The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Lane Boundary and Curb Estimation with Lateral Uncertainties

Albert S. Huang and Seth Teller
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA
Email: albert@csail.mit.edu, teller@csail.mit.edu

Abstract— **This paper describes an algorithm for estimating lane boundaries and curbs from a moving vehicle using noisy observations and a probabilistic model of curvature. The primary contribution of this paper is a curve model we call *lateral uncertainty*, which describes the uncertainty of a curve estimate along the lateral direction at various points on the curve, and does not attempt to capture uncertainty along the longitudinal direction of the curve. Additionally, our method incorporates expected road curvature information derived from an empirical study of a real road network.**

Our method is notable in that it accurately captures the geometry of arbitrarily complex lane boundary curves that are not well approximated by straight lines or low-order polynomial curves. Our method operates independently of the direction of travel of the vehicle, and incorporates sensor uncertainty associated with individual observations. We analyze the benefits and drawbacks of the approach, and show results of our algorithm applied to real world data sets.

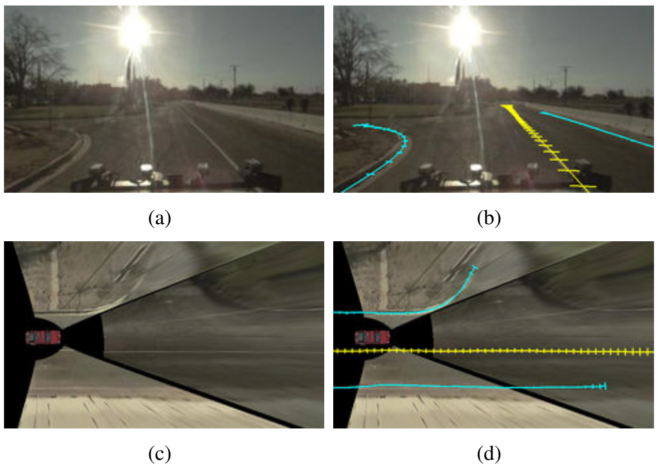


Fig. 1. Lateral uncertainty allows us to concisely model the uncertainty associated with a piecewise linear curve. (a) A camera image (b) Lidar-detected (cyan) and vision-detected (yellow) lane boundary estimates, and their lateral uncertainties projected into image space. (c) Three camera images from the same scene projected onto a ground plane. (d) The lane boundary estimates shown on the ground plane.

I. Introduction

The road networks of countries around the world carry countless numbers of people and goods to their destinations each day. To assist the safe and efficient transport of their travelers, roadways are typically marked with painted and physical boundaries that define the safe and legal regions of travel. The exact nature and appearance of these markings vary from region to region, but all serve to delineate the lanes within which a single file of vehicles is intended to travel.

A system able to automatically and reliably estimate the roadway and its lanes from a moving vehicle using on-board sensor data would have enormous benefits for land-based travel. It could be used for tasks ranging from wide-scale road and lane quality assessments, to providing inputs to a driver assistance safety system, to serving as a navigational component in a fully autonomous vehicle.

These tasks have slightly different requirements, but all require that the system be able to divine the shape and geometry of at least some part of the roadway and its lanes. To do so, the system must utilize information from its on-board sensors, and any other a priori information it has available (e.g. from a road map).

We divide the lane-finding problem into three individual sub-problems: Feature detection, boundary estimation, and lane tracking. The feature detection problem refers to the use of on-board sensors to detect road paint, curbs, and other environmental markings such as color or texture discontinuities that may be used to demarcate the roadway and its lanes. The boundary estimation problem is that of using the detected features to estimate the number and shape of each of the lane boundaries. Finally, the lane tracking problem is to infer the number and shape of the travel lanes.

In this paper, we consider the sub-problem of lane boundary estimation for an autonomous vehicle. Specifically, given a set of noisy observations that are likely to correspond to lane boundary fragments, we are concerned with fusing those observations into curve estimates of potential lane boundaries that are tracked and filtered over time and space. Our system should be able to estimate lane boundaries independently of their orientation with respect to the vehicle, so that the system has good situational awareness (e.g. when exiting parking lots and arriving at intersections). Additionally, our system must accurately model a wide variety of lane boundary geometries, to effectively capture the shapes of real roads and lanes.

The primary contribution of this paper is an algorithm for tracking curves from noisy observations. Central to this algorithm is a novel probabilistic curve model we call *lateral uncertainty*, and an empirically determined model of road curvature. The key insight is to describe the uncertainty of our system's estimates along the lateral direction of the curve, and not on the longitudinal direction, as illustrated in Fig. 1. This allows us to robustly and efficiently incorporate noisy observations into a tracked curve estimate.

II. Related Work

Aspects of the lane-finding problem have been studied for decades in the context of autonomous land vehicle development [1], [2] and driver-assistance technologies [3], [4], [5]. McCall and Trivedi provide an excellent survey [6]. Early work in autonomous vehicle development often made simplifying assumptions on road curvature and vehicle orientation relative to the road. These systems exhibited limited autonomy in the sense that a human driver had to "stage" the vehicle into a valid lane before enabling autonomous operation, and to take control whenever the system could not handle the required task, for example during highway entrance or exit maneuvers [2]. Recent work has attempted to relax many of these requirements [7], [8], [9].

Because sensing the real world is an inherently uncertain process, there has been much work on modeling uncertainty for lane estimation, starting from Dickmanns's original Kalman filter formulation [1]. Recently, Sehestedt et al. described the use of a particle filter for boundary tracking [10], and ZuWhan Kim presented a system to detect and track the left and right boundaries of a single lane using a combination of a support vector machine, RANSAC spline fitting, and a dynamic bayesian network [8].

Our approach to the boundary estimation problem differs from recent work in two key respects. Unlike previous approaches, which are primarily concerned with detecting and tracking boundaries in a frame relative to the vehicle or sensor (usually a camera) [9], [8], [10], our formulation estimates boundaries in a Cartesian frame fixed to the local environment [11]. This has simplifying ramifications, as it eases the fusion of multiple heterogeneous sensors, and the motion of boundary curves is minimized in this coordinate frame. We typically propagate our state estimates forward through time via the identity transformation. The second primary difference is that our piecewise linear uncertainty model allows us to model and track an arbitrary number of curves without any restrictions on their positions or orientations relative to the vehicle.

III. Approach

We represent a continuous parametric 2-D curve $\mathbf{f}$ as:

$$\mathbf{f}(s) = (f_x(s), f_y(s))^\top, s \in [s_1, s_n] \tag{1}$$

where $f_x(s)$ and $f_y(s)$ are the x and y coordinates of the curve, parametrized by the scalar value s, and defined on the domain $s \in [s_1, s_n]$. In the context of road and lane boundaries, we treat a boundary as a single parametric curve, with coordinates expressed in a Cartesian frame fixed to the local environment. The length of the curve could be on the order of meters for short streets or merge lanes, or thousands of kilometers for a transcontinental highway.

For simplicity, we consider only curve representations that are piecewise linear, and represent a curve $\mathbf{f}$ by its $n \times 2$ matrix of control points $\mathrm{F} = (\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n)^\top$. A point $\mathbf{f}(s)$ on the curve can then be determined by simple linear interpolation. We will refer to the curve alternately as either $\mathbf{f}$ or F, depending on context. Additionally, we assume some method of inferring first and second order spatial derivatives from the control points, such as by fitting quadratic or cubic splines [12].

The goal of our method is to produce a useful estimate $\hat{\mathbf{f}}$ of $\mathbf{f}$, given a set of noisy observations. We seek to estimate the region of $\mathbf{f}$ that is within sensor range of our vehicle. We note that it is typically the case that the domain on which $\hat{\mathbf{f}}$ is defined is a subset of the domain on which $\mathbf{f}$ is defined. Thus, we define our estimate $\hat{\mathbf{f}}$ as:

$$\hat{\mathbf{f}}(s) = (\hat{f}_x(s), \hat{f}_y(s))^\top, s \in [r_1, r_n] \tag{2}$$

and note that $s_1 \leq r_1 < r_n \leq s_n$.

A. Lateral Uncertainty

We do not typically have knowledge of the true form of a curve $\mathbf{f}$, and would like to define a probability distribution over its possible shapes. A straightforward approach is to assume that the control points of the polyline are normally distributed, and represent the uncertainty with a $2n \times 1$ mean vector and $2n \times 2n$ covariance matrix.

The major drawback of this method is that allowing the control points of the polyline to vary in all directions provides unnecessarily many degrees of freedom. Specifically, the shape of the curve does not change significantly if we move a polyline control point by a small amount along the longitudinal direction. In the case of three collinear control points, moving the middle control point longitudinally does not change the curve shape at all. Thus, using a covariance matrix on the control point coordinates is an inefficient way to represent uncertainty.

Instead, we propose to represent the probability distribution over the shape of a curve using a mean $\mathrm{F}_\mu = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_n)^\top$ and a *lateral uncertainty* term represented by the $n \times n$ covariance matrix[1] Σ. Given a random matrix $\mathrm{G} = (\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_n)^\top$ of control points representing a curve drawn from the distribution, such that each $\mathbf{g}_i$ can be expressed as:

$$\mathbf{g}_i = \boldsymbol{\mu}_i + w_i \bar{\boldsymbol{\mu}}_i \tag{3}$$

where w_i is a scalar, and $\bar{\boldsymbol{\mu}}_i$ is the unit normal vector of the curve at $\boldsymbol{\mu}_i$, we define the probability density of G as:

$$P_{\hat{\mathrm{F}},\Sigma}(\mathrm{G}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}\mathbf{w}^\top \Sigma^{-1} \mathbf{w}) \tag{4}$$

where $\mathbf{w} = (w_1, w_2, \dots, w_n)^\top$ is the vector of residuals, distributed according to $\mathbf{w} \sim N(0, \Sigma)$. For our purposes, the mean is the true curve, such that $\mathrm{F}_\mu = \mathrm{F}$ and $\bar{\mathbf{u}}_i = \bar{\mathbf{f}}_i$. Fig. 1 illustrates this with a number of lane boundary estimates, and their 3-σ lateral uncertainties marked by short line segments at the control points of the curves. The cross-covariances of Σ are not illustrated.

[1]Our implementation uses diagonal covariances for speed, but the derivation for dense covariances is given for generality and simplicity of explanation.

The intuition behind this formulation is that we allow the control points of the curve to vary only along the curve normal. Thus, each control point has one degree of freedom instead of two; even though G has $2n$ distinct components, it has only n degrees of freedom.

We have defined G such that each of its control points lies on the normal vector of a control point of F. To evaluate the probability density of an arbitrary curve $\mathbf{g}$, we can re-sample its control points to satisfy this requirement. This process preserves accuracy so long as long as the difference between the original curve $\mathbf{g}$ and the polyline representation G is small (e.g. as measured by the area between the two).

If we re-sample the control points of a curve on which a probability distribution has been defined, then we will also need to re-define the distribution in terms of the new control points. If H is a $k \times n$ matrix, where k is the number of new control points, then HG is a random variable with mean HF and lateral uncertainty $\mathrm{H}\Sigma\mathrm{H}^\top$. A re-sampling of curve control points simply amounts to choosing H such that each of its rows has at most two non-zero entries that are adjacent and sum to unity.

B. Estimation

In estimating and tracking curves, we do not have absolute knowledge of either the true curves, or the true probability distributions. Instead, we work with estimates of these values. We refer to our estimate of F as $\hat{\mathrm{F}} = (\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \ldots, \hat{\mathbf{f}}_n)^\top$, our estimate of Σ as Q, and our estimate of $\bar{\mathbf{f}}_i$ as $\breve{\mathbf{f}}_i$. These estimates are themselves random variables, and if our model is correct, the expected values of these estimates are the true values. As is standard practice in estimation, we will often use the estimates in place of the true values for our work.

C. Observations

We define a noisy observation $\mathbf{z}$ of $\mathbf{f}$ as:

$$\mathbf{z}(u) = \mathbf{f}(c(u)) + v(u)\bar{\mathbf{f}}(c(u)), u \in [u_1, u_m] \tag{5}$$

where $v(u)$ is a scalar noise term, $\bar{\mathbf{f}}(u)$ is the unit normal vector of $\mathbf{f}$ at u, and $c(u)$ is a parametrization function that maps values of u to values of s. It is typically the case that $\mathbf{z}$ is only a partial observation of $\mathbf{f}$, such that $s_1 \leq c(u_1) < c(u_m) \leq s_n$. Thus, $\mathbf{z}$ is a random curve whose probability distribution is determined by $\mathbf{f}$ and $v(u)$.

As before, we represent $\mathbf{z}$ with the control point matrix $\mathrm{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m)^\top$ and the lateral uncertainty covariance matrix R. In particular, each control point $\mathbf{z}_i$ can be written:

$$\mathbf{z}_i = \mathbf{f}_{c_i} + v_i\bar{\mathbf{f}}_{c_i} \tag{6}$$

where c_i is the discrete version of the parametrization function c, and v_i is a noise term such that the vector $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ is a random variable distributed according to $\mathbf{v} \sim N(0, \mathrm{R})$.

If we define F_z to be the $m \times 2$ matrix with $\mathbf{f}_{c_i}$ as its ith row, and $\bar{\mathrm{F}}_z$ to be the $m \times 2$ matrix with $\bar{\mathbf{f}}_{c_i}$ as its ith row, then we can re-write Z as:

$$\mathrm{Z} = \mathrm{F}_z + \mathrm{diag}(\mathbf{v})\bar{\mathrm{F}}_z \tag{7}$$

In short, we define an observation Z to be a polyline where each control point of the observation corresponds to a point on the true curve, plus a noise component along the lateral direction of the true curve.

D. Data Association

Given a curve estimate $\hat{\mathrm{F}}$ with lateral uncertainty Q, and an observation Z with lateral uncertainty R, we would like to determine if Z is an observation of $\mathbf{f}$, or if it is an observation of a different curve. We assume that Z is sampled such that each control point $\mathbf{z}_i$ lies on the normal vector of $\hat{\mathbf{f}}_{c_i}$. Thus, each $\mathbf{z}_i$ can be expressed as:

$$\mathbf{z}_i = \hat{\mathbf{f}}_{c_i} + e_i\breve{\mathbf{f}}_{c_i} \tag{8}$$

for a scalar value e_i. Denote P as the $m \times m$ sub-matrix of Q such that $\mathrm{P}_{j,k} = \mathrm{Q}_{c_j,c_k}$. Additionally, define the vector of residuals $\mathbf{e} = (e_1, e_2, \ldots, e_m)^\top$. Finally, define the scalar random variable y to be the Mahalanobis distance:

$$y = \mathbf{e}^\top(\mathrm{R} + \mathrm{P})^{-1}\mathbf{e} \tag{9}$$

If $\mathbf{z}$ is an observation of the curve $\mathbf{f}$, then $\mathbf{e}$ has zero mean, and y obeys a χ^2 distribution with m degrees of freedom [13]. With this, we can compute a p-value and apply a standard goodness-of-fit test to determine if $\mathbf{z}$ is an observation of $\mathbf{f}$, or if it is an observation of a different curve.

When simultaneously estimating and tracking multiple curves, we can apply a greedy matching procedure, whereby an observation is associated with the curve that best "explains" that observation. If no tracked curve could reasonably have been expected to generate the observation, then a new curve is defined, initialized with the mean and lateral uncertainty of the observation.

E. Curvature Prediction

The data association procedure described above is applicable only when the observation and tracked curve have some longitudinal overlap. It is often the case that the observation $\mathbf{z}$ is actually part of the true curve $\mathbf{f}$, but has no overlap with the current estimate $\hat{\mathbf{f}}$. Consider the case of tracking a lane boundary marked with a dashed line; when a new dash is observed, we would like to associate it with the existing curve estimate.

Intuitively, if we have observed one portion of a curve, we can reliably predict the shape of the nearby unobserved parts of the curve. The direction and curvature of lane boundaries do not change very rapidly, and are governed by the physical limitations of the vehicles they carry. In order to make this prediction, we must first have a model of how the curve evolves over space.

The state of Massachusetts publishes a dataset containing the geometry of more than 61,000 km of public roads in the state, produced by manual annotation of ortho-rectified aerial imagery [14]. We fit a simple first-order Markov model to

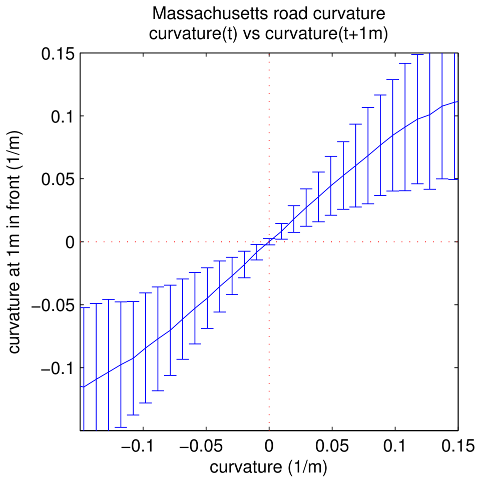


Fig. 2. A road curvature prediction model, fit to MassGIS data on public roads in the state of Massachusetts. 1-σ bounds are shown.

this data set to produce a generative model of road curvature. Specifically, given the signed curvature at one point on a lane boundary, this model predicts the curvature of the lane boundary one meter farther down the curve (Fig. 2). Higher-order models can be expected to improve prediction accuracy and reduce prediction variance.

Using this curve model, we can extend both our estimate $\hat{\mathbf{f}}$ of $\mathbf{f}$ and the observation $\mathbf{z}$. If the original observation $\mathbf{z}$ was reasonably close to the original curve estimate, but did not actually have any longitudinal overlap, then the extensions may have enough overlap to robustly determine if the two correspond to the same underlying curve. This is illustrated in Fig. 3.

F. Update

Once an observation $\mathbf{z}$ has been associated with an existing curve estimate $\hat{\mathbf{f}}$, we use $\mathbf{z}$ to update our estimate, analogous to the update step of a standard Kalman filter [13]. If the observation $\mathbf{z}$ does not span the length of $\hat{\mathbf{f}}$, we first augment its lateral uncertainty covariance matrix R with entries corresponding to unobserved parts of $\hat{\mathbf{f}}$ set to infinity. The control point matrix Z is also extended. Similarly, if $\mathbf{z}$ extends beyond $\hat{\mathbf{f}}$, then $\hat{\mathrm{F}}$ and Q are augmented accordingly. Without loss of generality, we assume that Z is an $n \times 2$ matrix, where each row of Z lies on the normal vector of $\hat{\mathbf{f}}$ at the same row in F.

The updated mean, which we denote as $\tilde{\mathrm{F}}$, is then:

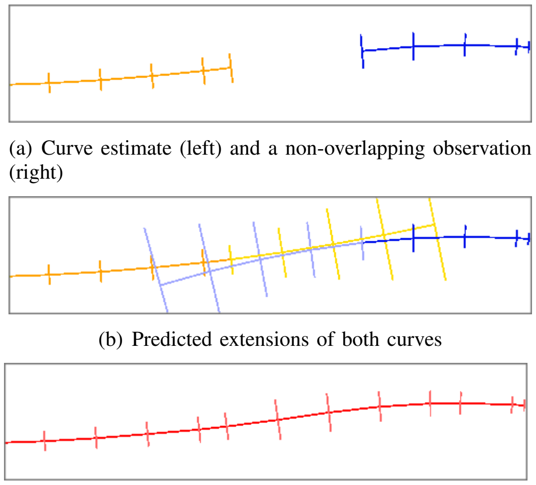
$$\tilde{\mathrm{F}} = \hat{\mathrm{F}} + \mathrm{diag}(\mathrm{Q}(\mathrm{Q}+\mathrm{R})^{-1}\mathbf{e})\breve{\mathrm{F}} \tag{10}$$

The updated lateral uncertainty covariance matrix, which we denote as $\tilde{\mathrm{Q}}$, is expressed as:

$$\tilde{\mathrm{Q}} = (\mathrm{I} - \mathrm{Q}(\mathrm{Q}+\mathrm{R})^{-1})\mathrm{Q} \tag{11}$$

Intuitively, this update allows our estimate of F to shift a control point only along the curve normal at that control point. The amount by which a control point is shifted is determined from the lateral uncertainties of the original curve and of the observation.

By shifting the control points, we have also changed the curve normals. Since our uncertainty is defined only along the direction of the curve normals, the actual probability distribution over the curve changes after re-computing the normal vectors. The amount of error introduced by this change is directly related to the angle change of the normal vector, and introduces approximation errors analogous to those found in the linearization step of an extended Kalman filter.



(a) Curve estimate (left) and a non-overlapping observation (right)

(b) Predicted extensions of both curves

(c) Final curve estimate

Fig. 3. A curvature prediction model allows us to associate observations with existing curve estimates when there is no overlap. Short line segments perpendicular to the curves indicate the 1-σ lateral uncertainty.

IV. Implementation

To assess its performance, we implemented our algorithm on a Land Rover LR3 passenger vehicle equipped with multiple cameras and laser range finders. As input to our system, we used road paint and curb detection algorithms developed at MIT [7], [15]. Both algorithms output piecewise linear curves in a local coordinate frame.

Curves corresponding to road paint and curbs were tracked separately; road paint and curbs were never mixed to update the same curve estimate. Polyline control points were spaced at 1m intervals, and each curve was re-sampled after every update to maintain longitudinal control point spacing.

Through a number of experiments, we arrived on several useful thresholds. To reduce false matches, we required that a detection and a tracked curve overlap by 4.0m. This threshold includes overlap of predicted curve extensions. When extending curves based on our prediction model, extensions were terminated when the 1-σ lateral uncertainty at the end of the curve exceeded $1.5m$. Additionally, we set a p-value threshold of 0.94 for the χ^2 error statistic y.

To prevent overconfidence as a result of not accounting for correlations across observations, we applied a minimum bound of 0.1m on the 1-σ lateral uncertainty of each control point.

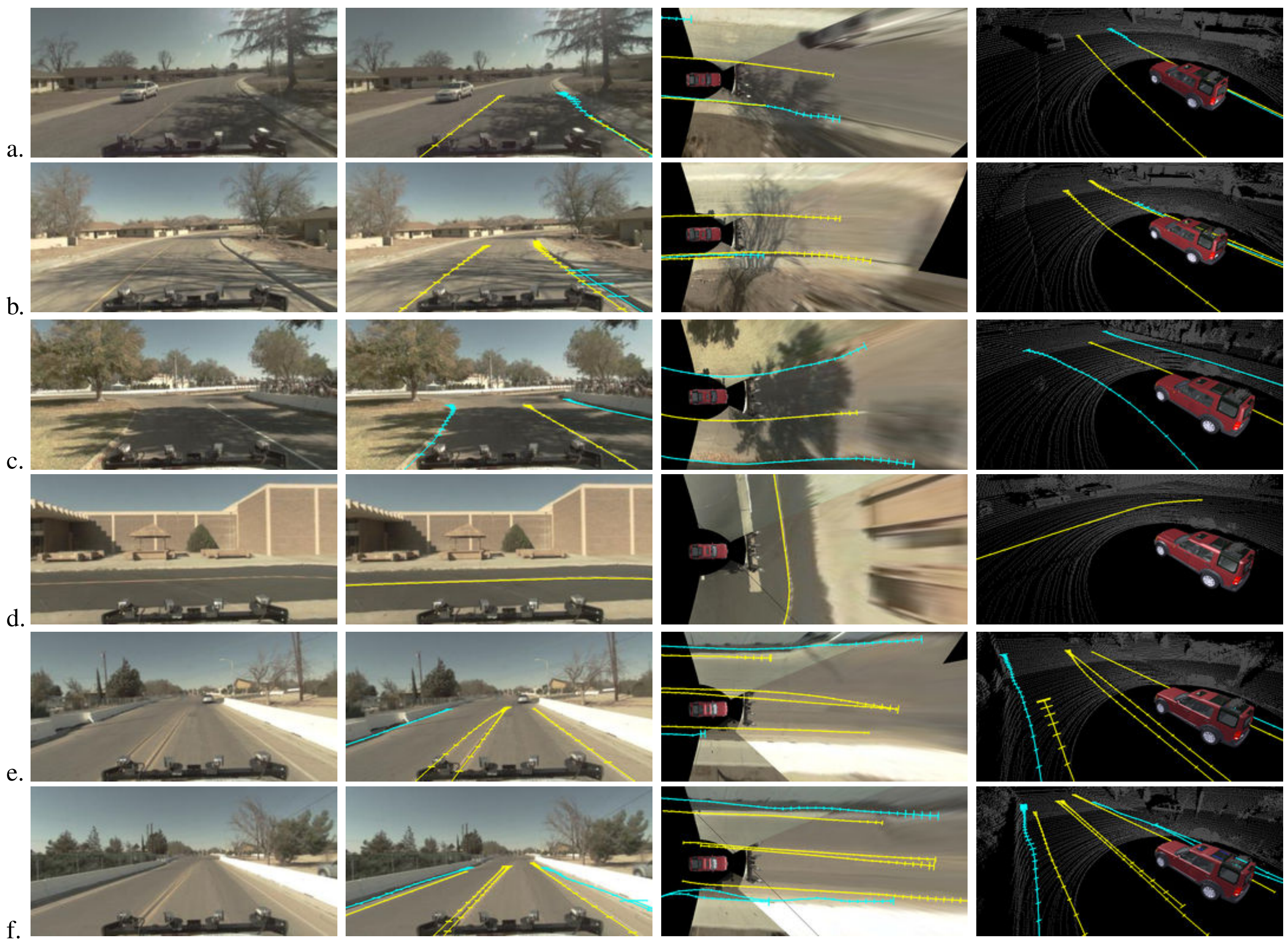


Fig. 4. Lane boundary tracking results, with curve estimates longer than 20m depicted. The first column shows a camera image. The second column shows tracked road paint (yellow) and curb (cyan) estimates superimposed on the image. The third column shows a simulated overhead view with three camera images inverse-perspective mapped onto an assumed ground plane, and the curves superimposed accordingly. The final column shows a simulated viewpoint left of and behind the vehicle with the lidar point cloud shown in grey.

V. Results

We tested our method on the MIT Urban Challenge dataset [15], which contains synchronized camera, lidar, and navigational data for 90 km of travel through a mock urban environment. This dataset also contains the output of the road paint and curb detection algorithms used as input to our method.

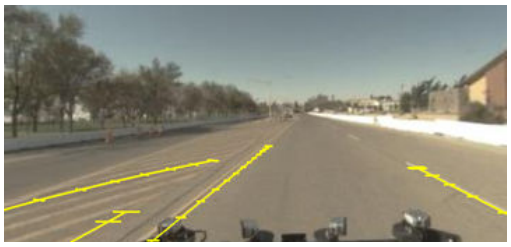
Fig. 4 shows a number of areas where our method performed well. Each row contains four images depicting the same scene. Vision-detected curves (road paint) are drawn in yellow, with lateral uncertainties represented by short perpendicular line segments. Lidar-detected curves (curbs) are similarly drawn in cyan. In both cases, only curves longer than 20m are depicted. Our system tracks curves of all lengths; we use the 20m threshold as an example of how one might further distinguish curves corresponding to lane boundaries.

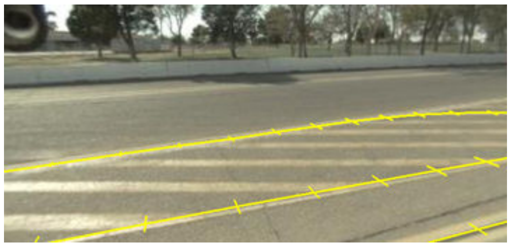
In Figs. 4a-4c, our method is able to successfully track a painted lane boundary in the presence of strong shadows. The road paint detection algorithms produce many false alarms in these cases, but only the detections that are highly likely to correspond to the tracked curve are used to update the painted boundary estimate. Fig. 4d illustrates our method's ability to track lane boundaries perpendicular to the vehicle's direction of travel, such as would be encountered when exiting a parking lot or at an intersection. In Figs. 4e and 4f, our method is able to track closely spaced curves.

A. Limitations

An immediately apparent limitation of our algorithm is that it does not account for error correlations between individual detections. Thus, if a series of detections are used to update a tracked curve, and each of the detections has highly correlated error, then our system will produce overconfident results. We are not aware of any other lane estimation system that robustly addresses this issue.

It is possible to employ heuristics to reduce, but not eliminate, this effect. For example, we enforced a lower bound on the lateral uncertainties to prevent overconfidence. In addition, we may be able to estimate and account for certain inter-detection error correlations by computing the distance between vehicle poses at detection time. A simple


(a) Forward camera view


(b) Leftward camera view

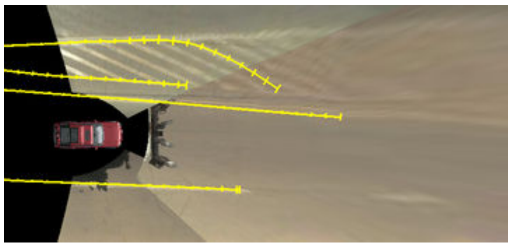

(c) Simulated overhead view

Fig. 5. Our method can fail when a detection is erroneously associated with a tracked curve. In this case, the road paint is detected correctly, but the wrong detection is used to update a curve estimate.

rule could be to ignore the second of two consecutive and identical detections made while the vehicle is halted.

A second limitation of our algorithm is that if a detection is incorrectly associated with a tracked curve, and is also used to extend the curve estimate by an amount that deviates significantly from the true curve, then our method is unlikely to recover. An example of this is shown in Fig. 5. To reduce the chances of this occurring, we could employ several methods. First, a more accurate model of road curvature could be used to reduce the overall matching score errors in some cases. Second, the order in which updates are applied could be determined by a ranking procedure, instead of simply applying updates in the order that matchings are generated. Third, we could apply a limit on how far a tracked curve can be extended by a single observation, as a means to filter spurious matches. Finally, semantic reasoning incorporating information such as marking type, appearance, and context could be used to guide the matching.

VI. Conclusion

Road and lane estimation can be divided into several sub-problems: detecting features such as road paint and curbs; finding and tracking boundary curves; and identifying and tracking the actual road and travel lanes. We have described an approach to the sub-problem of boundary curve estimation and tracking that uses lateral uncertainty to capture probability distributions over piecewise linear curves.

Our method matches curves with an empirically determined probabilistic curvature model, and is able to match curves that do not overlap longitudinally. Our method is robust to noise, successfully suppresses a range of falsely detected features (e.g. those caused by shadows), and is able to track an arbitrary number of curves independently of their position and orientation with respect to the vehicle.

Finally, we showed the results from applying our method to a variety of realistic driving scenarios, discussed its strengths and limitations, and described several possible improvements.

References

[1] E. Dickmanns and B. Mysliwetz, “Recursive 3-D road and ego-state recognition,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 199–213, Feb. 1992.

[2] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, “Vision and navigation for the Carnegie-Mellon Navlab,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, May 1988.

[3] L. Fletcher and A. Zelinsky, “Context sensitive driver assistance based on gaze - road scene correlation.” in *Int. Symposium on Experimental Robotics*, Rio De Janeiro, Brazil, Jul. 2006, pp. 287–296.

[4] N. Apostoloff and A. Zelinsky, “Vision in and out of vehicles: Integrated driver and road scene monitoring,” *International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 513–538, Apr. 2004.

[5] M. Bertozzi and A. Broggi, “GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection,” *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–80, Jan. 1998.

[6] J. C. McCall and M. M. Trivedi, “Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation,” *IEEE Transactions on Intelligent Transport Systems*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[7] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, “Multi-sensor lane finding in urban road networks,” in *Proceedings of Robotics: Science and Systems*, Zürich, Switzerland, June. 2008.

[8] Z. Kim, “Robust lane detection and tracking in challenging scenarios,” *IEEE Trans. Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, Mar. 2008.

[9] Y. Wang, E. K. Teoh, and D. Shen, “Lane detection and tracking using B-Snake,” *Image and Vision Computing*, vol. 22, no. 4, pp. 269 – 280, 2004.

[10] S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, “Robust lane detection in urban environments,” in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, USA, Oct 2007.

[11] D. Moore, A. S. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller, “Simultaneous local and global state estimation for robotic navigation,” in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, May 2009, pp. 3794–3799.

[12] R. H. Bartels and J. C. Beatty, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.

[13] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., 2001.

[14] Executive Office of Transportation, “MassGIS planning roads datalayer description,” http://www.mass.gov/mgis/eotroads.htm, Boston, MA, USA, 2008.

[15] J. Leonard *et al.*, “A perception-driven autonomous vehicle,” *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, Oct 2008.