Progress In Road Intersection Detection For Autonomous Vehicle Navigation

Todd R. Kushner, Sunil Puri

Todd R. Kushner, Sunil Puri, "Progress In Road Intersection Detection For Autonomous Vehicle Navigation," Proc. SPIE 0852, Mobile Robots II, (1 January 1987); doi: 10.1117/12.968232

Event: Advances in Intelligent Robotics Systems, 1987, Cambridge, CA, United States

Progress in Road Intersection Detection for Autonomous Vehicle Navigation

Todd R. Kushner
Sunil Puri

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

Abstract

The autonomous land vehicle (ALV) has successfully navigated simple non-branching roads. To traverse more complicated roadways, the ALV must be able to detect road intersections.

We present two methods of road intersection detection. The first recognizes an image road intersection by matching image road boundary data with a predefined intersection model. The predefined model is the intersection configuration derived from an *a priori* map database. Road boundaries are extracted from the image and processed to produce a set of linear segments. A heuristic method produces a set of combinations of these segments that correspond to segments in the model. The best match is determined by a minimum goodness-of-fit measure. The method was tried on a Y-intersection. The ALV uses appropriate models derived from a map database.

A second method, rather than recognizing the image road intersection, finds corridors of free space that the ALV can navigate through. A road height profile obtained from a range scanner is used to find a free space region. The height profile, representing height and normal road surface angle, is broken into solid horizontal planar segments. In a segmented image, distances to the road boundary in each of five principal directions (east, northeast, north, northwest, and west) are found for each road point. These image distances are converted to distances between the road point and the world plane boundary in each direction using the road height profile. The smallest directional distance approximates the world road boundary distance. We present results on synthetic and real road intersection images.

Introduction

Two goals of the DARPA-supported Vision Based Navigation for Autonomous Vehicles project at the University of Maryland have been the development of a vision system for autonomous navigation of roads and road networks, and support of Martin Marietta Aerospace, Denver, the integrating ALV program contractor [1]. Other research being done for the ALV program includes CMU's NAVLAB [2], which involves building a mobile vehicle laboratory for understanding color video and range images to allow the vehicle to drive over narrow asphalt paths near campus while avoiding obstacles.

An upcoming requirement in Martin Marietta's schedule of demonstrations is the negotiation of road intersections. Martin Marietta has done work on road-following for video images of straight roads [3]. This work has involved building symbolic descriptions of road boundaries using segmentation of color video and range imagery. Scene models are built by segmenting video images, extracting boundaries, and transforming boundaries in the image plane to a vehicle-centered three-dimensional system. Other work on segmenting color video and range imagery for the ALV project includes [4], [5]. The paved road of the ALV onroad test area intersects the following other types of traveled routes: paved roads, dirt roads, off-road vehicle trails, dirt paths, paved parking lots, and dirt parking lots.

We have developed a system that, using a road network map like that compiled by the U.S. Army Engineer Topographic Laboratories (ETL), can recognize paved road intersections in processed image data. In the event that the map indicates that the vehicle is approaching an intersection, the vision system has available precomputed predictions, indexed by world road coordinates, concerning the structure of the intersection in the immediate vicinity of the vehicle. Due to uncertainties in vehicle position and the limited accuracy of such map information, the vision system can use this information only in a qualitative fashion. The aforementioned CMU work addresses detecting intersections using a map. That work, however, uses a camera to sight intersections and sidewalks and uses these sightings to correct the estimate of the vehicle's position with respect to a campus map including intersections and sidewalks, rather than using the back-projected predicted geometry of intersections as an aid to recognition.

We present two algorithms for recognizing road intersections in visual images. The first classifies image road boundary points on the intersecting and current road by matching those points against a model of the road predicted from the map data. The information available to the vision system about the structure of the intersection from the map includes the widths and angles between the road boundaries at the intersection. The algorithm makes use of this information by making a series of hypotheses about which boundary points belong to which road segments, and then evaluates how well the road widths and angles resulting from these assignments match the road widths and angles predicted from the model. The second method detects navigable regions through a road intersection by converting a representation of the distances in the image between each road point and the nearest boundary to a representation of the distances between points on the road and the boundary in the ground plane. This method requires a road height profile from a range scanner, such as the ERIM laser range scanner developed for the ALV program [6]. A region sufficiently far from the road boundary that a vehicle can safely pass through it is identified and a detailed path through this region planned.

Boundary-fitting Method

The road intersection is observed from a camera mounted on top of the ALV as it approaches the road intersection from one of the branch roads. We assume that a model of the road intersection, obtained from *a priori* knowledge of the scene contained in a map database, is available.

The issues to be addressed in detecting the road intersection from the image data are :

(i) Given the extracted road boundaries from the image, which image features describe the scene?

(ii) How should the matching between image features and object models be done in order to recognize the road intersection in a scene?

There are three steps in the road boundary extraction process. The road is first segmented in the image. Next, the boundary of the road is traced. These boundary points are sampled at fixed intervals to reduce noise. Then, the sampled boundary points are back-projected onto a flat world plane.

The model of the road intersection that we use is two-dimensional, constructed from the geometric properties of the intersection projected from a map database. Geometric properties, e.g., widths and angles between intersecting roads, are computed from an overhead view of the three-dimensional back-projected map intersection.

Widths and angles between extracted boundaries of the intersecting roads were used as features in the scene to describe the object being modeled. The problem is attacked by combinatorially generating hypotheses about which boundary points in the image belong to which road segments in the model. Line segments are fit to these selected groups of boundary points. The road widths and angles of a hypothesis are measured, and these road widths and angles are evaluated against a model to choose a best match.

The example presented is a Y-intersection, so there are three road segments in our model. Two segments correspond to the intersection of the Y-intersecting road and the current road; one segment corresponds to the current road side opposite the Y-intersecting road. Each of the first two segments is further subdivided into two connected line segments, one corresponding to the Y-intersecting road and one corresponding to the current road (Figure 1). The purpose of matching is to simultaneously determine, for the two split segments, how to partition their corresponding data points into sets that best match the connected line segment.

In the image, the vehicle is pointed down the center of the road with the Y intersecting road going off the left image side. The segmented image border following produces three data segments: 1) one segment beginning where the left side of the main road enters the bottom of the image and ending where the Y-intersecting road exits the left side of the image (lower); 2) one segment from where the Y-intersecting road exits the left side of the image (upper) to where the main road intersects an artificial horizon drawn across the top of the image (left); 3) one segment from where the main road intersects the artificial horizon (right) to where the right side of the road exits the image (bottom; Figure 2).

Segments of the data are matched with the model using pairs of angles and distances between the two shapes, and figures of merit are assigned to these matches, as done in [7]. These features include the distances D_{mn}, D_{lm}, D_{ln}, and the angles θ_l and θ_m (Figure 1).

A *breakpoint* is defined as a point on one of the data segments that corresponds to one of the split model segments that partitions the data segment into two pieces. To find the best breakpoints to match the two model line segments, breakpoints for the two data segments are selected combinatorially, and lines are fit to the pieces of data that the breakpoints separate. The features d_{mn}, d_{lm}, d_{ln}, ϕ_l and ϕ_m, corresponding to D_{mn}, D_{ln}, D_{lm}, θ_l and θ_m of the model, are computed between these fit lines, and a goodness-of-fit is given by

$$E = a\left(\frac{\left|D_{mn} - d_{mn}\right|}{d_{mn}} + \frac{\left|D_{ln} - d_{ln}\right|}{d_{ln}} + \frac{\left|D_{lm} - d_{lm}\right|}{d_{lm}}\right) + b\left(\frac{\left|\theta_m - \phi_m\right|}{\phi_m} + \frac{\left|\theta_l - \phi_l\right|}{\phi_l}\right) \quad (1)$$

where a and b are user specified parameters (which for our purposes were set to 1.0). Every possible combination of breakpoints was considered and the combination with the minimum value of E was selected. Figure 3 shows the results of applying this algorithm to a sample intersection (with a goodness-of-fit value of 2.47).

Since the Y-intersecting road is curved, a more complex model shown in Figure 4 was tried. This model constrains segments to enclose the road area. The search space now consists of all combinations of four breakpoints. An additional feature, the length of the line fit to the data between the two breakpoints, was added to the matching. The search space was reduced with this feature by requiring that the line fit to the data between the two breakpoints must be within a prespecified error of this length. Figure 5 shows the result of matching this model to the Y-intersection using the new goodness-of-fit measure

$$F = a\left(\frac{\left|D_{mn} - d_{mn}\right|}{d_{mn}} + \frac{\left|D_{ln} - d_{ln}\right|}{d_{ln}} + \frac{\left|D_{lm1} - d_{lm1}\right|}{d_{lm1}} + \frac{\left|D_{lm2} - d_{lm2}\right|}{d_{lm2}}\right) + b\left(\frac{\left|\theta_{m1} - \phi_{m1}\right|}{\phi_{m1}} + \frac{\left|\theta_{m2} - \phi_{m2}\right|}{\phi_{m2}} + \frac{\left|\theta_{l1} - \phi_{l1}\right|}{\phi_{l1}} + \frac{\left|\theta_{l2} - \phi_{l2}\right|}{\phi_{l2}}\right) \quad (2)$$

The ALV, knowing its position in the world, can query the map database to determine what type of road intersection should appear in the scene and then use the appropriate model for matching with the image.

Distance Method

A second method finds corridors of free space in an intersection that the ALV can navigate through. Finding corridors of free space through which a vehicle will travel is part of *intermediate-range navigation*, by which a path is planned for a vehicle to travel from the current position to the location of the goal [8]. The ALV does this by finding distances in the image from any point in the road to the edge of the road in a set of fixed directions. Then, with a road height profile obtained from a range scanner, these image distances can be projected into world plane distances with inverse perspective geometry. The minimum of these directional world plane distances can be taken as the free space distance for that point in the road. In our model, we simplify the height profile of the road by breaking it into horizontal planar segments. Figure 6 shows a Y-intersection superimposed on a synthetic staircase terrain, with shading exaggerated to emphasize the angles between ground plane segments. Figure 7 shows a thresholded version of the synthetic image. The algorithm was implemented on a combination of a fast pipelined image processor, a Vicom, and a host computer, a VAX-11/785. The image distance from any road point to the border is determined for each of five principal directions: east, northeast, north, northwest, and west. In our example, since the Y-intersection contains roads that are vertically oriented in the image, these five directions are adequate to find an accurate road boundary distance. The image distances were computed on the Vicom. Figures 8 a-d, f show the image distances for the five directions for the segmented image. These distances were obtained on the Vicom by eroding points from the image border using fast pipelined image-processing operations, keeping count of the number of points so far eroded from the border at any point (the image distance), and summing the points in the image to detect when the all the points have been processed (completely eroded). The erosion is accomplished by blurring the image in a given direction and thresholding the image to remove the blurred border (both one frame-time operations). In a second image, those points corresponding to the eroded points are accumulated, each labeled with the value corresponding to the number of points so far eroded, the image distance. The image after the erosion can be quickly summed on the Vicom using a column-sum operation, and these column sums can be checked to see if the image has been completely eroded for that direction.

These distances are projected onto the world plane using the height profile from the range scanner with the help of a host VAX-11/785 with fast floating-point hardware. For a given height profile, a lookup image of the world distance for a unit pixel distance at each road point of the image in each of five principal directions is computed. Then, the image distances are converted to world distances by integrating along the appropriate direction in the lookup image for the image distance. Partial sums are maintained during integration so that later stages of the computation are speeded up. The lowest world distance for every direction at a point is taken as the final world boundary distance. This distance can be thresholded at, e.g., two meters from the edge of the road, to give navigable tracks of free passage through the intersection. Figure 8 e shows the final world road boundary distance for the synthetic image, contoured so that distances are displayed modulo two meters. Figure 9 a-h shows the method applied to a real road intersection image, using an estimated road height profile, also, contoured so that distances are displayed modulo two meters. Sample processing times for the Vicom are 28-32 seconds for one-time initialization and 128-150 seconds for processing a synthetic or real 128x128 image. These processing times, however, could be greatly speeded up on a computer that combines both a faster pipelined-processing capability with floating-point hardware, such as the WARP [9].

Conclusions

We have presented a model-based object recognition method for detecting road intersections in images and a distance-based method for detecting corridors of free space through intersections in images. The model-driven technique produces a set of parameter values for each hypothesized location of intersection in the data and then uses a numerical goodness-of-fit to determine how well the model matches the data. Further work includes reducing the search by ordering the combinations of breakpoints to be searched (for example, by first choosing breakpoints where the angle between the lines fit to the data pieces best matches the angle predicted by the model) and terminating the search when the goodness-of-fit value reaches a threshold. Further work for the distance-based method includes reimplementing the technique on fast pipelined floating-point computers such as the WARP, and testing the technique on real images with road height profiles taken from actual range scanners.

References

[1] S. Chandran, L. Davis, D. DeMenthon, S. Dickinson, S. Gajulapalli, S. Huang, T. Kushner, J. LeMoigne, S. Puri, T. Siddalingaiah, and P. Veatch, "An Overview of Vision-Based Navigation for Autonomous Land Vehicles 1986," University of Maryland Center for Automation Research TR-285, April, 1987.

[2] C. Thorpe, S. Shafer, and T. Kanade, "Vision and Navigation for the Carnegie Mellon NAVLAB," *Proc. DARPA Image Understanding Workshop*, Los Angeles, February, 1987.

[3] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "Video Road-Following for the Autonomous Land Vehicle," *Proc. IEEE International Conference on Robotics and Automation*, Raleigh, NC, April, 1987.

[4] K. Olin, F. Vilnrotter, M. Daily, and K. Reiser, "Developments in Knowledge-Based Vision for Obstacle Detection and Avoidance," *Proc. DARPA Image Understanding Workshop*, Los Angeles, February, 1987.

[5] M. Daily, J. Harris, and K. Reiser, "Detecting Obstacles in Range Imagery," *Proc. DARPA Image Understanding Workshop*, Los Angeles, February, 1987.

[6] R. Dunlay and D. Morgenthaler, "Robot Road-Following using Laser-Based Range Imagery," *Trans. SME Second World Conference on Robotics Research*, Scottsdale, AZ, August, 1986.

[7] L. Davis, "Shape Matching Using Relaxation Techniques," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-1** (1), pp. 60-72, January, 1979.

[8] A. Waxman, J. LeMoigne, L. Davis, B. Srinivasan, T. Kushner, E. Liang, and T. Siddalingaiah, "A Visual Navigation System for Autonomous Vehicles," *IEEE J. on Robotics and Automation*, **RA-3** (2), pp. 125-141, April, 1987.

[9] H. Kung and J. Webb, "Global Operations on the CMU Warp Machine," *Proc. 1985 AIAA Computers in Aerospace V Conference*, October, 1987.

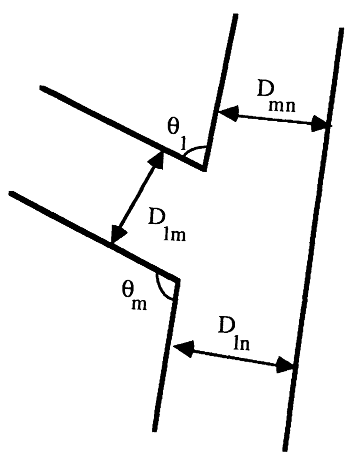

Figure 1 Model for the Y-shaped road intersection.

Figure 2 Road boundary points projected from the image onto a flat ground plane.

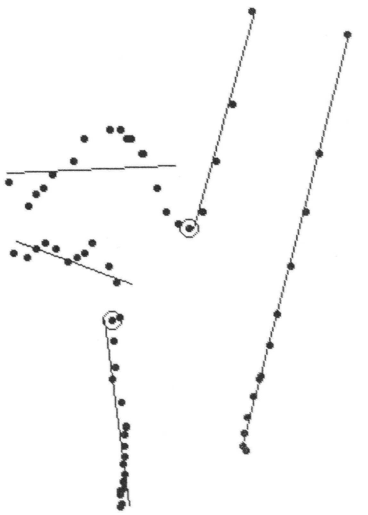

Figure 3 Result of matching the model to the data. The circled points in the data are the *breakpoints*.

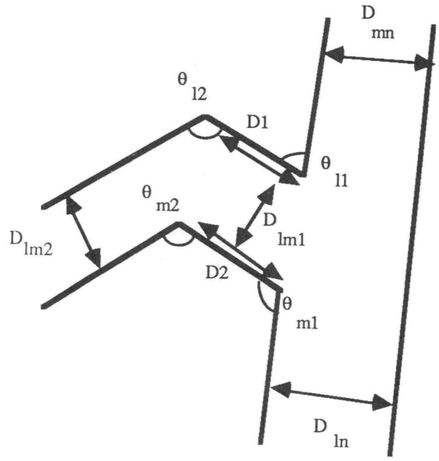

Figure 4 A more complex model for the Y-shaped road intersection.

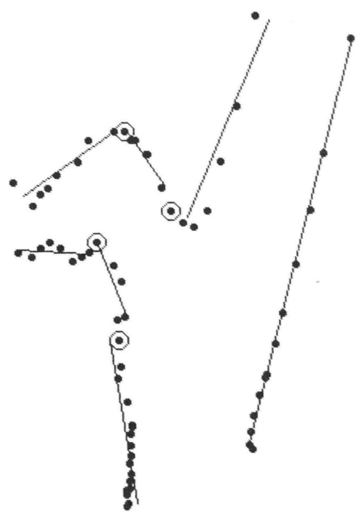

Figure 5 Result of matching the model shown in Figure 4 to the data. The circled points in the data are the *breakpoints*.

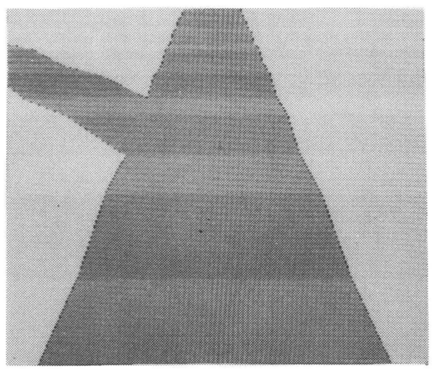

Figure 6 Y-intersection Superimposed on Synthetic Height Profile.

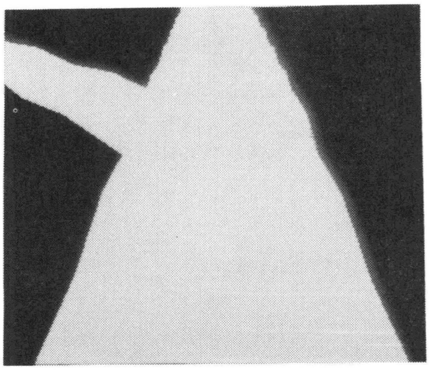

Figure 7 Thresholded Synthetic Image.

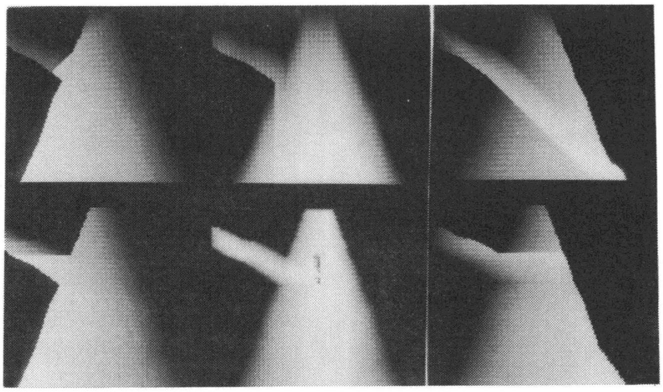

Figure 8 Image Distances for Five Directions in Segmented Synthetic Image: a) northeast direction; b) north direction; c) northwest direction; d) east direction; e) final world road boundary distance image; f) west direction.

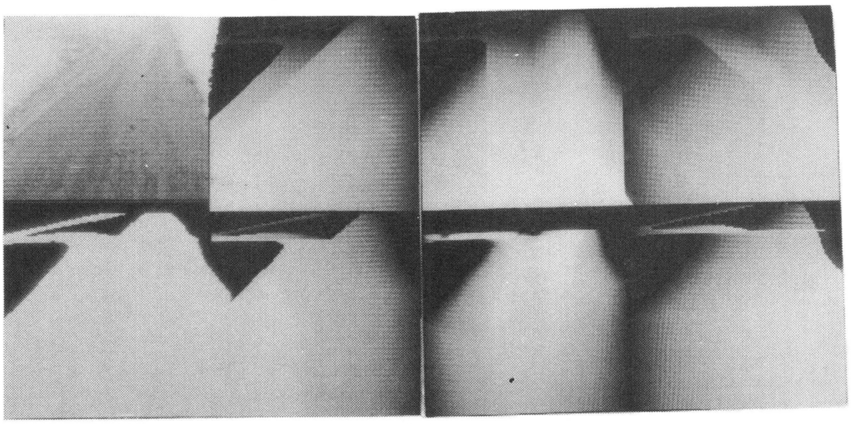

Figure 9 Image Distances in Segmented Real Image: a) Y-intersection image; b) northeast direction; c) north direction; d) northwest direction; e) thresholded Y-intersection image; f) east direction; g) final world road boundary distance image; h) west direction.