

Stochastic road shape estimation

B. Southall

C. J. Taylor

GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104

{southall,cjtaylor@grasp.cis.upenn.edu}

Abstract

We describe a new system for estimating road shape ahead of a vehicle for the purpose of driver assistance. The method utilises a single on board colour camera, together with inertial and velocity information, to estimate both the position of the host car with respect to the lane it is following and also the width and curvature of the lane ahead at distances of up to 80 metres. The system's image processing extracts a variety of different styles of lane markings from road imagery, and is able to compensate for a range of lighting conditions. Road shape and car position are estimated using a particle filter. The system, which runs at 10.5 frames per second, has been applied with some success to several hours' worth of data captured from highways under varying imaging conditions.

1. Introduction

Computer vision technology is of increasing interest to automobile manufacturers, both for completely autonomous driving and also for the more medium-term goal of driver assistance, where hazardous road situations such as accidental road exit or imminent obstacles may be identified ahead of the vehicle in sufficient time for the driver to take necessary action. The vision system described here is designed to be part of a collision warning system. Our experimental vehicle (a Buick LeSabre) is equipped with a single, forward looking, colour CCD camera mounted between the rear-view mirror and the windshield together with sensors which yield estimate of velocity and yaw rate, and a scanning radar. The speed and yaw-rate data is converted into an audio signal that can be recorded on the sound track of a video tape. This useful capability permits the synchronisation of image and proprioceptive information during data collection experiments. The eventual aim is to estimate the vehicle's road position and the shape of the lane ahead of the vehicle, and to use this information to determine whether

any of the targets detected by the radar system present an obstacle to the vehicle's motion, and to alert the driver if such conditions arise.

In this paper, we present a vision algorithm which is able to estimate the vehicle's position (bearing and lateral offset) with respect to the centre of its current highway lane together with the pitch of the camera and width of the lane. The curvature and rate of change of curvature of the lane up to 80 metres ahead is also estimated, to coincide with the field of view of the radar obstacle detection system. The system has been demonstrated on several hours' worth of data captured from the test vehicle and is found to perform stably in a range of road and weather conditions. We review related work below, before giving details of the algorithm and some examples of its operation.

1.1. Previous Work

Attention has previously been focussed on the topic of lane tracking, although much of this has concentrated on the use of vision information for steering control, a task which requires a much shorter look ahead distance than 80m. Dickmanns *et al*[4] have demonstrated a system that can provide position and curvature estimates using a Kalman filter estimator [5] whose observations are image edges – a controlled search for these features allows edges that do not correspond to useful road markings to be rejected. In subsequent work [3], they use two cameras for road shape estimation; one wide-angle to cover the near-field, and a longer focal length camera to obtain better far-field information.

Other algorithms based upon Kalman filtering include that of Özgüner *et al* [10], and Taylor *et al* [13]. Both of these methods use matched filters to extract lane markings from images which are then used to fit a road model. Özgüner *et al* fit a cubic model of road shape to the image data, with the Kalman filter providing prediction and smoothing (a data association policy is required to match predictions with image measurements). Taylor *et al* use a Hough transform algorithm to fit straight lines to extracted

features and then estimate offset and bearing to the road centre from these lines. Curvature is estimated from successive bearing measurements by their Kalman filter.

The CMU Navlab project [14] currently uses the RALPH algorithm [11], which operates on low-resolution rectified images of the road ahead of the vehicle. The rectification process corrects for the perspective projection, yielding a new image where road lane markings are parallel; by averaging over the rows of this image, a profile is obtained, and vehicle position information is obtained from this profile. To estimate road shape, RALPH hypothesises five different curvature values and uses these to straighten out any curvature present in the rectified lane images. The curvature is taken to be the hypothesis that performs this task best. Other methods [1, 12] fit road models to groups of pixels whose brightness exceeds a given threshold.

The largest challenge facing road trackers is the variability likely to be encountered, in the form of differing road conditions, weather conditions and the types (and quality) of road markings used to demarcate lane boundaries. Such variability makes feature extraction difficult and prone to error, in particular the presence of background features incorrectly classified as road markers. The techniques of data association, robust line fitting and averaging used in the work described above are all efforts to reduce the effects of these outliers prior to road shape estimation. By contrast, our approach is to diminish the effects of these outliers by using a robust estimation scheme, the CONDENSATION algorithm [6], that was designed specifically with estimation in the presence of clutter in mind. By dealing with clutter at the higher level of estimation, rather than with lower level image processing, a robust system should result that is less prone to distraction.

2. Road, car and observation models

At the heart of our system is a six-dimensional state vector (denoted $s(t)$ at discrete time t) that describes both the position of the vehicle and the geometry of the road ahead.

$$s(t) = [y_0(t), \tan \epsilon(t), C_0(t), C_1(t), W(t), \theta(t)]^T, \quad (1)$$

where y_0 denotes the lateral offset and ϵ the bearing of the vehicle with respect to the centre-line of the lane, C_0 and C_1 the curvature and rate of change of curvature of the lane ahead of the vehicle, W the width of the lane, and θ the pitch of the camera to the road surface, which is assumed to be locally (i.e. within the 80m working range of our system) flat.

2.1. The road model

Given a state vector $s(t)$, equation 2 describes the shape of the road ahead of the vehicle.

$$y(x) = y_0 + \tan(\epsilon)x + \frac{C_0}{2}x^2 + \frac{C_1}{6}x^3, \quad (2)$$

where y is the lateral position of the road centre with respect to the vehicle, and x the distance ahead, as illustrated in figure 1.

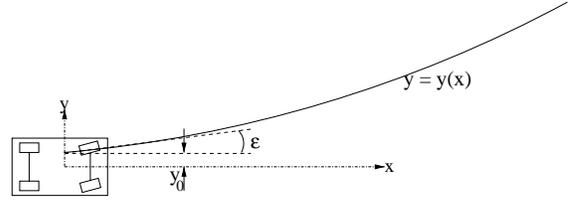


Figure 1. The camera, road and image coordinate systems.

2.2. State evolution model

Figure 2 shows the vehicle in motion between discrete time t and $t + \Delta t$. We can obtain equations for the evolution of the state vector $s(t)$ by using equation 2 in conjunction with the following relationships

$$\begin{aligned} x(t) &= x(t + \Delta t) + V(t)\Delta t \cos \epsilon(t) \\ y(t) &= y(t + \Delta t) + V(t)\Delta t \sin \epsilon(t), \end{aligned} \quad (3)$$

where $V(t)$ is the vehicle's velocity at time t .

Using the identity $\Delta x = V(t)\Delta t \cos \epsilon(t)$ and we arrive at the following set of equations:

$$s(t + \Delta t) = A(\Delta x)s(t) + \begin{bmatrix} 0 \\ -\dot{\psi}\Delta t \\ 0 \\ 0 \\ 0 \\ \Delta\theta \end{bmatrix} + \mathbf{w}(t), \quad (4)$$

where

$$A(\Delta x) = \begin{bmatrix} 1 & \Delta x & \frac{\Delta x^2}{2} & \frac{\Delta x^3}{6} & 0 & 0 \\ 0 & 1 & \Delta x & \frac{\Delta x^2}{2} & 0 & 0 \\ 0 & 0 & 1 & \Delta x & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

$\dot{\psi}$ is the vehicle's yaw rate and $\Delta\theta$ the change in pitch between time t and $t + \Delta t$ measured from the image as described in section 3.2. Finally, $\mathbf{w}(t)$ is a vector of exogenous

disturbances that reflect the uncertainty in the evolution of the system (see section 4.1 below). Recall from the introduction that we obtain estimates of both $V(t)$ and $\dot{\psi}$ from sensors in our vehicle.

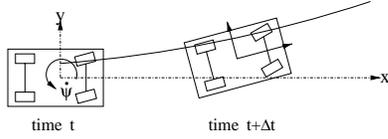


Figure 2. Car motion model.

2.3. Imaging model

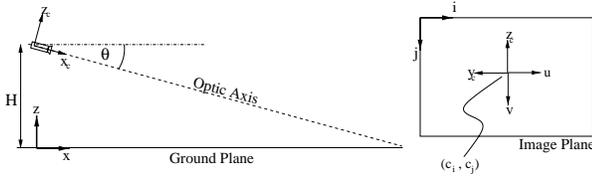


Figure 3. Vehicle, road and image co-ordinate systems. The road y axis points into the page.

Figure 3 depicts the relationship between the camera and the plane of the road ahead of the vehicle, together with a schematic diagram of the image plane co-ordinates $u = -y_c$ and $v = -z_c$. The relationships between u, v and ground plane co-ordinates x, y are:

$$u = \frac{-y}{x \cos \theta + H \sin \theta}, v = \frac{H \cos \theta - x \sin \theta}{x \cos \theta + H \sin \theta}, \quad (6)$$

where the relationships between image plane co-ordinates u and v and pixel co-ordinates i and j are

$$u = \frac{i - c_i}{f_i}, v = \frac{j - c_j}{f_j}. \quad (7)$$

c_i, c_j is the pixel location at which the camera's optic axis intersects the image plane, and f_i and f_j are the effective focal lengths of the camera in the i and j directions respectively.

2.4. Calibration

Our equations above require both intrinsic (f_i, f_j, c_i, c_j) and extrinsic (H, θ) properties of the car's camera to be calibrated. For intrinsic calibration, we use a calibration toolbox developed at the University of Southern California [2]. Features with known geometry are extracted from a set of images of a calibration target (in a range of poses) and then fed into an optimisation routine that produces an estimate of



Figure 4. Extrinsic calibration. A road of known width, with lane marking lines marked by the user.

both the camera's optic centre (c_i, c_j) and the effective focal lengths (f_i, f_j). Once these intrinsic quantities are known, a single image of a straight road of known width can be used to calculate both the height of the camera H , and an initial estimate of camera pitch θ . This procedure simply requires the user to specify the lines that define the width of the lane, as illustrated in figure 4. With the 'flat earth' assumption, these two lines, parallel on the road, intersect in the image at the horizon. The image in figure 4 has significant vertical curvature, the road is however, flat within the desired operating range of our system. The camera pitch θ is calculated using the following relationship

$$v_h = -\tan \theta, \quad (8)$$

where v_h is the v co-ordinate of the horizon line.

The height of the camera may be calculated using the image gradient $m = du/dv$ of the left and right hand lane marking lines, together with the camera pitch and the width W of the lane on the ground:

$$H = \frac{W \cos \theta}{m_r - m_l}, \quad (9)$$

where the subscripts r and l denote the left and right hand lane markers respectively.

3. Image processing

To extract information from images captured by the car's camera, we use two separate algorithms, both described below. The first of these aims to locate the lane markings that describe the shape of the lane in the road ahead. The second image processing algorithm measures the vertical disparity between successive images in order to provide an estimate of change in camera pitch angle.

3.1. Lane marking extraction

At first glance, the task of extracting white lane markings from the background of a dark road may seem straightforward. However, for a number of reasons, the task is surprisingly difficult. The contrast between lane markings and

road is extremely variable owing to factors such as differing road surface material (tarmac, concrete) and lane marking condition (freshly painted, worn, different colours, different shapes and sizes of markings). Also, as with any outdoor system, weather conditions have a great effect on imaging conditions; overcast days are most favourable because the lighting tends to be diffuse and uniform over the scene, whereas strong sunlight adds not only shadows but also yields near saturated images with low contrast between lane markings and road surface. Saturation effects are particularly notable in the far-field where the lane markings tend to be not only very small but also of a similar grey-level to the road surface.

We use a two-stage algorithm for extracting lane markings from the red channel of our colour images (this channel has good contrast properties for both white and yellow image markers). In the first stage, a filter matched to the expected profile of a lane marker (a bright peak on a darker background, roughly triangular in shape) is passed over the image one line at a time, to produce an array of normalised cross-correlation scores. If the cross-correlation score exceeds a threshold, the pixel is denoted as a possible lane marker.

The second stage of the process then inspects the grey-level of each such pixel, and if that also exceeds a threshold, then the pixel is confirmed as a lane marker. It should be noted that we calculate the grey level threshold dynamically in order to account for variations in lighting. This strategy has been found to be effective in a range of conditions from overcast to bright sunlight, and also allows the system to cater for complex shadows that are cast by trees and other road-side structures.



Figure 5. Feature extraction examples (extracted features marked with black dots)

Figure 5 shows example images of two different roads as processed by our algorithm. In both images, the grey-level brightness gradient from bottom to top is notable. In the top image, the lane markers include standard striped markers

and the smaller “Bott’s dots” markings. In the lower image, a scene illuminated by bright sunlight is shown – note the low contrast between lane markings and road surface. It is also clear from the images that the feature detection process produces many false positives, i.e. road surface pixels erroneously labelled as lane markings. This is a consequence of the requirement that the same detection system must extract lane markings of many different types under several different imaging conditions. Clutter is particularly prevalent in the far-field, where lane markings are both small and of low contrast. The system’s estimation scheme (section 4) offers significant robustness to this clutter.

3.2. Vertical disparity estimation

The evolution of the pitch of the camera as the car travels along the road is described by equation 4, and requires an estimate of the change of pitch between frames, $\Delta\theta$. If we can estimate the vertical disparity d_j (along the j image axis) between two successive frames, $\Delta\theta$ can be obtained as follows:

$$\Delta\theta = \frac{d_j}{f_j(1 + \tan^2 \theta)}, \quad (10)$$

where, as above, f_j is the effective focal length of the camera in the j axis of the image, and θ is the current estimate of camera pitch.

Our assumption here is that the disparity d_j reflects the motion of the horizon line between two images (equation 8 above relates the horizon position and pitch angle θ). We calculate d_j by using the sum of absolute differences algorithm on samples taken from around our latest estimate of the horizon line.

4. Estimation by particle filtering

In order to estimate the shape of the road ahead of the vehicle, we have chosen a particle filter, the CONDENSATION algorithm [6]. CONDENSATION was initially developed to address the problem of tracking curves in clutter, which is precisely the situation we have in road tracking (tracking a cubic model in the presence of false positives from the image processing). The term *particle filter* refers to a mechanism for estimating a probability distribution over the state space $s(t)$ given observations from a stream of images. The distribution is approximated by a set of N “particles”, pairs $\{s, \pi\}$, where, as above, s is a state vector, and π is a weight that reflects the plausibility of s as a representation of the true state of the system. Importantly, the method places no assumptions on the distributions involved, and it is this power to represent arbitrary, multi-modal distributions that proves useful when tracking in the presence of the clutter that often confounds uni-modal methods such as the Kalman filter.

The algorithm, described in full by Isard and Blake [6], may be summarised as follows:

Repeat N times for each image:

1. **Stochastically select** a sample s from the particle set, based upon corresponding weight π (a procedure known as *factored sampling* [6]).
2. **Predict** the motion of the sample in the state space using a stochastic state evolution model.
3. **Measure** the plausibility of the evolved sample by comparing its position to that of observations made from the image. Generate a new weight π .

Steps 2 and 3 require state evolution and observation models respectively; these were introduced above in section 2, and we revisit them below in the context of our estimation scheme.

4.1. State evolution

From section 2.2 above, we recall that the evolution of the state s is governed by the following equation:

$$s(t + \Delta t) = A(\Delta x)s(t) + \begin{bmatrix} 0 \\ -\dot{\psi}\Delta t \\ 0 \\ 0 \\ 0 \\ \Delta\theta \end{bmatrix} + w(t). \quad (11)$$

In step 2 of the CONDENSATION algorithm, this equation is used to predict the new position of states sampled from the particle set. The equation can be broken into two parts; deterministic (the matrix A) and stochastic (the noise vector $w(t)$). The deterministic element of the evolution equation forces a drift on the particle set such that it follows the system's overall dynamics.

To generate the prediction, we calculate the deterministic part of the equation and then produce a noise vector $w(t)$ by sampling from a distribution that reflects the uncertainty in the state evolution process. In our system, we assume that this uncertainty has zero mean (it does not introduce bias into the system) and is uni-modal. Also, it is important that the corrupting noise has limited extent, such that the state vector remains within a region of space that describes physically plausible road shapes and camera positions. For computational ease, $w(t)$ is drawn from a Gaussian distribution, and physical constraints are imposed on equation 11 simply by limiting the individual elements of $s(t)$ so that they remain within a region of the state space that describes physically realistic configurations.

4.2. Observation

The observation model is of the form

$$z(t) = h(s(t)) + v(t), \quad (12)$$

where z is the observation, $h(\cdot)$ describes the deterministic relationship between state and observation, and the distribution of the noise vector $v(t)$ specifies the stochastic element of the observation process. As is typical in particle filtering implementations [6], we choose the distribution of $v(t)$ to be uni-modal and zero mean with long tails to permit the presence of outliers in the observed data. In our implementation, z is the position of a lane marker feature, and $h(\cdot)$ describes the relationship between a point on the road (from equation 2) and a pixel in the image (from equations 6 and 7).

The purpose of observation is to score the plausibility of a particle's estimate s . We perform this efficiently by first constructing a score array, where each element corresponds to a pixel in the image of the road, and the score of that element reflects the distance to the nearest detected feature. The relationship between score and distance is based upon the distribution of $v(t)$, and is shown schematically in figure 6. To generate a total score for an estimate s , we simply project the estimated lane marking positions onto the score array (using $h(\cdot)$) and sum the values of the elements lying underneath the projected lines.

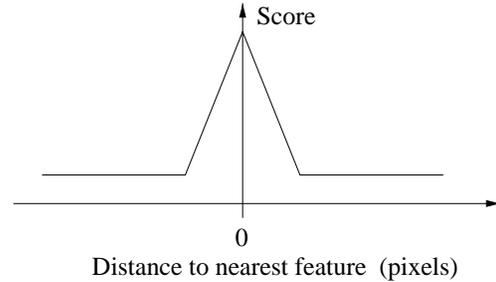


Figure 6. The relationship between score and distance to nearest feature

5. Performance enhancements

We have also incorporated two methods that improve the efficiency of the CONDENSATION algorithm into our implementation. These are known as partitioned sampling [9] and importance sampling [7]. The practical effect of both of these measures is to reduce the total number of particles required for effective tracking. Each will be dealt with separately below, together with a measure to facilitate automatic recovery from loss of track, if that should occur.

5.1. Partitioned sampling

It has been shown [9] that if the state space for a given problem can be partitioned into multiple sub-spaces that may be estimated hierarchically, then the efficiency and speed of the CONDENSATION algorithm can be increased. In our system, we partition the space into $s_1 = [y_0, \tan \epsilon, W, \theta]^T$ and $s_2 = [C_0, C_1]^T$. The first partition describes the width of the road ahead of the vehicle and the camera’s position relative to the centre of the road (the ‘straight road’ properties), whilst the second partition describes the curvature properties of the road.

In the CONDENSATION algorithm with partitioned sampling, described fully by MacCormick and Blake [8], a distribution is estimated over the first partition using the CONDENSATION algorithm described above, and then a second sampling (stochastic selection) stage occurs. In this second stage, particles are selected by factored sampling from the new distribution for the first partition and are paired with samples (again, selected by factored sampling) from the second partition, and these full state particles are then scored against observations to gain an estimate of the full state distribution.

In our system, this allows us to estimate the straight road properties s_1 on features from the lower, less cluttered part of the image where, owing to perspective effects, the road curvature does not appear significant (figure 5), before appending the curvature parameters (s_2) and scoring the full state over the far field features where curvature is more apparent. The algorithm also allows us to concentrate more particles in the “hard” part of the state space, so we use a smaller N for estimating s_1 , because of low clutter in the image closer to the vehicle, and a larger N for the full state estimate owing to the increased clutter in the far field.

5.2. Importance samplers

A second tool for increasing the efficiency of a particle filter is to introduce *importance sampling* functions [7]. These are auxiliary sources of information that allow the introduction of new samples into the particle set as desired. If introduced appropriately, these new samples do not alter the underlying distribution estimated by the particle set, but they allow a re-distribution of the particles so they are concentrated around “more likely” parts of the state space.

For example, in the process of estimating the s_1 partition, we run a Hough transform algorithm to extract image lines that correspond to the left and right hand lane markings, and calculate values for y_0 , ϵ , W and θ on the basis of the line information from a single image. We then introduce samples based upon this instantaneous estimate into the particle set that otherwise contains state estimates based upon a time history. The “new” particles ensure that part of the

state space that seems likely to contain the solution will be inspected during the particle scoring process.

We also use an importance sampler to introduce particles in the estimation of the road curvature C_0 . We have estimates of vehicle velocity v and yaw rate ψ which, using the usual formula for the velocity of an object travelling along a circular arc, yields the instantaneous estimate $C_0 = \psi/v$.

5.3. Initialisation samples

In addition to the importance samples, we introduce a small percentage of initialisation samples, drawn from a distribution of “reasonable” default road shapes, into the particle set at each time step. The inclusion of such samples allows the tracker to recover from any loss of lock that may occur; the algorithm will always be scoring some particles that are in likely positions to be supported by road features. This feature also permits the algorithm to initialise itself automatically.

6. Application to image sequences

We have successfully applied our lane tracking algorithm to image sequences and velocity/yaw rate data collected from the experimental vehicle as it was driven along various state and inter-state highways. The implementation used a set of 550 particles to estimate s_1 , of which 5% at each time-step were drawn from the initialisation distribution and 10% from the Hough transform importance sampling function. The combined s_1 and s_2 estimate was formed from 1450 particles of which 5% were initialisation samples and 15% importance samples from the instantaneous estimate of C_0 . The algorithm executed at a frame rate of approximately 10.5Hz on an off-the-shelf desktop PC (867 MHz Pentium III). Our estimate for the state is obtained from the mean of the particle set, and, if desired, a confidence measure in road geometry can be derived from the variance of the particle set.

It is difficult to obtain ground truth data for lane tracking systems, so quantitative measures of algorithm performance are not given here. The difficulties arise in attempting to measure the vehicle position with respect to a map of the road with sufficient accuracy to judge both offset and bearing relative to the lane in addition to predicting the curvature ahead of the car for comparison with system estimates. Whilst it is true that without this veridical information we cannot judge system accuracy, it is possible to assess robustness to different road and weather conditions by observing the algorithm’s performance on video sequences. We have found that the algorithm functions robustly under many conditions, although saturation caused by bright sunlight and specular reflections from the road during rain storms are problematic.

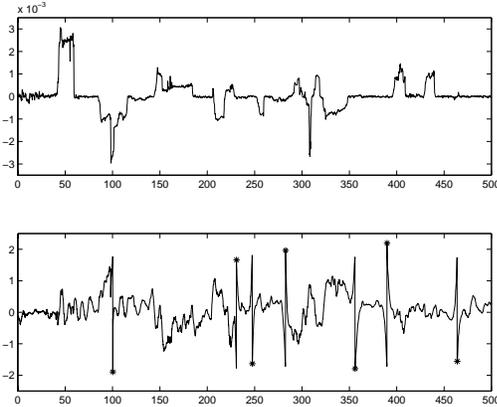


Figure 7. Estimates vs time (s). Top: Road curvature C_0 (m^{-1}). Bottom: Offset y_0 (m)(lane changes are marked with ‘*’).

An accompanying MPEG file shows the algorithm running on test data. (<http://www.cis.upenn.edu/~southall/mpegs/ICCV2001.mpg>). This short sequence illustrates typical behaviour of the algorithm, and includes a lane change, complex shadowing from a bridge and road-side trees, cornering and tracking from a single set of lane markings on the vehicle’s left hand side as it crosses an on-ramp. In this case, the lane width stays constant until the lane markings re-appear on the right, at which point it re-locks onto both set of lane markers. The sequence also highlights the most frequent (but mild) failure of the algorithm; that curvature estimates lag on entry and exit to bends in the road. The state evolution equation 4 does not predict non-random changes of curvature, so the particle set will not be “pushed” into a curve during prediction, and it typically takes a few frames before the image observed image data begins pull the mean of the distribution into the curve. We are currently investigating importance sampling functions for C_1 that will purely use image data. This should increase the system’s responsiveness to curvature changes.



Figure 8. Road shape estimate on a curve.

Example parameter estimates and shape road estimates are illustrated in figures 7 and 8. Figure 7 plots (against time) the estimates for curvature C_0 (top) and lateral offset y_0 (bottom). The y_0 graph is accentuated by a series

of spikes; these correspond to lane changes – the system detects a lane change and resets the y_0 origin to the centre of the new lane automatically. Each lane change is marked with a ‘*’; if this is at the top of a spike, there is a leftwards lane change, at the bottom rightwards.

Figure 8 shows the mean of the particle superimposed (in black) on an image captured as the host vehicle follows another car around a curve. Note that the curvature of the road is correctly estimated even when the majority of the right hand lane marking is occluded by the leading car.

7. Conclusions

We have presented an algorithm for estimating vehicle position and road shape from a single forward-looking camera mounted behind the windshield of a standard passenger car. The algorithm performs at 10.5 frames per second and has shown robustness to variations in both road properties and illumination conditions. Owing to the underlying estimation scheme, the algorithm is able to self-initialise and recover automatically in the few cases where track is lost. We are currently working to improve performance on curve entry and exit, and also on experimental procedure for measuring performance with respect to ground truth data.

Acknowledgments

This work was funded by a joint effort between the U.S. Department of Transportation, General Motors, and Delphi Automotive Systems. We also thank General Motors and Delphi Automotive Systems for the use of the test vehicle.

References

- [1] M. Bertozzi and A. Broggi. Vision-based vehicle guidance. *Computer*, 30(7):49–, July 1997.
- [2] J. Y. Bouget. Matlab camera calibration toolbox. http://www.vision.caltech.edu/bougetj/calib_doc/index.html.
- [3] E. D. Dickmanns. Vehicles capable of dynamic vision: a new breed of technical beings? *Artificial Intelligence*, 103(1–2), August 1998.
- [4] E. D. Dickmanns and B. D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):199–213, February 1992.
- [5] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
- [6] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *Int. J. Computer Vision*, 1998.
- [7] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. ECCV 1998*. Springer Verlag, 1998.
- [8] J. P. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *Int. J. Computer Vision*, 39(1), August 2000.

- [9] J. P. MacCormick and M. Isard. Partitioned sampling, articulated objects and interface-quality hand tracking. In *Proc. ECCV 2000*. Springer Verlag, 2000.
- [10] Ü. Özgüner, K. A. Ünyelioglu, and C. Hatipoğlu. An analytical study of vehicle steering control. In *Proc. 4th IEEE Conference on Control Applications*, pages 125–130, 1995.
- [11] D. Pomerleau. RALPH: Rapidly adapting lateral position handler. In *Proc. Intelligent Vehicles*, pages 54–59, 1995.
- [12] L. T. Schaaser and B. T. Thomas. Finding road lane boundaries for vision-guided vehicle navigation. In I. Masaki, editor, *Vision-based vehicle guidance*. Springer-Verlag, 1991.
- [13] C. J. Taylor, J. Košecká, R. Blasi, and J. Malik. A comparative study of vision-based lateral control strategies for autonomous highway driving. *Int. J. Robotics Research*, 18(5), May 1999.
- [14] C. E. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the Carnegie-Mellon navlab. *IEEE Trans. Pattern Anal. Machine Intell.*, 10(3):362–373, May 1988.