Probabilistic Lane Estimation using Basis Curves

Albert S. Huang   Seth Teller

MIT Computer Science and Artificial Intelligence Laboratory

Cambridge, MA 02139

albert@csail.mit.edu   teller@mit.edu

Abstract—Lane estimation for autonomous driving can be formulated as a curve estimation problem, where local sensor data provides partial and noisy observations of spatial curves. The number of curves to estimate may be initially unknown and many of the observations may be outliers or false detections (due e.g. to to tree shadows or lens flare). The challenges lie in detecting lanes when and where they exist, and updating lane estimates as new observations are made.

This paper describes an efficient probabilistic lane estimation algorithm based on a novel curve representation. The key advance is a principled mechanism to describe many similar curves as variations of a single *basis curve*. Locally observed road paint and curb features are then fused to detect and estimate all nearby travel lanes. The system handles roads with complex geometries and makes no assumptions about the position and orientation of the vehicle with respect to the roadway.

We evaluate our algorithm with a ground truth dataset containing manually-labeled, fine-grained lane geometries for vehicle travel in two large and diverse datasets that include more than 300,000 images and 44 km of roadway. The results illustrate the capabilities of our algorithm for robust lane estimation in the face of challenging conditions and unknown roadways.

I. Introduction

A system able to automatically and reliably estimate the roadway and its lanes from a moving vehicle using on-board sensors such as cameras or laser range scanners (LIDAR) would have enormous benefits for land-based travel. It could be used for tasks ranging from wide-scale road and lane quality assessment, to providing inputs to a driver assistance safety or navigation system, to serving as a situational awareness component in a fully autonomous vehicle.

Aspects of the lane estimation problem have been studied for decades in the context of autonomous land vehicle development [6, 13] and driver-assistance technologies [3, 4, 1, 9]. The majority of lane estimation systems have been developed for lane keeping and lane departure warning (LDW) in highway environments [11]. The goal of these systems is to estimate the vehicle's current travel lane and, in the case of lane keeping systems, apply minor corrective steering when the vehicle detects an unsignaled lane departure.

LDW systems have achieved success by restricting the problem domain in numerous ways, such as assuming the vehicle begins in a lane and is oriented accordingly, estimating only the vehicle's current travel lane, and optimizing for well-marked highway environments with low road curvature and few, if any, inflection points. High failure rates are acceptable, as a human driver can choose to ignore the LDW output.

However, these simplifications also limit applicability of LDW systems to fully autonomous driving, which has a significantly lower margin for error, and requires more situational awareness than an estimate of the current travel lane. Autonomous vehicle operation on arbitrary road networks will require estimates of all nearby lane geometries, even under challenging environmental conditions.

II. Recent Work

Recent work on lane estimation has focused on operation in more varied and complex environments. Wang et al. used B-Splines to model roads with significant curvature [15], while still retaining the single lane and other LDW assumptions. Matsushita and Miura use a particle filter to detect certain types of intersections [10], in a step towards estimating road network topology. Our work in the DARPA Urban Challenge used a RANSAC-based method to estimate multiple travel lanes from sensor data [7]. The primary challenge to all of these approaches lies in determining which (noisy) sensor observations correspond to lane markings.

In previous work [8], we described the lateral uncertainty algorithm for estimating potential lane boundaries. The algorithm uses probabilistic methods to estimate individual curves from noisy observations, but does not address how to group curves to form lane estimates or track whole lanes over time. Neither does it distinguish true lane boundaries from long tree shadows or other non-boundary painted lines such as stop lines and pedestrian crossings.

This paper generalizes the lateral uncertainty method to perform joint inference of multiple curves (i.e., the left and right boundaries of a lane). It provides a principled framework for ignoring sensor observations that are similar to, but do not correspond to, lane boundaries, and for using observations of one curve to update estimates of another (e.g., when one lane boundary is faded or occluded by traffic). We formulate lane estimation as a curve estimation problem, describe a novel representation for open 2D curves, and present a Bayesian lane estimation algorithm that is robust to the noise and outliers typical of image and LIDAR data.

III. Method Overview

The geometry of a travel lane can be described by a single centerline curve $\mathbf{f}(s) : [s_1, s_n] \to \mathbb{R}^3$ whose width varies along the length of the curve:

$$\mathbf{f}(s) = (f_x(s), f_y(s), f_h(s))^\top \qquad (1)$$

where $f_x(s)$ and $f_y(s)$ describe the lane centerline, and $f_h(s)$ describes the lane half-width, all of which are defined on the

domain $s \in [s_1, s_n]$. The interpretation of s depends on the curve model chosen, and may not have any physical meaning. We describe the curve in a coordinate system fixed with respect to the local environment, such that lane geometry is nearly invariant to time and the vehicle's own state.

Initially, the number of lanes and their geometries are unknown. The goal of our system is to detect all nearby lanes, estimate their geometries, and update these estimates as the vehicle travels. The system takes as input curve fragments corresponding to potential lane boundary observations, such as those detected by a vision-based road-paint detection algorithm, or a LIDAR-based curb-detection algorithm. While many of these curve fragments are true observations of the road and lanes, some may arise from tree shadows or lens flare, and others may simply be due to painted markings that do not correspond to lane boundaries (e.g., crosswalk stripes, parking spot delimiters, or stop lines).

We approximate $\mathbf{f}(s)$ as piecewise-linear, such that it can be described by a series of control points connected by line segments, and note that approximation error decreases with control point density. To reason about lanes, their boundaries, and observations, we introduce the notion of *basis curves*.

The insight behind this representation is that nearby curves often share common structure. Factoring out this structure leaves a set of residuals which are computationally convenient to manipulate. Within the space of these residuals, which we call the basis curve space, lane boundary observations can be expressed as linear transformations of the true lane geometry, and the Gaussian distribution can be used to model uncertainty over curve geometry. Standard estimation algorithms such as the Kalman filter can then be used to estimate complex lane geometries [2].

IV. BASIS CURVES

Consider a piecewise-linear curve $\mathbf{b}$, represented by the matrix of control points $\mathrm{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)^\top$. Denote the unit normal vectors to the curve at each of the control points[1] by the matrix $\bar{\mathrm{B}} = (\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_n)^\top$. We consider a curve $\mathbf{g}$, represented as $\mathrm{G} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n)^\top$, to be a *variation* of $\mathbf{b}$ if each $\mathbf{g}_i, i \in \{1 \ldots n\}$, can be expressed as:

$$\mathbf{g}_i = \mathbf{b}_i + g_i^{\mathrm{B}}\bar{\mathbf{b}}_i \qquad (2)$$

where each g_i^{B} is a signed scalar that indicates how much $\mathbf{g}$ varies from $\mathbf{b}$ along the normal vector at a particular control point. In this context, $\mathbf{b}$ is a *basis curve*, and the vector $\mathbf{g}^{\mathrm{B}} = (g_1^{\mathrm{B}}, g_2^{\mathrm{B}}, \ldots, g_n^{\mathrm{B}})^\top$ is the *offset vector* of $\mathbf{g}$ from $\mathbf{b}$.

Here we have defined $\mathbf{g}$ as a variation of the basis curve $\mathbf{b}$. Suppose $\mathbf{g}$ is instead expressed using a different curve representation, possibly a spline or even a polyline with a different number of control points. If $\mathbf{g}$ is geometrically similar to $\mathbf{b}$, then we can choose a polyline approximation of $\mathbf{g}$ as a variation of $\mathbf{b}$. When the matrix of control points G is defined in this manner, we call the resulting offset vector $\mathbf{g}^{\mathrm{B}}$ the *projection of* $\mathbf{g}$ *onto* $\mathbf{b}$ (Fig. 1).

[1]We use $\bar{\mathbf{b}}_i = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \frac{\mathbf{b}_{i+1}-\mathbf{b}_i}{||\mathbf{b}_{i+1}-\mathbf{b}_i||}, i \in \{1 \ldots n{-}1\}$ and $\bar{\mathbf{b}}_n = \bar{\mathbf{b}}_{n-1}$.

(a) A basis curve (blue) represented by polyline B, and a uniform quadratic B-Spline $\mathbf{g}$.

(b) The projection of $\mathbf{g}$ onto $\mathbf{b}$, producing the g_i^{B}.

(c) Approximation of curve $\mathbf{g}$ as a variation G of the basis curve $\mathbf{b}$.
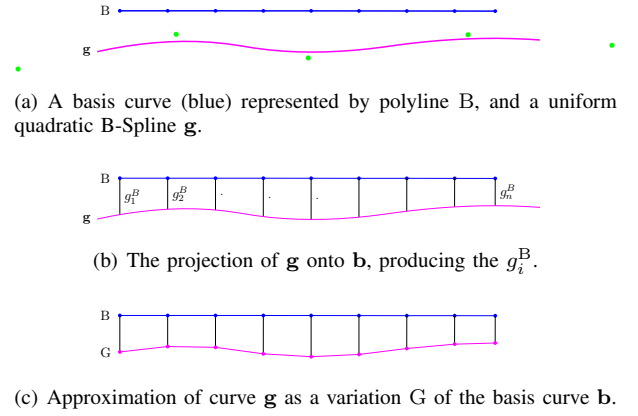
Fig. 1. A uniform quadratic B-Spline (magenta) defined by the green control points can be projected onto a basis curve (blue). Curves using other representations can be projected using analogous intersection procedures.

The error of this approximation (e.g., as measured by the area between the polyline approximation and the original curve) is small if $\mathbf{g}$ and $\mathbf{b}$ have similar geometry, and if the control point spacing of $\mathbf{b}$ is small relative to the curvatures of the two curves. If the curves are very different (e.g., nearly orthogonal or without overlap), it may not be meaningful to project one onto the other.

To review, thus far we have defined:

$\mathbf{b}$	A basis curve.
B	The $n \times 2$ matrix of control points representing $\mathbf{b}$.
$\mathbf{b}_i$	The i-th control point of B.
$\bar{\mathrm{B}}$	The $n \times 2$ matrix of unit normal vectors of $\mathbf{b}$ at each control point.
$\bar{\mathbf{b}}_i$	The i-th unit normal vector of $\bar{\mathrm{B}}$.
$\mathbf{g}$	A curve geometrically similar to $\mathbf{b}$.
G	The $n \times 2$ matrix of control points approximating or representing $\mathbf{g}$.
$\mathbf{g}_i$	The i-th control point of G.
$\mathbf{g}^{\mathrm{B}}$	The projection of $\mathbf{g}$ onto $\mathbf{b}$, also referred to as the offset of $\mathbf{g}$ from $\mathbf{b}$.

We can now use $\mathbf{b}$ to represent common structure shared by many curves, and offset vectors to reason about the differences among curves. One interpretation of this model is that $\mathbf{b}$ induces an n-dimensional vector space, into which we can project any curve. We call this the *basis curve space* of $\mathbf{b}$. When projected into a basis curve space, a curve can be represented by a single n-dimensional point – its offset vector.

In general, any curve that spans the length of $\mathbf{b}$ can be projected onto $\mathbf{b}$. It is also useful to reason about curves that do not run the length of $\mathbf{b}$, but do have some "longitudinal overlap" in that they intersect with one or more support lines of normal vectors of $\mathbf{b}$. In this case, each of the $\frac{n(n-1)}{2}$ consecutive subsets of the control points of $\mathbf{b}$ can be treated as an individual basis curve with its own basis curve space. These smaller basis curve spaces are subspaces of the basis
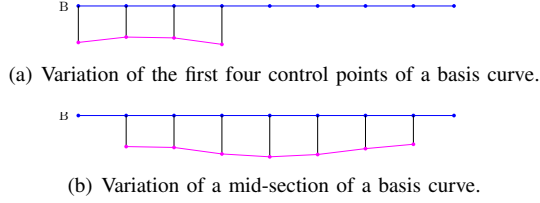
curve space of $\mathbf{b}$ (Fig. 2).

(a) Variation of the first four control points of a basis curve.

(b) Variation of a mid-section of a basis curve.

Fig. 2. Variations (magenta) of a subset of the basis curve (blue) can be described using subspaces of the vector space of the full basis curve.

A. Basis curve normal distributions

Next, we consider probability distributions on curves. Specifically, consider a random variation of a basis curve $\mathbf{b}$, where the offset vector $\mathbf{g}^{\mathrm{B}}$ is normally distributed according to $\mathbf{g}^{\mathrm{B}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Fig. 3 shows a basis curve and several hundred variations, where the offset vectors of each variation are drawn from a multivariate Gaussian.
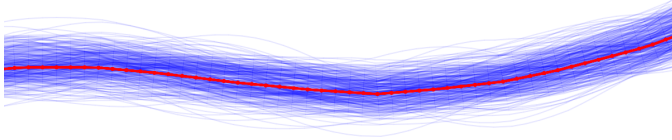

Fig. 3. Three hundred normally distributed variations of a basis curve (red).

Together, $\mathbf{b}$, $\boldsymbol{\mu}$, and Σ define a distribution over curves. We refer to a distribution of this form as a *basis curve normal distribution*, and represent it with the term $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$. The probability density of a curve $\mathbf{g}$ is then defined as:

$$P_{\mathbf{g}}(\mathbf{g}; \mathbf{b}, \boldsymbol{\mu}, \Sigma) = \mathcal{N}(\mathbf{g}^{\mathrm{B}}; \boldsymbol{\mu}, \Sigma) \tag{3}$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$ refers to the probability density function of the normal distribution with mean $\boldsymbol{\mu}$ and covariance Σ.

The intuition behind this formulation is that control point uncertainty is expressed only along the curve normals of the basis curve. 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. The covariance matrix Σ represents the degree and structure of variation between curves drawn from the distribution.

If a distribution is instead defined on the entire control point space such that each control point has 2 degrees of freedom [5], it is possible to sample two curves that have the same shape, but different probability densities. Thus, distributions over the entire space of control points are not necessarily good distributions over curve shape.

Restricting each control point to one degree of freedom has useful properties. In our case, two random curves drawn from a basis curve normal distribution with different probability densities will also exhibit different shapes. Evaluating the likelihood that an arbitrary curve is drawn from a distribution becomes a simple process of computing curve and ray intersections. There are some exceptions to this that arise when the covariance at a control point is large relative to the curvature,

but this has not had a noticeable impact in our usage for lane estimation.

B. Changing basis curves

It is sometimes desirable to switch the basis curve upon which a curve distribution is defined, but without changing the underlying distribution. In general, it is not possible to match the original distribution exactly, and the approximation error introduced by the reparameterization is directly related to the amount by which the basis curve normal vectors change: approximation error is smallest when the new basis curve is locally parallel to the original basis curve. However, if the new basis curve $\mathbf{b}'$ is similar to the original, then reasonable choices of a new mean and covariance, $\boldsymbol{\mu}'$ and Σ', are possible.

Variation: If $\mathbf{b}'$ is a variation of the original basis curve $\mathbf{b}$, then B' can be described by:

$$\mathrm{B}' = \mathrm{B} + \mathrm{diag}(\mathbf{b}'^{\mathrm{B}})\bar{\mathrm{B}} \tag{4}$$

where $\mathbf{b}'^{\mathrm{B}}$ is the projection of $\mathbf{b}'$ onto $\mathbf{b}$. In this case, a new mean can be defined by subtracting the projection of the new basis curve, and the covariance can remain unchanged:

$$\begin{aligned} \boldsymbol{\mu}' &= \boldsymbol{\mu} - \mathbf{b}'^{\mathrm{B}} \\ \Sigma' &= \Sigma \end{aligned} \tag{5}$$

Re-sampling: If $\mathbf{b}'$ is defined by a re-sampling of the control points of $\mathbf{b}$, then each of the m control points of $\mathbf{b}'$ is a convex linear combination of two neighboring control points on the original curve $\mathbf{b}$. Thus, B' can be related to B by:

$$\mathrm{B}' = \mathrm{HB} \tag{6}$$

where H is a $m \times n$ matrix whose rows have at most two positive entries that are adjacent and sum to unity (representing the convex linear combination of neighboring control points). Additionally, $\mathrm{H}_{i,j}$ may be non-zero only if $\mathrm{H}_{i-1,k}$ is zero for all $k > j$ (control points may not be re-sampled out of order).

Since every point on a polyline is either a control point or an interpolation of neighboring control points, we can define a new mean and covariance by applying the re-sampling transformation:

$$\begin{aligned} \boldsymbol{\mu}' &= \mathrm{H}\boldsymbol{\mu} \\ \Sigma' &= \mathrm{H}\Sigma\mathrm{H}^{\top} \end{aligned} \tag{7}$$

V. LANE BOUNDARY ESTIMATION

To detect and estimate lanes, our method first estimates potential lane boundaries, then groups boundary curves together to form lane estimates. Here, we develop a recursive algorithm that estimates a single unknown curve $\mathbf{f}$, which we use to model a potential lane boundary. The algorithm takes as input an initial curve estimate and a noisy observation of the curve, and produces an updated estimate that reflects the novel information provided by the observation.

We refer to our initial belief about $\mathbf{f}$ using the basis curve normal distribution $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$. The matrix of control points $\hat{\mathrm{F}} = (\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \ldots, \hat{\mathbf{f}}_n)^{\top}$ describes our mean estimate $\hat{\mathbf{f}}$, and is obtained from $\mathbf{b}$ and $\boldsymbol{\mu}$ as:

$$\hat{\mathbf{f}}_i = \mathbf{b}_i + \mu_i \bar{\mathbf{b}}_i \tag{8}$$

We use $\mathbf{f}^{B} = (f_1^{B}, f_2^{B}, \ldots, f_n^{B})^{\top}$ to denote the projection of $\mathbf{f}$ onto $\mathbf{b}$. Given a fixed basis curve $\mathbf{b}$, the vector $\boldsymbol{\mu}$ is effectively an estimate of $\mathbf{f}^{B}$. In this sense, there are two estimates: the estimate $\boldsymbol{\mu}$ of $\mathbf{f}^{B}$, and the estimate $\hat{\mathbf{f}}$ of $\mathbf{f}$, where $\hat{\mathbf{f}}$ is defined in terms of $\boldsymbol{\mu}$ and $\mathbf{b}$ as in Eq. (8).

A. Observations

We model an observation $\mathbf{z}$ of $\mathbf{f}$ as a subset of $\mathbf{f}$, corrupted by zero-mean i.i.d. noise. Note that $\mathbf{z}$ is typically a partial observation of $\mathbf{f}$, since only a portion of a painted road boundary may be observed at any given moment. For simplicity, we first consider the case where $\mathbf{z}$ is a full observation of $\mathbf{f}$, and later consider partial observations.

If we project $\mathbf{z}$ onto $\mathbf{b}$ and refer to the piecewise linear approximation of $\mathbf{z}$ as $Z = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)^{\top}$, then each $\mathbf{z}_i$ can be expressed as:

$$\begin{aligned} \mathbf{z}_i &= \mathbf{b}_i + (f_i^{B} + v_i)\bar{\mathbf{b}}_i \\ &= \mathbf{b}_i + z_i^{B}\bar{\mathbf{b}}_i \end{aligned} \quad (9)$$

where we model the noise terms $\mathbf{v} = (v_1, v_2, \ldots, v_n)^{\top}$ as jointly Gaussian with zero mean and covariance R, such that $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, R)$. The projection $\mathbf{z}^{B} = (z_1^{B}, z_2^{B}, \ldots, z_n^{B})^{\top}$ of $\mathbf{z}$ onto $\mathbf{b}$ can then be described as:

$$z_i^{B} = f_i^{B} + v_i \quad (10)$$

and:

$$\mathbf{z}^{B} = A\mathbf{f}^{B} + \mathbf{v} \quad (11)$$

where the observation matrix $A = I_{n \times n}$ is a trivial linear transformation. If $\mathbf{z}$ is a partial observation of $\mathbf{f}$, then it can still be related to $\mathbf{f}$ by choosing an observation matrix A of size $m \times n$, where $m \leq n$, and A has an $m \times m$ identity sub-matrix and is zero elsewhere. For example, if the basis curve has three control points and the observation curve spans only the first 2 of these control points, then the corresponding observation matrix $A_{2 \times 3}$ is:

$$A_{2 \times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (12)$$

B. Data Association

Not all observations correspond to the desired curve $\mathbf{f}$. To avoid corrupting the curve estimate, only true observations should be incorporated, and non-observations ignored or used to update a different curve estimate. Determining if an observation corresponds to a tracked curve is the well-known data association problem [14].

We use a standard χ^2 outlier rejection approach. Define the scalar random variable y as the following Mahalanobis distance [2]:

$$y = (\mathbf{z}^{B} - A\boldsymbol{\mu})^{\top}(R + A\Sigma A^{\top})^{-1}(\mathbf{z}^{B} - A\boldsymbol{\mu}) \quad (13)$$

If $\mathbf{z}$ is an observation of the curve $\mathbf{f}$, then y obeys a χ^2 distribution with m degrees of freedom. Observations that fall in the extremities of this distribution are rejected as outliers.

When simultaneously estimating and tracking multiple curves, we apply a gated greedy matching procedure, in which each observation is associated with the curve that best "explains" that observation. Other techniques, such as joint-compatibility branch and bound [12], may also be used. If no tracked curve is likely to have generated the observation according to the current estimates, then the observation is used to initialize a new curve for tracking.

C. Curvature Prediction

As the vehicle travels, it may receive observations that extend beyond the end of its current curve estimate. To incorporate these observations, the observation and curve estimate are first extended (Fig. 4b). We use a method described in previous work [8], briefly summarized here.

Once we have observed and estimated one portion of a curve, we can reliably predict the shape of the nearby unobserved parts of the curve. To do so, we fit a first-order Markov model of road curvature to a dataset containing the manually annotated coarse geometry of existing public roadways. Given the signed curvature at one point in a road, the model predicts a distribution over the curvature at a further point along the road. This model can then be used to extend both the curve estimates and observations as the vehicle travels.

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

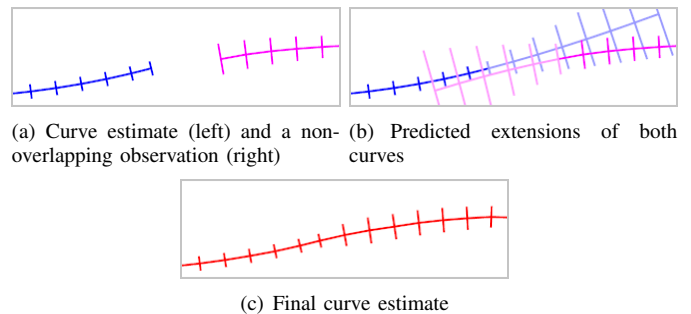(b) Predicted extensions of both curves

(c) Final curve estimate

Fig. 4. A curvature prediction model allows us to associate observations with existing curve estimates when there is little or no overlap. The short line segments perpendicular to each curve indicate 1-σ uncertainty.

D. Update

Once an observation $\mathbf{z}$ has been associated with an existing curve estimate, the algorithm revises that estimate using the update step of a standard Kalman filter [2].

As mentioned earlier, approximation errors resulting from projection onto a basis curve are minimized when the basis curve geometry matches the true curve geometry. Therefore, we reparameterize the curve estimate, so that the basis curve coincides with the newly updated maximum likelihood curve estimate. Since this estimate is a variation of the current basis curve, reparameterization consists of offsetting the basis curve by the mean vector, then setting the mean vector to zero (Sec. IV-B). We also re-sample the basis curve control points to maintain nearly uniform control point spacing.

Finally, we note that propagating the curve estimate forward through time consists of a simple identity transform, a consequence of expressing the curve geometry in a co-ordinate system fixed to the local environment. Algorithm `update_curve_estimate` shows a full update cycle. It

takes as input the initial belief $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$ over a single curve, and an observation. If the observation passes the data association test, then it is used to generate the updated curve estimate $\tilde{\mathcal{N}}(\mathbf{b}^+, \boldsymbol{\mu}^+, \Sigma^+)$. Fig. 4 illustrates an update step for a single curve estimate.

Algorithm `update_curve_estimate`

Input: an initial belief $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$
Input: an observation $\tilde{\mathcal{N}}(\mathbf{b}, \mathbf{z}^{\mathrm{B}}, \mathrm{R})$
Input: an observation matrix A
Output: the new belief $\tilde{\mathcal{N}}(\mathbf{b}^+, \boldsymbol{\mu}^+, \Sigma^+)$

1: Augment the initial estimate and observation as necessary via curve prediction
2: Let m be the dimensionality of $\mathbf{z}^{\mathrm{B}}$
3: Let k be the 0.95 percentile of the χ_m^2 distribution
4: **if** $(\mathbf{z}^{\mathrm{B}} - \mathrm{A}\boldsymbol{\mu})^\top (\mathrm{R} + \mathrm{A}\Sigma\mathrm{A}^\top)^{-1} (\mathbf{z}^{\mathrm{B}} - \mathrm{A}\boldsymbol{\mu}) > k$ **then**
5:     Reject observation as an outlier
6:     **return** $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$
7: $\mathrm{K} \leftarrow \Sigma\mathrm{A}^\top (\mathrm{A}\Sigma\mathrm{A}^\top + \mathrm{R})^{-1}$
8: $\tilde{\boldsymbol{\mu}} \leftarrow \boldsymbol{\mu} + \mathrm{K}(\mathbf{z}^{\mathrm{B}} - \mathrm{A}\boldsymbol{\mu})$
9: $\tilde{\Sigma} \leftarrow (\mathrm{I} - \mathrm{KA})\Sigma$
10: $\mathbf{b}^+ \leftarrow \mathrm{B} + \mathrm{diag}(\tilde{\boldsymbol{\mu}})\bar{\mathrm{B}}$
11: $\boldsymbol{\mu}^+ \leftarrow \mathbf{0}$
12: $\Sigma^+ \leftarrow \tilde{\Sigma}$
13: **return** $\tilde{\mathcal{N}}(\mathbf{b}^+, \boldsymbol{\mu}^+, \Sigma^+)$

VI. LANE ESTIMATION

The boundary curves of a single lane are highly correlated, and information about one boundary gives valuable information about the other. We represent a lane as a piecewise linear centerline curve whose width varies along the curve, and describe its control points with the matrix $\mathrm{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n)^\top$, where each $\mathbf{f}_i$ is defined as $\mathbf{f}_i = (f_{xi}, f_{yi}, f_{hi})^\top$.

Using the convention that the normal vectors of a curve point "left", two points $\mathbf{f}_{li}$ and $\mathbf{f}_{ri}$ on the left and right boundaries, respectively, can be described as:

$$\mathbf{f}_{li} = \begin{bmatrix} f_{xi} + f_{hi}\bar{f}_{xi} \\ f_{yi} + f_{hi}\bar{f}_{yi} \end{bmatrix} \qquad \mathbf{f}_{ri} = \begin{bmatrix} f_{xi} - f_{hi}\bar{f}_{xi} \\ f_{yi} - f_{hi}\bar{f}_{yi} \end{bmatrix} \qquad (14)$$

where $\bar{\mathbf{f}}_i = (\bar{f}_{xi}, \bar{f}_{yi})^\top$ is the normal vector to the centerline curve at point i.

A. Lane distributions

As with zero-width curves, a basis curve can be used to represent and approximate lanes. We describe the projection $\mathbf{f}^{\mathrm{B}}$ of $\mathbf{f}$ onto $\mathbf{b}$ as:

$$\mathbf{f}^{\mathrm{B}} = (f_{c1}^{\mathrm{B}}, f_{h1}, f_{c2}^{\mathrm{B}}, f_{h2}, \ldots, f_{cn}^{\mathrm{B}}, f_{hn})^\top \qquad (15)$$

where $f_{c1}^{\mathrm{B}}, \ldots, f_{cn}^{\mathrm{B}}$ is the projection of the centerline of $\mathbf{f}$ onto $\mathbf{b}$. Thus, the projection of a lane onto $\mathbf{b}$ is simply the projection of its centerline augmented by its half-width terms.

A normal distribution over the projections of $\mathbf{f}$ onto $\mathbf{b}$ defines a distribution over lanes. We use such a distribution, parameterized by $\boldsymbol{\mu}$ and Σ, to represent a belief over the true geometry of $\mathbf{f}$. The mean estimate can be represented by a matrix of control points $\hat{\mathrm{F}} = (\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \ldots, \hat{\mathbf{f}}_n)^\top$, where each control point $\hat{\mathbf{f}}_i = (\hat{f}_{xi}, \hat{f}_{yi}, \hat{f}_{hi})^\top$ can be expressed as:

$$\hat{\mathbf{f}}_i = \begin{bmatrix} \hat{f}_{xi} \\ \hat{f}_{yi} \\ \hat{f}_{hi} \end{bmatrix} = \begin{bmatrix} b_{xi} + \mu_{ci}\bar{b}_{xi} \\ b_{yi} + \mu_{ci}\bar{b}_{yi} \\ \mu_{hi} \end{bmatrix} \qquad (16)$$

where each μ_{ci} describes the mean offset of a centerline control point, and each μ_{hi} describes the mean half-width of the lane estimate at the control point.

It is sometimes useful to change the basis curve upon which a lane distribution has been defined, while incurring minimal changes to the actual distribution. Choosing a new mean and covariance in the cases of re-sampled and offset basis curves follows the same procedure as in Sec. IV-B, with minor modifications. When the new basis curve is a variation of the original basis curve, the width components of the mean lane do not change. When the new basis curve is a re-sampling of the original basis curve, the re-sampling matrix H must account for re-sampling the width components in addition to the centerline offset values.

B. Observation model, data association, and update

A full boundary observation of $\mathbf{f}$ is a curve, which we describe with the matrix of control points $\mathrm{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)^\top$, where each $\mathbf{z}_i$ can be written:

$$\begin{aligned} \mathbf{z}_i &= \mathbf{b}_i + (f_{ci}^{\mathrm{B}} + af_{hi} + v_i)\bar{\mathbf{b}}_i \\ &= \mathbf{b}_i + z_i^{\mathrm{B}}\bar{\mathbf{b}}_i \end{aligned} \qquad (17)$$

where a has value $+1$ or -1 for an observation of the left or right boundary, respectively, and we model the noise terms $\mathbf{v} = (v_1, v_2, \ldots, v_n)^\top$ jointly as $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathrm{R})$.

Collectively, the offset vector $\mathbf{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, z_2^{\mathrm{B}}, \ldots, z_n^{\mathrm{B}})^\top$ can then be expressed as:

$$\mathbf{z}^{\mathrm{B}} = \mathrm{A}\mathbf{f}^{\mathrm{B}} + \mathbf{v} \qquad (18)$$

where the elements of the observation matrix A are chosen to satisfy Eq. (17). If $\mathbf{z}$ is a partial observation of the boundary, such that it projects onto only m control points of $\mathbf{b}$, then A has size $2m \times 2n$, similar to the case for zero-width curves.

Data association and update steps can be approached in the same way as for zero-width curves. Given a lane distribution and observation as expressed above, we can apply a χ^2 test to determine if $\mathbf{z}$ is an observation of $\mathbf{f}$. When estimating multiple lanes, we use a gated greedy assignment procedure to assign observations to lanes. Once an observation has been associated with a lane estimate, the standard Kalman update steps are used to update the mean and covariance.

After the updated estimates have been computed, we once again reparameterize the distribution such that the basis curve coincides with the updated maximum likelihood estimate, to minimize approximation error in future update steps. Fig. 5 shows a full update cycle, where an observation of a lane boundary is used to both update and extend the lane.

(a) A lane estimate

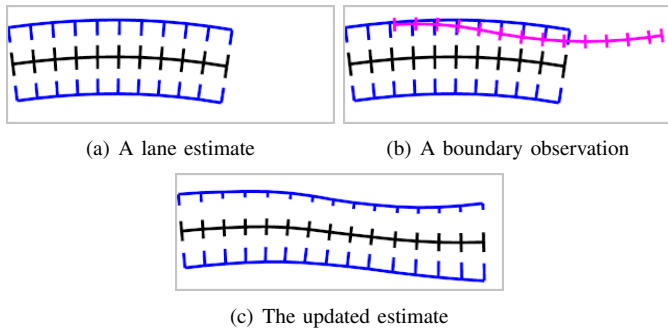(b) A boundary observation

(c) The updated estimate

Fig. 5. A boundary observation is used to update the lane estimate. The middle (black) curve marks the lane centerline, and the outer (blue) curves mark the left and right boundary marginal distributions. Short line segments along the curves mark control points, and the length of these segments indicate 1-σ uncertainty. Note that both boundaries are updated even though only one boundary is observed.

C. Initial estimate

We initialize a lane estimate by independently estimating many zero-width curves as described in Sec. V, while periodically searching for curve pairs that are sufficiently long, parallel, and separated by an appropriate distance. Once a suitable pair of boundary curves is identified, they are used to initialize a lane estimate. The initial lane basis curve $\mathbf{b}$ is chosen by projecting one boundary curve onto the other and scaling the offset vector by 0.5 (an approximation of the medial axis), and both curve estimates are then reparameterized with $\mathbf{b}$. Referring to the left and right curve estimates as $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}_l, \Sigma_l)$ and $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}_r, \Sigma_r)$, we treat them as independent observations of the same lane, and express them jointly as:

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\mu}_l \\ \boldsymbol{\mu}_r \end{bmatrix} = \begin{bmatrix} A_l \\ A_r \end{bmatrix} \mathbf{f}^B + \mathbf{v} = A_z \mathbf{f}^B + \mathbf{v} \qquad (19)$$

where A_l and A_r are the transformation matrices relating a lane to its left and right boundary observations (Sec. VI-B), $\mathbf{f}^B$ is the projection of the unobserved true lane onto $\mathbf{b}$, and $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Sigma_z)$ is a noise term described by:

$$\Sigma_z = \begin{bmatrix} \Sigma_l & 0 \\ 0 & \Sigma_r \end{bmatrix} \qquad (20)$$

Using the information filter [14], we can see that the initial distribution parameters best representing the information provided by the boundary curves can be expressed as:

$$\begin{aligned} \Sigma_0 &= (A_z^\top \Sigma_z^{-1} A_z)^{-1} \\ \boldsymbol{\mu}_0 &= \Sigma A_z^\top \Sigma_z^{-1} \mathbf{z} \end{aligned} \qquad (21)$$

VII. EXPERIMENTS

To quantitatively assess the performance of our system, we evaluated it against ground truth across two datasets containing data from a forward-facing wide-angle camera (Point Grey Firefly MV, 752x480 @22.8 Hz), and a Velodyne HDL-64E laser range scanner. As input to our system, we used vision- and LIDAR-based road paint and curb detection algorithms described in previous work [7].

The first dataset consists of 30.2 km of travel in 182 minutes, and can be characterized by wide suburban lanes, no pedestrians, generally well-marked roads, few vehicles, and a bright early morning sun. The vehicle also traverses a short 0.4 km dirt road and a 1.7 km stretch of highway. The second dataset consists of 13.6 km of travel in 58 minutes through a densely populated city during afternoon rush hour. This dataset can be characterized by roads of varying quality, large numbers of parked and moving vehicles, and many pedestrians.

To produce ground truth, we annotated high-resolution geo-registered ortho-rectified aerial imagery with lane geometry. The vehicle's GPS estimates during a data collection provide an initial guess as to the vehicle's pose; these were corrected by manually aligning sensor data (i.e., camera and LIDAR data) with the aerial imagery at various points in the data collection. The result is a dataset containing ground truth lane geometry relative to the vehicle at every moment of travel. We emphasize that our algorithm uses only local sensor data – GPS and the ground truth map were used only for evaluation purposes.

We compare the results of our algorithm, which we refer to as the basis curve (BasCurv) algorithm, with our previous work in the DARPA Urban Challenge [7], which we refer to as the evidence image (EvImg) algorithm. The evidence image algorithm can be used as a standalone lane estimation system by using the output of the first of its two stages, which performs lane detection from sensor data only. Both algorithms use the same features as input.

For computational speed, our implementation of the basis curve algorithm used diagonal covariance matrices when estimating lane boundaries, and block-diagonal covariance matrices (2×2 blocks) for lane estimation. This introduces additional approximation errors, but yielded good performance in our experiments. After each observation update, basis curves are re-sampled to maintain a uniform (1 m) control point spacing. Parameters such as covariances and data association thresholds were determined experimentally. The basis curve algorithm was implemented in Java and runs at real-time speeds.

A. Centerline error

The centerline error of a lane estimate at a given point on the estimate is defined as the shortest distance from the estimated lane centerline point to the true centerline of the nearest lane. Fig. 6 shows the 50th and 90th percentile values for the centerline error of the two algorithms as a function of distance from the vehicle. The basis curve algorithm has significantly lower error at all distances.

Fig. 7 shows the centerline error as a function of true lane width. The evidence image algorithm assumes a fixed lane width of 3.66 m, and its performance degrades as the true lane width departs from this assumed value. Since the basis curve algorithm jointly estimates lane width and centerline geometry, it remains relatively invariant to changes in lane width.
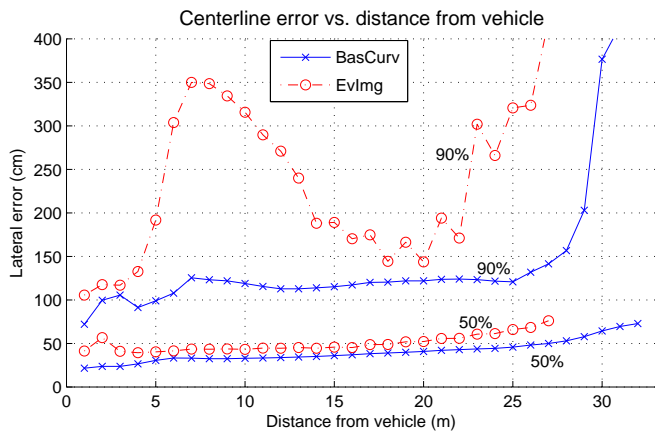
Fig. 6. The 50 and 90 percentile values for centerline lateral error, as a function of increasing distance from the vehicle.
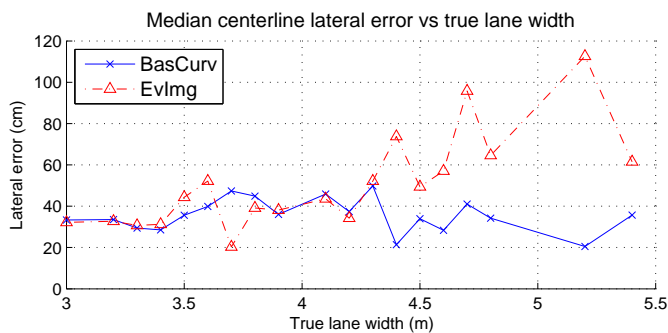

Fig. 7. Median centerline lateral error as a function of true lane width.

B. Lookahead distance and time

The *lookahead distance* and *lookahead time* metrics measure how much farther the vehicle can travel before reaching the end of its current lane estimate, assuming constant speed. Lookahead distance is computed by measuring the distance from the vehicle to the farthest point ahead on the current lane estimate, and lookahead time is computed by dividing the lookahead distance by the vehicle's instantaneous speed.

Fig. 8 aggregates lookahead statistics over both datasets, and shows the lookahead distance and lookahead time cumulative distributions for the two algorithms. In all cases, the basis curve algorithm outperforms the evidence image algorithm. For example, the basis curve algorithm provided some lane estimate forward of the vehicle for 71% of distance traveled, compared to 36% for the evidence image algorithm.

C. Qualitative results

Fig. 9 shows the output of the basis curve lane estimation algorithm in a variety of challenging scenarios. In (a) and (b), tree shadows and an erroneous curb detection are detected and rejected as outliers, leaving the lane estimates intact. In (b), the median strip separating opposite lanes of traffic is correctly excluded from the lane estimates. In (c) and (d), correctly detected road paint is successfully excluded from
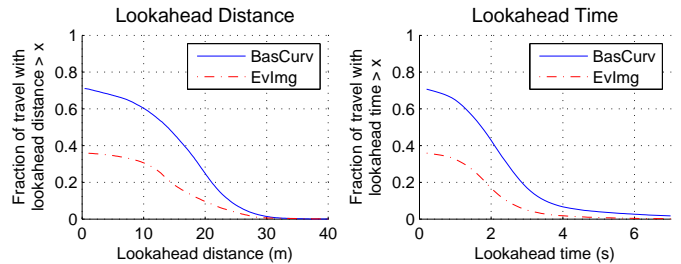

Fig. 8. Lane estimate lookahead distance (left) and time (right) CDFs.

the lane estimates. In (e-g), our method performs well despite adverse lighting scenarios and a snow-covered environment.

VIII. DISCUSSION

Overall, the basis curve algorithm provides lane estimates of equal or better accuracy to those produced by the evidence image approach, and does so more often and with a greater lookahead. We attribute this to the data association and outlier rejection properties of the basis curve algorithm, and to the joint estimation of lane width and centerline geometry. The evidence image algorithm performs no outlier rejection, attempting to fit lanes to both true lane boundary detections and false detections such as tree shadows and non-boundary road paint; nor does it estimate lane width.

We have formulated the lane estimation problem in such a way that standard estimation and tracking algorithms based on the Kalman filter can be used for complex lane geometries. In doing so, we gain the advantages of the Kalman filter, and also invite all of its shortcomings. Cases where outliers appear very similar to inliers, such as long shadows nearly parallel to the road, can cause the lane estimates to diverge. Another failure mode arises when one road marking appears initially to be the correct boundary, but the true lane boundary then comes into view. In this case, our method will converge upon the first marking as the lane boundary, since the unimodal nature of the Kalman filter will prevent it from assigning substantial weight to the true boundary.

These difficulties are similar to those studied in other estimation domains, and it should also be possible to apply lessons learned in those domains to lane estimation with basis curves. One approach is particle filtering, which has been successfully applied in many estimation and tracking problems to model complex distributions and enable multi-hypothesis tracking. However, the high dimensionality of the lane estimates would require careful treatment.

Finally, in using a Gaussian noise model, we are simplifying and approximating the true system errors. This has proved successful in practice, although more careful study is required to understand the extent to which our simplifications and approximations are valid, and when other models may be more appropriate.

IX. CONCLUSION

This paper introduced the notion of basis curves for curve estimation, and described an application to the lane estimation
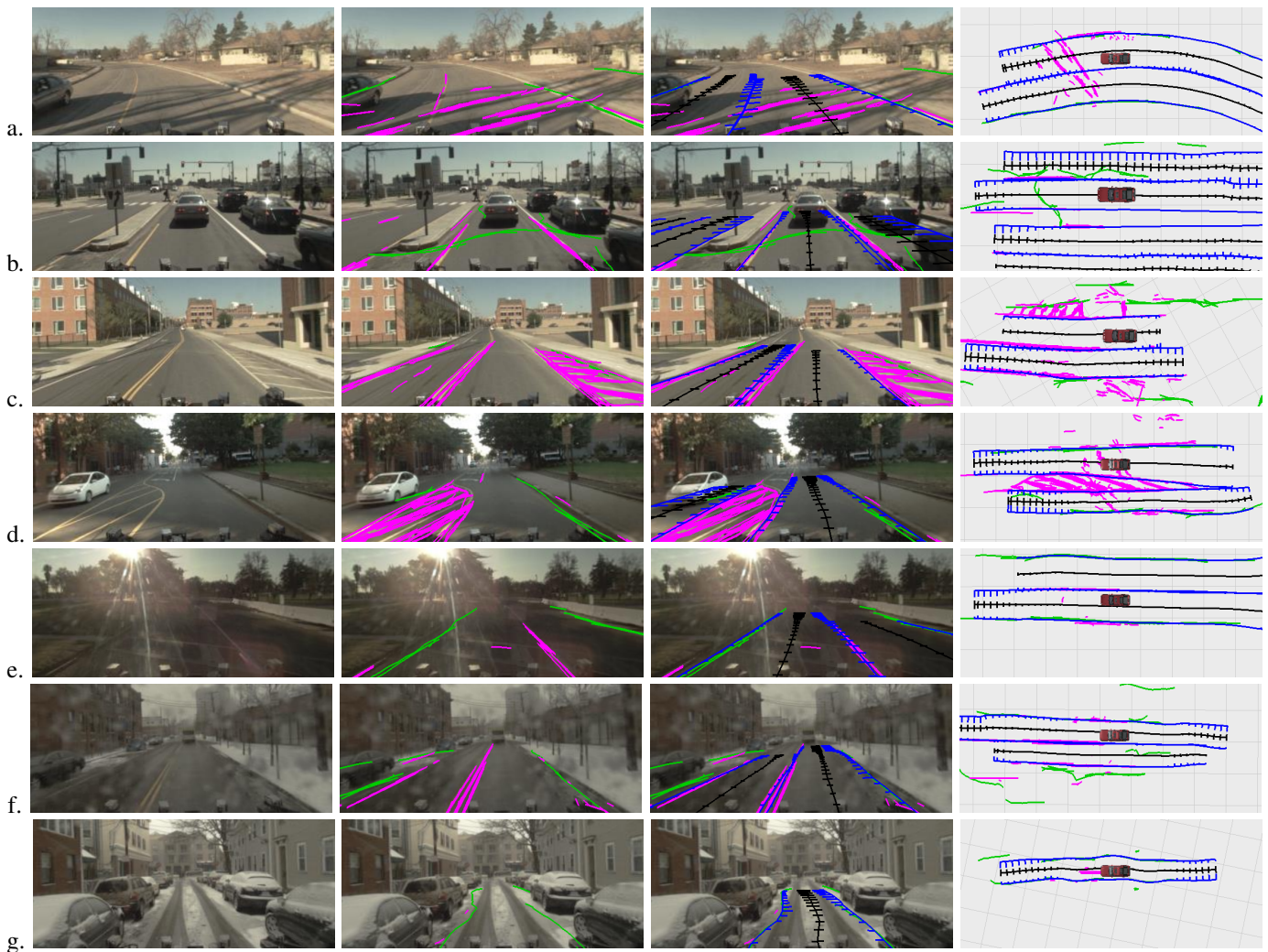
Fig. 9. Lane estimation in a variety of environments. *Column 1:* Camera images. *Column 2:* Detections of road paint (magenta) and curbs (green). *Column 3:* Lane centerline estimates (black) and boundary curve estimates (blue) projected into the image. *Column 4:* Synthesized overhead view of lane estimates.

problem. A detailed evaluation of our method's performance on a real-world dataset, and a quantitative comparison against ground truth and a previous approach, shows distinct advantages of the basis curve algorithm, particularly for estimating lanes using partial observations, for handling noisy data with high false-positive rates, and for jointly estimating centerline geometry and lane width.

REFERENCES

[1] N. Apostoloff and A. Zelinsky. Vision in and out of vehicles: Integrated driver and road scene monitoring. *Int. Journal of Robotics Research*, 23(4-5):513–538, Apr. 2004.

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

[3] M. Bertozzi and A. Broggi. GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Transactions on Image Processing*, 7(1):62–80, Jan. 1998.

[4] M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 1:1–16, 2000.

[5] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[6] E. Dickmanns and B. Mysliwetz. Recursive 3-D road and ego-state recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):199–213, Feb. 1992.

[7] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller. Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, 26(2-3):103–122, Apr. 2009.

[8] A. S. Huang and S. Teller. Lane boundary and curb estimation with lateral uncertainties. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, St. Louis, Missouri, Oct. 2009.

[9] Z. Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intelligent Transportation Systems*, 9(1):16–26, Mar. 2008.

[10] Y. Matsushita and J. Miura. On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter. In *Proc. European Conference on Mobile Robots*, Sep. 2009.

[11] 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*, 7(1):20– 37, Mar. 2006.

[12] J. Neira and J. D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics and Automation*, 17(6):890–897, Dec 2001.

[13] 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*, 10(3):362–373, May 1988.

[14] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[15] Y. Wang, E. K. Teoh, and D. Shen. Lane detection and tracking using B-Snake. *Image and Vision Computing*, 22(4):269 – 280, 2004.