Proceedings o'f' SPIE Conference on Mobile Robots, November 10-11, 1988,
Cambridge, Massachusetts.

Color Vision for Road Following *

Jill D. Crisman and Charles E. Thorpe
Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213

October 12, 1988

Abstract

At Carnegie Mellon University, we have two new vision systems for outdoor road following. The first system, called SCARF (Supervised Classification Applied to Road Following), is designed to be fast and robust when the vehicle is running in both sunshine and shadows under constant illumination. The second system, UNSCARF (UNSupervised Classification Applied to Road Following), is slower, but provides good results even if the sun is alternately covered by clouds or uncovered. SCARF incorporates our results from our previous experience with road tracking by supervised classification. It is an adaptive supervised classification scheme using color data from two cameras to form a new six dimensional color space. The road is localized by a Hough space technique. SCARF is specifically designed for fast implementation on the WARP supercomputer, an experimental parallel architecture developed at Carnegie Mellon.

UNSCARF uses an unsupervised classification algorithm to group the pixels in the image into regions. The road is detected by finding the set of regions which, grouped together, best match the road shape. UNSCARF can be expanded easily to perform unsupervised classification on any number of features. and to use any combination of constraints to select the best combination of regions. The basic unsupervised classification segmentation will also have applications outside the realm of road following.

1 Introduction

At Carnegie Mellon University, we have been building successful, color vision based, road following systems for several years [6,7,9,10]. The main emphasis of our road following research is to find unstructured roads in images that are complicated by shadows, leaves or *dirt* lying on the mad, lighting changes, and the like. We initially used edge based techniques, that searched for edges in the image to correspond with road edges in the scene. This proved inadequate for our Schenley Park test site, since often image edges caused by shadows were more distinctive than edges formed from road boundries. Currently we have been using a color classification system, SCARF (Supervised Classification Applied to Road Following), where each pixel in the image is labeled as road or non-road based on the match of its color to previously learned colors. The road is found by looking for the road shape that contains the most 'mad' labeled pixels. We also use an unsupervised classification algorithm, UNSCARF (UNSupervised Classification Applied to Road Following), that groups pixels that have similar color and location, and then searches for the combination of groups that best matches the road shape. This paper discusses these two systems.

Other groups have also been working on road-following. In Germany. Dickmanns and Grafe [3,4] view mad following as a control problem. They have developed an elegant control formulation that incorporates a simple road edge detector with the vehicle model to drive their vehicle at speeds up to 100 kph. They also use constraints of the autobahn shape

'This research is sponsored by the Strategic Computing Initiative of DARPA. DoD, through ARPA Order 5351. and monitored by the US Army Engineer Topographic Laboratories under contract DACA76-85-C-0003, titled "Road Following." Portions of this research were also partially sponsored by the National Science Foundation contract DCR-8604199 and by the Digital Equipment Corporation External Research Program.

and markings. The autobahns are of constant width and are either straight, constant curvature. or clothoid in shape. The rapid processing and structured road model help to limit a search window in the image, and discard the extraneous edges normally found by edge detectors. However, it seems that their trackers could be distracted by the shadows, puddles and road imperfections that plague our test site.

The University of Maryland [2] has also been working on mad following. Their system drove an autonomous vehicle based on edge detection. Image edges where tracked from the bottom of the image to the top using an edge detector in a window of the image. Once an edge is located, it is used to constrain the position and orientation of the next window. Then the edges were grouped using a Hough transform to determine which image edges form the best road edge. This system worked well when the dominant edges in the image are road edges, but similar systems at CMU have failed when tracking edges in strong shadows or when leaves or *dirt* lie *on* the roads.

At Martin Marietta, the VITS system [8] has achieved impressive speeds *on* fairly unstructured roads. It projects the three-dimensional color space (red, green, blue) into one dimension, or in later systems two dimensions. It then differentiates the road from non-road by a linear discriminant function. The road/non-road threshold is selected by sampling a part of the image that is guaranteed to be road This system is similar to CMU road following, but emphasizes speed rather than general capability. Their system works fast, up to *20* kph, on their test site, but it is doubtful that it will work on other test sites, since the color projection is tuned for the features that are best to discriminate their road from their non-road.

Our goal is to build general color vision algorithms that work in a wide variety of situations. In particular, we are working on recognizing unstructured mads in various types of illumination and weather conditions. To give our system general capabilities, we must address the following problems:

- ***The objects in the scene undergo spatial changes in appearance.*** For example, under sunlight, roads appear to be a different color than they appear in the shade.
- ***Objects in the scene undergo temporal changes in appearance.*** This may occur when clouds pass over the sun for instance. The change in illumination will cause identical road segments to have different colors from frame to frame.
- ***The dynamic range of our cameras is limited.*** We cannot digitize meaningful data in dark shadows of a brightly sunlit image, nor *can* we capture data in the brightly sunlit regions of a dark image.
- ***The roads in Schenley Park are very unstructured.*** There are no center or bordering lines painted on our roads, as *on* highways. Many of the road edges are obscured or uneven. The pavement of our roads is deteriorating in places. and the pavement may be covered with the same leaves, dirt, or snow, that appear off road.

Our two new systems, ***SCARF*** and UNSCARF, were built to address these problems. Both systems deal with the limited dynamic range of the cameras by using two *cameras* with different iris settings to capture both dark and bright areas of the scene. ***SCARF*** is designed to be a fast, adaptive system. Even though algorithm speed is not a goal of our research, faster algorithms have the advantage of more overlap between frames, if the vehicle speed is constant. When the images are processed closer in time and distance, the lighting conditions are less likely to change dramatically and the road position in the image will not move far between frames. UNSCARF tackles the temporal and spatial changes by processing each image independently of the others. No color models are tracked from frame to frame, making this algorithm insensitive to spatial or temporal changes in color.

In the next section of *this* paper, we describe the ***SCARF*** algorithms and discuss results. UNSCARF is detailed and discussed in the following section, and finally, general conclusions are drawn in the final section of this paper.

2 SCARF

SCARF has evolved by adding more and more capabilities to a simple road following system. A block diagram of this system is shown in figure 1. ***SCARF*** uses two cameras to digitize one frame of data. The two color images are reduced by an averaging filter and sent to the classifier. For each pixel in *the* reduced images. the classifier calculates a likelihood value that describes how well the pixel matches remembered road and non-road colors. The interpreter uses the likelihood values to find the road shape that contains the most likely road pixels. The road location is then used to update the remembered colors for the next frame. The road location is also used to steer the vehicle. This system has been implemented on the WARP supercomputer.

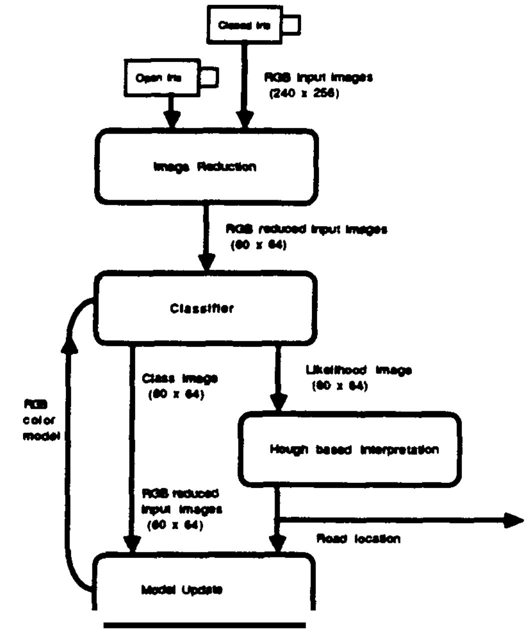

Figure 1: **Block** diagram of ***SCARF***

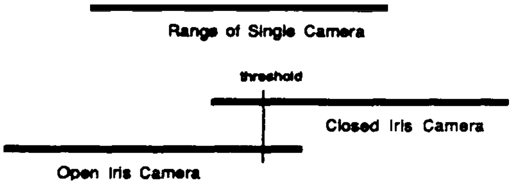

Figure 2: **Extending** the **dynamic** range using 2 cameras

2.1 Two Camera System

To extend the dynamic range of a single camera, we **are** using two images of the same scene digitized from the two cameras mounted on our test vehicle, **the** Navlab. The *cameras* were positioned **as** closely together **as** possible, **and** bore-sighted, minimizing the difference between the camera images. To avoid calibrating **the** two cameras, we treat the images **as** if they were taken from the same camera. This approximation is **adequate** for our purposes since the baseline of our cameras is much smaller than the **distance** to **the road.**

The improvement in dynamic range **results from** the different **iris** settings of the two camera **as** shown in figure 2. One of the cameras is set to capture the shadowed **area** of **the** scene by opening its iris. The second camera captures the sunny areas by using a closed iris.

When the two color images ***are*** digitized, they ***are*** first reduced by using **an** averaging filter. This not only reduces the **data** size, but will also reduce the noise content of the image. The reduced input images **are** used throughout the program, to increase the speed of the processing.

Two different methods **for** using **the** ***two*** input images **were** ***tested.*** The first approach is to combine the two reduced images into one. We used a simple thresholding technique to extend the dynamic range **as** shown in **figure** 2. If the **closed iris** image pixel was too ***dark,*** then the pixel was selected **from** the open **iris** image, otherwise it was copied from the closed iris image. The second approach was to use both **reduced** input images to **form** a six-dimensional color space. Then all six features, the red, green, and blue bands from the two images, ***are*** used in SCARF.

2.2 Classifier

In standard pattern recognition theory, a classifier takes a d-dimensional measurement vector, $\mathbf{x}$, and chooses the best Class label, ω_j, from a set of K classes, using a previously computed, *class conditional* probability, $P(\mathbf{x}|\omega_j)$, for each class [5]. The best class is the class that maximizes the a *posteriori* probability, $P(\omega_j|\mathbf{x})$. The expression for the *a posteriori* probability is normally derived from Bayes rule:

$$P(\omega_j|\mathbf{x}) = \frac{P(\mathbf{x}|\omega_j)P(\omega_j)}{P(\mathbf{x})}$$

In our case, each pixel provides a 6 dimensional measurement vector (d = 6), $\mathbf{x} = [R_1G_1B_1R_2G_2B_2]^T$, formed by concatenating the red, *green,* and blue bands of the two reduced input images. We use several classes to model road and non-road colors, typically 12 road models and 12 non-road models, giving 24 total color models (K = 24). We assume that the *class conditional* probability models for each class are Gaussian distributions, therefore, $P(\mathbf{x}|\omega_j)$ *can* be completely characterized by the mean vector and the covariance *matrix* of the sampled points for class ω_j. We also assume that the $P(\omega_j)$ is the ratio of the number of samples in w„ N_j, over the total number of samples, N. Therefore,

$$P(\omega_j|\mathbf{x}) = \frac{2\pi^{-d/2}||\mathbf{C}||^{-1/2}e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}}{P(\mathbf{x})}\frac{N_j}{N}.$$

Rather than calculating $P(\omega_j|\mathbf{x})$ at each pixel, we simplify the calculations by chosing the class, ω_j, that has the maximum $\ln P(\omega_j|\mathbf{x})$. This can be further simplified by noticing that $P(\mathbf{x})$ is identical for all of the classes, so that it *can* be eliminated from all of the terms. Therefore our classifier selects the class that maximizes the following likelihood measure:

$$\lambda_j = \ln(\frac{N_j}{N}) - \frac{1}{2}\ln((2\pi)^d||\mathbf{C}_j||) - \frac{1}{2}(\mathbf{x}-\mathbf{m_j})^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m_j}).$$

2.3 Interpretation

The interpretation receives a likelihood image, containing A, and a classification image, containing w„ from the classifier. By looking at the classification image, we can label each pixel in the image as either 'road' or 'non-road'. The interpreter searches for the road having the highest combined likelihood using a voting scheme similar to the Hough technique. The standard Hough algorithm searches for a line by voting for all of the lines passing through an edge point. However, we find a road by voting for all the possible roads containing 'road' pixels and by voting against all possible roads containing 'non-road' pixels. The main difference is that all of our pixels vote, not just pixels lying on the edge of the road. We also use the likelihood measure *to* determine the weight of each vote.

We assume the road is locally nearly straight, and *can* be parameterized using v, the column where the road center line crosses the horizon row, (or the vanishing point) and θ, the angle between the road center line and a vertical image line. These two parameters are *the* dimensions of an accumulator used for collecting votes. Each pixel in the likelihood image votes for all the roads that contain that pixel by adding its likelihood to the proper positions in the accumulator.

For each angle θ_i, a given pixel location (r, c) will vote for a set of vanishing points lying between v_s and v_e as shown in figure 3. The starting column position, v_s, corresponds the interpretation that pixel (r, c) lies on on the right hand edge of the road in the image, and the ending column position, v_e corresponds *to* the (r, c) pixel lying on the left hand edge of the road. The v positions are calculated by:

$$v_s = c + (r - horiz)\tan e - (w/l)(r - horiz)$$

$$v_e = c + (r - horiz)\tan \ell + (w/l)(r - horiz).$$

where $horiz$ is the horizon row in the image, w is the road width at the bottom of the image, and l is the length from the horizon row to the bottom of the image. The maximum value of the accumulator is chosen to be the road.

2.4 Model Formation

The new model can be calculated using standard statistical equations for mean and covariance:

$$\mathbf{m}_j = \frac{1}{N_j}\sum_{i=1}^{N_j}\mathbf{x}_i \qquad (1)$$

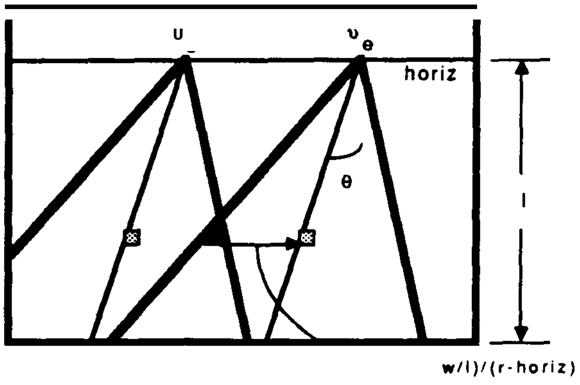

Figure 3: Hough voting scheme

$$C_j = \frac{1}{N_j}\sum_{i=1}^{N_j} \mathbf{x}_i^T\mathbf{x}_i - \mathbf{m}_j^T\mathbf{m}_j. \qquad (2)$$

First we have to decide which pixels belong to each class. Once we have a 'road'/'non-road' labeling, we calculate statistics for the road and non-road classes. Then we reclassify each 'road'or 'non-road' pixel using only the road classes for 'road' pixels and only the non-road classes for 'non-road' pixels. We iterate the calculate statistics and reclassify steps until the classes converge.

The road location is given from the user in initialization or from the interpretation on subsequent steps. Using this location, each pixel can be identified as road or non-road. The road region and the non-road region of the image are shrunk, forming a safety margin at the edge of the road. This is important so that the new color model is not corrupted due to the discretization of road locations or inaccurate fitting of a straight road model to a gently curving road. The reduced road and non-road areas are used to sample the colors of road and non-road.

An iterative clustering technique is applied to the road region and an identical procedure is followed for the non-road region. First, the mad pixels are arbitrarily given one of the road class labels. We assign the classes by indexing through the road pixels and assigning the next mad class. The color model, consisting of $\{N_j, \mathbf{m}_j, C_j\}$ is calculated for each of the classes using equations (1) and (2). Then all of the road pixels are relabeled by the class whose mean color is closest to the pixel value, and a new color model is calculated using the new labeling. This 'label/sample' loop is repeated until most of the pixels remain in the same class.

2.5 WARP Implementation

We have implemented one of our supervised classification systems on the wire-wrapped, prototype WARP supercomputer [1]. The increase in processing speed, although significant, was limited by the small memory on each cell. Much time was spent down-loading code and data, at each function call, typically 14 calls per step. Our new PC WARP has more memory on each processing unit, allowing larger programs and larger global data structures. Therefore, we have one large WARP function rather than multiple WARP functions, taking advantage of the larger program memory. This results in greater speeds since data is only downloaded once and the WARP start-up sequence is executed once per image frame.

The inputs to the WARP program are the six reduced images and the statistical model for each class. The WARP function segments, interprets, and produces the new color model. It outputs the road location and the new color model.

2.6 Discussion

This program has been tested on several sequences of images. The SCARF system has driven over all of the roads of our test site successfully. We have driven the system a variety of weather conditions, on different road surfaces, and under different lighting conditions. It adapts very well to different road surface types and differing off-road objects. Figure 4 shows SCARF running through severe shadow conditions from our test site.

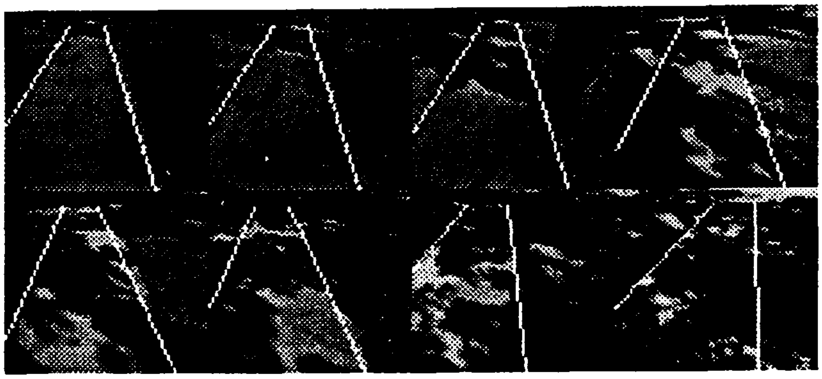

Figure 4: *SCARF* examples in dark shadows: The lines show the resulting road location.

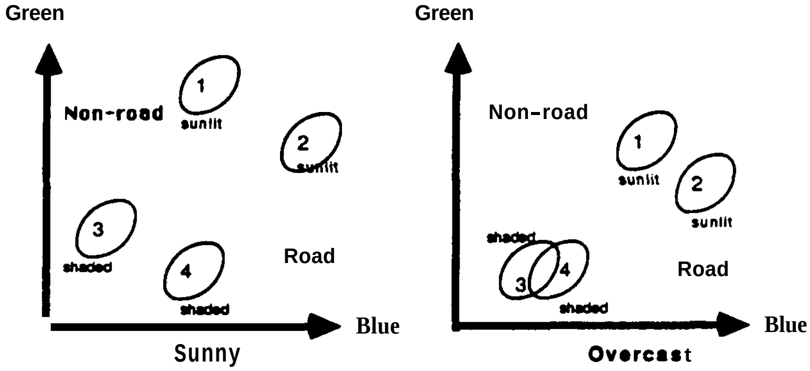

Figure 5: Effects of rapid illumination change

Using two input images does increase the dynamic range beyond that of a single camera. We found that combining the data into a single image provided a fast means of extending the dynamic range, however, using both input images was more reliable. Not only does the use of two images increase the effective dynamic range, it also increased the data available to classify each pixel, thereby increasing classification confidence and accuracy. Moreover, using all the data from both cameras avoids the potential problems of picking a threshold for selecting data to form a single image.

The classification works well as long as the lighting conditions or the road type does not change drastically between frames. As the time and distance between frames decrease, the results from the classification improve.

The Hough interpretation provides the robustness of the *SCARF* system. Since it is an area based technique, there is more data used in the interpretation than an edge based technique. This makes the system less sensitive to noise. Using this interpretation, we have been able to drive our Navlab vehicle in a variety of weather conditions. The results are good even when the road may be partially covered with the same leaves, dirt, or snow that is on the non-road parts of the image.

3 UNSCARF

UNSCARF was designed to attack the problem with temporal and spatial color appearance changes. In SCARF, models of road and non-road colors, taken from the previous image, were used to label pixels in the current image. However, if the color appearance of these objects changed dramatically. for any reason, then the color models no longer represented the colors of the objects in the new image. *SCARF* performed well as long as the illumination did not change too quickly. An example of a failure situation due to rapid illumination changes is shown in figure 5. A classifier is calculated for sunny and shaded road and non-road classes in a sunlit scene as shown on the left. In the next image, that classifier will fail, since the sun is hidden by clouds and the colors of the mad and non-road classes have shifted.

UNSCARF does not use pre-computed color models, instead it groups pixels that are similar in color and location in the image using an unsupervised classification algorithm as shown in figure 6. Then the pixels with the same labels, or

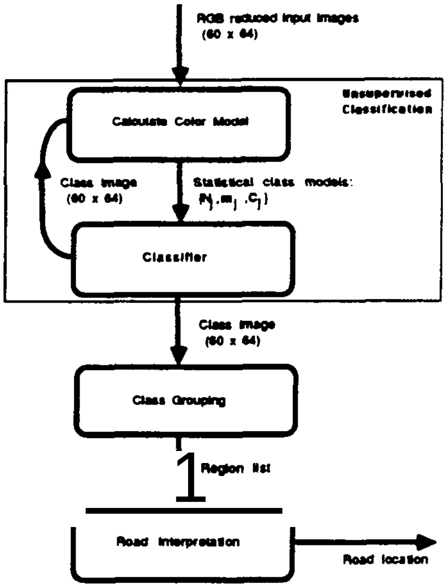

Figure 6: **UNSCARF** block diagram

classes, are collected into regions using a connected components algorithm. and polygon approximations are fit to the pixel regions. Finally we search for the combination of polygons that, taken together, forms the best road shape.

3.1 Unsupervised Classification Segmentation

The unsupervised classification algorithm groups pixels having similar colors and locations by an iterative clustering technique similar to the model update of the *SCARF* system. The main difference is that none of the pixels have a 'road' or 'non-road' label. The pixels are given an initial classification. Then a statistical model is calculated for each class, and the pixels are reclassified using the new model. This is repeated until the classes converge.

Each pixel of the input image has five features ($d = 5$) that are used in the clustering:

$$\mathbf{x} = [RED, GREEN, BLUE, row, column]^T.$$

This *can* easily be expanded *to* a eight dimensional space by using the RGB bands of a second image. The system has a fixed number of classes in each image, in *our* typically 24. First it labels each pixel of the image with a class, insuring that the classes are evenly scattered throughout the image. Next, a statistical color model, $\{N_j, \mathbf{m}_j, C_j\}$, is formulated for each class, ω_j, for this class assignment using equations (1) and (2). Then the pixels are labeled using a classification scheme similar to that of *SCARF* and a new statistical model is calculated. The 'classify/sample' loop is repeated until most of the pixels in the image remain in the same class. This usually converges quite quickly, taking between 8 and 15 iterations until 95% of the pixels remain in the same class.

The classification scheme *can* have several different flavors. The first scheme used was a nearest mean classifier. In other words, the pixels were labeled with the class whose mean was closest *to* the pixel value. This has a tendency to form spherical clusters in the feature space. Since we were using the spatial parameters of *(row. column)* all of the regions formed from the final class labeling have a approximately circular shapes in the image. To allow elliptical shapes to represent elongated or linear objects we used the nearest class as given by the Mahalanobis distance:

$$d_j = (\mathbf{x} - \mathbf{m}_j)^T C^{-1} (\mathbf{x} - \mathbf{m}_j).$$

This distance metric needed to be normalized since once one of the classes had a larger covariance than any of the other classes, then in the reclassification, even more pixels would be classified as the large class. This would balloon until all of

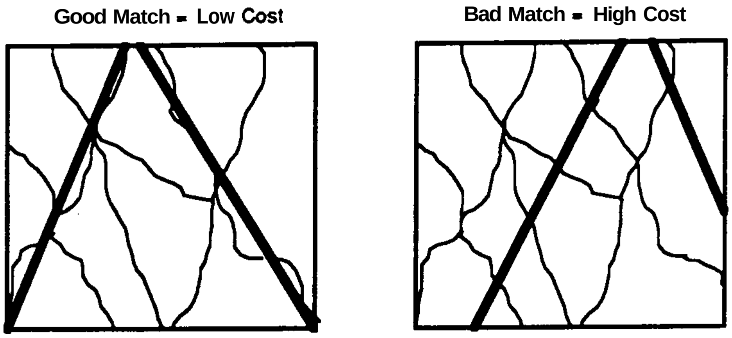

Figure 7: Road costs

the pixels were described by one class. To avoid this, we normalize the distance metric, by dividing each element of the covariance matrices of each class by the d^{th} root of the determinant of that mamx :

$$C'_{ij} = C_{ij}/||C||.$$

Then the Mahalanobis distance metric is calculated using C':

$$d_j = (\mathbf{x} - \mathbf{m}_j)^T C'^{-1}(\mathbf{x} - \mathbf{m}_j).$$

This allows each class to have a different elliptical shape, while maintaining the same size for each class and thus preventing one class from dominating the others.

Selecting the initial classes scattered throughout the image, causes the (row,*column)* parameter statistics to be almost identical for all classes. Therefore the initial classification is based solely on color. In later steps, however the *(row.* column) parameters *are* valuable. By clustering with color, we assume that an object will have a similar color throughout. By adding the *(row,column)* parameters, we are exploiting the constraint that objects are localized in the image. The positional constraint made segmentations cleaner than strictly color constraints. by eliminating small noisy regions.

3.2 Interpretation

The interpreter is based on evaluating all possible roads that will could appear in the image. The evaluation function looks at the difference between the road shape and the region edges in the image. The interpreter uses the same two road parameters as in SCARF: v, the column at which the road crosses the horizon, and θ, the angle between the center line and a vertical line in the image. However, instead of building an accumulator, we step through all of the interpretations and evaluate how well that interpretation fits the regions of the image.

To evaluate a candidate road, we first decide which regions would be part of theat candidate road. This is done by testing if the center of mass of the region lies on the road. All of the regions lying in the candidate are then grouped together and approximated with a conglomerate polygon. The area between the road model and the edge of the conglomerate polygon is used as a cost metric of the interpretation. The candidate road whose conglomerate polygon has the lowest cost is selected as the result. Figure 7 shows the cost metric of a *good* fit and a bad fit for the road.

3.3 Discussion and Future Work

In this system, the low-level segmentation uses mainly color constraints to segment the image, while the high-level interpretation uses only geometric constraints to localize the road. Therefore, the different levels of the system are using completely different constraints. Figure 8 shows an example of the unsupervised classification segmentation running on a road scene. The images to the right is the class image, where each class label is represented by a different intensity value. The left image shows the pixels colored by the mean values of their class labeling. The top pair of images is the initial scattered classification used to build the initial models. The middle pair of images shows the classification and mean class colors after two 'classify/sample' loops, and the bottom pair shows the results after five 'classify/sample' steps.

The advantage of the cost evaluation scheme of our interpretation is that new constraints *can* easily be added to the total cost. For example, we could add costs so that all of the road regions sould have a similar color, different that those

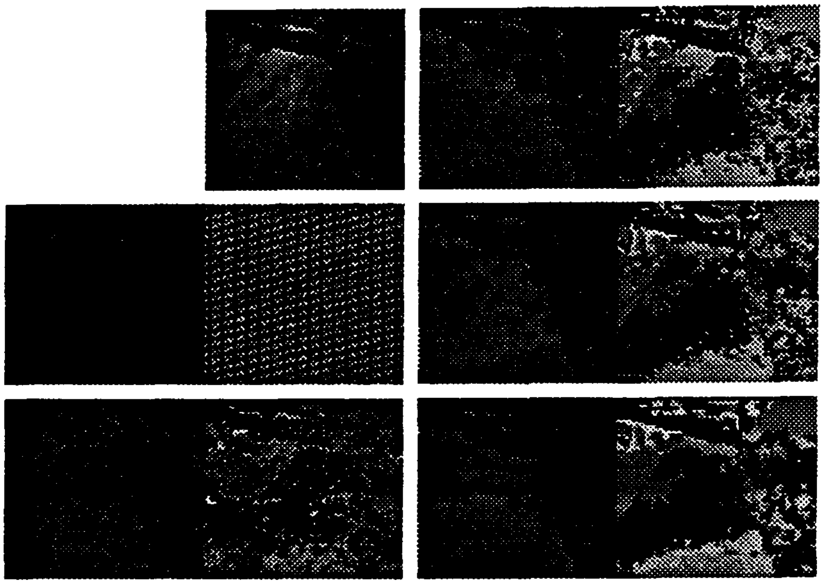

Figure 8: Example Clustering: Top left image is the original image. Each pair of class images corresponds to an iteration of the clustering algorithm. The right images have each class is assigned a different intensity, and the left images have each pixel is colored by the mean RGB class value. The bottom right pair is the final class images.

of the non-road regions. We could also add a cost insuring that the road region is similar in color to the road seen in the previous image. A cost can be added so that conglomerate polygons with straight edges are preferred over those with jagged edges. Although these additional cost have not been necessary on the images tested, they may become more important as we become more experienced with this interpretation system.

The system takes about 3-20 minutes to process one frame of the sequence. To speed up the processing, we have implemented the unsupervised clustering algorithm distributed on a multiple number of Sun workstations. Using this method, we have achieved a speed up proportional to the number of Suns used.

We will expand the unsupervised classification algorithm in several ways.

- First, if the system could decide the number of classes needed to characterize the data, rather than having a fixed number of classes specified, *then* the regions would be more representative of the data. As an initial attempt, we will split and merge regions after each reclassification step. Large regions will be split, and regions with very close mean values will be merged. This way, the system will decide how many regions it needs to adequately represent the data.
- We will expand the road interpretation to detect intersections. We will apply the road searching that we have currently implemented. Then we will enumerate all of the possible branches from the road, and search for intersection branches with the same cost evaluation method used for the main road. We may need to add a color constraint *to* the algorithm, since in our test site sometimes the shadows of the trees can form intersection shapes.
- We believe that the basic unsupervised segmentation algorithm described here *can* be used for many different vision applications. To show this we will use this system to do terrain typing for our cross-country navigation experiments.

4 Results and Conclusions

SCARF and UNSCARF have been prototyped and tested individually. Our current efforts include speed and algorithmic improvements to each system. We are also considering cooperation between SCARF and UNSCARF. One idea is to use

UNSCARF as a bootstrapping program and use SCARF as the general road-follower. If SCARF should realize that its results are not good, then control can be returned to UNSCARF.

The second possibility is to use lessons leamed from one system to improve the other. We intend, for instance, to track the colors of the road regions detected by UNSCARF. Analyzing the changes in colors over time may provide cues which can improve the supervised classification in SCARF. This combination of different methods for scene segmentation will continue to expand the Navlab's capabilities for paved roads, dirt roads, and for terrain typing for cross-country navigation.

References

[1] E. Arnould, H. T. Kung, O. Menzilcioglu, and K. Sarocky. A systolic array computer. In *Proceedings of 1985 IEEE International Conference on Acoustics. Speech and Signal Processing,* pages 232–235, March 1985.

[2] D. DeMenthon. *Inverse Perspective of a Road from a Single Image.* Technical Report CAR-TR-210, University of Maryland, 1986.

[3] E. Dickmanns and A. Zapp. Autonomous high-speed road vehicle guidance by computer vision. In *Proc. 10th* IFAC, Munich, 1987.

[4] E. Dickmanns and A. Zapp. A curvature-based scheme for improving road vehicle guidance by computer vision. In *Proc. SPIE Conference 727 on Mobile Robots,* Cambridge, 1986.

[5] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, Inc., 1973.

[6] T. Kanade, C. Thorpe, and W. Whittaker. Autonomous land vehicle at CMU. In *Proc. ACM* Computer *Conference,* Cincinnati, February 1986.

[7] C. Thorpe. M. Hebert, T. Kanade. and S. Shafer. Vision and navigation for the Carnegie-Mellon Navlab. *PAMI*, 10(3), 1988.

[8] M. Turk, D. Morgenthaler, K. Gremban. and M. Marra. VITS–a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* May 1988.

[9] R. Wallace, K. Matsuzaki, Y. Goto, J. Webb, J. Crisman, and T. Kanade. Progress on robot road following. In *IEEE Conf. on Robotics and Automation*, San Francisco. 1986.

[10] R. Wallace, A. Stentz, C. Thorpe, H. Moravec, W. Whittaker, and T. Kanade. First results in robot road following. In *Proceedings of IJCAI 85,* August 1985.