intersection interpretation. Otherwise, the next best branch is considered. One by one, new branches are added to the intersection only if their inclusion provides a better match between the interpretation and the road-surface likelihood image.

4) Discussion: This section discussed how the multidimensional intersection model matching was reduced using a series of shape-matching steps. It first found the main branch of the intersection and then used this branch as a constraint on the kernel location of the intersection. The multiple branches of the intersection were determined using the kernel location found by previous steps.

The new intersection branch is tested to insure that it does not excessively overlap the other branches already included in the intersection. Each pixel on the line is located (using a modified Bresenhams line drawing algorithm [17]) to count the total number of pixels n_o on the new branch center line that overlap the road surfaces of the other branches. This algorithm also computes the total length n of the center line appearing in the image. The percentage of center line pixels contained in the other branches n_o/n must be less than some threshold while the number of nonoverlapping pixels $n - n_o$ must be greater than another threshold. If either of these tests fail, the new intersection branch is not added to the candidate interpretation and the overall interpretation likelihood is not effected. These thresholds were determined empirically through testing.

To perform a exhaustive search of all road and intersection candidates is computationally expensive. To locate a four-branch intersection exhaustively, we would need to search over six parameters $(kr, kc, \theta_1, \theta_2, \theta_3, \theta_4)$. Allowing k possible values for each of the parameters, we would then need to compute k^6 candidate matches. Using our heuristic search method, we still require two parameters to find the main road, two parameters to find the first branch, and one parameter to locate all intersection branches. Again allowing k possible values for each of the parameters, we now need to test only $2k^2 + k$ candidate intersections. This is 210 candidates rather than 10^6 candidate matches for $k = 10$. The assumption underlying the heuristic search is that we can find a good match to a main road first and that this match will contain the kernel location of the intersection. We also assume that the main branch can correctly identify the kernel location of the intersection. While these assumptions worked well on the intersections in which we tested the algorithm, we can envision scenarios, such as T-shaped intersections (y-shaped intersections with parallel branches) where this heuristic may fail.

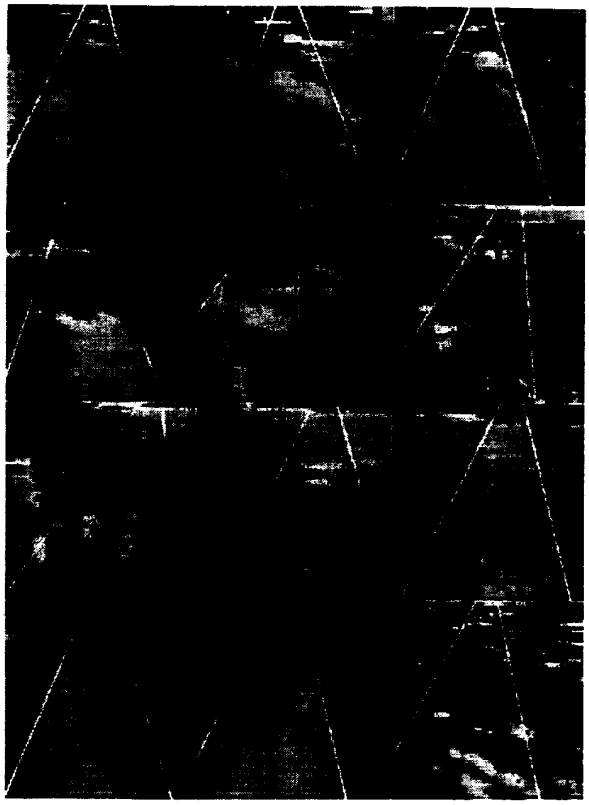

Fig. 5. SCARF detects unstructured roads in dark shadows: SCARF can detect a road in widely fluctuating illumination. In all images (except the fourth), the road is correctly identified. In the fourth image, the road detected is not that far from the true road location so that the robot has no problem remaining on the road. This sequence illustrates the difficulty with dark shadows, since in the shadowed areas of the image it is difficult for a person to distinguish between road and off-road.

Another difficulty with this matching procedure is the limited field of view of the cameras. We had to limit our tests to single-lane roads so that in the field of view we could get enough pixels to form our off-road color models. Moreover, if we would have switched to a fish-eye lens, we would introduce distortion that would limit our straight-road modeling assumption. Even with our normal lens, if the branches have curved pavement around the junction of the branches, the whole image could be composed of the intersection branch with no off-road pixels to sample. Typically in this case, we retain the off-road color models from the previous image. We also had some difficulty with multiple intersection interpretations being equally likely when the intersections are near the bottom in the image. We have had to add constraints to track the intersections from image to image, as the vehicle approaches, so that, by the

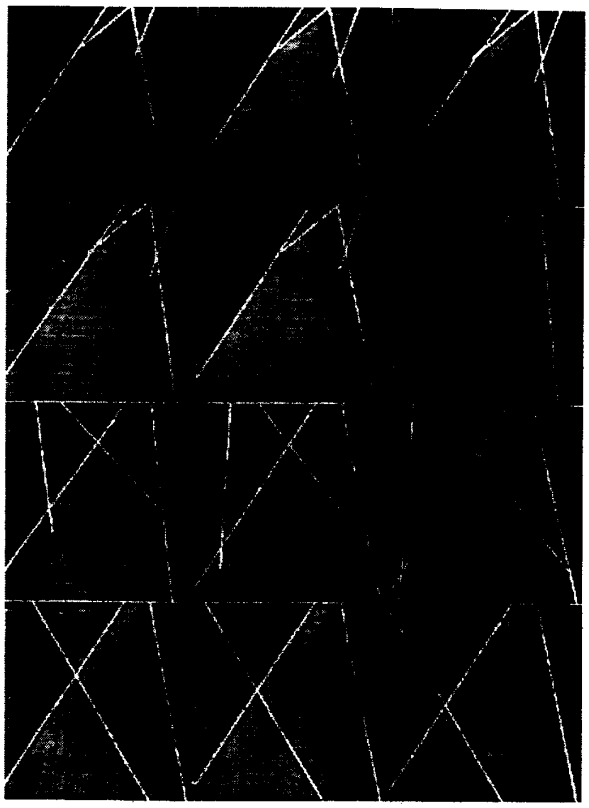

Fig. 6. SCARF results on a y-shaped intersection: This intersection is interesting in that the main road has a width of 3.1 m while the branch has only a width of 1.7 m. While this intersection violates the assumption that the roads must be the same width, SCARF can still detect it. A curve in the road is initially detected as an intersection, but as the intersection becomes larger in the image, SCARF correctly detects it and tracks its location.

time the intersection reaches the bottom of the image, only a small number of branch angles around the previously detected angle are considered. This was needed because the camera field of view does not include much of an intersection at the bottom of the image.

The limitations of the SCARF algorithm are emphasized by the assumptions of the system:

- colors of the road differ from colors off-road:
- roads are locally straight, constant width, and planar; and
- the speed of processing is fast enough to sample the scene for the vehicle speed.

SCARF can have difficulties in the classification steps if there are not distinguishable colors between road and off-road. If, for example, there are scattered leaves on the road and mostly leaves off-road, SCARF would have no difficulties since the volume of leaf colors off-road will be much greater than those on the road. Therefore, the statistical models of leaf color will be much stronger for off-road. However, if the volume of leaves on and off-road are identical, the color classification algorithm cannot distinguish road and off-road leaves. Leaves in this image will be assigned a 0.5 or "unknown" likelihood. While we have been able to demonstrate robustness of the SCARF algorithm even though the road shape assumptions

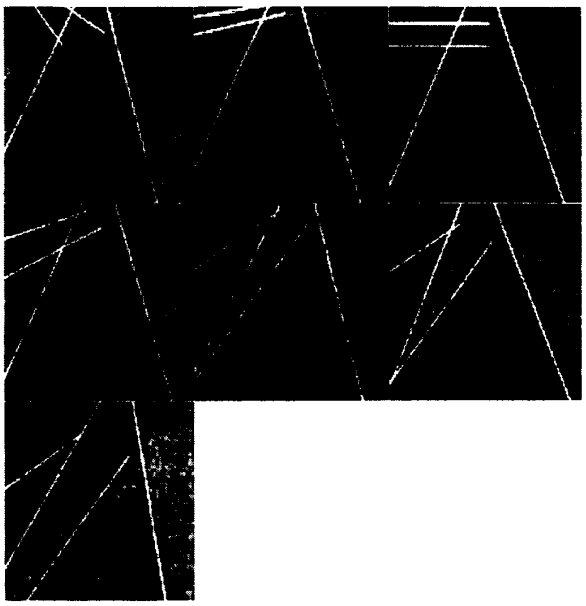

Fig. 7. SCARF results on a λ-shaped intersection: This shows the first branch detection of a λ-shaped intersection. This also shows the difficulties with unstructured roads with the degraded road surface and rugged road edges. In this case, the leaves on the off-road areas of the scene also cover part of the road. The SCARF intersection models capture the shape of the intersection well enough for vehicle navigation.

were violated, SCARF would fail to recognize a four-lane road when the vehicle has been traveling on a two-lane road. We have also noticed control instabilities as the speed increases, especially around curves in our test site. This demonstrates that the processing speed of the algorithm must be adequately fast to permit the locally straight-road assumption to be valid.

IV. Results

Results of SCARF processing several unstructured road and intersections are shown in Figs. 5–9. The results are discussed in the captions associated with each figure. Fig. 5 shows SCARF's processing in difficult shadows. SCARF locates the road well enough to navigate through the sequence. SCARF has succeeded in detecting this road as well as others in difficult shadows. Figs. 6–8 show the results of SCARF on some unstructured intersections. In all of these examples, we show only the first branch of the detected intersection. Often we used only the initial branch since this was sufficient to navigate the Navlab through the test site. By not detecting the location of all the branches, we saved computation time and could therefore process images faster. An example of multibranch detection is shown in Fig. 9. Detection of straight roads takes SCARF 6 s running on a Sun 4. To detect intersections requires 20 s on the average.

This sequence of figures is just an example of the flexibility of the SCARF algorithm at recognizing unstructured roads and intersections. We have tested this system extensively on our test vehicle, the Navlab, on our primary test site (a winding bicycle path in Schenley Park) in all seasons under all types of weather conditions. It has successfully navigated during rainy

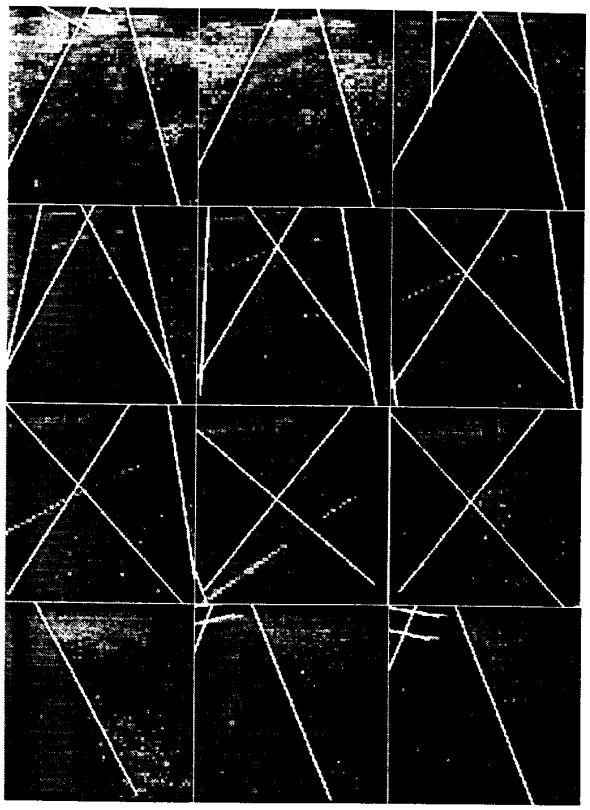

Fig. 8. SCARF results on a Y-shaped intersection: The center of this intersection is located on the top of a hill that defies the ground plane assumption of the SCARF system. Yet, SCARF still detects this intersection. Also note that the right branch of the intersection is degraded and the color of this branch appears different than the other road surfaces in the image. The intersection detected in the last images of this sequence actually represent a sharp curve in the road.

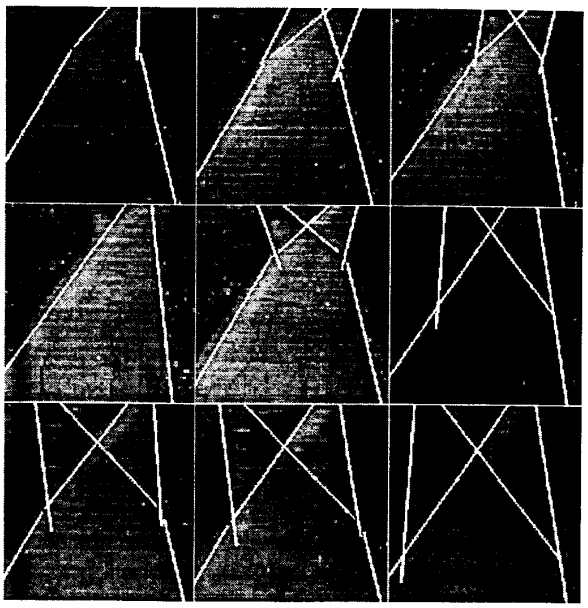

Fig. 9. SCARF multibranch detection on a y-shaped intersection: This shows the results of SCARF multibranch intersection detection. Note that in the fourth image, a single road is detected. In the remaining images, however, the intersection is correctly located. We have found that an intersection branch that is matched near the top of the image is more susceptible to noise since there are fewer intersection pixels in the image. Therefore, if using SCARF's results to build a map, a navigation system should rely more on later intersection branch locations.

weather, when the images appear grey and colorless; on the brightest sunny days, when the nearby trees cast extremely dark shadows across the road; and in the fall, when leaves cover the ground off-road and sparsely cover the road.

The SCARF system has been integrated into several road-navigation systems including the Park I demonstration system [21] and the Park II demonstration [20]. SCARF has been implemented on the WARP Supercomputer [4] where it achieved processing times as low as 3 s image. The increased processing speed can be used to increase the speed of the robot vehicle or it can be used to sample the environment closer in space for more reliable predictions. SCARF has more recently been integrated into the Autonomous Mail Vehicle [19]. SCARF has proved useful for a number of navigation scenarios.

V. Conclusions

This paper described the SCARF road- and intersection-detection system. This system specializes in detecing unstructured roads, the most difficult road-following scenario. This is the first road-detection system that can detect intersections without map shape and location information. Other systems rely on knowing the angles and shapes of the intersection branches. SCARF determines the branch angles strictly by the color data in the input image. SCARF has successfully driven the Navlab mobile robot on numerous occasions in a variety of weather conditions. SCARF has also been integrated into both mapped and unmapped navigation systems at Carnegie Mellon.

The success of this system is due to several factors: a likelihood measure associated with classification, an area-based matching technique, the straight-road model, and sampling and adjusting the color models of road and off-road in each image. The likelihood measure from the classification causes shadowed pixels (whose color matches pretty well with both shadowed road and shadowed off-road colors) to be weighted less in determining the road location than those pixels having distinctive road or off-road colors. The area-based matching does not depend on the roads having clean edges. Also, since more data points are used for locating the road, the area-based technique is less sensitive to noise. The straight-road models are also less sensitive when matching noisy data since there are fewer parameters to the model. A curved road, if sampled often, can be represented by a piecewise straight-road model. Since SCARF adjusts the color model at each step of the processing, it can adjust to changing illumination and current road and off-road surface appearances.

Acknowledgment

The authors wish to thank Dr. Takeo Kanade at Carnegie Mellon University who supervised this research. They would

also like to thank the numerous members of the Navlab and related projects at Carnegie Mellon, without whom this work would not have been possible.

REFERENCES

[1] D. Ballard and C. Brown, *Computer Vision*. Englewood Cliffs. NJ: Prentice-Hall, 1982.

[2] K. Castleman, *Digital Image Processing*, A. Oppenheim, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[3] J. Crisman, "Color vision for the detection of unstructured road and intersections," Ph.D. dissertation, Dept. of Electrical and Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, May 1990.

[4] J. Crisman and J. Webb, "The Warp machine on Navlab," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 5, May 1991; also in *Vision and Navigation: The Carnegie Mellon Navlab*, C. Thorpe Ed. Norwell, MA: Kluwer, 1990.

[5] E. Dickmanns and A. Zapp, "A curvature-based scheme for improving road vehicle guidance by computer vision," in *Proc. 10th IFAC* (Munich), 1987.

[6] ——, "Guiding land vehicles along roadways by computer vision," presented at AFCET Conf. Automatique 85—The Tools for Tomorrow, Toulouse, France, Oct. 1985.

[7] K. Dowling, *et. al.*, "Navlab: An autonomous navigation testbed," in *Vision and Navigation: The Carnegie Mellon Navlab*, C. Thorpe, Ed. Norwell, MA: Kluwer, 1990.

[8] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[9] Y. Goto, K. Matsuzaki, I. Kweon, and T. Obatake, "CMU sidewalk navigation system: A blackboard-based outdoor navigation system using sensor fusion with color-range images," in *Proc. 1st Joint Conf. ACM/IEEE*, Nov. 1986.

[10] S. Kenue, "Lanlok: Dectection of land boundaries and vehicle tracking using image-processing techniques. Part I: Hough-transform, region-tracking, and correlation algorithms," in *Proc. SPIE, Mobile Robots IV*, Nov. 1989.

[11] ——, "Lanelok: detection of lane boundaries and vehicle tracking using image-processing techniques. Part II: Template matching algorithms," in *Proc. SPIE, Mobile Robots IV*, Nov. 1989.

[12] K. Kluge and C. Thorpe, "Explicit models for robot road following," in *Proc IEEE Conf. Robotics Automat.*, May 1988.

[13] D. Kuan, G. Phipps, and A. Hsueh, "Autonomous land vehicle road following," in *Proc. 1st Int. Conf. Comput. Vision* (London), June 1987, pp. 557–566.

[14] T. Kushner and S. Puri, "Progress in road intersection detection for autonomous vehicle navigation," in *Proc. SPIE, Mobile Robots*, 1987, pp. 19–24.

[15] S. Liou and R. Jain, "Road following using vanishing points," in *Proc IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.* (Miami Beach, FL), June 1986, pp. 41–46.

[16] B. Mysliwetz and E. Dickmanns, "Distributed scene analysis for autonomous road vehicle guidance," in *Proc. SPIE, Mobile Robots II* (Cambridge, MA), Nov. 1987, pp. 72–79.

[17] W. M. Newman and R. F. Sproull, *Principles of Interactive Computer Graphics*, 2nd ed. New York: McGraw-Hill, 1979.

[18] D. Pomerleau, "Neural network based autonomous navigation," in *Vision and Navigation: The Carnegie Mellon Navlab*, C. Thorpe, Ed. Norwell, MA: Kluwer, 1990.

[19] C. Thorpe, "Outdoor visual navigation for autonomous robots," in *Vision and Navigation: The Carnegie Mellon Navlab*, C. Thorpe, Ed. Norwell, MA: Kluwer, 1990.

[20] C. Thorpe and T. Kanade, "1987 year end report for road following at Carnegie Mellon," Carnegie Mellon Univ., Robotics Institute, Tech. Rep. CMU-RI-TR-8, 1988.

[21] ——, "1986 year end report for road following at Carnegie-Mellon," Carnegie Mellon Univ., Robotics Institute, Tech. Rep. CMU-RI-TR-87-11, 1987.

[22] S. Tsugawa, T. Yatabe, T. Hirose, and S. Matsumoto, "An automobile with artificial intelligence," in *Proc. 6th Int. Joint Conf. Artificial Intell.* (Tokyo), Aug. 1979, pp. 893–895.

[23] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "VITS: A vision system for autonomous land vehicle navigation," *IEEE Trans. Pattern Anal. Machine. Intell.*, vol. 10, no. 3, pp. 342–361, May 1988.

[24] R. Wallace, K. Matsuzaki, Y. Goto, J. Crisman, J. Webb, and T. Kanade, "Progress in robot road-following," in *Proc. IEEE Conf. Robotics Automat.*, Apr. 1986, pp. 1615–1621.

[25] A. Waxman *et. al.*, "A visual navigation system for autonomous land vehicles," *IEEE J. Robotics Automat.*, vol. RA-3, no. 2, pp. 124–141, Apr. 1987.

[26] H. Wunshce, "Detection and control of mobile robot motion by real time computer vision," in *Proc. SPIE, Mobile Robots II* (Cambridge, MA), Oct. 1986.

Jill D. Crisman (S'89–M'90) received the B.S. and M.S. degrees in electrical engineering from the University of Pittsburgh, Pittsburgh, PA in 1984 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1990.

She was an active researcher in computer vision on the Navlab project at Carnegie Mellon University from 1985 to 1990. She is currently an Assistant Professor in the Electrical and Computer Engineering Department at Northeastern University, Boston, MA, where she is directing the Robotic and Vision Systems Laboratory. Her research interests include computer vision, robotics, pattern recognition, and artificial intelligence.

Dr. Crisman is a member of Eta Kappa Nu, Tau Beta Pi, and the Classification Society of America.

Charles E. Thorpe (M'87) received the B.A. degree in natural science from North Park College, Chicago, IL, and the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 1984.

He is currently a Senior Research Scientist at the Robotics Institute of Carnegie Mellon University. His interests are in computer vision, planning and control of robot vehicles operating in unstructured outdoor environments. He directs the research on the Navlab project, which is developing mobile robots that use perception and mapping for outdoor autonomous driving. He is also involved with robots for planetary exploration and for underwater mapping. He is editor of the book, *Vision and Navigation: The Carnegie Mellon Navlab.*

Dr. Thorpe serves as a technical editor for IEEE Transactions on Robotics and Automation.