A Discriminating Feature Tracker for Vision-Based Autonomous Driving

Henry Schneiderman and Marilyn Nashman

National Institute of Standards and Technology
Robot Systems Division
Building 220, Room B127
Gaithersburg, MD 20899
e-mail: hws@cme.nist.gov, nashman@cme.nist.gov

ABSTRACT

A new vision-based technique for autonomous driving is described. This approach explicitly addresses and compensates for two forms of uncertainty: uncertainty about changes in road direction and uncertainty in the measurements of the road derived in each image. Autonomous driving has been demonstrated on both local roads and highways at speeds up to 100 km/h. The algorithm has performed well in the presence of non-ideal road conditions including gaps in the lane markers, sharp curves, shadows, cracks in the pavement, and wet roads. It has also performed well in rain, dusk, and nighttime driving with headlights.

1. Introduction

Feature tracking algorithms developed for autonomous driving [1] do not satisfactorily address the issue of modelling and compensating for uncertainty in their measurements. In particular, they overlook the fact that road visibility can vary from image to image. For example, there will be images in which the lane markers are clearly visible and others in which they are less visible, obscured, or absent (see Figure 1 and Figure 2). Our approach explicitly addresses uncertainty of two forms. The first is the uncertainty about how quickly the road changes as a function of time (assuming a nominally constant speed). The second is uncertainty in the visibility of the lane markers in each individual image. Both contributions of uncertainty are formulated into a criterion of optimality for the estimate of the lane marker models. This criterion is satisfied by a weighted recursive least squares (WRLS) with exponential decay estimator. Using this method, the road following algorithm achieves robust behavior in the presence of small gaps in the lane markers and momentary loss in visibility of lane markers. The algorithm also performs well in the presence of other non-ideal conditions such as sharp curves, shadows, cracks in the pavement, wet roads, rain, dusk, and nighttime driving.

2. Lane marker tracking algorithm

This algorithm requires that lane markings be present and attempts to track the lane markings on each of two lane boundaries in the lane of travel. There are three successive stages of computation (also see Figure 1):

1) Edge extraction - Extracting edge point position and orientation.

2) Data association - Determining likely groupings of edge points to each lane marker.

3) Model update - Updating the lane marker models.

This sequence of operations is repeated for each new image. Video imagery from a camera mounted above the cab of the vehicle provides the input to Stage (1). Stages (2) and (3) interact with geometric models of each lane marker. Stage (2) attempts to group the extracted edge points obtained from stage (1) with each lane marker by comparing each edge point against the current models. Stage (3) then updates the models using those edge points that have been grouped to each lane marker.

Section 2.1 describes the geometric representation of the lane markers. Section 2.2 describes how the model is initialized to a road scene. Section 2.3 describes the edge extraction algorithm (stage 1). Section 2.4 describes the data association algorithm (stage 2). Section 2.5 describes the method for updating the lane marker models (stage 3).

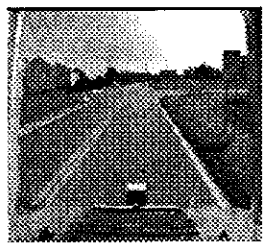

Figure 1. Lane markers are clearly visible

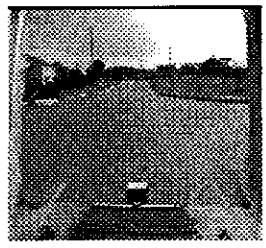

Figure 2. Gap in lane markers

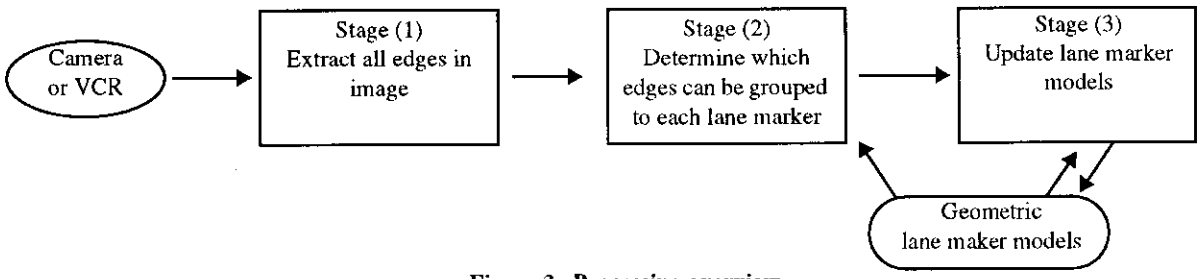

Figure 3. Processing overview

2.1. Representation of the lane markers

All representations are maintained in 2D with respect to the image plane throughout all computations in this algorithm. Both the left and right lane markings in the lane of travel are modeled. These markings correspond to the white or yellow lines painted on the road. These lines are either solid or striped (see Figure 4).

Each of these lane boundaries is modeled by a second order polynomial in the image plane:

$$x = a_1 + a_2 y + a_3 y^2 \quad (1)$$

The parameters, a_1, a_2, a_3, govern the shape and position of the lane marker model. The endpoints of each lane marker model are given by the intersection of the model equation with the boundary of the window of interest (see section 2.4).

A second order model was chosen because it provides an adequate representation of shape within the constraints of real-time performance required for the autonomous driving problem.

2.2. Initial conditions

The algorithm requires an initially approximate model of the lane markers before tracking can begin. This initial correspondence between the road markings and the models of the road markings is established by a teleoperator. The teleoperator manually positions the models to align them with the appearance of the lane markers in the image.

2.3. Edge extraction

In the first processing step, edge extraction is performed on each image (stage (1) in Figure 1). For every point in the image, edge magnitude and edge orientation are computed using a two-dimensional *3 X 3* spatial Sobel operator. A binary edge image is produced by thresholding the edge points in magnitude. Figure 5 shows a typical image of a road viewed from a camera mounted on a vehicle. The thresholded edges in this image are shown in Figure 6.

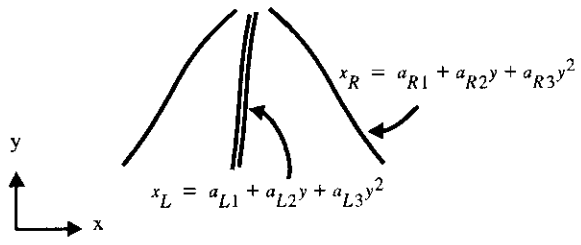

Figure 4. Lane marker models

2.4. Data association

The raw edges in each image are produced by various visual entities including lane markers, shadows, pot holes, cracks, and other vehicles. A data association algorithm (stage (2) in Figure 1) is used to determine which of these raw edge points are likely to be associated with each lane marker and to discard those edges that do not seem to be associated with either lane marker.

The data algorithm compares each edge pixel to the current model of each lane marker. An edge pixel must satisfy two criteria to be associated with a lane marker. The first criterion is two-dimensional spatial proximity of the edge point to the model. The second criterion is similarity of direction of the edge point with the angular orientation of the model.

In each image, many edges can be discarded immediately on the basis of the spatial proximity criterion. This is done by eliminating all edges that fall outside a window of interest. This eliminates many, but not all edges that violate the spatial proximity criterion. Figure 7 represents the results of masking the thresholded edge image with the window of interest.

For all edge points falling within the window of interest, the two data association criteria are applied on a point by point basis. First, the data association procedure compares the edge direction of each candidate edge point with the angular direction of the model. When this angular criterion is satisfied, the distance d is computed between the edge point and the model. If this distance is less than the pre-specified threshold, the edge point is associated to that lane marker. Figure 8 shows the thresholded edges extracted from the original image that are grouped to the right lane marker. (In Figure 8, edges forming the right lane marker do not appear as continuous as they do in Figure 7. This is due to sampling of the edge image).

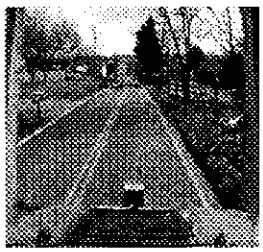

Figure 5. Original image

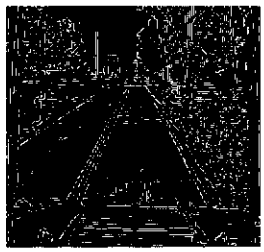

Figure 6. Thresholded edge image

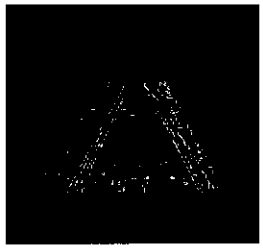

Figure 7. Edges within window of interest

Figure 8. Edges grouped to right lane marker.

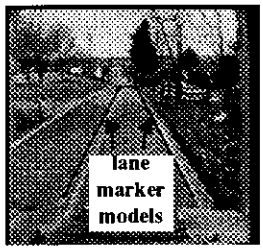

Figure 9. Computed lane markers graphically superimposed on image

2.5. Lane marker model update

Using the associated edge points, each of the two polynomial lane marker models is updated independently (except for a case described in 2.5.7). Several principles are followed to obtain robust lane marker updates. Section 2.5.1 describes how an estimate improves with more data. In updating the lane marker models, two types of uncertainty are addressed. There is uncertainty in how quickly the road changes as a function of time (2.5.2) and there is uncertainty in the visibility of the lane markers in each image (2.5.3). These factors are formulated into a criterion of optimality for the lane marker estimate (2.5.4). This criterion of optimality is satisfied by a weighted recursive least squares (WRLS) with exponential decay estimator (2.5.5). The WRLS with exponential decay solution is compared to the Kalman filter solution (2.5.6). In 2.5.7 some additional constraints between the lane markers are discussed.

2.5.1. More data improve the estimate

In general, an estimate can be improved by using more data. It can be shown that if a measurement consists of a sum of a stationary signal and unbiased noise, the estimate of the signal will improve — the variance in the estimate will decrease — as more measurements are averaged [10], [11], [12].

To obtain the best possible estimate from each image, it is important to make use of all visible portions of the lane markers. Valuable information is wasted if visible portions of the lane markers are partially masked out as is done by YARF [8][9].

An estimate can be further improved by using data from multiple images in a sequence. This is particularly, important for lane marker sensing where the edge data from any one image may be too weak or contaminated by incorrect data associations.

However, as mentioned above, in order to improve an estimate by using more data, measurement noise must be unbiased. In lane marker tracking this assumption is fairly viable due to a combination of two phenomena. First, the immediate surroundings of the lane markers (as viewed in the image) are constantly changing because of the motion of the vehicle. Secondly, any surrounding spurious edge points, e.g., shadows and cracks in the pavement, are not inherently biased to either side of the lane markers.

2.5.2. Uncertainty in road change

The lane markers are not strictly stationary signals across successive images. However, they change relatively slowly assuming a nominal vehicle speed. Therefore, a compromise must be mediated between robustness of the estimate — by using data over a large temporal span — and responsiveness to actual changes in the lane markers — by using data over a smaller temporal span. This compromise is achieved by the relative weighting of new data with respect to older data in the estimate.

This relative weighting between new data and old data may be simply governed by an exponential decay factor where the weight contributed by the exponential decay, λ, for each edge point is:

$$\lambda^{t-to} \qquad (2)$$

$0.0 < \lambda \leq 1.0$
t is the current time
to is the time the image was sampled

For example, if $\lambda = 0.5$, all edge points in the current image, $to = t$, have a weight of 1.0. All edge points in the image read at time $to = t - 1$ have a weight of 0.5, etc. Values of λ anywhere in the range $0.5 < \lambda < 0.75$ produced acceptable tracking.

2.5.3. Uncertainty in lane marker visibility

In each image, lane marker visibility is measured by the number of edge points matched to the lane marker model. This measure of visibility acts as an additional

weight when edge data are combined temporally in an estimate. Figure 10, shows images sampled from a sequence in which there are gaps in both lane markers. Figure 11 and Figure 12 graph the number of points matched to the lane markers during this interval of time. Not surprisingly, the number of matched points drops during the gap. These images will therefore carry relatively less weight in the estimates and do not greatly perturb the estimated lane marker models.

The measure described above is limited in that it only measures the visibility of the lane markers. Ideally, a broader measure of confidence is desirable. Such a measure of confidence should also penalize for the presence of clutter such as shadow edge points mistakenly grouped to the lane markers. One potential measure of clutter is the variance about the estimated function $x = a_1 + a_2 y + a_3 y^2$:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\left[x_i - \left(a_1 + a_2 y_i + a_3 y_i^2\right)\right]^2 \qquad (3)$$

N - Number of matched edges points.

A higher variance indicates a more scattered distribution of edge points. This usually indicates the presence of spurious edge points. Confidence could then be computed as an inverse function of variance as in [14][15]. However, in [14][15] the detected features were edges corresponding to only one light-to-dark transition. Unfortunately, lane markers give rise to 2 spatially distinct edges for single stripes and 4 for double stripes. There are also edges caused by the boundary between the pavement and the shoulder. Since there is spatial spread inherent in the sensed lane marker, a high variance in its location does not necessarily indicate the presence of clutter.

2.5.4. Criterion of optimality

In updating a lane marker model, the two contributions of uncertainty are directly formulated into a criterion of optimality, J_R:

$$J_R = \sum_{j=0}^{t}\left(\lambda^{t-j}\sum_{i=1}^{N_j}\left[x_{j,i} - \left(a_1 + a_2 y_{j,i} + a_3 y_{j,i}^2\right)\right]^2\right) \qquad (4)$$

This represents the residual least squares error as weighted by the uncertainty measures. By computing a_1, a_2, a_3 such that J_R is minimized, these uncertainties are minimized in the least squares sense in the estimate.

Expanding the outer summation in equation (4) in reverse order gives:

$$J_R = \sum_{i=1}^{N_t}\left[x_{t,i} - \left(a_1 + a_2 y_{t,i} + a_3 y_{t,i}^2\right)\right]^2 + \qquad (5)$$

$$\lambda\sum_{i=1}^{N_{t-1}}\left[x_{t-1,i} - \left(a_1 + a_2 y_{t-1,i} + a_3 y_{t-1,i}^2\right)\right]^2 +$$

$$\lambda^2\sum_{i=1}^{N_{t-2}}\left[x_{t-2,i} - \left(a_1 + a_2 y_{t-2,i} + a_3 y_{t-2,i}^2\right)\right]^2 + \ldots$$

t - Current time

N_t - Number of matched edges points in image t

Each summation in (5) represents the edge points from one image grouped to the lane marker. The weight measuring image visibility is implicitly included; that is, each summation is over the number of edge points matched to the lane marker for that image, N_t points. An image in which many edge points are matched will therefore contribute more terms to the residual and thereby carry more influence in the determination of a_1, a_2, a_3. The exponential decay is achieved by multiplying each summation by a power of λ where older images are multiplied by increasing powers of λ.

2.5.5. Weighted recursive least squares with exponential decay solution

For each new image, a_1, a_2, a_3 must be recomputed for each lane marker model such that J_R is minimized. Excepting pathological cases, a unique solution for a_1, a_2, a_3 exists. This solution can be recursively formulated. In the recursive formulation, the current solution for a_1, a_2, a_3 is called the state. When a set of edge points from a new image is acquired, the state is recomputed using only this new data explicitly. All past data are completely represented by the previous state and the estimated covariance in the previous state. This update consists of three successive computations [16]:

Update prediction

$$\hat{y}('t) = \phi^{T}('t)\,\theta(t-1) \qquad (6)$$

Update state estimate

$$\theta(t) = \theta(t-1) + P(t-1)\,\phi(t)\,[y(t) - \hat{y}(t)]\,['I + \phi^{T}(t)\,P(t-1)\,\phi(t)]^{-1} \qquad (7)$$

Update state covariance estimate

$$P(t) = \frac{1}{\lambda}\Big(P(t-1) - [P(t-1)\,\phi(t)\,\phi^{T}(t)\,P^{T}(t-1)]\,['I + \phi^{T}(t)\,P(t-1)\,\phi(t)]^{-1}\Big) \qquad (8)$$

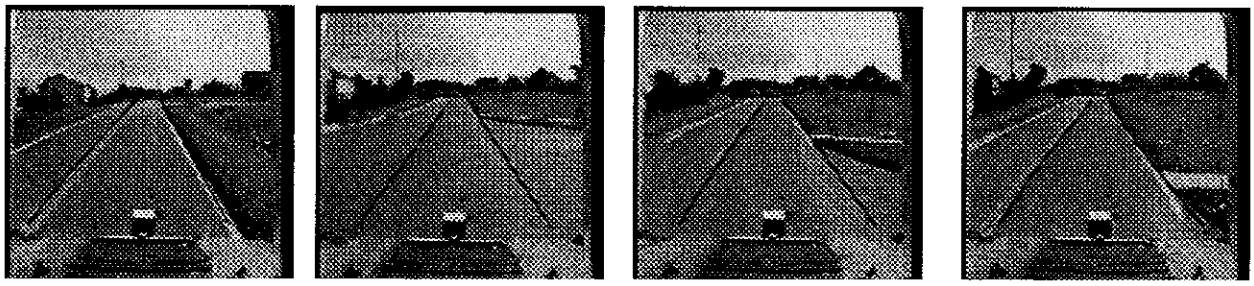

Figure 10. Images sampled from a sequence in which there is a gap in the lane markers. Lane marker models are graphically superimposed.

where

$$y('t) = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \cdots \\ x_{t,N_t} \end{bmatrix} \qquad \phi^T('t) = \begin{bmatrix} 1 & y_{t,1} & y_{t,1}^2 \\ 1 & y_{t,2} & y_{t,2}^2 \\ \cdots & \cdots & \cdots \\ 1 & y_{t,N_t} & y_{t,N_t}^2 \end{bmatrix}$$

$$\theta(t-1) = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

This represents the state estimate at time = t - 1, and P (t - 1) represents an estimate of its covariance.

This computation is known as weighted recursive least squares (WRLS) with exponential decay [16].

Although equations (6), (7), (8) compute the solution that minimizes J_R in equation (4), an equivalent algorithm, the square root information filter (SRIF) algorithm [17], is used in practice. The SRIF is also recursive but has much less sensitivity to finite precision round-off error and is also less computationally demanding than equations (6), (7), (8).

2.5.6. Comparison between Kalman filter and WRLS with exponential decay

For the purposes of the Kalman filter, the system can also be modelled by the following state space model:

State transition equation

$$\theta(t) = \theta(t-1) + v(t) \tag{9}$$

Measurement equation

$$y(t) = \phi^T(t)\,\theta(t-1) + w(t) \tag{10}$$

θ(t), ϕ(t), y(t) are defined above
v (t) and w (t) are additive noise

This represents a stationary signal with a drift, *v(t)*. The covariance of the drift represents uncertainty in the rate of change in the lane markers. These equations will be the basis of the Kalman filter discussion to follow. However, to incorporate vehicle motion into this formulation, the state equation could have been expressed as:

$$\theta(t) = A\theta(t-1) + v(t) \tag{11}$$

Where *A*, the 3x3 state transition matrix, represents the known change in state that occurs each cycle. In autonomous driving, the vehicle motion affects the state in a

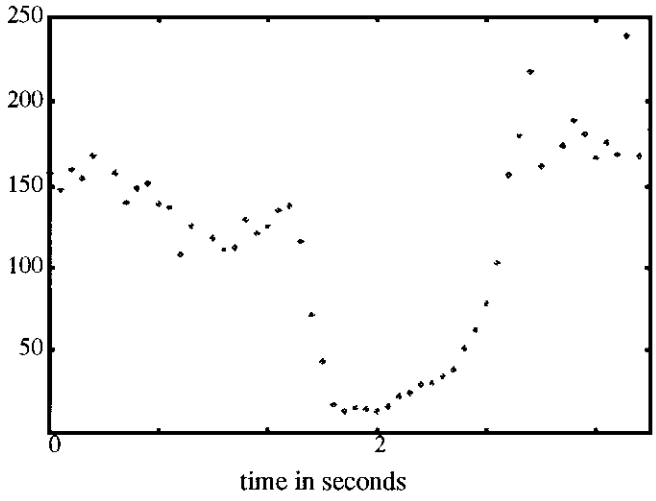

Figure 11. Number of points grouped to left lane marker during image sequence.

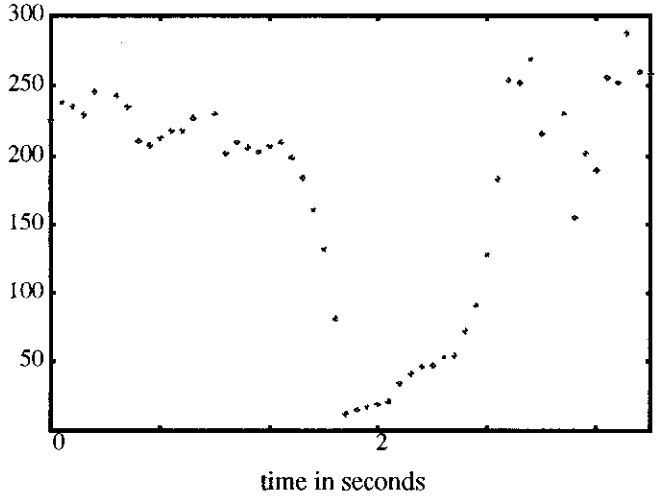

Figure 12. Number of points grouped to right lane marker during image sequence

known manner and such knowledge could be conceivably linearized and expressed in A. In the VaMoRs system [2][3][4][5][6][7], vehicle motion was formulated into the model update in this way. However, this method departed from the classical Kalman filter in its interpretation of A. A was chosen as a function of the state itself making the state update non-linear, sub-optimal and possibly unstable.

The Kalman filter provides the minimum variance linear estimate of the state (equations (8) and (9)) when $v(t)$ and $w(t)$ are white unbiased Gaussian noise.

The Kalman filter solution is similar to that given by WRLS with exponential decay. In the Kalman filter solution, the update of the prediction and the update of the state are identical to those for WRLS with exponential decay as given by equations (6) and (7). However, the Kalman update of the state covariance differs and is given by:

$$P(t) = P(t-1) - [P(t-1)\phi(t)\phi^T(t)P^T(t-1)] \left[{}^tR_2(t) + \phi^T(t)P(t-1)\phi(t)\right]^{-1} + {}^tR_1(t) \quad (12)$$

R_1 (t) and R_2 (t) are the covariances of v(t) and w(t) in equation (10) respectively.

In both algorithms, the state covariance, $P(t)$, serves the same functionality. In the state update (equation (7)), the state covariance behaves as a gain that multiplies the error between the model and the current measurement, $[y(t) - \hat{y}(t)]$. Therefore a large state covariance (i.e. large eigenvalues) means there is more uncertainty in the current estimate and consequently the state estimate will be more responsive to the error. Conversely, a smaller state covariance means there is more certainty in the current estimate and the state estimate will be less responsive to the error.

Both algorithms seek to maintain the size of the covariance such that a desirable trade-off is achieved between responsiveness to changes in the state and robustness in the presence of noise. However, they differ in the mechanism by which this covariance adjustment is achieved. In the Kalman filter, covariance size management is achieved through the relative choices of the terms $R_1(t)$ and $R_2(t)$ in equation (12) (the covariances of $v(t)$ and $w(t)$ in equation (10), respectively). In the WRLS with exponential decay, covariance size management is achieved through the choice of λ.

The problem faced in implementing the Kalman filter for this application is in deriving physically accurate models for the covariances $R_1(t)$ and $R_2(t)$. Considering $R_1(t)$ first, $R_1(t)$ is the covariance in the state equation, equation (10). In less precise terms, $R_1(t)$ represents prior knowledge in how much the road is expected to change and specifically how these changes affect the uncertainty in each state parameter. $R_2(t)$ represents the amount of uncertainty in the measurement such as that caused by visibility and clutter as discussed in 2.5.3. In particular, $R_2(t)$ accounts for the uncertainty in each individual edge point and the correlations in uncertainty among all edge points.

In the VaMoRs system [2][3][4][5][6][7], a Kalman filter like formulation is used for road following, but none of these publications mention how the covariances $R_1(t)$ and $R_2(t)$ are physically derived or which values are used for these covariances. Such an omission seems serious in that the behavior of the algorithm is largely governed by these choices. Moreover, the whole question of uncertainty modeling is left unanswered.

2.5.7. Road width constraint

If sufficient data are matched to both lane markers, the two lane markers are treated independently. However, a road width constraint is used when the lane marker data are sparse for one lane marker and strong for the other lane marker. This constraint is designed to handle the situation where one lane marker momentarily disappears. Currently a road width constraint is applied when the number of edge points associated to the weaker lane marker is under a predefined threshold and the stronger lane marker exceeds the threshold. The threshold is currently set to 40 points.

Road width is modeled as a first order polynomial in the image plane:

$$x_w = a_{1w} + a_{2w}y \quad (13)$$

Each coefficient of this polynomial is computed from the difference of the corresponding coefficients of the two lane marker models. This difference is then averaged over time with exponential decay. For example, for the first coefficient this gives:

$$a_{1w}(j) = \frac{a_{1R}(j) - a_{1L}(j) + Na_{1w}(j-1)}{1+N} \quad (14)$$

j - Time at which image was sampled
N - Decay factor, currently N=20

These coefficients are recomputed for each new image.

The road width constraint involves using the road width model to compute a synthetic lane marker model by adding this offset to the location of the other lane marker model. The synthetic lane marker model is then weighted and combined into the lane marker model update for that lane marker. The weight this synthetic lane marker carries is a function of the number of edge points associated to lane marker versus the number of points associated to the other lane marker.

3. Evaluation of algorithm

Actual autonomous driving experiments require a complete autonomous navigation system which include a perception system, a steering/control system, and a robotic vehicle. Such a system has been developed at the National Institute of Standards and Technology (NIST) using an Army High Mobility Multipurpose Wheeled Vehicle (HMMWV). The HMMWV and steering/control system are described in [22] and [23].

3.1. Testing

It is very difficult to objectively measure autonomous driving performance. One possible measure might be the centering of the vehicle between the lane markings. A method for performing this measurement is currently under development at NIST [24]. However, in objectively comparing different systems and techniques using such a method, testing must be performed on the same roads under the same environmental conditions (e.g., angle of sun, cloud cover, etc.). Therefore, to compare the algorithm described in this paper with other similar algorithms would require duplication of the computational systems these other algorithms use. These computer systems often use custom or special purpose real-time computational hardware that is not readily available. For example, the VaMoRs system described in [2][3][4][5][6][7] uses a customized transputer network.

Since objective measures of performance are not currently feasible, a subjective description of this system's performance can be provided. Autonomous driving experiments using the NIST HMMWV have been performed on several roads including the NIST campus, Great Seneca Highway in Gaithersburg, Maryland, and Montgomery County Police Test track in Rockville, Maryland. These roads contained standard lane markings such as double yellow lines, single white lines, and dashed white lines. On all these roads the algorithm performed quite successfully and reliably. The only failure occurred on one portion of road where the pavement abruptly changed from dark asphalt to light cement for an overpass. The lane markings did not provide enough contrast to be detected on the cement pavement. Otherwise, the vehicle maintained centering in the lane of travel. It was able to successfully traverse sections in which there were significant gaps of 6 - 7 meters in the lane marking for small intersections and through an underpass. The roads were moderately shadowed by surrounding trees and varied from mild curvature and hills to severe rises and drops and sharp curves on the Montgomery Police test track. Pavement quality varied significantly, including stretches of old pavement with many cracks and discolorations. Top speeds of 100 km/h were also demonstrated. On all roads the vehicle was able to travel at the legal speed limit. Testing also included a wide variety of conditions including rain (wet roads), night time driving with headlights, and driving at dusk into the sunlight.

The algorithm has also been widely tested using video taped road scenes. Tracking was maintained on video tapes of roads with sharp curves, hills and moderate shadows. However, on one portion of video taped road, tracking was temporarily lost when the vehicle travelled through a sharply curved hilly portion of road that was shadowed by a heavily wooded area. Tracking was maintained in typical traffic situations: on-coming traffic, passing vehicles, and traveling behind other vehicles.

3.2. Timing

The update rate of the system is 15 Hz and the worst case computational delay (between image capture and computation of steering) is 150 milliseconds (ms). Edge extraction required 66.7 ms.The number of edge points extracted varies from scene to scene and consequently the processing times for the algorithms in stages (2) and (3) (Figure 1) vary depending on the number of data points present. For a representative road scene containing approximately 300 edge points, the edge matching is performed in 21 ms and the road model update is performed in 51 ms. The steering process [22][23] requires less than 1 ms computation time.

4. Conclusion and future work

An algorithm has been described that robustly tracks road lane markers. It is assumed that the lane markers are visible with either solid, double, or dashed lines. All visual processing is done in two dimensional image coordinates. Processing is performed in sequential stages: extracting edges, associating edge points to the lane markers, and updating models of the lane markers. Uncertainty in the visibility of the lane markers and uncertainty in the change of the road are modelled and incorporated into a weighted recursive least squares with exponential decay estimate used to update the lane marker models. The system update rate is 15 Hz.

Although this system performed very well, it can only be considered a first step in autonomous driving. The challenge in future work is to develop algorithms of increasing reliability and to strive toward the goal of robustness under all possible driving conditions.

Acknowledgments

We would like to give special thanks to Thomas Wheatley, Karl Murphy, Harry Scott, Steve Legowik, and Chuck

Giauque for their help. We would also like to thank all members of the Robot Systems Division at the National Institute of Standards and Technology (NIST) for their helpful comments.

References

[1] H. Schneiderman and J. Albus. "Progress and Prospects for Vision-based Automatic Driving." Proceedings of IVHS America 3rd Annual meeting,Washington, DC. April, 1993

[2] E.D. Dickmanns and V. Graefe. "Dynamic Monocular Machine Vision." Machine Vision Applications. Vol. 1, pp 223 - 240. 1988.

[3] E.D. Dickmanns and V. Graefe. "Applications of Dynamic Monocular Machine Vision." Machine Vision Applications. Vol. 1, pp 241-261. 1988.

[4] E. D. Dickmann, B. Mysliwetz, T. Christians. "An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles." IEEE Transactions on Systems, Man, and Cybernetics. Vol. 20, No. 6. pp. 1273-1284, 1990.

[5] E. Dickmanns, A. Zapp. "A Curvature-based Scheme for Improving Road Vehicle Guidance by Computer Vision." SPIE Vol. 727. Mobile Robots, 1986.

[6] B. Mysliwetz, E. Dickmanns. "Distributed Scene Analysis for Autonomous Road Vehicle Guidance." SPIE Vol. 852. Mobile Robots II, 1987.

[7] E. D. Dickmanns and B. D. Mysliwetz. "Recursive 3-D Road and Relative Ego-State Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 14, No. 2. Feb., 1992.

[8] K. Kluge and C. Thorpe. "Explicit Models for Road Following." In *Vision and Navigation: The Carnegie Mellon Navlab*. Kluwer. Norwell, MA. 1990. Also in IEEE Conference on Robotics and Automation, 1989.

[9] Karl Kluge and Charles Thorpe. "Representation and Recovery of Road Geometry in YARF." Proceedings of the Intelligent Vehicles '92. Detroit. pp 114 - .

[10] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, Englewood Cliffs, New Jersey. 1993.

[11] C. Helstrom. *Probability and Stochastic Processes for Engineers*. Macmillan Publishing Company. New York, 1984.

[12] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. 2nd Ed. Magraw-Hill. New York, 1984.

[13] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons. New York, 1973.

[14] H. Schneiderman, A. J. Wavering, M. Nashman, R. Lumia. "Real-Time Model-based Visual Tracking." 1994 International Workshop on Intelligent Robotic Systems.

[15] H. Schneiderman, M. Nashman, R. Lumia. "Model-based Vision for Car Following." Proceedings of SPIE Conference on Sensor Fusion. Boston. September, 1993.

[16] G. Goodwin, K. Sin. *Adaptive Filtering, Prediction and Control*. Prentice-Hall. Englewood Cliffs, New Jersey, 1984.

[17] G. Bierman. *Factorization Methods for Discrete Sequential Estimation*. Academic Press. New York, 1977.

[18] C. Lawson and R. Hanson. *Solving Least Squares Problems*. Prentice-Hall. Englewood Cliffs, New Jersey, 1974.

[19] G. Golub and C. Van Loan. *Matrix Computations*. 2nd Ed. John Hopkins University Press. Baltimore, 1989.

[20] G. Strang. *Linear Algebra and Its Applications*. 3rd Ed. Harcourt Brace Jovanovich. Orlando, Florida. 1988.

[21] B. Noble and J. Daniel. *Applied Linear Algebra*. 3rd Ed. Prentice-Hall. Englewood Cliffs, New Jersey. 1988.

[22] S. Szabo, H. Scott, K. Murphy, S. Legowik, R. Bostelman. "High-Level Mobility Controller for a Remotely Operated Land Vehicle." Journal of Intelligent and Robotic Systems. Vol. 5, pp 63-77. 1992

[23] K. Murphy. "Navigation and Retro-Traverse on a Remotely Operated Vehicle." Proceedings of the IEEE Conference on Intelligent Control and Instrumentation. Singapore, February, 1992.

[24] M. Herman, et al. "Recommendations for Performance Evaluation of Unmanned Ground Vehicle Technologies."