A REAL-TIME ROAD FOLLOWING AND ROAD JUNCTION DETECTION VISION SYSTEM FOR AUTONOMOUS VEHICLES

Darwin Kuan, Gary Phipps, and A-Chuan Hsueh

Artificial Intelligence Center
Central Engineering Laboratories
FMC Corporation
1185 Coleman Ave.
Santa Clara, CA 95052

ABSTRACT

This paper describes a real-time road following and road junction detection vision system for autonomous vehicles. Vision-guided road following requires extracting road boundaries from images in real-time to guide the navigation of autonomous vehicles on the roadway. We use a histogram-based pixel classification algorithm to classify road and non-road regions in the image. The most likely road region is selected and a polygonal representation of the detected road region boundary is used as the input to a geometric reasoning module that performs model-based reasoning to accurately identify consistent road segments and road junctions. In this module, local geometric supports for each road edge segment are collected and recorded and a global consistency checking is performed to obtain a consistent interpretation of the raw data. Limited cases of incorrect image segmentation due to shadows or unusual road conditions can be detected and corrected based on the road model. Similarly, road junctions can be detected using the same principle. The real-time road following vision system has been implemented on a high-speed image processor connected to a host computer. We have tested our road following vision system and vehicle control system on a gravel road. The vehicle can travel up to 8 kilometers per hour speed on the road.

I INTRODUCTION

There are increasing interests on intelligent navigation of autonomous vehicles in a complex environment as a technology development test bed to integrate artificial intelligence research on planning, reasoning, perception, mobility control, and learning. An autonomous vehicle needs to plan its action, perceive its surroundings, execute its plan, and adapt itself to the environment for survival. Given a high level mission goal, the planning system needs to generate a plan to achieve the goal. Based on this plan, the autonomous vehicle starts to execute the plan in the real world. It collects information from sensors to perceive its environment, to follow a road, to navigate through obstacles, to identify terrain types, to recognize objects and landmarks, and to understand scenes. If some unexpected events happen that interfere with the current plan, the autonomous vehicle needs to replan in order to adjust itself to the current situations.

There are several efforts on autonomous vehicle development at CMU, University of Maryland, Martin Marietta, and FMC. Under the Autonomous Vehicle Test Bed Program, we at FMC have developed a mission planning system [3] and a path planning system [1] on Symbolics Lisp Machines, a reflexive pilot system on SUN workstations [2], a high speed sonic imaging sensor, and a computer-controlled M113 armored personnel vehicle. The vehicle can perform real-time obstacle avoidance using the sonic imaging sensor at 8 kilometers per hour vehicle speed. In this paper, we describe our implementation of a real-time road following vision system that can follow a gravel road at 8 kilometers per hour vehcile speed.

Visual navigation of autonomous vehicles on road networks is an important problem. Results on vision-guided road following have been reported in [5] [6]. These approaches use a predictive edge tracking technique to follow paved roads. In the so called "feed-forward" mode [6], a prior detected road boundary taken together with the current vehicle motion, is used to predict the approximate location of important road features and place a window in a subsequent image. Only those pixels in the prediction window are processed. The detected edge location and orientation in the window combined with the road continuity constraint are sufficient to determine the next window location in the same image for road boundary tracking. Because only a small portion of the whole image needs to be processed, this approach significantly speeds up the computation. However, due to the sequential nature of the road boundary tracking operation and its heavy reliance on prediction, the road boundary tracker may be confused by shadows, vehicle tracks and tire marks, and fuzzy road boundaries to lock on the wrong edge features.

Our autonomous vehicle is a tracked vehicle (M113 armored personnel carrier) that usually travels on dirt or gravel roads with fuzzy road boundaries and many vehicle tracks. These conditions make the use of prediction difficult. Consequently, we take a consistency checking approach that aggregates all the consistent evidence to reach a final interpretation. No attempt is made to optimize the image segmentation algorithm. Instead, we put our emphasis on developing a geometric reasoning module that can accurately identify road segments and road junctions based on imperfect image segmentation results.

The vision system operates in a loop (see Figure 1). It first acquires a color image from a camera and the current vehicle location from an inertial navigation system. The image segmentation module uses a pixel classification algorithm to segment the image into road and non-road regions. The road boundary tracking module finds the most likely road region and traces the contour of the region. The contour is then represented as a sequence of line segments using a line fitting algorithm. These line segments are then transformed from the image coordinate system to the local vehicle coordinate system and sent to the geometric reasoning module. The geometric reasoning module aggregates local geometric supports and assigns a consistent interpretation to these line segments. The resulting road interpretation is then fused with other sensor interpretation results (e.g., obstacle map from range sensor) and sent to the pilot system to generate a local path and perform the actual vehicle navigation.

We have implemented this vision system on a high-speed image processor connected to a host computer. All the image segmentation functions are implemented on the image processor that operates at 30 frames per second. All the road boundary tracking, line fitting, and geometric reasoning functions are implemented on the host computer. The vision system currently takes approximately three seconds to process each road image. The pilot system takes the road description and generate a local path within 200 ms. We have successfully integrated the vision, planning, and pilot systems with the vehicle control system and the vehicle can travel at 8 kilometers per hour vehicle speed on a gravel road.

II ROAD IMAGE SEGMENTATION

The vision system first acquires the blue image from a color camera. The reason for selecting the blue band is because it gives the best result for distinguishing the road from the background. We use a pixel classification technique to segment the image into road and non-road regions. There are four possible classifications for each pixel:

1. an actual road pixel is classified as road
2. an actual non-road pixel is classified as non-road
3. an actual road pixel is classified as non-road
4. an actual non-road pixel is classified as road.

The first two cases are correct classifications and the last two correspond to miss and false alarm respectively [4]. Cost factors are defined for each case and the classifier is designed to minimize the total cost. The resulting classifier is the ratio of two conditional probability density functions of pixel intensity distribution - one is conditional on the hypothesis that all the pixels are from the road class and the other is conditional on that all the pixels are from the non-road class. A pixel is classified as road if the conditional probability ratio of its intensity value is greater than a threshold that is determined by the cost factors and the a priori probabilities of the road and non-road classes.

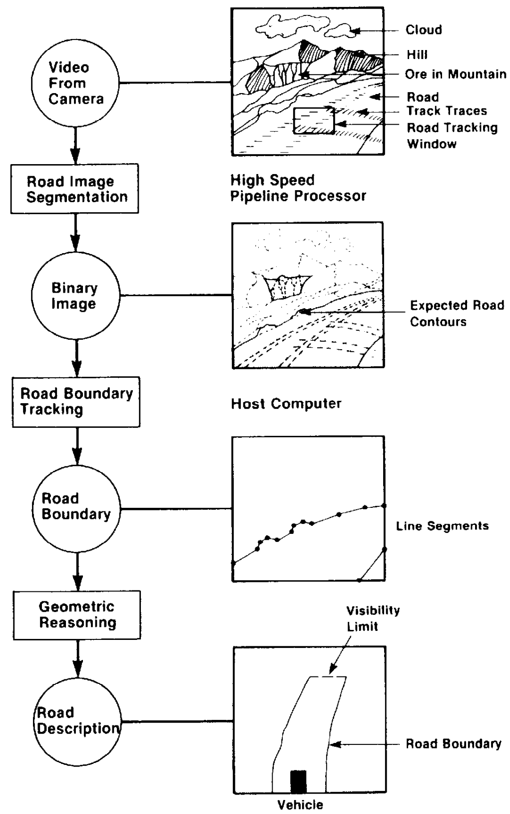

Figure 1: road following vision system architecture

For each new image, a properly selected reference window, usually positioned at the center bottom of the image, is used to lock on a portion of the road. The normalized histogram of the pixels inside the reference window is calculated and used to approximate the conditional probability density function of the road pixel class. The normalized histogram of the whole image is then used to approximate the probability density function of pixel intensity for road plus non-road background. The conditional probability density function of the non-road class can be obtained as a weighted linear combination of the two histograms according to the assumed a priori probabilities of the road and background. These conditional probability density functions are then substituted into the classifier to set up a lookup table for pixel classification in the current image. The segmented image is then smoothed to remove noise pixels and fill gaps.

III ROAD BOUNDARY TRACKING

The function of the road boundary tracking module is to find the most likely road region based on segmentation results and track its boundary.

The image segmentation module returns a segmented binary image that contains several classified road regions. It takes a lot of time to track the boundary of each region in the image. Because the real road region is usually large compared to misclassified noise regions and spreads across the image near the bottom of the image, it is most likely to intersect with a vertical scan line starting at the center bottom of the image. The road boundary tracking module uses this heuristic and scans along the center column from the bottom of the image. If there is a road class region, the road boundary tracking module starts to follow the region contour until it returns to the same starting point. If the region contour length is greater than a threshold, then it is assumed to be the actual road region. Otherwise, the road boundary tracker continues to scan and track the next road class region.

The road region contour is then represented in terms of a sequence of line segments by using a line fitting routine. These line segments are then sent to the geometric reasoning module for detailed shape analysis.

Figure 2 shows a typical road image obtained from camera. Figure 3 shows all the detected road class regions in the region of interest using the pixel classification algorithm. The large region with linear border is selected by the road boundary tracker as the road region. The vectors in the center of the picture show the projected road boundaries on the ground plane. The quadrangle that bounds the projected road boundaries delimits the visibility limit and the camera's field of view.

IV GEOMETRIC REASONING

The image segmentation module extracts road regions only based on local intensity variation without reasoning about the global geometric properties of the road boundary. There are many situations where the image segmentation module does not work properly. For example, different lighting conditions, seasonal changes, puddles on the road, shadows on the road, just to name a few. The development of more sophisticated image segmentation techniques is certainly important. However, in some situations, geometric reasoning can eliminate erroneous data based on road model and shape analysis. The image segmentation results are what the vision system "sees." The geometric reasoning results are what the vision system "perceives."

The road model we use for geometric reasoning can be described in terms of three constraints:

1. road sides consistency constraint - each road edge on the left side has at least one right side road edge that is parallel to and overlaps (along its orientation) the given edge.

2. smoothness constraint - both the left and right sides of a road change direction smoothly even for curved road.

Figure 2: typical road image.

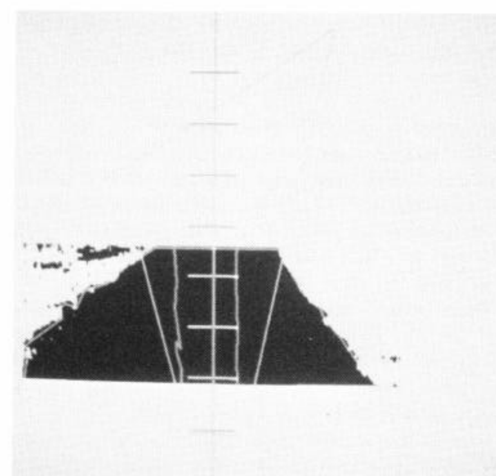

Figure 3: segmented road image and the projection of the road boundaries on the ground plane.

3. continuity constraint - a road spans continuously on the ground plane, therefore, continuity between road boundaries exists in two images taken in sequence.

The geometric reasoning module needs to use this road model to

- find the road left and right boundaries
- check the road sides consistency constraint
- check the smoothness constraint on each side of the road

- check the road continuity constraint between image frames
- return a road interpretation and its goodness factor.

A. Find Road Sides

Given a polygonal description of a road region contour, we first transform these line segments from the image coordinate system to the local vehicle coordinate system since the road model constraints are defined on the ground plane. This transformation is obtained by first calibrating the camera and assuming the road is on the ground plane, then projecting image points to the ground plane.

To determine the left and right sides of a road, we first find the closest and farthest points of the road boundary on the ground plane. These two points divide the road boundary into two parts - the left and right road boundaries.

B. Road Sides Consistency Constraint

The corresponding edge segments on two sides of a road are locally parallel. This property is used to remove erroneous road regions included due to imperfect image segmentation. For each edge segment on the left side, we check to see if there is any right side edge segment that supports the road model. That is, the two segments are locally parallel and have the correct distance interval between them. If there is one, then the amount of overlap along their orientation and other geometric information are recorded in the segment support structure. This is done for each edge segment on the left and right sides. If an edge segment has sufficient support from the other side to cover its extent, then it is labeled as consistent.

If both sides of the road are smooth and every edge segment has support from the other side, then they are used as the final road interpretation. However, If some edge segments do not have geometric support from the other side, then we start to trace each side of the road to find consecutive consistent edge segments. If there is a break between two sequences of consistent edge segments, a "perceived" edge segment is created to link the two sequences and the original edge segments in between are removed. The road sides consistency constraint is then slightly relaxed and applied to the newly created "perceived" edge segments to make sure that they agree with the road model. This approach has the ability of data selection before model fitting. Locally consistent data are selected to reach a global interpretation, while inconsistent data are thrown away before interpretation.

This step of geometric reasoning makes the road following vision system capable of working with imperfect segmentation results. Typical cases it can handle includes shadows casted on the road and fuzzy road boundary. Figure 4 shows a puddle on the right side of the road. Figure 5(a) shows the road boundary on the ground plane. Figure 5(b) shows the final road interpretation after geometric reasoning with the newly created "perceived" edges drawn in dashed lines.

Figure 4: segmented road image with a puddle on the right side.

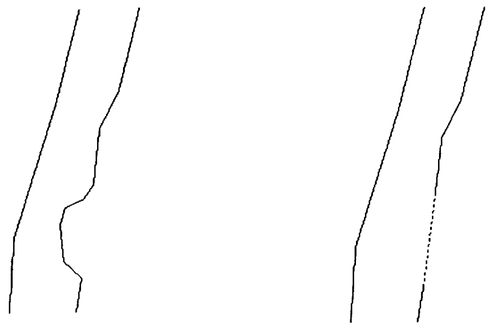

Figure 5: (a) road boundaries before geometric reasoning. (b) final road interpretation.

C. Smoothness Constraint

Typical road boundary changes direction very smoothly. The geometric reasoning module checks the left and right road boundaries and returns a smoothness factor for each side. To check the smoothness constraint, the angle between two adjacent edge segments is calculated. If the angle is greater than a threshold that is a function of edge segment length, then the edge segments are labeled as not smooth. The reason to make the angle threshold vary according to edge segment length is to allow more tolerance for short edge segments. The smoothness factors of road sides are then calculated as a normalized measure of the smoothness factors of its component edge segments.

D. Continuity Constraint

The two constraints we discussed are applied in a single image frame. The continuity constraint is applied between adjacent image frames to enforce consistency in the time axis. This is useful in several ways. First, if the adjacent frame road boundaries are not consistent (e.g., there is no smooth transition between road segments), a warning is signaled to the road following system to slow down the vehicle. In this case, if both image frames have consistent road interpretation, the new frame road boundary is used because the current information is more accurate than the old road information. Second, continuity between frames is also used to evaluate the goodness of each road side in the current image. This makes the road following system work even if only one side of the road is visible.

V ROAD JUNCTION DETECTION

Visual navigation of autonomous vehicles on a road network not only needs to follow a single road, but also detect road junctions and turn to one of the intersecting roads. Recent results on road junction detection are reported in [5]. In this approach, road junction appearance is first predicted based on the vehicle location and a road network map. Prominent road junction features are then used to guide the match of image features detected. In here, we only use a general road junction model without map prediction.

If there is a road junction on the map and we want to turn to another road, the planning system will issue a road junction detection task to the vision system when the vehicle is near that region. This task command will trigger the road junction detection module in the vision system to perform additional road junction consistency constraint checking. On the other hand, if the vehicle wants to stay on the same road, the road following vision system will automatically treat the road junction region as erroneous data and try to ignore it.

The road junction detection algorithm is very similar to the road sides consistency constraint technique we discussed in the last section. If the road junction detection module is not triggered, the road following system will treat junctions as imperfect road regions and the smoothness and road sides consistency constraints will remove them to form a final road interpretation. However, if the road junction detection module is triggered, instead of removing edge segments that do not have support from the other side, it tries to find support from edge segments on the same side. If it successfully finds supports for these edge segments, then they are the boundaries of the other road. In principle, this will work; however, in our case, road junctions usually have round corners and grass and trees may break the other road at the junction. We currently use more relaxed constraint that only checks if the perceived edges on both sides of the road support each other. Figure 6 shows a road junction scene. Figure 7(a) shows the road region and junction boundary. Figure 7(b) shows the final road interpretation and the perceived edges in dashed lines. In this case, edges on the same side of the road do not provide enough supports for junction detection. However, the perceived edges on two sides of the road support each other and is a weak evidence of the existence of a road junction. The road junction detection module and part of the geometric reasoning module are still in the experimental stage and is currently in the process of being optimized for real-time operation.

VI PILOT SYSTEM

Given a road scene description from the vision system, the pilot system is responsible for guiding the vehicle to follow the road and avoid obstacles. The pilot system used is a real-time reflexive pilot described in [2]. The road scene model contains left and right road boundaries and an artificial visibility limit placed at the end of the road. Candidate subgoals are positioned on the visibility limit line segment. A subgoal is found to be reachable by the vehicle without getting off the road if its left and right limiting rays bound a non-empty free-space cone. For each subgoal, a local path from the vehicle to the subgoal is generated in terms of executable vehicle commands and the subgoal that maximizes a predefined objective function is selected for execution. The pilot system currently takes approximately 200 ms to process one road scene. Figure 8, 9, and 10 show a time sequence of autonomous road following action.

VII CONCLUSIONS

In this paper, we described the implementation of a real-time road following vision system for autonomous vehicles. We have integrated the vision, planning, and pilot systems with the vehicle control system and the vehicle can travel at 8 kilometers per hour vehicle speed on a gravel road. We are currently working on obstacle avoidance on the roadway by fusing information obtained from a color camera and a sonic imaging sensor. We are also reimplementing our road following vision system on a more powerful pipeline image processor to achive 20 km/hr road following.

Figure 6: segmented road image with road junction.

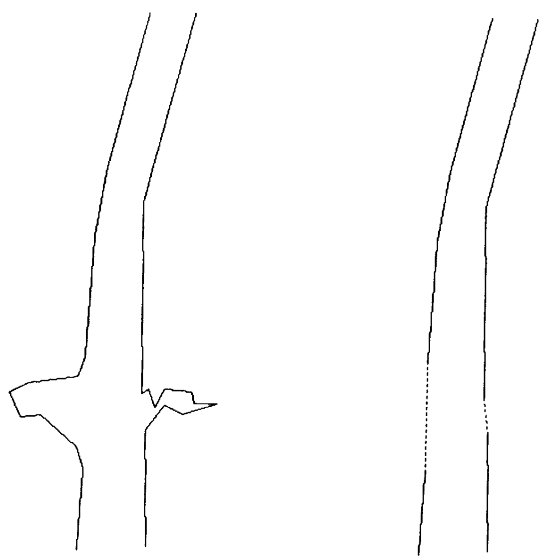

Figure 7: (a) road and junction boundaries before geometric reasoning (b) final road interpretation.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the encouragement of FMC management especially Andy Chang and Lou McTamaney. We would also like to thank the superb vision software support from Mary Cole and Darrell Smith, and the pilot software support from John Nitao and Steve Quen.

REFERENCES

[1]. Kuan, D., Brooks, R. A., and Zamiska, J. C., "Natural Decomposition of Free Space for Path Planning," Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, Missouri, March 1985.

[2]. Nitao, J. J., Parodi, A. M., "A Real-Time Reflexive Pilot for an Autonomous Land Vehicle," IEEE Control Systems Magazine, December 1985.

[3]. Pearson, G., and Kuan, D., "Mission Planning System for an Autonomous Vehicle", Proceedings of the IEEE Second Conference on Artificial Intelligence Applications, Miami, Florida, December 1985.

[4]. Van Trees, H. L., *Detection, Estimation, and Modulation Theory: Part I*, Wiley, New York, 1968.

[5]. Wallace, R., Matsuzaki, K., Goto, Y., Crisman, J., Webb, J., and Kanade, T., "Progress in Robot Road-Following," Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, April 1986.

[6]. Waxman, A. M., Le Moigne, J., and Srinivasan, B., "Visual Navigation of Roadways," Proceedings of the 1985 IEEE International Conference on Robotics and Automation, April 1985.

Figure 8: road-following sequence (first image).

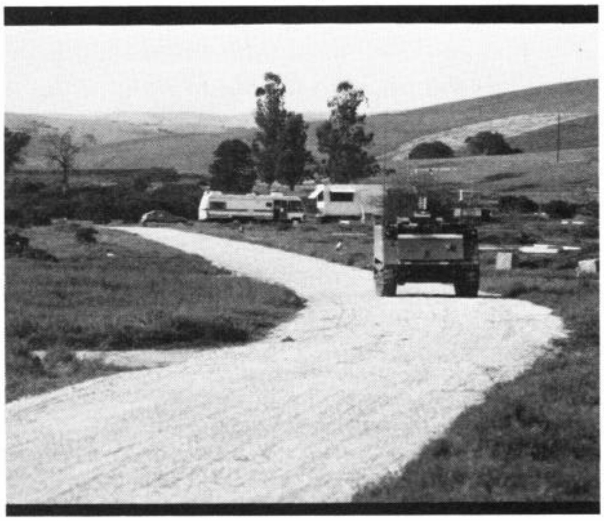

Figure 9: road-following sequence (second image).

Figure 10: road-following sequence (third image).