Extracting Road Curvature and Orientation From Image Edge Points Without Perceptual Grouping Into Features

Karl Kluge

Artificial Intelligence Laboratory, University of Michigan, 1101 Beal Avenue, Ann Arbor, MI 48109-2110, USA

Phone: (+1)(313) 763-5439; FAX: (+1)(313) 763-1260; email: kckluge@eecs.umich.edu

Abstract

The ARCADE algorithm (Automated Road Curvature And Direction Estimation) estimates road curvature and tangential road orientation relative to the camera line-of-sight. The input to ARCADE consists of edge point locations and orientations extracted from an image, and it performs the estimation without the need for any prior perceptual grouping of the edge points into individual lane boundaries. It is able to achieve this through the use of global constraints on the individual lane boundary shapes derived from an explicit model of road structure in the world. The use of the Least Median Squares robust estimation technique allows the algorithm to function correctly in cases where up to 50% of the input edge data points are contaminating noise. Two applications of ARCADE as the first stage of processing for a lane sensing task are described in this paper. The first application is extraction of the locations of the features defining the visible lane structure of the road. The second application is the generation of training instances for an ALVINN-like neural network road follower without the need for a training signal from a human driver, and without the need for any camera calibration beyond the location of the horizon row in the image plane.

1 Introduction

The ability to detect lane boundaries in color images of a road scene is an enabling or enhancing technology for a number of driver's assistant applications. Among these applications are lane excursion warning, headway maintenance for intelligent cruise control, and cambered power steering. Lane detection, (the extraction of the lane structure visible in an image without a prior model of that specific road) is distinct from lane tracking (tracking lane boundaries from image to image based on a prior model of the lane structure of that road, as performed by systems such as the system developed at NIST [13], VaMoRs [1], VITA [3], and YARF [7]).

There are a number of desirable properties which should be possessed by a solution to the lane detection problem. First, it should handle curved lane boundaries rather than assuming straight lane boundaries. Second, it should be able to handle the full range of ways in which a lane boundary can be marked: painted lines, whether single or double, solid or broken; and pavement edges. Third, it should take advantage of global scene constraints as a way of improving robustness in the face of noise in the images. Finally, it should produce an explicit measure of the reliability of the result it has produced. While existing solutions meet some of these criteria, none fully meet all of them.

The ARCADE algorithm (*Automated Road Curvature And Direction Estimation*) was developed to provide a robust and reliable method of determining road curvature and orientation as the first stage of processing for a lane detection scheme which fully addresses the criteria listed above. In addition, it can also be applied to other tasks, such as generating training samples for a neural network road follower without the need for a training signal from a human driver. ARCADE estimates the curvature and orientation of a road directly from image edge point locations and orientations, without the need to perform any prior perceptual grouping of those edge points into individual lane boundaries. It can do this in cases where up to 50% of the edge points are noise edges from shadows, cracks, etc. ARCADE does this by enforcing global scene constraints derived from an explicit model of the shape of lane boundaries in the image plane, and by using Least Median of Squares estimation [11] in order to prevent outliers in the edge data from corrupting the estimate of the road shape parameters.

Section 2 reviews existing lane detection techniques, examining which of the criteria described above they embody. Section 3 presents the derivation of the global constraints on the shape parameters of the individual lane boundaries in the image plane. Section 4 presents the ARCADE algorithm, which uses those constraints to robustly estimate road curvature and orientation from image edge data without prior perceptual grouping of the edge points. Section 5 describes the application of ARCADE to the tasks of extracting lane boundary locations in an image and generating training examples for a neural net road follower.

2 Related work

A number of systems make the assumption that the lane boundaries are straight lines, and detect them by extracting edge points from the image and using the Hough transform. An example of such a system is a lane detector developed at Honda [4]. The system developed at Matsushita [14] and later versions of the LANELOK system developed at General Motors [6] enforce a global scene constraint between the left and right lane edges (that they should meet at a vanishing point on the horizon row in the image plane). A system by Polk and Jain [9] and the SHIVA system [7] further extend these methods to better handle curved roads by dividing the image into a small number of horizontal sections and finding linear approximations to the lane edges within each section. None of these systems provides an explicit estimate of the reliability of the results it produces, and the use of a linear or piecewise-linear model of the lane boundaries limits the accuracy of the feature location in the image.

Two systems explicitly use curved models of the lane boundaries. The system developed at the University of Bristol [12] performs an intensity-based segmentation to extract regions corresponding to solid painted lines or sections of dashed painted lines. The system looks for the set of concentric arcs which define the lane structure. While the system handles curved lane boundaries and applies a global scene constraint to filter the results of the image segmentation, it assumes that there are painted stripes marking both edges of all the lanes, and provides no reliability estimate.

Initial location of lane boundaries for tracking by the VaMoRs system [1] is described in [2]. A direction-insensitive line detector is applied near the bottom of the image in order to locate the left and right lane edges. The orientation of the lane edges is determined, and the lane edges are then tracked upward in the image. A direction-specific mask is chosen adaptively to accommodate changes in the lane boundary orientation. There is no constraint imposed linking the shapes of the left and right lane boundaries. The assumption is that use of a direction-sensitive line detector will prevent the tracking process from being misled by noise in the image. Also, there is no reliability estimate.

The ARCADE algorithm was developed to provide the first stage of processing for a lane detection technique which would meet all the criteria listed in the introduction (model curved lane boundaries of all types; exploit global scene constraints; and provide an estimate of the reliability of the result). Central to ARCADE is the use of global road shape constraints derived from an explicit model of how the features defining a road appear in the image plane.

3 Derivation of global constraints on the image shape parameters of individual lane boundaries

The ARCADE algorithm is based on a model of road structure in the world which assumes that the markings and pavement boundaries defining the road and its lane structure can be approximated by circular arcs on a flat ground plane over the length of the road visible in a single image. A circular arc of curvature k is approximated by a parabola of the form

$$x = 0.5 \times k \times y^2 + m \times y + b$$

This is essentially the same model of road structure described in [8].

In the case where the camera is not tilted the derivation of the shape of individual lane boundaries in the image plane is straightforward. Assuming perspective projection, a pixel (r, c) in the image plane projects onto a point (x, y) on the ground plane according to the equations

$$x = c \times cf \times y, y = H/(r \times rf)$$

where H is the height of the focal point above the ground plane, rf is the height of a pixel on the image plane divided by the focal length, and cf is the width of a pixel on the image plane divided by the focal length. Substitution and a little algebra shows that the arc on the ground plane

$$x = 0.5 \times k \times y^2 + m \times y + b$$

corresponds to the curve

$$c = \frac{0.5 \times k \times H}{r \times rf \times cf} + \frac{b \times rf \times r}{H \times cf} + \frac{m}{cf}$$

in the image plane. Combining the camera calibration and road shape parameters together, the feature projects into the image plane as a curve of the form

$$c = k'/r + b' \times r + vp$$

where $r = 0$ is the row in the image plane containing the horizon.

In the case where the camera is tilted downward, the derivation is more complicated but the family of curves in the image plane is the same. The curve in the image plane has the form

$$c = \frac{1}{rf \times cf \times H \times \sqrt{1 + cr^2 \times rf^2}} \times$$

$$\left(\frac{1}{r} \times 0.5 \times k \times H^2 \times (1 + cr^2 \times rf^2) +\right.$$

$$r \times rf^2 \times (b - m \times H \times cr \times rf + 0.5k \times H^2 \times cr^2 \times rf^2) +$$

$$\left. H \times rf \times (1 + cr^2 \times rf^2) \times (m - k \times H \times cr \times rf)\right)$$

where cr is the row corresponding to the center of the field of view of the camera, and the other camera parameters are the same as in the untilted case.

If the feature arcs are assumed to have approximately the same curvature and to have parallel tangents at their x intercepts, then the global constraint can be imposed that all the road features will have the same k' and vp parameters in the image plane. It is possible to eliminate the feature-specific b' parameter by making use of local feature orientation information. Taking the derivative of the feature equation, solving for b', and substituting into the feature equation gives the relationship between a point on any road feature, the local feature tangent, and the k' and vp parameters shared by all the road features,

$$c = \frac{2 \times k'}{r} + \frac{dc}{dr} \times r + vp$$

Thus, the k' and vp parameters can be estimated directly from the raw edge point location and orientation data without the need to group the edge points together into individual features. The feature locations can then be extracted by averaging together pixels with the same b' value to construct a one-dimensional intensity profile of the road and segmenting that profile.

The next section of the paper describes how the ARCADE algorithm uses this model of road feature shape in the image plane to estimate road curvature and orientation. Then the application of ARCADE as the first stage of processing for extracting the visible lane structure in an image is described. Finally, the use of ARCADE to generate training data for an ALVINN-like road following neural network is described.

4 Processing stages in ARCADE

4.1 Extracting edge point locations and orientations

A simple one-dimensional operator is used to detect edges along a subset of the rows of the image. A kernel of the form [-1 -1 ... -1 -1 +1 +1 ... +1 +1] is cross-correlated with the pixels of the selected rows. This averages the image along the row, reducing the effects of texture in the image. The resulting edge magnitude signal is thresholded, and local extrema are detected. Edge orientation is estimated by matching edge points across several adjacent rows and fitting a line to the points. Since the estimation procedure used to recover the road shape parameters is robust in the presence of contaminating data observations, occasional errors in the local edge point matching do not have an effect on the overall result produced by the algorithm.

The segmentation used in the current implementation was selected for simplicity and speed. While this implementation of the algorithm uses this particular form of edge detection, any segmentation technique which detected boundaries in the image and provided an estimate of local boundary orientation could be used.

4.2 Estimating the road parameters from noisy edge data

The edge data extracted from the image is likely to be contaminated by bad data points. Some of the edge directions may be in error, and some of the edge points will not belong to road features. The Least Median Squares procedure [11] is used to estimate k' and vp from the image edge data in order to avoid distortions in the recovered road shape due to bogus edges. Consider the linear system $y_i = \beta x_i + \varepsilon$, where β is the vector of parameters to be estimated, and ε is a noise term. Standard least squares finds the estimate $\hat{\beta}$ which minimizes $\Sigma(r_i^2)$, where r_i is the residual $r_i = \hat{\beta} x_i - y_i$. Least Median of Squares tries to find the estimate $\hat{\beta}$ which minimizes $median(r_i^2)$. The case of fitting a line in two dimensions provides a simple geometric intuition for what the LMS estimate is. The LMS estimate is the line such that a band containing half the data points and centered on the line has the minimum height in y (the dependent variable). The computation of the LMS estimate is straightforward. Random subsets of the data are chosen. The standard least squares estimate of the parameters is made for each subset, and the median squared residual of the entire data set is calculated for that estimate. The estimate which produced the lowest median squared residual is selected as the final estimate. The LMS estimation technique can accommodate data sets where up to 50% of the data consists of outliers.

4.3 Estimating the reliability of the result

Given a set of extracted edges, the algorithm described will produce some estimate of the road curvature and orientation parameters in the scene. Those estimated parameters may have little resemblance to reality if there were more noise edges than road feature edges in the image. As a result, it is important to produce a value which indicates the extent to which the estimated road parameters are a good fit to the edge data. The quality measure chosen is the median absolute error in edge orientation given the estimated k' and vp parameters, i.e.

$$median\left(\left|\operatorname{atan}\left(\frac{dc}{dr_i}\right) - \operatorname{atan}\left(\left(c_i - vp - 2k'/r_i\right)/r_i\right)\right|\right)$$

A threshold for rejecting the parameter estimate for an image as unreliable could be derived experimentally based on the orientation error distribution for the edge detection scheme used.

5 Applications of ARCADE

5.1 Extracting the locations of visible lane markings and pavement edges

Once the road curvature related parameter k' and road orientation related parameter vp have been recovered, the image pixels corresponding to a particular b' value can be averaged together to create a one-dimensional intensity profile of the road cross section. An arbitrary row is selected for the cross section, r_x. For pixel (r_P, c_P) the corresponding b' parameter for that pixel is computed by the equation

$$b'_P = (c_P - vp - k'/r_P)/r_P$$

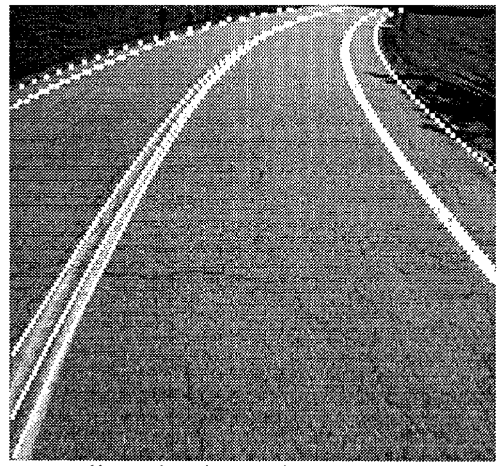
The corresponding column in the cross section row is then given by the equation

$$c_X = k'/r_X + b'_P \times r_X + vp$$

Averaged signals are produced for the red, green, and blue pixel components as well as the pixel intensity. The resulting one-dimensional signals are then segmented to locate the individual pavement and painted stripe edges visible in the image.

The segmentation is done using algorithms similar to those used in the feature tracking of the YARF system [7]. The columns of the averaged RGB signal are classified as yellow-stripe color or non-yellow-stripe color based on hue and saturation. The same one-dimensional edge kernel used to detect edges in the image is applied to the averaged intensity profile, and local extrema are extracted from the resulting edge magnitudes. Painted stripes are identified by looking for a dark-to-light edge followed after a short distance by a light-to-dark edge of similar magnitude. The stripes are classified as white or yellow based on the fraction of pixels between the edges in the RGB signal which were classified as yellow-stripe color. Experiments are in progress to distinguish between solid and broken stripes. Other edges which do not appear to be painted stripe edges are considered as possible pavement boundaries.

Figure 1 shows two examples of the results of the algorithm. The top image shows a road with painted lane markings. The feature edges detected by the algorithm are highlighted in white. The algorithm achieves very good localization of the road feature edges, and successfully finds both shoulders, both white stripes, and the edges of the double yellow line. Note that the median absolute error between the edge point orientations in the edge data and the orientations predicted by the estimated shape parameters was only 3.89 degrees. The bottom image was taken on an unmarked asphalt bicycle path. Even though there is a great deal of texture in the image due to the grass on either side of the road, the averaging performed by the edge detection avoids producing large numbers of noise edges. A muddy rut along the right edge of the path produces the spurious edge to the right of the pavement edge.

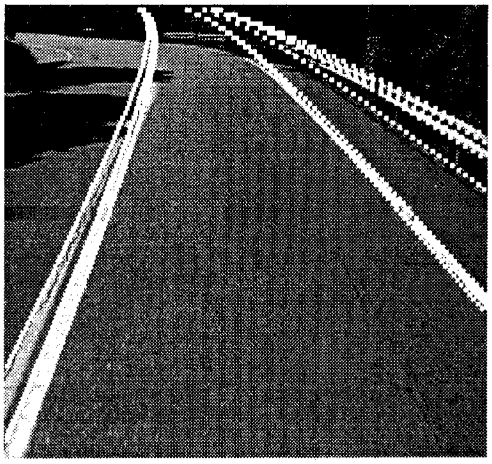


median absolute edge orientation
error = 3.89 deg

median absolute edge orientation
error = 5.24 deg

Figure 1: Feature extraction on a marked divided road (top) and unmarked asphalt bicycle path (bottom)

Figure 2 shows two images illustrating the limitations of the algorithm. In the top image, a sharp curve combined with a hill creates a situation where the estimated feature locations are badly in error near the top of the image. One extension planned for the algorithm is to include the effects of vertical curvature in the model. In the bottom image, the complex shadows and texture in the image result in a data set in which the majority of the detected edges are contaminants. As a result, the algorithm does not correctly locate the road. In this case, the median absolute edge orientation error is 30 degrees, compared to the low values in the other images shown. This demonstrates the utility of the reliability criterion for detection of such bogus results.



median absolute edge orientation
error = 4.17 deg

median absolute edge orientation
error = 30.6 deg

Figure 2: Examples showing limitations of the flat earth model (top) and use of the median absolute edge orientation error to detect unreliable results (bottom)

This algorithm has currently been tested on a set of 50 images taken on the two roads illustrated in the figures. These images contain a variety of shadow, curvature, and clutter conditions. Testing on a wider variety of roads will be done when we complete upgrading an Army HMMWV for use as a mobile testbed.

5.2 Application of ARCADE to train a road following neural net

The neural network approach to road following developed by Pomerleau in the ALVINN system [10] has shown the ability to learn to follow a broad variety of roads robustly. ALVINN uses a three-layer backprop network. The input unit activations encode a reduced resolution version of the image seen by the camera. The input units feed into a layer of hidden units. The hidden layer, in turn, is connected to a layer of output units which represent various steering directions. The simplicity of the network computations allows the system to drive a vehicle at highway speed using a standard workstation to perform the computation.

While ALVINN can provide a very rapid steering reflex, a separate network must be trained ahead of time for each road the system will encounter in the course of a run. Jochem, Pomerleau and Thorpe [5] describe a technique in which the hidden units of a set of ALVINN networks trained on different road types serve as the inputs to a network which is trained to combine them and produce the correct steering response for the original set of roads in the hope that this new network will generalize to previously unseen roads.

The road detection algorithm described in Section 5.1 has been applied as an alternative approach to handling unfamiliar roads in a context where the system is running on an autonomous vehicle without a human driver to provide training. Training examples can be generated showing a transformed image of the road with different curvature and offset relative to the vehicle. Given knowledge of the location of the edges of the lane which the vehicle is supposed to follow, the desired steering direction can be computed from measurements in the image plane using the pure pursuit technique [15].

Central to ALVINN's training scheme is the ability to transform input images and the corresponding steering output in order to generate examples of situations the human trainer is unlikely to put the vehicle into while driving. Given the image curvature and orientation parameters k' and vp, it is possible to warp the input image to produce synthetic images of the road with different curvatures and offsets of the camera.

Warping the input image to produce the synthetic training examples is done as follows. Suppose the synthetic image has a curvature term of k'_W. To find the pixel (r, c) in the original image corresponding to pixel (r, c_W) in the warped image, solve for b'_W by the equation

$$b'_W = (c_W - vp - k'_W/r)/r$$

Then compute the corresponding column in the original image by the equation

$$c = k'/r + b'_W \times r + vp$$

(In the case where the camera is tilted, there are small couplings of the feature curvature on the ground plane to the b' and vp parameters in the image plane, but for realistic camera calibration parameters these couplings can be neglected.) Shifting the offset of the camera relative to the road adds a constant offset

b'_{incr} to the b' value of a pixel, allowing the corresponding pixel in the unwarped image to be determined similarly. Changes in curvature and offset can be composed to simultaneously change both.

Initial results appear promising, with the network converging after a comparable number of training presentations as the original ALVINN. Figure 3 shows an example of the output of such a network after training. The net has been presented with two novel images, with the desired steering response indicated below the network input and the network's output response below that. As the location of the road shifts to the further left in the image, the steering direction also shifts further to the left.

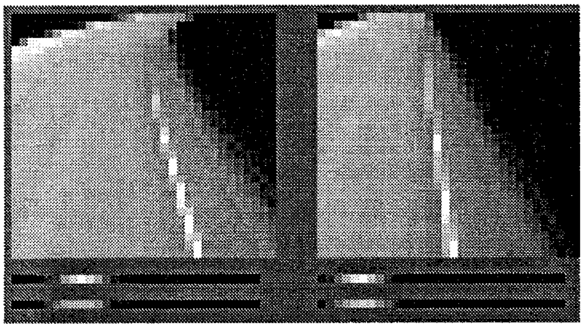


Figure 3: Example road following net trained using the road detection algorithm

6 Conclusion

Global constraints on the shape parameters of individual lane boundaries in an image can be derived from an explicit model of how the features defining a road in the scene project onto the image plane. This permits estimation of road curvature and orientation directly from edge point locations and orientations without any prior perceptual grouping of the edge points into individual lane boundaries. Use of the Least Median Squares estimation procedure makes the technique robust in the presence of noise edges in the data, and the reliability of the result produced can be estimated. The algorithm has a number of applications, from initializing symbolic road followers which track features from image to image, to generating the training examples for a neural net road follower.

7 Acknowledgments

This work was funded by ARPA under contract DAAE07-92-CR012, monitored by TACOM.

8 References

[1] Dickmanns, Ernst D.; and Mysliwetz, Birger D. Recursive 3-D Road and Relative Ego-State Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14:199-213, 1992.

[2] Efenberger, W.; Ta, Q.-H.; Tsinas, L.; and Graefe, V. Automatic Recognition of Vehicles Approaching from Behind. In *Proceedings of the Intelligent Vehicles '92 Symposium.* June, 1992.

[3] Franke, U.; Fritz, H.; and Mehring, S. Long Distance Driving with the Daimler-Benz Autonomous Vehicle VITA. In *PROMETHEUS Workshop.* December, 1991.

[4] Hashimoto, K.; Nakayama, S.; Saito, T.; Oono, N.; Ishida, S.; Unoura, K.; Ishii, J.; and Okada, Y. An Image-Processing Architecture and a Motion Control Method for an Autonomous Vehicle. In *Proceedings of the Intelligent Vehicles '92 Symposium.* June, 1992.

[5] Jochem, Todd; Pomerleau, Dean; and Thorpe, Charles. MANIAC: A Next Generation Neurally Based Autonomous Road Follower. *Proceedings, Intelligent Autonomous Systems 3.* February, 1993.

[6] Kenue, Surender K. LANELOK: Detection of Lane boundaries and Vehicle Tracking Using Image-Processing Techniques -- Parts I and II. In *SPIE Mobile Robots IV.* 1989.

[7] Kluge, Karl. *YARF: An Open-Ended Framework for Robot Road Following.* PhD thesis, Carnegie Mellon University, 1993.

[8] Mysliwetz, Birger D.; and Dickmanns, E. D. Distributed Scene Analysis for Autonomous Road Vehicle Guidance. *Proceedings SPIE Conference on Mobile Robots.* November, 1987.

[9] Polk, Amy; and Jain, Ramesh. A Parallel Architecture for Curvature-Based Road Scene Classification. In *Roundtable Discussion on Vision-Based Vehicle Guidance '90* (in conjunction with IROS). July, 1990.

[10] Pomerleau, Dean A. *Neural Network Perception for Mobile Robot Guidance.* PhD thesis, Carnegie Mellon University, 1992.

[11] Rousseeuw, Peter J.; and Leroy, Annick M. *Robust Regression and Outlier Detection.* John Wiley & Sons, Inc., 1987.

[12] Schaaser, L. T.; and Thomas, B. T. Finding Road Lane Boundaries for Vision Guided Vehicle Navigation. In *Vision-Based Vehicle Guidance.* Ichiro Masaki (ed), Springer-Verlag, 1992, Chapter 11.

[13] Schniederman, H.; and Nasman, M. Visual Processing for Autonomous Driving. In *Proceedings, IEEE Workshop on Applications of Computer Vision.* 1992.

[14] Suzuki, A.; Yasui, N.; Nakano, N.; and Kaneko, M. Lane Recognition System for Guiding of Autonomous Vehicle. In *Proceedings of the Intelligent Vehicles '92 Symposium.* June, 1992.

[15] Wallace, R.; Matsuzaki, K.; Goto, Y.; Crisman, J.; Webb, J.; and Kanade, T. Progress in Robot Road Following. *Proceedings IEEE International Conference on Robotics and Automation.* April, 1986.