Autonomous Vehicle Navigation in Rural Environments without Detailed Prior Maps

Teddy Ort[1], Liam Paull[1,2], Daniela Rus[1]

***Abstract*— State-of-the-art autonomous driving systems rely heavily on detailed and highly accurate prior maps. However, outside of small urban areas, it is very challenging to build, store, and transmit detailed maps since the spatial scales are so large. Furthermore, maintaining detailed maps of large rural areas can be impracticable due to the rapid rate at which these environments can change. This is a significant limitation for the widespread applicability of autonomous driving technology, which has the potential for an incredibly positive societal impact. In this paper, we address the problem of autonomous navigation in rural environments through a novel mapless driving framework that combines sparse topological maps for global navigation with a sensor-based perception system for local navigation. First, a local navigation goal within the sensor view of the vehicle is chosen as a waypoint leading towards the global goal. Next, the local perception system generates a feasible trajectory in the vehicle frame to reach the waypoint while abiding by the rules of the road for the segment being traversed. These trajectories are updated to remain in the local frame using the vehicle's odometry and the associated uncertainty based on the least-squares residual and a recursive filtering approach, which allows the vehicle to navigate road networks reliably, and at high speed, without detailed prior maps. We demonstrate the performance of the system on a full-scale autonomous vehicle navigating in a challenging rural environment and benchmark the system on a large amount of collected data.**

I. Introduction

Autonomous driving has the potential to drastically improve our lives. To date, the vast majority of fielded autonomous vehicles focus on one of two scenarios:

1) Lane following on well-marked structured highways
2) Urban navigation based on extremely precise and manually annotated detailed global maps

However, over a third of the roads in the United States are unpaved [1], and 65% do not possess reliable lane markings. These rural roads are challenging for autonomous driving because they are sparsely connected and cover vast areas. Thus, while the detailed global mapping approach becomes impractical as the maps grow prohibitively large, the lane following approach is also infeasible since lane markings and road curb geometry are frequently unavailable for reliable road lane following.

In this work, we build an efficient framework for autonomous driving on rural roads that combines the crowd-sourced *topological* map, open street map (OSM), with a *local* perception system for navigating individual road segments. These two capabilities (OSM + local perception) combine to enable global navigation over vast areas with a manageable amount of required preloaded data (just the OSM). The key insight which enables this lies in the fact that although GPS data is not precise enough for autonomous driving, it is precise enough to enable topological localization, and consequently can be augmented with local perception to solve the full autonomous navigation problem since the OSM contains all the rules associated with each road segment. For comparison, note as an example Levinson and Thrun [2] who use a compression method to shrink the size of their high-resolution 2D maps. They are able to store a 20,000 mile map in 200 GB. In contrast, a typical topological map of the same area would only require 3.5 GB. Given that the US alone contains more than 4,000,000 miles of roadways, the storage size of large maps is clearly a significant challenge.

Fig. 1: Mapless Navigation using Sparse Topological Maps. Top: The crowd-sourced topological map from openstreetmap.org is shown as red segments connecting yellow vertices. The point cloud obtained from a Velodyne HDL-64 laser scanner shows the area of the global map visible to the vehicle's sensors. Bottom: The full end-to-end mapless autonomous control system has been demonstrated in a rural setting.

This work was partially supported by the Toyota Research Institute (TRI). However, this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

[1] Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology {teddy,rus}@mit.edu

[2] Département d'informatique et de recherche opérationelle, Université de Montréal {paulll}@iro.umontreal.ca

We take a holistic approach in our application of this framework using one possible method of road segmentation that is particularly suitable for rural environments. It is based on robustly tracking road boundaries using a 3D LiDAR sensor that is capable of estimating the road surface edges without any assumptions about road markings and only a very loose prior knowledge about the road geometry (that a road is relatively flat).

Our method is very efficient despite the large rate of data collection from the sensor since we use the current road boundary estimate as a prior for detection at the next time step. We fuse the individual road boundary detections, together with the vehicle odometry, in a probabilistic framework. We have tested our algorithm on a full-scale autonomous Prius [3] in a rural environment. We have also evaluated our method offline on datasets collected from our test site. The full perception system runs on a standard PC at 5Hz and can reliably detect the road up to 35m in advance, meaning that it can enable the car to travel at speeds well exceeding 30m/s (67mph), and could be much more if the method was parallelized and implemented on a GPU.

In summary, we claim the following contributions:

- A framework for autonomous driving that relies on a topological rather than metric prior map;
- An example implementation using LiDAR-based road segmentation and temporal road boundary estimation that is particularly effective in unstructured environments;
- A demonstration on a full-scale autonomous car in an unstructured environment.

The remainder of the paper is structured as follows: In Sec. II, we review the related literature, in Sec. III, we introduce our generalized framework for autonomous driving based on topological maps and local perception, in Sec. IV, we describe in detail our specific application of this framework to enable autonomous driving in rural environments, in Sec. V, we show results from our experiments, and in Sec.VI we discuss our conclusions and future work.

II. Related Work

This work is inspired by previous work in the areas of *hybrid metric-topological mapping*, as well as *local perception and localization for autonomous driving*. In this section, we will highlight some of the most related works.

A. Hybrid Metric/Topological Mapping

The notion of hybrid metric/topological estimation has been an active field of research in the simultaneous localization and mapping (SLAM) community for two decades or more. For example, seminal works such as [4] and [5] laid the foundation in this field. The fundamental idea here is to use *local* perception for small-scale localization and topological perception for global localization and navigation. In some sense, these works are faced with a larger problem than what we are attempting to tackle because they attempt to simultaneously build and localize within both the small-scale and the topological maps in an online fashion. In this work, we exploit the fact that, in the driving context, there already exists a curated and open topological map that is actually much more expressive than the topological representations used in these works (such as the Generalized Voronoi Graph) since the edges in OSM are also labeled with important semantic information.

Many other related works in the SLAM field associate one submap with each edge in the topological graph and focus on solving the relative transformation between the sub-graphs [6] [7]. The challenge here relates to allowing a robot to robustly transition from one submap to another.

We embrace a fully relative representation of the local map. In other words, the frame used for small-scale localization is always attached to the robot (and is updated each time the robot moves). Others have embraced this fully relative view [8]–[10] for SLAM by parameterizing the robot trajectory in continuous time.

In this work, our goal is not to generate maps (in relative or global frame) but rather to generate feasible trajectories within the map. By explicitly connecting the local relative perception problem with the task of navigating through the environment at that instant, we can derive a much more lightweight representation for the robot state.

B. Localization for Autonomous Driving

The existing state-of-the-art localization and perception systems can be categorized based on whether they require a *prior metric map* [2], [11]–[13], and, if so, by the representation of that map. For autonomous mobility-on-demand systems that are severely restricted in their operational area, maintaining a highly accurate globally referenced metric map may be possible, but this approach does not scale well either temporally or spatially.

An entirely separate approach, more closely related to what we are proposing here, is to perceive the local environment for autonomous vehicle navigation. This type of approach is appealing since it does not require construction, maintenance, and storage of very dense maps. However, these algorithms require a mechanism for accurately detecting the drivable road surface or lanes reliably and then connecting this information to a global topological map for robust navigation. The vast majority of local perception systems utilizing such approaches rely heavily on road markings [14]–[16], the very predictable geometry of the road curbs [17], [18], or both [19]. Vision-based approaches to road segmentation not based on road markings include both model-based [20] and learning-based [21] methods and can achieve good performance in ideal conditions, but fail under poor illumination.

In Sec. IV we propose a LiDAR-based road segmentation algorithm that makes minimal assumptions on the lane markings or road geometry and describe how this applies to our mapless navigation framework for rural autonomous driving.

III. Mapless Autonomous Driving Framework

A. Framework Overview

In this section, we describe our general "mapless" autonomous driving framework. By mapless we refer to the fact that, unlike most autonomous driving systems (e.g. [2],

[11]–[13], [22]–[24]), we assume no precise metric high-resolution global map. However, we do assume that we have access to a "topological" map of the environment in which the vehicle is operating (such as the OSM shown in Fig. 1). This topological graph is represented as $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$, where each vertex $v_i \in \mathcal{V}$ contains a (possibly imprecise) GPS location $v_i = (lat_i, lon_i)$. Each edge $e_{ij} \in \mathcal{E}$ indicates that there is a direct feasible path from vertex v_i to vertex v_j that does not pass through any other vertex. These edges may also be weighted according to the traversal cost. We define the navigation task as finding a feasible trajectory from the current location x_{start} to the goal location x_{end} also in GPS coordinates.

We assume that each edge e_{ij} contains a unique set of rules of the road that should be adhered to in order for safe traversal. For a detailed treatment of how these rules can be encoded please see [25].

We decompose this problem into the following sub-tasks: 1) Global topological localization 2) Graph search, and 3) Edge traversal. The topological localization and graph search problems are relatively well formulated and can be solved with commercial GPS/navigation systems. Therefore, we only give a brief overview of their functionality here.

The topological localization problem is that of adding the current location x_{start} to the topological graph as a vertex, v_{start}, and connecting it to the graph by finding at least one feasible path to an existing vertex.

The same is done for the end location x_{end} and vertex v_{end}. The graph search procedure produces $\{v_{start}...v_{end}\} \in \mathcal{V}$, a sequence of connected vertices that have the shortest path. This problem can be solved with standard graph search algorithms, for example Djikstra's algorithm.

B. Edge Traversal

This work focuses primarily on the edge traversal sub-task of the full navigation problem. At this stage, we assume we have obtained the next edge that must be traversed, e_{ij}, along with any associated rules-of-the-road from the result of the previous topological localization and graph search sub-tasks. The goal of edge traversal is to enable the robot to traverse the edge using only local sensor information. An overview of the procedure is given in Algorithm 1.

First, define the local coordinate frame of the robot at time t to be X_t, the set of valid configurations that adhere to the rules of the road on edge e_{ij} to be $\mathcal{X}_{ij}$, and the sensor swath at time t to be $\mathcal{S}_t$. Furthermore, for brevity of notation, we will assume in this section that $\mathcal{X}_{ij}$ is 2D, that is, that the road segment is approximately flat. However, this approach could easily be extended to 3D if the elevation change in the edge traversal were significant.

Next, we find the local goal as the point in X_t that is as close as possible (in the Euclidean sense) to v_j, the next waypoint in the list of graph vertices:

$$(x_{goal}, y_{goal}) = \underset{(x_t, y_t) \in \mathcal{S}_t \bigcap \mathcal{X}_{ij}}{\operatorname{argmin}} ||^{X_t}v_j - [x_t, y_t]^T||_2 \quad (1)$$

where ${}^{X_t}v_j$ is the Cartesian location of vertex v_j in the X_t frame.

Algorithm 1 Edge Traversal Algorithm: Local Road Perception, Tracking and Following

Inputs:
- next global waypoint
- stream of local road detection measurements
- stream of odometry measurements

while Current state not equal to global waypoint **do**
  while New sensor input not ready **do**
    Update reference trajectory estimate $\mathbf{f}_{k+1}(\alpha, \bar{\Theta}_{k+1})$ where $\bar{\Theta}_{t+1} = \mathbf{T}^{-1}\Theta_t$
    Generate control commands with trajectory following controller
  end while
  Generate a new reference trajectory based on the current sensor input $\mathbf{f}_{k+1}(\alpha, \hat{\Theta}_{k+1})$
  Probabilistically fuse reference trajectory estimates
end while

This formulation accounts for two important possibilities that result from the fact that we are planning in the local frame: 1) The GPS coordinates in the OSM map may be sufficiently inaccurate that navigating precisely to those coordinates would be dangerous and 2) The next vertex to be attained in the graph may be sufficiently far away that it is outside of the sensor swath of the vehicle. In such cases, it is impossible to generate a feasible reference path from the current vehicle location all the way to the next waypoint. We solve both of these problems by projecting the goal waypoint onto the feasible space in the sensor swath.

The reference trajectory therefore satisfies:

$$\tau(\alpha) : [0, 1] \mapsto \mathcal{X}_{ij} \quad (2)$$

with $\tau(0) = [0, 0]^T$ and $\tau(1) = [x_{goal}, y_{goal}]^T$.

Often this trajectory will be internally parameterized by arc length or curvature; however, we will assume that there is some mapping $\mathcal{M}$ to Cartesian coordinates:

$$\tau(\alpha) \xmapsto{\mathcal{M}} \mathbf{f}_t(\alpha, \Theta_t) = \begin{bmatrix} f_x(\alpha, \Theta_t) \\ f_y(\alpha, \Theta_t) \end{bmatrix} \quad (3)$$

where $\mathbf{f}(\alpha, \Theta)$ defines the model that is being used to represent the trajectory and Θ are the model parameters that are recursively estimated.

Furthermore, since the reference trajectory is generated using noisy sensor data, we also obtain $\sigma_\tau(\alpha)$, which contains the associated uncertainty for each point along $\alpha = [0, 1]$.

1) When a new odometry measurement arrives: For modeling the motion of the vehicle we use a zero-slip kinematic odometry motion model and assume that we have access to a noisy measurement of linear forward change in position, $\Delta_x + \epsilon_x$, and a noisy measurement of the change in heading, $\Delta_\phi + \epsilon_\phi$. The kinematic model is then given by:

$$\begin{aligned} x^r_{t+1} &= x^r_t + (\Delta_x + \epsilon_x)\cos\phi^r_t \\ y^r_{t+1} &= y^r_t + (\Delta_x + \epsilon_x)\sin\phi^r_t \\ \phi^r_{t+1} &= \phi^r_t + (\Delta_\phi + \epsilon^r_\phi) \end{aligned} \quad (4)$$

where $X^r = [x^r_t, y^r_t, \phi^r_t]^T$ is the pose of the vehicle at time t. However, in contrast to the standard approach where this motion model is used to estimate the position of the robot in

some global reference frame, in this case we are re-aligning the "global frame" with the local frame at every timestep. In other words, $x_t^r = y_t^r = \phi_t^r = 0$, and the motion model can be significantly simplified:

$$\begin{aligned} x_{t+1}^r &= (\Delta_x + \epsilon_x) \\ y_{t+1}^r &= 0 \\ \phi_{t+1}^r &= (\Delta_\phi + \epsilon_\phi) \end{aligned} \tag{5}$$

We update the local reference frame to coincide with the vehicle frame. Define the measurement as (Δ_x, Δ_ϕ), with associated uncertainty σ_Δ. This is achieved through a standard homogeneous transformation matrix, $[\bar{x}_{t+1}, \bar{y}_{t+1}, 1]^T = \mathbf{T}^{-1}[x_t, y_t, 1]^T$: where

$$\mathbf{T} = \begin{bmatrix} \cos(\Delta_\phi) & -\sin(\Delta_\phi) & \Delta_x \\ \sin(\Delta_\phi) & \cos(\Delta_\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Then if we restrict the mapping $\mathcal{M}$ to a linear function on the model parameters of the form $\mathbf{f}(\alpha, \Theta) = \Theta\mathbf{A}(\alpha)$ then the model parameters in the new coordinate frame are given by:

$$\begin{aligned} \overline{\boldsymbol{x}}_{t+1} &= \boldsymbol{T}^{-1}\boldsymbol{x}_t \\ &= \boldsymbol{T}^{-1}\boldsymbol{f}\left(\alpha, \Theta_t\right) \\ &= T^{-1}\Theta_t \boldsymbol{A}\left(\alpha\right) \\ &= \bar{\Theta}_{t+1}\boldsymbol{A}(\alpha) \end{aligned} \tag{7}$$

where $\bar{\Theta}_{t+1} = \mathbf{T}^{-1}\Theta_t$ and the modifications induced by the change of reference frame are propagated to the parameters in order to maintain the fixed model form of $\mathbf{f}$.

In practice, the frequency of this odometry loop can be much faster than the measurement loop described in the next section, as shown in Fig. 2.

2) When a new sensor measurement arrives: We assume that the vehicle is equipped with a sensor or suite of sensors that can perceive the local environmental geometry and even potentially semantics. These could be vision, LiDAR, or radar sensors, or any combination thereof.

Each new batch of sensor data (e.g. an image from a camera or a scan from a laser), is fed through a perception system to generate "feature" detections in the sensor frame (i.e. no registration is necessary). At this stage, the *local motion planning algorithm* is run to generate a suitable reference path to $\mathbf{x}_{goal}$ as described above. Since the perception system is generally imperfect, the uncertainty generated by perception can be propagated onto the trajectory parameters.

Consequently, we arrive at a new local reference trajectory, $\mathbf{f}_{k+1}(\alpha, \hat{\Theta}_{k+1})$ where once again the predefined model has been fit and the trajectory uncertainty has been propagated to the parameters. Finally, since the trajectory from odometry propagation $\mathbf{f}_{k+1}(\alpha, \bar{\Theta}_{k+1})$ and the new trajectory generated from the sensor measurement $\mathbf{f}_{k+1}(\alpha, \hat{\Theta}_{k+1})$ are in the same local reference frame, they can be probabilistically fused.

IV. Application: Edge Traversal on Rural Road Segments

An instance of the above framework was used to navigate autonomously on rural road segments as outlined in Fig. 2.

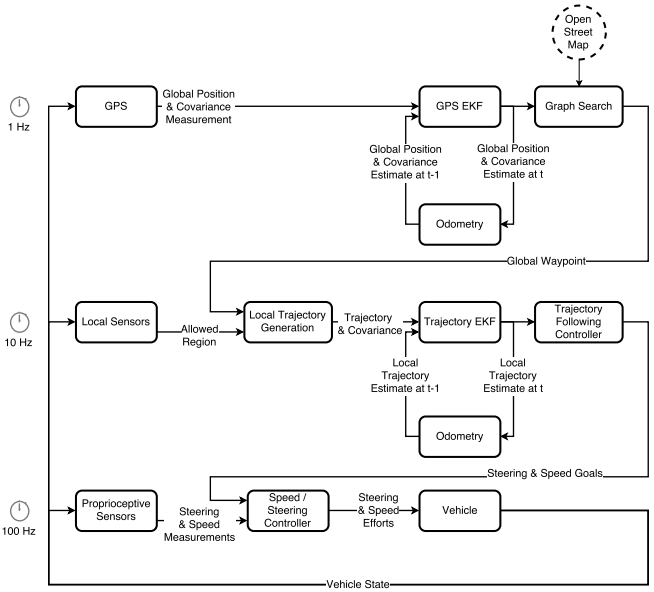

Fig. 2: An overview of the mapless driving algorithm showing the three levels of hierarchy: Level 1 incorporates coarse GPS measurements for global navigation at 1Hz. Level 2 uses sensor measurements to generate a trajectory towards a local waypoint at 10Hz. Level 3 performs low-level control of the vehicle speed and steering at 100Hz.

In this application, OSM is used for mapping and the approximate location on the OSM is obtained using GPS. The sensor used for planning safe trajectories in the local frame is a Velodyne HDL-64 laser scanner. Each revolution of the sensor generates a pointcloud. A road segmentation algorithm was developed to obtain the road boundary points in the sensor swath of the vehicle. These points were then fit using a RANSAC/least-squares approach to obtain an optimal trajectory within the road boundary points. Importantly, the quality of the fit was also obtained from the residuals output from the least-squares minimization. This allows the trajectory estimates to be probabilistically fused during the next iteration. The vehicle is also equipped with odometry sensors used to propagate the previous trajectory estimates to the local vehicle frame after the vehicle has moved. Thus, the reference trajectory is always in the vehicle local frame. Since the odometry is typically updated much faster than the sensor measurements arrive (in this case 100Hz), the vehicle can continue to follow the reference trajectory for some time even while no new measurements have been received. Finally, the reference trajectory is used to steer the vehicle towards the local goal for edge traversal (x_{goal}, y_{goal}) which is within the sensor swath. As the sensor reveals new areas of the map, the local goal will move closer to the global goal v_{end} as described in Sec. III.

A. Preliminaries

As is common in autonomous vehicle navigaition, we represent the reference trajectory as a clothoid [26]. Specifically, we choose to model the road as a clothoid of degree one whereby the curvature of the road varies linearly. We can recover the heading direction along the path by integrating

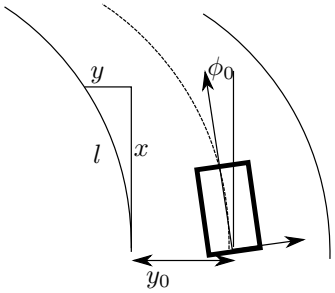

Fig. 3: Clothoid geometry

the curvature along the path:

$$\begin{aligned}\phi(l) &= \phi_0 + \int_0^l C(t)dt \\ &= \phi_0 + lC^0 + \frac{l^2}{2}C^1\end{aligned} \tag{8}$$

Referring to Fig. 3, based on differential geometry we can reparameterize the clothoid in Cartesian coordinates [26]:

$$\begin{aligned}x &= x_0 + \int_0^l \cos\phi(t)dt \\ y &= y_0 + \int_0^l \sin\phi(t)dt\end{aligned} \tag{9}$$

Collectively (8) and (9) combine to define the mapping $\mathcal{M}$ as defined in 3.

By assuming that: (a) the origin of the curve in Cartesian coordinates is in line with the vehicle ($x_0 = 0$) and (b) that the change in heading over the course of the curve is relatively small ($\cos\phi(t) \approx 1, \sin\phi(t) \approx \phi(t) \forall t$[1]), we can arrive at the following spline representation in Cartesian coordinates:

$$y = y_0 + \phi_0 x + \frac{C^0}{2}x^2 + \frac{C^1}{6}x^3, \tag{10}$$

which corresponds to selecting $\mathbf{f}(\alpha, \Theta)$ according to

$$\mathbf{f}(\alpha,\Theta) = \begin{bmatrix} 0 & x_{goal} & 0 & 0 \\ y_0 & \phi_0 x_{goal} & \frac{C^0}{2}x_{goal}^2 & \frac{C^1}{2}x_{goal}^3 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \alpha \\ \alpha^2 \\ \alpha^3 \end{bmatrix} \tag{11}$$

as described in Sec. III-B.1.

The four-dimensional state that we will use to estimate the allowable configurations within the sensor swath will then be: $\mathcal{S}_t \bigcap \mathcal{X}_{ij} = \{y_0, \phi_0, C^0, C^1\}$. Note that this region is the maximum drivable region in the current sensor swath and rules of the road constraints in $\mathcal{X}_{ij}$ could be used to further restrict the allowable trajectories to a particular portion of the road. In this application, since the roads are in a test facility without oncoming traffic, the trajectory was chosen to follow the center of the road.

[1]as is noted in [26] this second assumption can be explicitly enforced by truncating the curve at such a point as this approximation error grows too large - typically $\phi(t) > 15^o$ is used as a threshold

B. Trajectory Propagation from Odometry

When new vehicle odometry arrives (at high frequency), the spline estimated is updated based on the information from the onboard sensors of the vehicle. The process model is given by

$$X_{t+1}^s = f(X_t^s, X_{t+1}^r, \epsilon_t) \tag{12}$$

where $\epsilon_t = [\epsilon^x, \epsilon^\phi, \epsilon^{C^0}, \epsilon^{C^1}]^T$ are all the additive zero-mean Gaussian sources of noise in the process model, and f is derived by substituting the updated vehicle estimate from odometry (5) into the spline function (10), and, differently from [20], we also incorporate the change in heading resulting from the odometry, which results in an additional offset added to the ϕ_0 parameter:

$$\begin{aligned}y =& y_0 + (\phi_0 + \Delta_\phi + \epsilon_\phi)(x + \Delta_x + \epsilon_x) \\ &+ \frac{C^0}{2}(x + \Delta_x + \epsilon_x)^2 + \frac{C^1}{6}(x + \Delta_x + \epsilon_x)^3\end{aligned} \tag{13}$$

which is solved and then the coefficients are collected relative to the appropriate terms to yield the noiseless process model:

$$X_{t+1}^s = \begin{bmatrix} 1 & \Delta_x & \frac{1}{2}\Delta_x^2 & \frac{1}{6}\Delta_x^3 \\ 0 & 1 & \Delta_x & \frac{1}{2}\Delta_x^2 \\ 0 & 0 & 1 & \Delta_x \\ 0 & 0 & 0 & 1 \end{bmatrix} X_t^s + \begin{bmatrix} \Delta_\phi \Delta_x \\ \Delta_\phi \\ 0 \\ 0 \end{bmatrix} \tag{14}$$

Note that this process model is linear with respect to the state but not with respect to the noise parameters. Therefore the noise Jacobians, $W = \frac{\partial f}{\partial \epsilon_t}$ are calculated at each time step in the prediction phase:

$$W = \begin{bmatrix} \phi_t^0 + \Delta_x C_t^0 + \frac{1}{2}\Delta_x^2 C_t^1 + \Delta_\phi & \Delta_x & 0 & 0 \\ C_t^0 & \Delta_x C_t^1 & 0 & 0 \\ C_t^1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

and the final additive noise term in the filter update is given by $W^T Q W$, where Q is a diagonal matrix of noise covariances for each of the sources of noise.

C. Single Scan Road Boundary Detection

In this section, we describe our method for robustly detecting road regions in rural areas from LiDAR data. The key insight in this method is that we can detect the road area in each individual laser scan by thresholding in the *frequency domain*. The road has a much smoother texture relative to surrounding areas, the points in each ring exhibit less variation in their position compared to the points lying in the grass or trees on the side as shown in Fig. 5. To characterize the texture of a region, the distance from the vehicle at each point is calculated where for the i^{th} point x_i the distance is $d_i = \|x_i\|$. Next, the distance signal is processed by taking the absolute value of the first derivative of the signal in order to remove the mean and amplify the noise. The resulting signal is a measure of the surface texture where larger values indicate rougher surfaces. The top graph in Fig. 4 shows the raw signal representing a single ring in a laser scan while the processed signal is shown in the bottom.

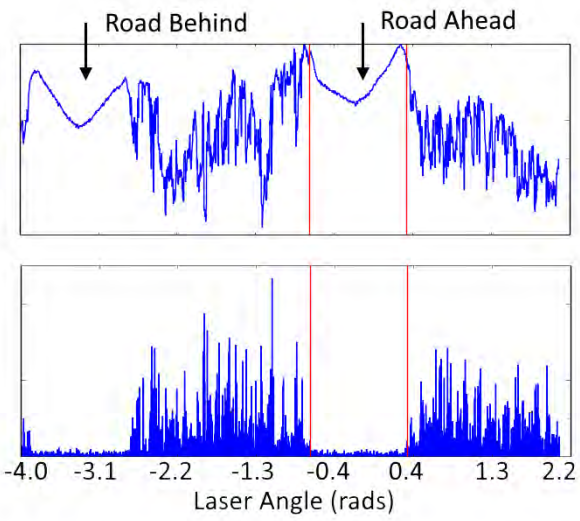

Fig. 4: A single ring from the pointcloud viewed as a 1 dimensional signal the road texture is clearly distinguishable from the surroundings. The top graph shows the raw signal and the bottom shows the signal after post-processing (see IV-C). The vertical red lines indicate the location marked as the road boundary in front of the vehicle.

The vertical red lines show the location of the actual road edges for the road in front of the vehicle.

Finally, the processed signal is searched for the road edges by first finding the standard deviation of the signal in the road area and then returning the first point that is larger than n standard deviations where n is a tunable parameter used to choose the sensitivity of the edge detection. In practice, this value could remain constant over a variety of roads and weather conditions.

Once the edges in a single ring are found, the algorithm moves on to the next ring. In each case, the initial guess for the location of the edge is chosen as the point that has the shortest Euclidean distance to the one in the prior ring. In this way, the edge detection moves progressively along the edges of the road without the need to explore the regions of the point cloud that are far from the road. Fig. 5 shows the result of running the edgepoint detection algorithm on the pointclouds.

D. Trajectory Update from New Local Estimate

We use RANSAC to fit the boundary points to a spline (one for each road boundary). The reference spline trajectory is then generated as the average between the two boundary splines as shown in blue in Fig. 5. The result of each single scan curve fit is treated as a new incoming measurement. After performing RANSAC, the inliers are fit to the spline using a least-squares fitting procedure, and the *residual* of this fitting procedure is used to scale the covariance of the measurement update.

$$z_t = HX_t^s + \eta_t \tag{16}$$

where $H = [x, \frac{1}{2}x^2, \frac{1}{6}x^3]^T$ and each data point used to fit the curve can be treated as an independent measurement. For point i:

$$\eta_t^i = ||z_t^i - H^i X_t^s||^2 \tag{17}$$

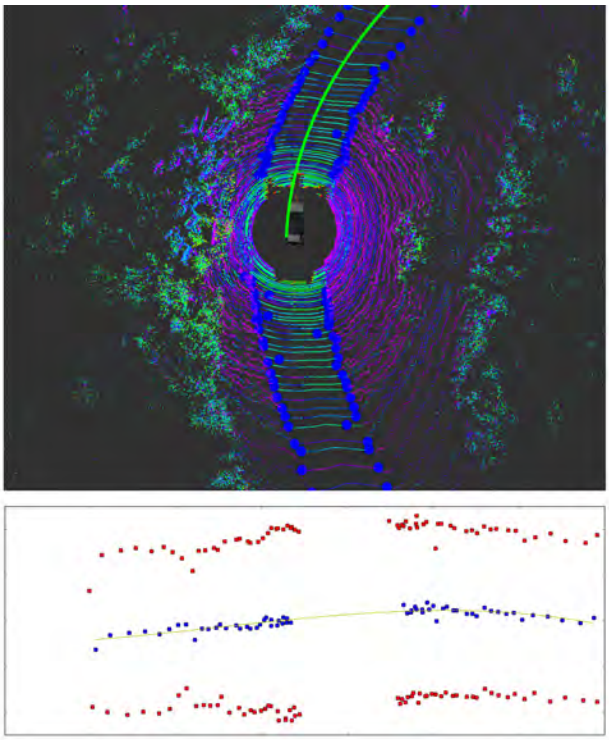

Fig. 5: Top: We use a texture based approach to detect the road boundaries (blue spheres) from single laser scans. Bottom: The result from a RANSAC implementation used to fit a spline to the edge points detected at the road boundaries. Note that the empty area in the center is the location of the vehicle, which occludes the sensor.

Thus, the detected road is used to generate a feasible driving trajectory along with a measure for the certainty that the trajectory is correct.

E. Trajectory Following Controller

The final step to enable autonomous navigation is the trajectory following controller. This lies between the output of the road boundary segmentation and the low-level vehicle steering controller as seen in Fig. 2. The trajectory follower takes as input a parameterized trajectory spline along the road and determines the steering angle the vehicle should take to follow that trajectory. Since the trajectory generation is happening in a frame always coincident with the vehicle, we use a moving goal-point approach where the vehicle is always following a point at a fixed distance, δ, on the input trajectory. To calculate the position and orientation errors at δ, first the spline model in (10) is evaluated to get the position error $\Delta y_0 = y(\delta)$, then the derivative of that model is used to calculate the angle error.

$$\begin{aligned} \frac{dy}{dt} &= \phi_0 + C^0 x + \frac{C^1}{2}x^2 \\ \Delta\phi_0 &= \tan^{-1}\left(\frac{\mathrm{dy}}{dt}\bigg|_{x=\delta}\right) \end{aligned} \tag{18}$$

Finally, these angle errors are passed to the position and orientation PID controllers, which were tuned to drive these errors to 0.

V. Results

A. Experimental Setup

The vehicle used for testing and data collection was a 2015 Toyota Prius V retrofitted for autonomous driving (see [3] for details). A Velodyne HDL-64S3 LiDAR sensor was used for perception, while a Microstrain 3DM-GX4 inertial measurement unit (IMU) and external wheel encoders provided onboard odometry measurements. Since the road segmentation is able to perceive the road up to 35m ahead, and the vehicle reached maximum speeds of $10\frac{\text{m}}{\text{s}}$, at 5hz, the frequency of the road segmentation was fast enough to ensure that the vehicle traveled at most 2m or $< 5\%$ along the known road spline before it was ready to incorporate new sensor data.

B. Localization Accuracy

In order to evaluate the accuracy of the local trajectories generated by the local perception system, the parameters $\hat{X}_t^s$ and the uncertainty P_t at each time step t were recorded. Furthermore, the raw parameters estimated directly from the model-fitting step without temporal filtering were also saved. Finally, the vehicle was manually driven down the center of the road and the path taken was recorded using the odometry measurements. These measurements served as a ground truth allowing us to compare the path predicted by the segmentation process with the path the vehicle actually took. The root mean squared distance (RMSD) was evaluated for each point along the vehicle path, as compared to the true road center to determine how well the predicted path matched. These measurements allowed us to compare both the unfiltered and filtered road parameters as estimated by the segmentation system for evaluation. In the top of Fig. 6, we show these road spline estimates overlaid on a map of the test site and the corresponding deviation from the road center is shown in the bottom. Notice that while the unfiltered parameters occasionally show spikes in the deviation, the filtered estimate shown in red tends to attenuate those spikes due to the robustness gained by fusing the new measurements with the prior estimate.

To better quantify the overall quality of the results and the effect of the filter, refer to Fig. 7. The histogram shows that for both methods more than 80% of the predicted trajectories were within 1m of the road center and more than 97% within the road boundaries at 3 meters. While we do see a drawback due to utilizing the filter in a shift of the histogram of the filtered parameters toward the right side, this is expected because the filtered parameters do not respond as quickly to new measurements giving them a larger deviation overall. As seen previously in Fig. 6 however, this is critical in order to robustly reject the outliers in the raw estimate caused by the highly unstructured environment.

Table I shows a summary of the results from processing $n = 700$ scans. The first row indicates that while, on average, the error was < 1m, there was a slight increase in the average deviation when using the filter. The next row shows that the vast majority of the road trajectories estimated were within the road boundaries both with and without the filter. The last column captures the benefit of the filtered estimates. Amongst the 2.4% of the raw estimates that were *not* within

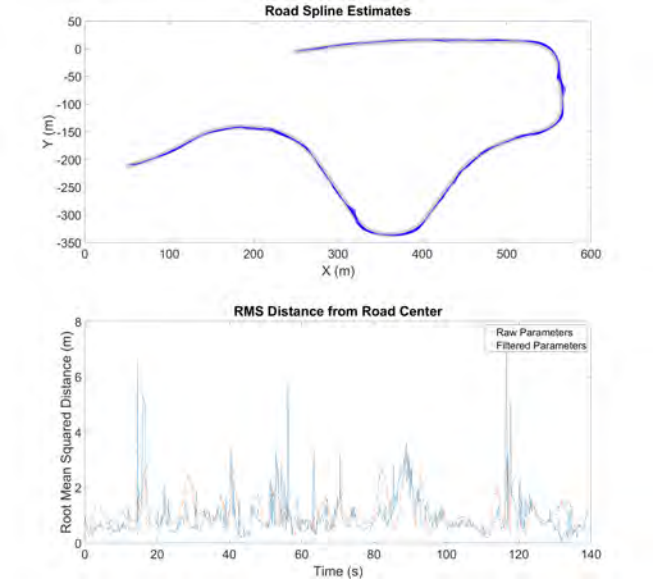

Fig. 6: Evaluation of 700 laser scans taken over a kilometer of driving in Devens, MA. Top: Each of the road splines estimated by the segmentation process at each time step. Bottom: The Root Mean Squared Distance evaluated along each estimated spline compared to the true road center.

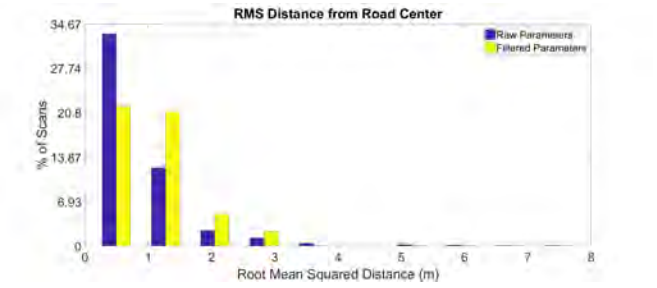

Fig. 7: Histogram of the evaluation of the RMS Distance from the Road center for each of the 700 scans evaluated. For a half-road-width of 3 meters, more than 97% of the predicted trajectories lie within the road boundaries.

the road boundary in the raw estimate, the average deviation beyond the road boundary was 1.39m. On the other hand, for the filtered parameters, $< 1\%$ of estimates crossed the road boundary, and those that did had on average, 0.1m distance. It is precisely for outliers such as these that necessitate a temporal filter to ensure the system can run continuously, and robustly, for an extended period of time.

C. Full-Scale Deployment

The system was evaluated at a test site in Devens, MA. The roads were single-lane, unmarked roads which do not have curbs or barriers at the boundaries (see Fig. 8). Note that although Fig. 6 shows that the planned trajectories occasionally deviated from the road. These deviations were typically far ahead and since the trajectories are replanned after the vehicle traverses only a short distance as described in V-A, the car was able to reliably follow the 1km path without any need for human intervention or any predefined map.

	Raw Estimate	Filtered Estimate
Mean RMS Distance from Center (m)	0.75	0.90
% of Scans Within Road Boundary	97.6	99.3
Mean RMS Distance beyond Road Boundary	1.39	0.10

TABLE I: The RMS distances over $n = 700$ segmented scans. The first row gives the mean RMS Distance from the road center over all the scans. The second gives the percentage of the estimates that did not cross over the road boundary, while the last row shows the mean RMS distance beyond the road boundary amongst those trajectories that did go beyond the boundary.

Fig. 8: Snapshots of the car operating at the test site in Devens, MA. It is a rural area where single-lane, unmarked roads, without curbs, are surrounded by forest.

VI. Conclusion

We have presented a novel and comprehensive framework for autonomous driving without detailed prior maps. Moreover, our method makes no assumptions about road markings and only minimal assumptions about road geometry. We have designed and incorporated a LiDAR-based local trajectory generation algorithm for traversing road segments using only an on-board sensor, which is robust to poor measurements in rural environments and very efficient. We have demonstrated the utility of the method on a full-scale autonomous car in a challenging rural setting.

References

[1] "Trading economics." URL: http://www.tradingeconomics.com/united-states/roads-paved-percent-of-total-roads-wb-data.html.

[2] J. Levinson and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Robotics: Science and Systems*, 2007.

[3] F. Naser, D. Dorhout, S. Proulx, S. D. Pendleton, H. Andersen, W. Schwarting, L. Paull, J. Alonso-Mora, M. H. A. Jr., S. Karaman, R. Tedrake, J. Leonard, and D. Rus, "A parallel autonomy research platform," in *2017 IEEE Intelligent Vehicles Symposium*, pp. 1–8, 2017.

[4] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 125–137, Apr 2001.

[5] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 5, pp. 4845–4851 Vol.5, April 2004.

[6] A. Ranganathan and F. Dellaert, "Online probabilistic topological mapping," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 755–771, 2011.

[7] M. Bosse, P. Newman, J. Leonard, and S. Teller, "Simultaneous localization and map building in large-scale cyclic environments using the atlas framework," *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2004.

[8] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 958–980, 2010.

[9] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid, "Rslam: A system for large-scale mapping in constant-time using stereo," *International Journal of Computer Vision*, pp. 1 – 17, June 2010.

[10] S. Anderson and T. D. Barfoot, "Towards relative continuous-time slam," in *2013 IEEE International Conference on Robotics and Automation*, pp. 1033–1040, May 2013.

[11] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *International Conference on Robotics and Automation*, 2010.

[12] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 176–183, Sept 2014.

[13] R. W. Wolcott and R. M. Eustice, "Fast lidar localization using multiresolution gaussian mixture maps," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2814–2821, May 2015.

[14] W. Liu, H. Zhang, B. Duan, H. Yuan, and H. Zhao, "Vision-based real-time lane marking detection and tracking," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pp. 49–54, Oct 2008.

[15] M. Aly, "Real time detection of lane markers in urban streets," in *2008 IEEE Intelligent Vehicles Symposium*, pp. 7–12, June 2008.

[16] J. C. McCall and M. M. Trivedi, "An integrated, robust approach to lane marking detection and lane tracking," in *IEEE Intelligent Vehicles Symposium, 2004*, pp. 533–537, June 2004.

[17] J. Byun, K.-i. Na, B.-s. Seo, and M. Roh, *Drivable Road Detection with 3D Point Clouds Based on the MRF for Intelligent Vehicle*, pp. 49–60. Cham: Springer International Publishing, 2015.

[18] K. Peterson, J. Ziglar, and P. E. Rybski, "Fast feature detection and stochastic parameter estimation of road shape using multiple lidar," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 612–619, Sept 2008.

[19] J. Leonard, J. How, S. Teller, *et al.*, *A Perception-Driven Autonomous Urban Vehicle*, pp. 163–230. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[20] Y. W. Seo and R. R. Rajkumar, "Detection and tracking of boundary of unmarked roads," in *17th International Conference on Information Fusion (FUSION)*, pp. 1–6, July 2014.

[21] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4885–4891, Oct 2016.

[22] I. Baldwin and P. Newman, "Laser-only road-vehicle localization with dual 2d push-broom lidars and 3d priors," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2490–2497, Oct 2012.

[23] I. Baldwin and P. Newman, "Road vehicle localization with 2d push-broom lidar and 3d priors," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2611–2617, May 2012.

[24] Z. J. Chong, B. Qin, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "Synthetic 2d lidar for precise vehicle localization in 3d urban environment," in *2013 IEEE International Conference on Robotics and Automation*, pp. 1554–1559, May 2013.

[25] C.-I. Vasile, J. Tumova, S. Karaman, C. Belta, and D. Rus, "Minimum-violation scltl motion planning for mobility-on-demand," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 1481–1488, IEEE, 2017.

[26] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-d road and relative ego-state recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 199–213, Feb 1992.