Autonomous Robots manuscript No.
(will be inserted by the editor)

Probabilistic lane estimation for autonomous driving using basis curves

Albert S. Huang ·
Seth Teller

Received: date / Accepted: date

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 forming lane boundaries. The number of lanes to estimate are initially unknown and many observations may be outliers or false detections (due e.g. to shadows or non-boundary road paint). 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 multi-lane geometries and makes no assumptions about the position and orientation of the vehicle with respect to the roadway.

We evaluate our algorithm using 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.

Keywords Lane estimation · Lane tracking · Autonomous vehicles · Intelligent vehicles

This research was sponsored by DARPA, Program: Urban Challenge, ARPA Order No. W369/00, Program Code: DIRO, issued by DARPA/CMO under Contract No. HR0011- 06-C-0149.

A. S. Huang · S. Teller
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA USA

E-mail: ashuang@alum.mit.edu, teller@csail.mit.edu

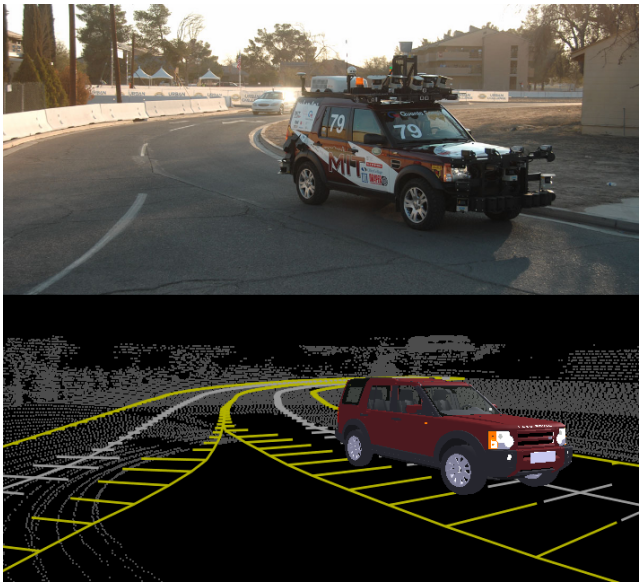


Fig. 1 (top) MIT's autonomous vehicle. (bottom) Lane estimates; uncertainty indicated by short perpendicular lines.

1 Introduction

The past decade has seen enormous progress in the development of autonomous land vehicles. From the first DARPA Grand Challenge in 2004, where vehicles struggled to drive more than a few miles autonomously, vehicles today have been tested in continuous operation for hundreds of miles in complex environments.

Of fundamental importance to autonomous driving is an accurate model of the roadway and its lanes as the vehicle moves. A system able to automatically and reliably construct such a model (Figure 1) using on-board sensors such as cameras or laser range scanners (LIDAR) would provide enormous benefits to an autonomous vehicle. A lane estimator could also be used for tasks ranging from wide-scale

road and lane quality assessment, to providing inputs to a driver assistance safety or navigation system.

In some cases, a model of the roadway and nearby lanes could be obtained by localizing within a previously constructed map using a global positioning system (GPS) receiver and/or local environmental cues (Urmson et al., 2009). However, this approach fails when the map becomes obsolete – e.g., when lanes are reconfigured by construction or resurfacing, or the road itself is significantly displaced by an earthquake (Bates, 2010; Voigt, 2011) – or when the localization process itself fails. In these situations, estimates constructed from vehicle sensor data could be used to correct the map, improve localization, or simply guide the vehicle without a map.

As with many robot perception tasks in real-world settings, accurately detecting and estimating lanes from local sensor data is a complex problem. Roadways and the conditions under which they are observed can range from well-marked surfaces on clear and sunny days, to poorly marked lanes obscured by snow, rain, shadow, strong sunlight, and a host of other factors. Algorithms designed for one scenario may not work well in others; the ability to robustly handle new and unfamiliar situations is critically important. Faded road paint lines still visible after re-painting must be distinguished from fresh road paint, and a functional system must accurately model the width, curvature, topology and other aspects of the road network required for the task at hand.

In previous work (Huang and Teller, 2009), 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 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.[1]

[1] Some of the methods and results of this paper were described in an earlier form at RSS 2010 (Huang and Teller, 2010).

2 Related Work

Aspects of the lane estimation problem have been studied for decades in the context of autonomous land vehicle development (Dickmanns and Mysliwetz, 1992; Thorpe et al., 1988) and driver-assistance (Bertozzi and Broggi, 1998; Bertozzi et al., 2000; Apostoloff and Zelinsky, 2004; Kim, 2008). Most lane estimation systems have been developed for lane keeping and lane departure warning (LDW) in highway environments (McCall and Trivedi, 2006). 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 (especially in poor observability conditions), 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 much 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.

Recent work on lane estimation has focused on operation in more varied and complex environments. Wang et al. (2004) used B-Splines to model roads with significant curvature, while still retaining the single lane and other LDW assumptions. Matsushita and Miura (2009) use a particle filter to detect certain types of intersections, 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 (Huang et al., 2009). The primary challenge in all of these approaches lies in determining which (noisy) sensor observations correspond to lane markings.

Crucial to the effective evaluation of any estimation algorithm is an objective way to assess its performance. However, existing evaluations of lane estimation systems have ranged from simply presenting lane estimates superimposed onto camera images (Sehestedt et al., 2007; Southall and Taylor, 2001), to providing summary statistics on closed-loop experiments (Bertozzi and Broggi, 1998; Pomerleau and Jochem, 1996), to rough analysis of short data sequences (Kim, 2008; Wang et al., 2004). In the limited cases where comparisons are made against a set of ground truth images, there are typically only a few hundred labeled images (McCall and Trivedi, 2006). In contrast, we provide a detailed

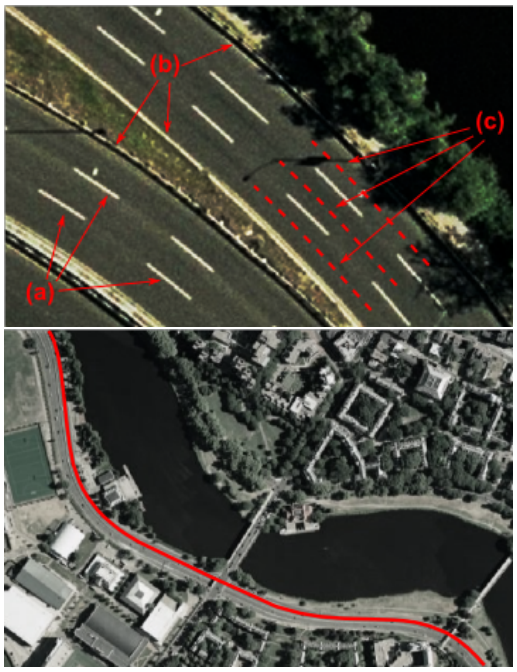


Fig. 2 Top: Curves of interest in the context of lane estimation include painted lane boundaries (a), raised curbs (b), and (typically invisible) lane centerlines (c). Bottom: The spatial extent of curves can be much larger than a vehicle's sensor range; in many cases estimates of the entire curve are not necessary.

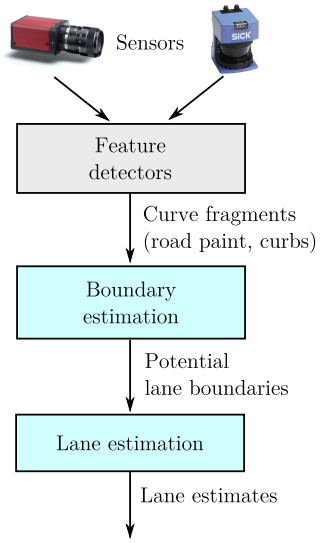


Fig. 3 System diagram. The focus of this paper is on using curve fragments to estimate potential lane boundaries (boundary estimation), and then combine them into lane estimates (lane estimation). The curve fragments can come from different feature detectors, such as road paint detectors or curb detectors, and need only be observations of potential lane boundaries..

analysis of our system on a ground truth dataset containing precise lane geometry for over 44 km of vehicle travel under a wide variety of environmental conditions.

3 Method Overview

The geometry of a travel lane can be described by a single centerline curve $\mathbf{f}(t) : [t_1, t_n] \rightarrow \mathbb{R}^3$ whose width varies along the length of the curve:

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

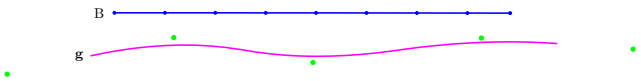
where $f_x(t)$ and $f_y(t)$ describe the lane centerline, and $f_h(t)$ describes the lane half-width, all of which are defined on the domain $t \in [t_1, t_n]$. The interpretation of t depends on the curve model chosen. We describe the curve in a coordinate system fixed with respect to the local environment, such that lane geometry is nearly invariant to time and the vehicle's own state.

In addition to the lane centerline, several other objects of interest can also be represented as parametric curves. Painted lane boundaries, physical road edges such as curbs, and the invisible road centerline can all be expressed as curves (Figure 2). Curves can be on the order of meters long for short streets or merge lanes, or hundreds of meters long for stretches of highway.

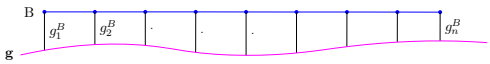
Initially, the number of lanes and their geometries are unknown. The goal of our system is to detect all nearby lanes, estimate their geometries, and update these estimates as the vehicle moves. The system takes as input curve fragments corresponding to potential lane boundary observations, such as those detected by a vision-based road-paint detection algorithm, or a LIDAR-based curb-detection algorithm. While many curve fragments are true observations of the road and lanes, some may arise from 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). Our system uses the curve fragments to estimate potential lane boundaries, which are then combined into lane estimates (Figure 3).

We approximate $\mathbf{f}(t)$ as piecewise-linear, describing it 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*.

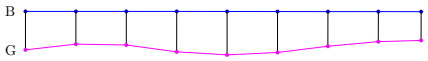
We chose this representation because nearby curves often share common structure. Factoring out this structure leaves a set of residuals which are computationally convenient to manipulate. Within the space of these residuals, which we call the basis curve space, lane boundary observations can be expressed 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 (Bar-Shalom and Li, 2001).



(a) A basis curve $\mathbf{b}$ represented by polyline B (blue), 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. 4 A uniform quadratic B-spline (magenta) defined by control points (green) can be projected onto a basis curve (blue).

4 Basis Curves

Consider a piecewise-linear curve $\mathbf{b}(t) : [t_1, t_n] \rightarrow \mathbb{R}^2$. As it is piecewise-linear (polyline), $\mathbf{b}$ can be represented by the matrix of control points $\mathrm{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)^\top$, where each control point $\mathbf{b}_i$ is a 2-D point: $\mathbf{b}_i \in \mathbb{R}^2, i \in \{1 \ldots n\}$. Denote the unit normal vectors to the curve at each of the control points[2] by the matrix $\bar{\mathrm{B}} = (\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \ldots, \bar{\mathbf{b}}_n)^\top$. We consider another piecewise-linear curve $\mathbf{g} : [t_1, t_n] \rightarrow \mathbb{R}^2$, represented as $\mathrm{G} = (\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n)^\top$, to be a *variation* of $\mathbf{b}$ if $\mathbf{g}_i, i \in \{1 \ldots n\}$, can be expressed as:

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

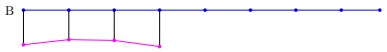
where each g_i^{B} is a signed scalar that indicates how much $\mathbf{g}$ varies from $\mathbf{b}$ along the normal vector at a particular control point. In this context, $\mathbf{b}$ is a *basis curve*, and the vector $\mathbf{g}^{\mathrm{B}} = (g_1^{\mathrm{B}}, g_2^{\mathrm{B}}, \ldots, g_n^{\mathrm{B}})^\top$ 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 not piecewise-linear, or is 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* $\mathbf{b}$ (Figure 4).

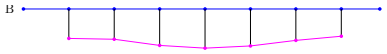
The error of this approximation (e.g., as measured by the area between the polyline approximation and the original curve) is small if $\mathbf{g}$ and $\mathbf{b}$ have similar geometry, and if the control point spacing of $\mathbf{b}$ is small relative to the curvatures of the two curves. If the curves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

We can now use $\mathbf{b}$ to represent common structure shared by many curves, and use offset vectors to reason about differences among curves. In this model, $\mathbf{b}$ 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.

[2] 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) Variation of the first four control points of a basis curve.



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

Fig. 5 Variations (magenta) of a subset of the basis curve (blue) can be described using subspaces of the vector space of the full basis curve.

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 $\mathbf{b}$ (Figure 5).

To review, thus far we have defined:

$\mathbf{b}$	A basis curve.
B	The $n \times 2$ control point matrix 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 $\mathbf{b}$.

4.1 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 $\mathbf{g}^{\mathrm{B}} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Figure 6 shows a basis curve and several hundred variations, where the offset vectors of each variation are drawn from a multivariate Gaussian.

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 $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$.

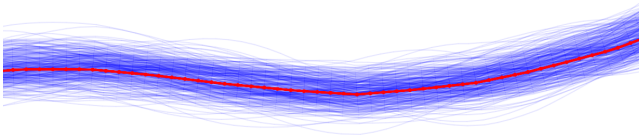


Fig. 6 Normally distributed variations (blue) of a basis curve (red).

The probability density of a curve $\mathbf{g}$ is then defined as:

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

where $\mathcal{N}\big(\mathbf{x}; \boldsymbol{\mu}, \Sigma\big)$ 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 orthogonally to 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. Figure 7 shows samples drawn from basis curve normal distributions with different covariance matrices.

If a distribution is instead defined on the entire control point space such that each control point has 2 degrees of freedom (Blake and Isard, 1998), it is possible to draw two samples with the same curve 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.

4.2 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 everywhere nearly 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.

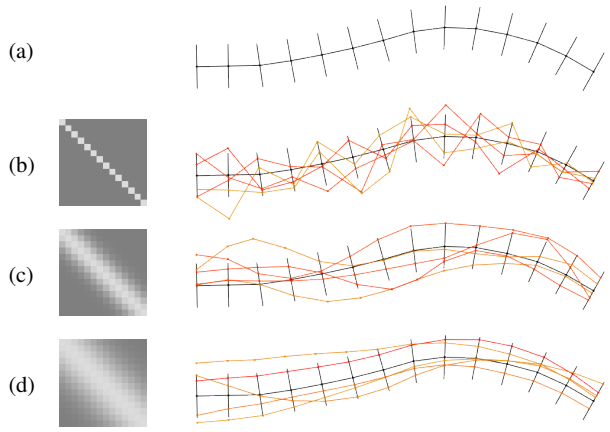


Fig. 7 (a) A basis curve, with normal vectors shown at each control point. (b-d) Three covariance matrices, and five samples drawn from the distributions defined by the basis curve, $\boldsymbol{\mu} = \mathbf{0}$, and each of the covariances. Lighter colors indicate higher correlation.

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:

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

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 $k > j$ (i.e., 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:

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

5 Lane boundary estimation

To detect and estimate lanes, our method first estimates potential lane boundaries, then groups boundary curves to form lane estimates. Here, we develop a recursive algorithm that estimates an unknown curve $\mathbf{f}(t) : [t_1, t_n] \rightarrow \mathbb{R}^2$, 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 $\mathbf{f}^{\mathrm{B}} = (f_1^{\mathrm{B}}, f_2^{\mathrm{B}}, \ldots, f_n^{\mathrm{B}})^\top$ 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 $\mathbf{f}^{\mathrm{B}}$. In this sense, there are two estimates: the estimate $\boldsymbol{\mu}$ of $\mathbf{f}^{\mathrm{B}}$, 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).

5.1 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 $\mathrm{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)^\top$, then each $\mathbf{z}_i$ can be expressed as:

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

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 $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathrm{R})$. The projection $\mathbf{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, z_2^{\mathrm{B}}, \ldots, z_n^{\mathrm{B}})^\top$ of $\mathbf{z}$ onto $\mathbf{b}$ can then be described as:

$$z_i^{\mathrm{B}} = f_i^{\mathrm{B}} + v_i \tag{10}$$

and:

$$\mathbf{z}^{\mathrm{B}} = \mathrm{A}\mathbf{f}^{\mathrm{B}} + \mathbf{v} \tag{11}$$

where the observation matrix $\mathrm{A} = \mathrm{I}_{n\times n}$ 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 submatrix and is zero elsewhere. For example, if the basis curve has three control points and the observation curve spans only the first two, then the corresponding observation matrix is:

$$\mathrm{A}_{2\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{12}$$

5.2 Data Association

Not all observations correspond to the desired curve $\mathbf{f}$. To avoid corrupting the curve estimate, only true observations should be used to update the curve estimate. Determining whether an observation corresponds to a tracked curve is the well-known data association problem (Thrun et al., 2005).

We use a standard χ^2 outlier rejection approach. Define the scalar random variable y as the following Mahalanobis distance (Bar-Shalom and Li, 2001):

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

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 (Neira and Tardos, 2001), 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.

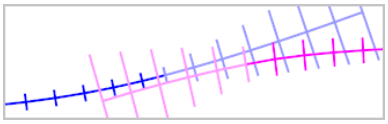
5.3 Curvature Prediction

As the vehicle travels, it may make observations that extend beyond the end of its current curve estimate. To incorporate such data, the observation and curve estimate are first extended (Figure 8b). We use a method described in previous work (Huang and Teller, 2009), summarized here.

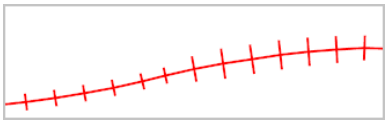
Once we have observed and estimated one portion of a curve, we can reliably predict the shape of the nearby unobserved parts of the curve. To do so, we fit a first-order Markov model of road curvature to a dataset containing the manually annotated coarse geometry of existing public roadways (Commonwealth of Massachusetts Office of Geographic and Environmental Information, 2008). 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.

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.

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



(b) Predicted extensions of both curves



(c) Final curve estimate

Fig. 8 A curvature prediction model allows us to associate observations with existing curve estimates when there is little or no overlap. The short perpendicular line segments indicate one-σ uncertainty.

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 $\tilde{\mathcal{N}}(\mathbf{b}^{+}, \boldsymbol{\mu}^{+}, \Sigma^{+})$

Augment the initial estimate and observation as necessary via curve prediction
Let m be the dimensionality of $\mathbf{z}^{\mathrm{B}}$
Let k be the 0.95 percentile of the χ^2_m distribution
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**
  Reject observation as an outlier
  return $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}, \Sigma)$
$\mathrm{K} \leftarrow \Sigma\mathrm{A}^{\top}(\mathrm{A}\Sigma\mathrm{A}^{\top} + \mathrm{R})^{-1}$
$\tilde{\boldsymbol{\mu}} \leftarrow \boldsymbol{\mu} + \mathrm{K}(\mathbf{z}^{\mathrm{B}} - \mathrm{A}\boldsymbol{\mu})$
$\tilde{\Sigma} \leftarrow (\mathrm{I} - \mathrm{K}\mathrm{A})\Sigma$
$\mathbf{b}^{+} \leftarrow \mathrm{B} + \mathrm{diag}(\tilde{\boldsymbol{\mu}})\bar{\mathrm{B}}$
$\boldsymbol{\mu}^{+} \leftarrow \mathbf{0}$
$\Sigma^{+} \leftarrow \tilde{\Sigma}$
return $\tilde{\mathcal{N}}(\mathbf{b}^{+}, \boldsymbol{\mu}^{+}, \Sigma^{+})$

5.4 Update

Once an observation $\mathbf{z}$ has been associated with an existing curve estimate, the algorithm revises that estimate using a standard Kalman filter (Bar-Shalom and Li, 2001).

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. 4.2). 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 coordinate 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 $\tilde{\mathcal{N}}(\mathbf{b}^{+}, \boldsymbol{\mu}^{+}, \Sigma^{+})$. Figure 8 illustrates an update step for a single curve estimate.

5.5 Relationship to Gaussian Processes

One way to view a basis curve normal distribution is as a specialized Gaussian process (GP) in the basis curve space. Gaussian processes provide a way to specify distributions over functions such that any finite sample of points are jointly Gaussian (Rasmussen and Williams, 2006). Our lane boundary estimation algorithm is grounded in traditional Kalman filtering techniques extended to distributions over curves, but could also be motivated from a machine learning perspective using Gaussian processes. The key issues would be in maintaining a sufficiently low number of samples to preserve computational performance, and in choosing the GP covariance function.

6 Combining boundaries into lanes

The boundary curves of a single lane are highly correlated; information about one boundary gives valuable information about the other. We would like to leverage this correlation in a lane estimation algorithm that does more than independently estimate multiple lane boundaries. To do this, we represent a lane as a piecewise linear centerline curve $\mathbf{s}(t) : [t_1, t_n] \rightarrow \mathbb{R}^3$ whose width varies with t. We represent $\mathbf{s}$ with the matrix $\mathrm{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n)^{\top}$, where each control point $\mathbf{s}_i$ is defined as $\mathbf{s}_i = (s_{xi}, s_{yi}, s_{hi})^{\top}$. The s_{xi} and s_{yi} terms describe the centerline geometry of $\mathbf{s}$, and s_{hi} describes the half-width of $\mathbf{s}$.

Using the convention that the normal vectors of a curve point "left", two points $\mathbf{s}_{li} \in \mathbb{R}^2$ and $\mathbf{s}_{ri} \in \mathbb{R}^2$ on the left and right boundaries, respectively, can be described as:

$$\mathbf{s}_{li} = \begin{bmatrix} s_{xi} + s_{hi}\bar{s}_{xi} \\ s_{yi} + s_{hi}\bar{s}_{yi} \end{bmatrix} \qquad \mathbf{s}_{ri} = \begin{bmatrix} s_{xi} - s_{hi}\bar{s}_{xi} \\ s_{yi} - s_{hi}\bar{s}_{yi} \end{bmatrix} \tag{14}$$

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

6.1 Lane distributions

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

$$\mathbf{s}^{\mathrm{B}} = (s^{\mathrm{B}}_{c1}, s_{h1}, s^{\mathrm{B}}_{c2}, s_{h2}, \ldots, s^{\mathrm{B}}_{cn}, s_{hn})^{\top} \tag{15}$$

where $s^{\mathrm{B}}_{c1}, \ldots, s^{\mathrm{B}}_{cn}$ is the projection of the centerline of $\mathbf{s}$ 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 $\mathbf{s}$ 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 $\mathbf{s}$. The mean estimate can be represented by a matrix of control points $\hat{\mathrm{S}} = (\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \ldots, \hat{\mathbf{s}}_n)^{\top}$, where each control point $\hat{\mathbf{s}}_i = (\hat{s}_{xi}, \hat{s}_{yi}, \hat{s}_{hi})^{\top}$ can be expressed as:

$$\hat{\mathbf{s}}_i = \begin{bmatrix} \hat{s}_{xi} \\ \hat{s}_{yi} \\ \hat{s}_{hi} \end{bmatrix} = \begin{bmatrix} b_{xi} + \mu_{ci}\bar{b}_{xi} \\ b_{yi} + \mu_{ci}\bar{b}_{yi} \\ \mu_{hi} \end{bmatrix} \tag{16}$$

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. 4.2, 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.

6.2 Observation model, data association, and update

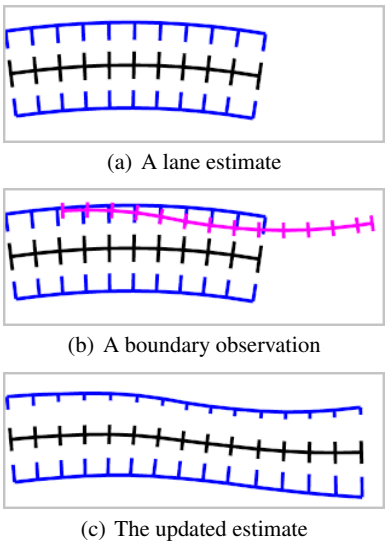
A full boundary observation of $\mathbf{s}$ is a curve, which we describe with the matrix of control points $\mathrm{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)^{\top}$, where each $\mathbf{z}_i$ can be written:

$$\begin{aligned} \mathbf{z}_i &= \mathbf{b}_i + (s^{\mathrm{B}}_{ci} + a s_{hi} + v_i)\bar{\mathbf{b}}_i \\ &= \mathbf{b}_i + z^{\mathrm{B}}_i \bar{\mathbf{b}}_i \end{aligned} \tag{17}$$

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 $\mathbf{z}^{\mathrm{B}} = (z^{\mathrm{B}}_1, z^{\mathrm{B}}_2, \ldots, z^{\mathrm{B}}_n)^{\top}$ can then be expressed as:

$$\mathbf{z}^{\mathrm{B}} = \mathrm{A}\mathbf{s}^{\mathrm{B}} + \mathbf{v} \tag{18}$$



(a) A lane estimate

(b) A boundary observation

(c) The updated estimate

Fig. 9 A boundary observation is used to update the lane estimate. The black curve marks the lane centerline, and the 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.

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 $\mathbf{s}$. 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. Figure 9 shows a full update cycle, where an observation of a lane boundary is used both to update and extend the lane. Figure 10 shows a real-world example where the vehicle is able to use observations of a single boundary to maintain an accurate estimate of an adjacent lane that is mostly occluded by vehicles.

6.3 Initial estimate

We initialize a lane estimate by independently estimating many zero-width curves as described in Sec. 5, while peri-

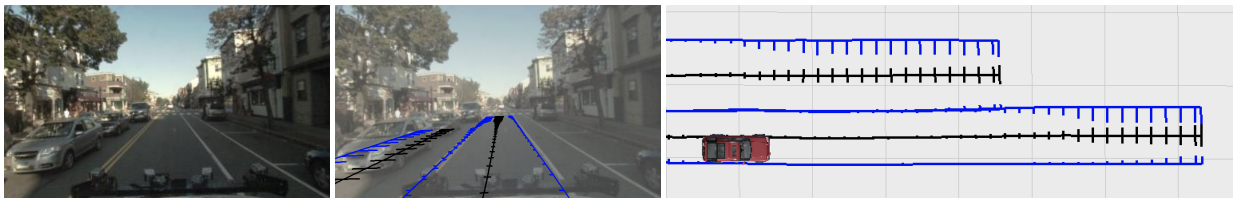


Fig. 10 (left) The farthest boundary of the opposite lane is occluded by other vehicles. However, by maintaining a joint distribution over centerline geometry and lane width, (middle) the system can still maintain an estimate of the opposite lane, using observations of the closer boundary. (right) A synthesized overhead view of the vehicle and its lane estimates. Black curves correspond to the estimated lane centerlines, and blue curves denote the marginal distributions of the lane boundaries. Short line segments extending laterally from the curves indicate control point uncertainty.

odically 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); both curve estimates are then reparameterized with $\mathbf{b}$. Referring to the left and right curve estimates as $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}_l, \Sigma_\mathrm{l})$ and $\tilde{\mathcal{N}}(\mathbf{b}, \boldsymbol{\mu}_r, \Sigma_\mathrm{r})$, we treat them as independent observations of a single lane, and express them jointly as:

$$\mathbf{z} = \begin{bmatrix} \boldsymbol{\mu}_l \\ \boldsymbol{\mu}_r \end{bmatrix} = \begin{bmatrix} \mathrm{A}_l \\ \mathrm{A}_r \end{bmatrix} \mathbf{s}^\mathrm{B} + \mathbf{v} = \mathrm{A}_z \mathbf{s}^\mathrm{B} + \mathbf{v} \tag{19}$$

where A_l and A_r are the transformation matrices relating a lane to its left and right boundary observations (Sec. 6.2), $\mathbf{s}^\mathrm{B}$ is the projection of the unobserved true lane onto $\mathbf{b}$, and $\mathbf{v} \sim \mathcal{N}\big(\mathbf{0}, \Sigma_z\big)$ is a noise term described by:

$$\Sigma_z = \begin{bmatrix} \Sigma_l & 0 \\ 0 & \Sigma_r \end{bmatrix} \tag{20}$$

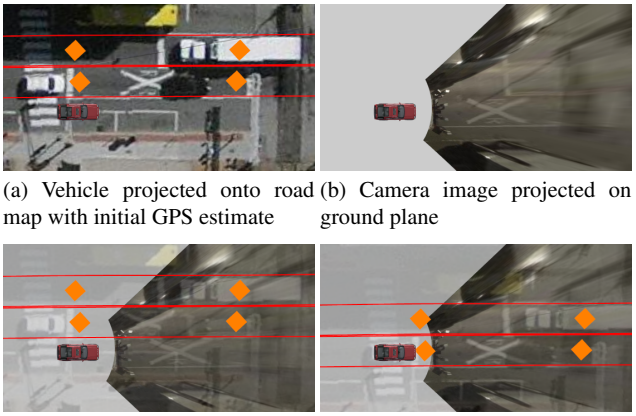
Using the information filter (Thrun et al., 2005), we can see that the initial distribution parameters best representing the information provided by the boundary curves can be expressed as:

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

7 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 (Huang et al., 2009).

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) Vehicle projected onto road map with initial GPS estimate (b) Camera image projected on ground plane

(c) Image overlaid on road map (d) Corrected GPS estimate

Fig. 11 To produce ground truth lane geometry, we created detailed road maps from aerial imagery, and localized the vehicle within these road maps by manually correcting the initial GPS estimates. (a) According to the initial GPS estimate, the vehicle is straddling a parking spot. (b) The camera image shows the vehicle clearly in the middle of a lane. (c) Overlaying the two images shows a clear offset in the GPS estimate. (d) Aligning the images corrects the GPS estimate.

a bright early morning sun. The vehicle also traversed a short 0.4 km dirt road and a 1.7 km stretch of highway. This dataset was collected on November 3, 2007 at the DARPA Urban Challenge in Victorville, CA. The second dataset, collected on September 2, 2009 in Cambridge, MA, 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. GPS estimates during 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 (Figure 11). 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.

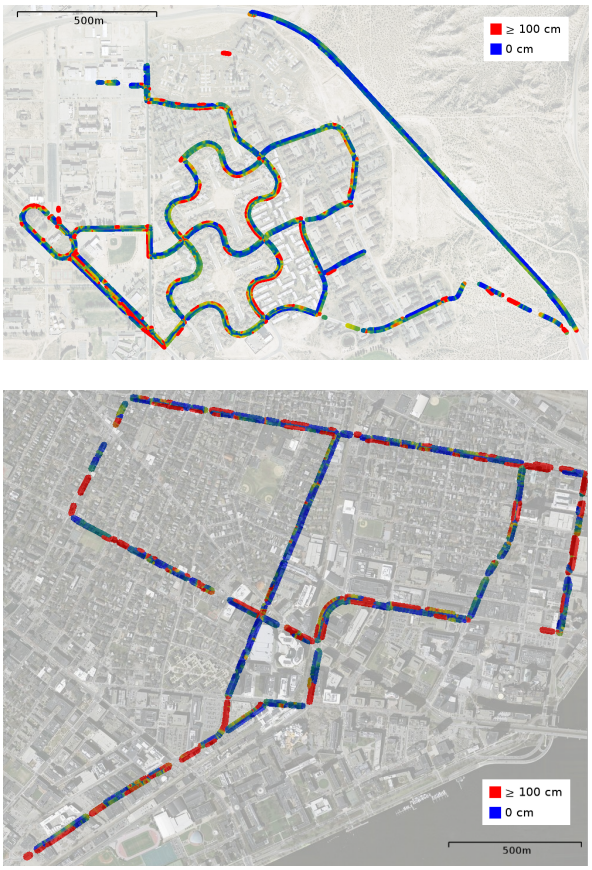


Fig. 12 Mean lane centerline error for the (top) Urban Challenge and (bottom) Cambridge datasets for the basis curve algorithm. Mean centerline error is indicated by color – areas with no observable error are shown in blue, and areas where the mean error reached or exceeded 100 cm shown in red.

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 (Huang et al., 2009), 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 algorithm was implemented in Java and handles sensor data in real time.

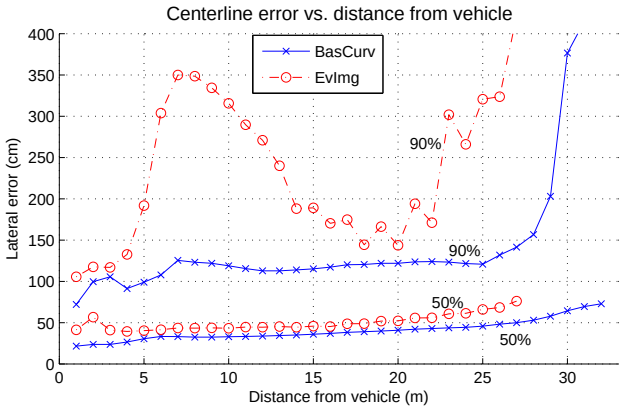


Fig. 13 The 50 and 90 percentile values for centerline lateral error, as a function of increasing distance from the vehicle. The 50% values indicate the median centerline error, and 90% of an algorithm's lane centerline estimates at a given distance from the vehicle have an error less than or equal to the 90% curve.

7.1 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. 12 shows the basis curve algorithm's mean centerline error as a function of vehicle position for the two datasets. Fig. 13 shows the 50th and 90th percentile values for the centerline error of the two algorithms as a function of distance from the vehicle, which indicate the median error and its 90% bounds. The basis curve algorithm has significantly lower error at all distances, and significantly lower variation.

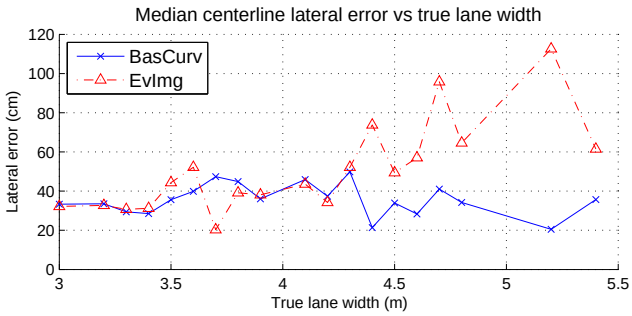


Fig. 14 Median centerline lateral error as a function of true lane width.

Fig. 14 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.

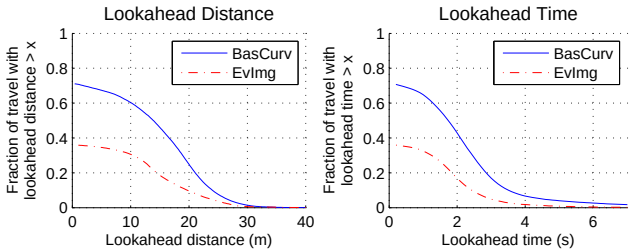
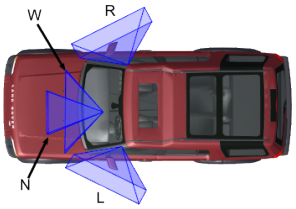


Fig. 15 Lane estimate lookahead distance (left) and time (right). The plots show how frequently the system maintained a minimum lookahead, as measured by fraction of the vehicle's total travel distance.



Camera	HFOV	Pitch
W	87.4°	2.7°
N	41.3°	3.0°
R	87.1°	14.9°
L	87.4°	14.7°

Fig. 16 (left) Camera viewing volumes. (right) Horizontal field-of-view and pitch for each camera.

7.2 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. 15 aggregates lookahead statistics over both datasets, and shows the lookahead distance and lookahead time 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.

7.3 Multiple cameras

The lookahead distance of a lane estimation algorithm is limited by the range of its input features. The feature detectors in turn are limited by the range and resolution of the sensors on which they operate. Here, we consider the use of additional cameras in a second experiment.

The road-paint feature detectors were run on four onboard Point Grey Firefly MV cameras. Cameras varied by focal length, position, and orientation. Two forward-facing cameras mounted above the rear-view mirror differed by focal length. Two additional cameras were placed above the left and right side mirrors, respectively, and were oriented to point slightly down and away from the vehicle. Figure 16 shows the camera viewing volumes. All cameras had an image resolution of 752×480 pixels, and acquired images at 22.8 Hz.

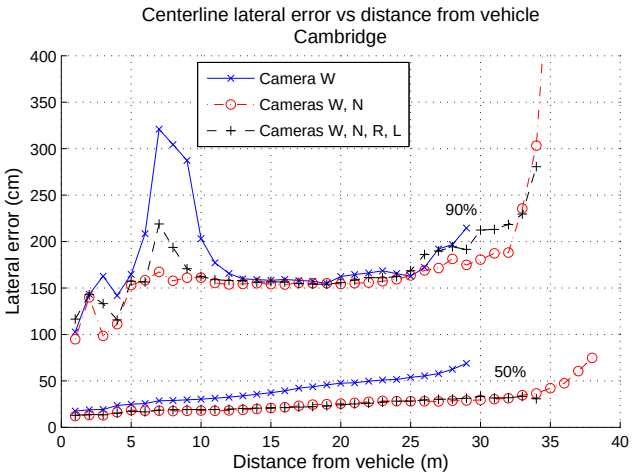


Fig. 17 The 50 and 90 percentile values for centerline error when using one, two, and four cameras. Camera calibrations are given in Fig. 17.

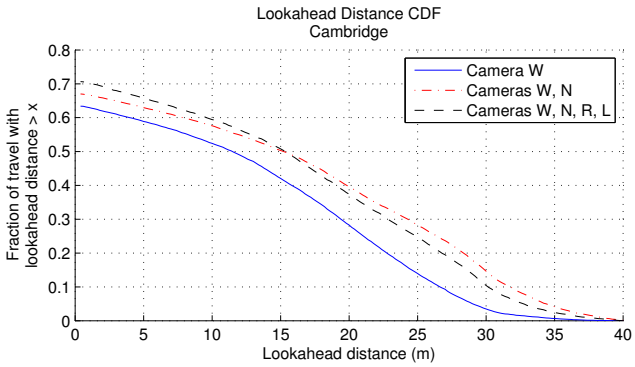


Fig. 18 Lookahead distance with using one, two, and four cameras.

The curb detector was also run on the Velodyne HDL-64E data, identical to the first experiment described.

Figure 17 shows the median and 90% centerline error and when the basis curve lane estimation algorithm was run with one, two, and four cameras as input to the road-paint detectors and the Velodyne as input to the curb detector. Figure 18 plots the lookahead distance achieved with different camera configurations.

The results show that using two forward-facing cameras with different focal lengths provides a modest improvement in centerline error over using a single wide-angle camera alone. For example, median centerline error at 25 m from the vehicle was reduced from 54 cm to 28 cm. As expected, the lookahead distance shows a significant improvement, with the vehicle maintaining some forward lane estimate for 68% of distance traveled using the two cameras, as opposed to 63% of distance traveled using one camera. The median lane estimate lookahead increased from 11.3 m to 15.6 m.

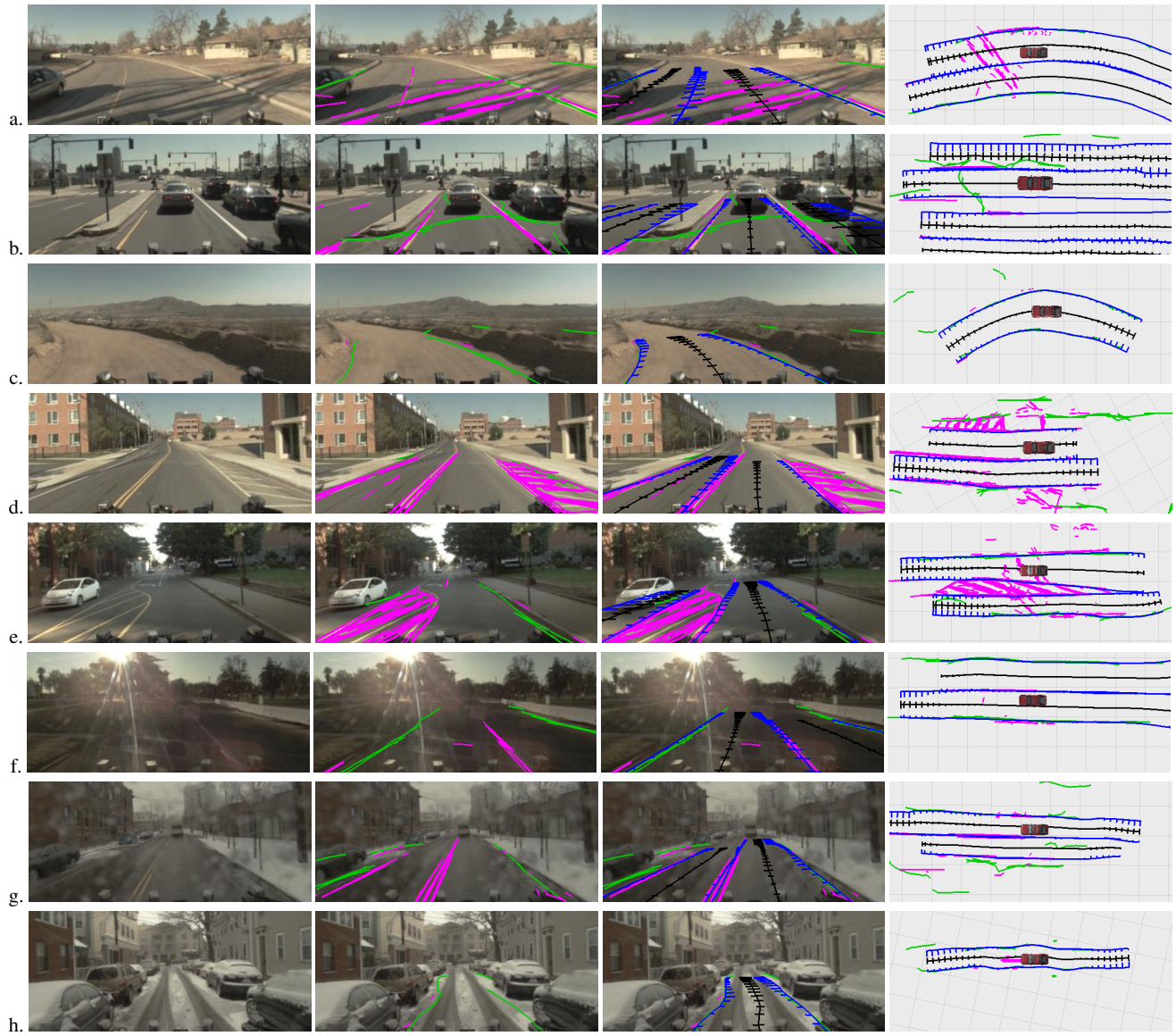


Fig. 19 Lane estimation in a variety of environments. *Column 1:* Camera images. *Column 2:* Detections of road paint (magenta) and curbs (green). *Column 3:* Lane centerline estimates (black) and boundary curve estimates (blue) projected into images. *Column 4:* Synthesized overhead views.

Two additional side-facing cameras did not substantially improve the centerline error, and in some cases resulted in an increased error. This can be attributed to a slight increase in the number of falsely detected lanes on the side of the road such as in breakdown lanes and parking spots. While the side-facing cameras increased the fraction of travel with lane lookahead up to 15 m forward of the vehicle, the lookahead rate for beyond 15 m decreased slightly over using the two forward-facing cameras alone. This can be attributed to processing time; as the number of feature observations near the vehicle increased as a result of the additional side-facing cameras, the estimation algorithms were slower to incorporate updates from the forward-facing cameras.

Overall, using additional forward-facing narrow-angle camera significantly reduced centerline error and increased lookahead distance. Using side-facing cameras improved overall "near-field" awareness, but not centerline error.

7.4 Qualitative results

Fig. 19 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), lane estimates are formed from physical boundaries on a dirt road

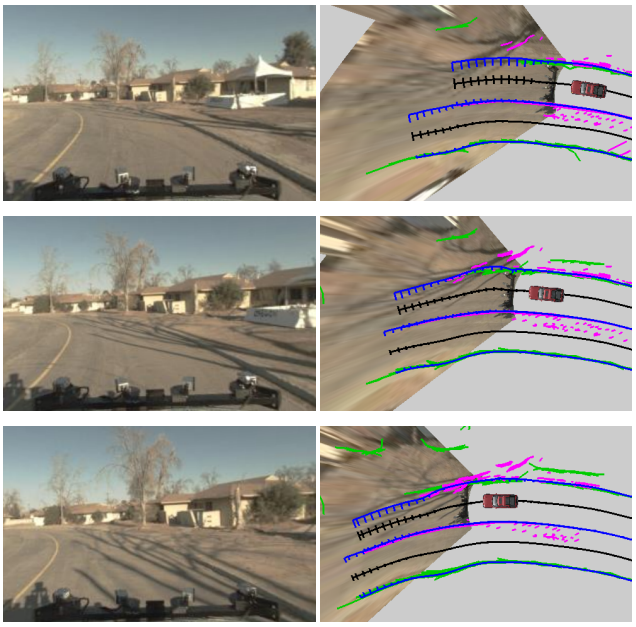


Fig. 20 Outlier observations, such as long shadows nearly parallel to the road, can result in incorrect data association. This may cause lane estimates diverge significantly from the true lane. Here, the true lane is wider than estimated.

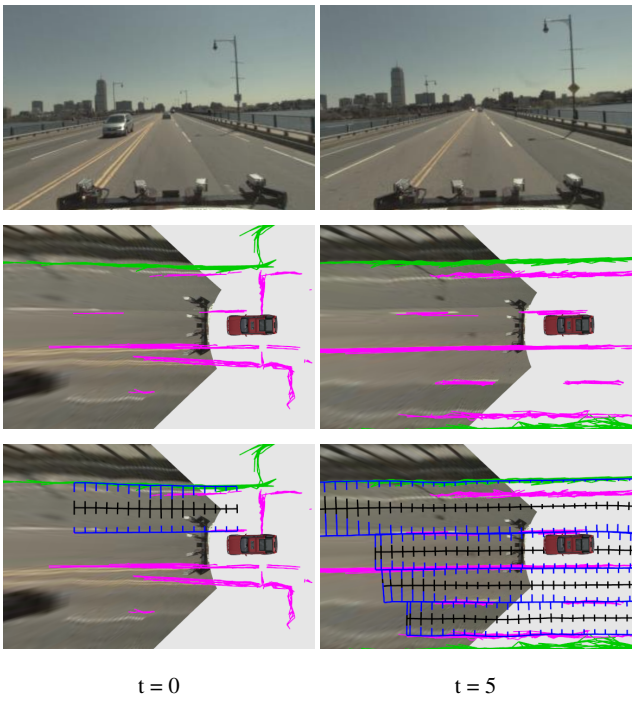


Fig. 21 Simple model tracking is not always sufficient for lane estimation. (left) Strong curb detections on the right side of the road result in a lane detection that spans both the correct lane and also the road shoulder. (right) Even though the correct boundary is observed later, continued persistent curb detections result in a high-confidence lane estimate that has "locked" on to the wrong boundary curve.

in the absence of painted markings. In (d) and (e), correctly detected road paint is successfully excluded from the lane estimates. In (f-h), the method performs well despite adverse lighting conditions and a snow-covered environment.

8 Discussion

The evidence image algorithm performs no outlier rejection, attempting to fit lanes to both true lane boundary detections and false detections such as long shadows and non-boundary road paint; nor does it estimate lane width. 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 its joint estimation of lane width and centerline geometry.

8.1 Limitations and future work

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. As is the case with many data association problems, the most difficult outliers to detect are those that are very similar to true inliers. Cases where outliers appear very similar to inliers, such as long shadows nearly parallel to the road, can cause the lane estimates to diverge (Figure 20).

Another failure mode arises when one road marking appears initially to be the correct boundary, but the true lane boundary then comes into view (Figure 21). In this case, our method will converge upon the first marking as the lane boundary, since the unimodal nature of the Kalman filter prevents it from assigning substantial weight to the true boundary. While a simple heuristic to re-initialize the lane estimate on the closest two boundary curves might address this scenario, other situations may be more difficult. For example, if the old lane markings on a recently repainted road are still visible, then it may not be obvious even to human drivers which markings denote true lanes. These situations illustrate the need for an explicit procedure to reason about the most likely set of nearby lanes, given the current observations.

It should also be possible to apply lessons learned in other estimation 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.

9 Conclusion

This paper introduced the notion of basis curves for curve estimation, and described an application to the lane estimation 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

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

Bar-Shalom Y, Li XR (2001) Estimation with Applications to Tracking and Navigation. John Wiley & Sons, Inc.

Bates C (2010) Chile earthquake moved whole city 10 feet to the west. http://www.dailymail.co.uk/-sciencetech/article-1256597/Chile-earthquake-moved-Concepcion-city-10ft-west.html

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

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

Blake A, Isard M (1998) Active Contours. Springer-Verlag

Commonwealth of Massachusetts Office of Geographic and Environmental Information (2008) MassGIS planning roads datalayer description. http://www.mass.gov/mgis/eotroads.htm

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

Huang AS, Teller S (2009) Lane boundary and curb estimation with lateral uncertainties. In: Proc. IEEE Int. Conf. on Intelligent Robots and Systems, St. Louis, Missouri

Huang AS, Teller S (2010) Probabilistic lane estimation using basis curves. In: Robotics: Science and Systems (RSS), Zaragoza, Spain

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

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

Matsushita Y, Miura J (2009) On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter. In: Proc. European Conference on Mobile Robots

McCall JC, Trivedi MM (2006) Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. IEEE Transactions on Intelligent Transport Systems 7(1):20– 37

Neira J, Tardos JD (2001) Data association in stochastic mapping using the joint compatibility test. IEEE Trans Robotics and Automation 17(6):890–897, DOI 10.1109/70.976019

Pomerleau D, Jochem T (1996) Rapidly adapting machine vision for automated vehicle steering. IEEE Expert: Special Issue on Intelligent Systems and their Applications 11(2):19–27, see also IEEE Intelligent Systems

Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. MIT Press

Sehestedt S, Kodagoda S, Alempijevic A, Dissanayake G (2007) Robust lane detection in urban environments. In: Proc. IEEE Int. Conf. on Intelligent Robots and Systems, San Diego, CA, USA

Southall B, Taylor CJ (2001) Stochastic road shape estimation. In: Proc. Int. Conference on Computer Vision, vol 1, pp 205–212, DOI 10.1109/ICCV.2001.937519

Thorpe C, Hebert M, Kanade T, Shafer S (1988) Vision and navigation for the Carnegie-Mellon Navlab. IEEE Transactions on Pattern Analysis and Machine Intelligence 10(3):362–373

Thrun S, Burgard W, Fox D (2005) Probabilistic Robotics. MIT Press

Urmson C, Anhalt J, Bagnell D, Baker C, Bittner R, Clark M, Dolan J, Duggins D, Galatali T, Geyer C, Gittleman M, Harbaugh S, Hebert M, Howard T, Kolski S, Kelly A, Likhachev M, McNaughton M, Miller N, Peterson K, Pilnick B, Rajkumar R, Rybski P, Salesky B, Seo YW, Singh S, Snider J, Stentz A, Whittaker W, Wolkowicki Z, Ziglar J, Bae H, Brown T, Demitrish D, Litkouhi B, Nickolaou J, Sadekar V, Zhang W, Struble J, Taylor M, Darms M, Ferguson D (2009) Autonomous driving in urban environments: Boss and the urban challenge. In: Buehler M, Iagnemma K, Singh S (eds) The DARPA Urban Challenge, Springer Tracts in Advanced Robotics, vol 56, Springer Berlin / Heidelberg, pp 1–59

Voigt K (2011) Quake moved japan coast 8 feet, shifted earth's axis. http://www.cnn.com/2011/WORLD/-asiapcf/03/12/japan.earthquake.tsunami.earth

Wang Y, Teoh EK, Shen D (2004) Lane detection and tracking using B-Snake. Image and Vision Computing 22(4):269 – 280