Fast Feature Detection and Stochastic Parameter Estimation of Road Shape using Multiple LIDAR

Kevin Peterson, Jason Ziglar, and Paul E. Rybski
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
kp@andrew.cmu.edu, jpz@cmu.edu, prybski@cs.cmu.edu

Abstract— **This paper describes an algorithm for an autonomous car to identify the shape of a roadway by detecting geometric features via LIDAR. The data from multiple LIDAR are fused together to detect both obstacles as well as geometric features such as curbs, berms, and shoulders. These features identify the boundaries of the roadway and are used by a stochastic state estimator to identify the most likely road shape. This algorithm has been used successfully to allow an autonomous car to drive on paved roadways as well as on off-road trails without requiring different sets of parameters for the different domains.**

I. Introduction

Developers of autonomous vehicles must overcome significant challenges before these vehicles can operate around human drivers. In urban environments autonomous cars will be required to follow complex traffic rules regarding merging and queuing, navigate in close proximity to other drivers, and safely avoid collision with pedestrians and fixed obstacles near or in the road. Knowledge of the location and shape of the roadway near the autonomous car is fundamental to these behaviors.

Detection and tracking of urban roads poses a complex challenge. Urban roads are typically characterized by boundaries such as curbs, berms, shoulders, and other geometric features that differ in shape and texture from the relatively flat road surface. These geometric features are not always present, may be occluded by parked cars or traffic, and can vary significantly along the length of the road. Clutter outside of the road such as parked cars, telephone poles, trees and bushes often indicates where the road is *not*, but these indicators are not easy to incorporate into standard filtering routines. Painted lines also serve as good indicators for the presence of the lanes within the road, but are not necessarily always present. This is particularly true on suburban roads where often the only indication of the road is the curb or other geometric feature at the edge of the road.

With so much variability in the appearance of the road, it is tempting to build an *a priori* GPS-registered map of the road network. Given a map like this, a GPS-enabled autonomous vehicle could easily follow the road without actually doing any exterioceptive sensing. Unfortunately, this is not a feasible long-term solution. GPS measurements drift and can be lost altogether when the sky is occluded by tunnels, trees, or nearby buildings. In addition to these challenges, keeping an up-to-date map of the world would be expensive and would require significant infrastructure. Temporary re-routing of traffic due to construction would also foil such a map-following solution. Finally, such road network maps could not realistically contain perfect information of all roads (particularly in rural areas) at all times.

In this paper, we describe a fast geometric road tracking system which can estimate the shape of both urban and unimproved roads. The system implements a stochastic state estimator to find the best fit road shape given a dense set of geometric features. The geometric features, derived from 2D and 3D LIDAR, include maps of nearby "lethal" obstacles, curbs, berms, and shoulders. Obstacles are detected by segmenting a point cloud into ground and non-ground clusters. Objects that protrude more than 0.5 meters from the ground are declared lethal. Curbs, berms, and shoulders are detected using a novel wavelet-based edge detector described in detail below. These features are transformed using several efficient dense point evaluators into an objective function which is optimized to find the "best fit" road using a particle filter state estimator. The tracker was developed as part of the system that won the 2007 DARPA Urban Challenge[1] and has allowed the vehicle to drive autonomously on both paved and dirt roads.

This paper is structured as follows. Section II lists related work. Section III defines the road tracking problem. Section IV details the sensor processing algorithms. Section V discusses the objective functions. Section VI describes the particle filter. Experimental results for both on-road and off-road trials are presented in Section VII. The paper is concluded in Section VIII.

II. Related Work

Nearly two decades of research on autonomous cars that are able to detect and follow roads precedes this work. We do not have space to reference all work in this area, but rather hope to do the diverse field justice by providing a broad sampling of the different approaches.

Some of the earliest successful vehicles such as NAVLAB used computer vision algorithms for finding the shapes of roads and intersections in residential streets [19]. Work in

[1] http://www.darpa.mil/GRANDCHALLENGE/

autonomous vehicle and lane following expanded to high-speed driving on highways where vehicles used cameras to track lanes based on the detection of painted lines [20]. Research in vision-based road detection has expanded to include handling of tracking both marked and unmarked roadways [3], as well as cases where roads split and join [15].

Single monocular cameras are typically more cost effective than laser range finders but each has their advantages and disadvantages. Cameras can capture visual edges and color and therefore have been used successfully for texture and color-based road following [14]. Three dimensional structure can be inferred from the movement of the vehicle [8]. However, cameras are passive sensors and are limited by lighting and shadows. Typically, cameras are better suited for tracking roads where specific features such as road lines can be detected [13], [25] directly.

Lasers have the advantage of being able to directly measure distances to objects but until only recently have been limited to only observing a narrow plane of data at a time. Regardless of visual appearance and lighting, lasers can detect and track physical boundaries along roads. Thus, they are well suited for off-road navigation and tracking [6]. Our approach uses geometry-based detection using dense LIDAR points to both find obstacles as well as road boundary features.

One method for detecting road shape utilizes the geometric features bounding a road surface. Geometric features–such as curbs, berms, ditches, and walls–bound a road with changes in slope or height of the ground surface. A scanning laser range-finder can sample the road geometry with feature extraction using edge detection to recover the road edges. Road detection has previously been performed through geometric cues for robots [26], [11]. In our application, however, the amount of data to process is significantly greater than before. As a result, the system must implement a fast, efficient, and robust edge detector.

For edge detection, an algorithm based around the Haar wavelet transform, is presented. Given the broad range of geometric features possible for road edges, edge detection must robustly detect both large scale, gentle edges (e.g. berms) and small scale, sharp edges (e.g. curbs.) Wavelets provide an efficient, robust method for performing multi-scale analysis of signals, resulting in a unified approach to detection road edges at multiple scales [17], [7]. In this context, the Haar wavelet can be seen as a multi-scale numerical derivatives, which is ideal for edge detection [10], [22].

Sensor fusion approaches to road following that include vision systems as well as laser or structured light system have been used successfully in an attempt to build on the strengths of both sensor modalities. This has been demonstrated successfully for both on-road [4] and off-road [21] tracking. Other sensors such as radar have successfully been fused with visual data [16] for finding and tracking a road.

We use a state representation and particle filter approach that is very similar to the work of [23]. A third degree polynomial is capable of representing the sorts of typical road curvatures. This parameterization of this curve is a Taylor series expansion of a clothoid curve [1] which has the benefit of specifying a vehicle trajectory that has constant velocity and yaw (steering) rate. However, our primary contributions are a fast mechanism for evaluating very dense laser datasets that allow the robot to detect road shape under a wide variety of on- and off-road conditions.

III. Road Detection Overview

Internally, roads are modeled as continuous entities containing a set of one or more lanes. In the Urban Challenge race, the organizers explicitly specified the location of all intersections ahead of time. Thus, our algorithm is based on the assumption that intersections are known *a priori*. The number of lanes in the road was also provided to use ahead of time, although it would be straightforward to track the number of lanes as an additional state.

We represent the road shape by the Taylor series expansion of a clothoid [23]:

$$y(x) = y_0 + tan(\phi)x + C_0\frac{x^2}{2} + C_1\frac{x^3}{6} \qquad (1)$$

where y_0 is the lateral offset between the vehicle and the road, ϕ is the angle of the road relative to the vehicle, and x is the longitudinal distance from the vehicle IMU to the road. The polynomial coefficients, C_0 and C_1, represent the curvature and curvature rate of the road. Additionally, the width of the road W is independently tracked and estimated.

The observation model for tracking trails is complicated and non-linear. Obstacles may lie at the edge of the road or not. Bushes may be scattered around the road indicating locations where the road could not possibly lie. Roads are, by definition, on average smoother than the surrounding terrain, but the absolute "smoothness" can change across the road. In general, it is difficult to transform this description of a road into a positive indication of the location of the road center as is required by standard linear filtering techniques.

For these reasons, we formulate our tracker as a particle filter that attempts to satisfy an objective function considering several features indicative of both trails and roads. The objective function incorporates three features: 1) The number of obstacles within the proposed road, 2) The number of laser edge detections within the road, and 3) The distance between the edge of the proposed road and the closest edge detection. These features are measured along the proposed road at regularly spaced intervals. The overall cost of the proposed road is a weighted sum of these features. See section VI for more details about the particle filter.

IV. Sensor Processing

Raw LIDAR sensor data is converted into two specific forms that are handled independently by the road shape estimator. The first form is an obstacle map which contains information about lethal obstacles that the vehicle must avoid. The second form is a feature map which contains information about small geometric features which delimit the shape and curvature of the road. These features include curbs, berms, shoulders, and ditches.

A. Obstacle Maps

Obstacle maps encode multi-valued cells which describe different types of potentially lethal obstacle. Each cell in the map represents one of five possible values, $M_{ij} \in \{Unseen, Empty, Small, Medium, Lethal\}$. The maps are centered on the vehicle and range in size from 100 meters to 200 meters with quarter meter resolution. Obstacle maps are generated by each individual LIDAR system and are fused together before being processed by the road shape estimator. During the fusion process, if two maps disagree on the characterization of a particular cell, the map fusion algorithm chooses the worst of the two classifications for that cell. Figure 1 shows an example of obstacles surrounding the robot found while traveling down a road.

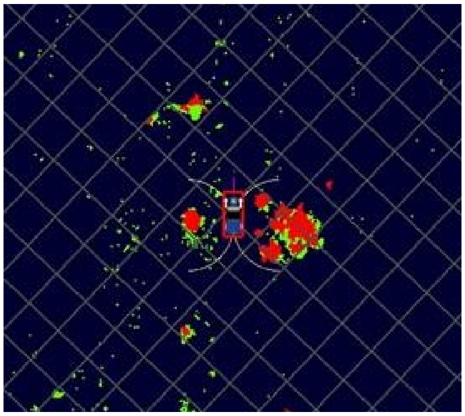


Fig. 1: Obstacle map representation of the area around the robot. Red pixels represent “Lethal” obstacles that must be avoided at all costs. Green pixels are smaller obstacles that have a lower cost associated with them. Black areas are pixels that have no cost.

B. Wavelet Curb Detection

The variability in geometric road boundaries complicates robust detection. Road edges are defined by changes in geometry of the ground surface - change in height (curbs), change in slope (berms), or by the presence of impassable barriers (walls). These cues can also exist as negative obstacles, such as ditches or culverts. All of these changes in ground surface are measurable via LIDAR.

The curb detection algorithm operates on a few base assumptions. First, the road surface is assumed to be relatively flat and slow changing. Second, the LIDAR returns from a given laser are treated as single scan lines through the world, slicing across the world in front of the robot at some angle. This is in contrast to techniques which treat the data as a cloud of points to be processed in three dimensions. These assumptions are used throughout the algorithm to simplify the feature extraction. With these assumptions in mind, the algorithm breaks down into three main steps: preprocessing, wavelet-based feature extraction, and post-processing.

In preprocessing, data is filtered to eliminate artifacts due to occlusion, and resampled to meet the input criteria of 2^n points for wavelet transform [5]. First, points within some threshold distance of neighboring points are clustered. Clusters with too few points are immediately removed. Second, the distance between consecutive remaining points is evaluated and points whose spacing is smaller than a threshold value are labeled dense. Finally, the dense points are linearly resampled to the next highest power of 2.

The preprocessing steps eliminate several artifacts in the data which we have found to be undesirable. The clustering step removes thin objects such as telephone poles and cones. These objects are typically found in the obstacle map (and so would be redundant) and do not constitute an edge of the road. The check for point density eliminates large steps in the data caused by discontinuities in the laser data. The linear resampling is necessary for the wavelet transform described below.

Feature extraction for curb data is based on the discrete wavelet transform using the Haar wavelet. The Haar wavelet is defined by a mother wavelet and scaling function:

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \le t < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \le t < 1, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$\varphi(t) = \begin{cases} 1 & \text{if } 0 \le t < 1 \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

These functions are used to generate a series of waveforms, known as daughter wavelets, which are convolved with the input signal. In the case of the Haar wavelet, the result of these convolutions are scaled average slopes within various windows of the data [5]. Importantly, these windows are the recursive subdivisions of the input signal - the first window contains the entire signal, the next two windows contain the left and right halves of the signal, and so forth. The windows are grouped into detail levels based on size, with the first detail level occurring at the coarsest scale.

In order to process the data, the scan line is transformed into the local frame, and the Haar wavelet transform is performed over the height values of the points, resulting in an array of coefficients, y. These coefficients are then fed into the following algorithm.

1) Collect coefficients for the current detail level, i.
2) Label each coefficient with the label of the parent detail level, $i-1$.
3) Calculate $\hat{y}_{road}$, using the previously propagated labels
4) Re-label coefficients by absolute distance from $\hat{y}_{road}$, where the distance threshold for level i given as d_i. That is, label points according to:

$$class(y[n], i) = \begin{cases} 1 \text{ if } |y[n] - \hat{y}_{road}| \ge d_i \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

5) Step to detail level $i+1$

This algorithm starts with the largest window, assuming $\hat{y}_{road} = 0$, and continues until it reaches the smallest window. This results in the points being split into two classes: road points (class 1) and non-road points. A single iteration

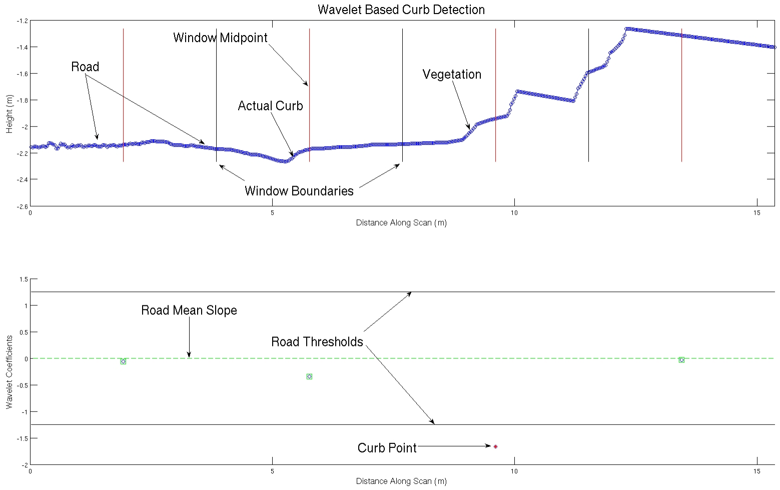


Fig. 2: Single frame of the feature extraction algorithm. The top frame contains the original height signal from a single LIDAR scan. The black vertical lines bound windows defined by a single detail level of the wavelet transform. The red vertical lines define the midpoints of these windows. The bottom frame shows the resulting wavelet coefficient for each window, as well as the classification after thresholding.

of this algorithm, with both the input signal and wavelet coefficients, is shown in Figure 2.

Post-processing performs a few simple steps to eliminate false positives due to sparse data. The labeling performed in preprocessing is used to find sparse regions classified as curb points. In these regions, the two closest dense points are found, and the point closest to the robot is labeled as a curb point. This heuristic is based on the fact that the LIDAR sensors are downward looking, which means that higher surfaces (e.g. curbs, berms, walls) will appear closer to the robot. In addition to this, the sparse points are deleted from the list of classified points. Finally, the scan line is iterated over one final time, tagging points where the class changes as edge points. This final data product of the curb detection is a list of points in global coordinates, classified as either road or non-road, and as edge transitions. Figure 3 shows a map of the curb points identified around the robot while it travels down a road.

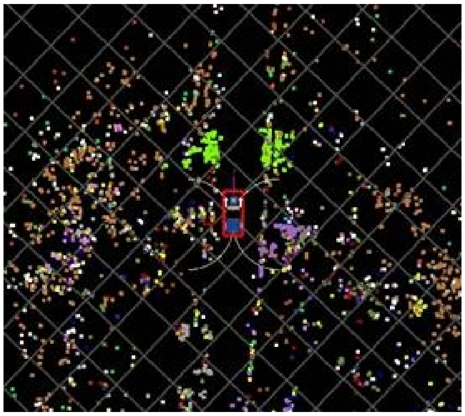


Fig. 3: Curb point map representation of the area around the robot. Colored pixels represent curbs as seen by the different LIDARs where the colors represent which LIDAR was responsible for identifying that point as a curb.

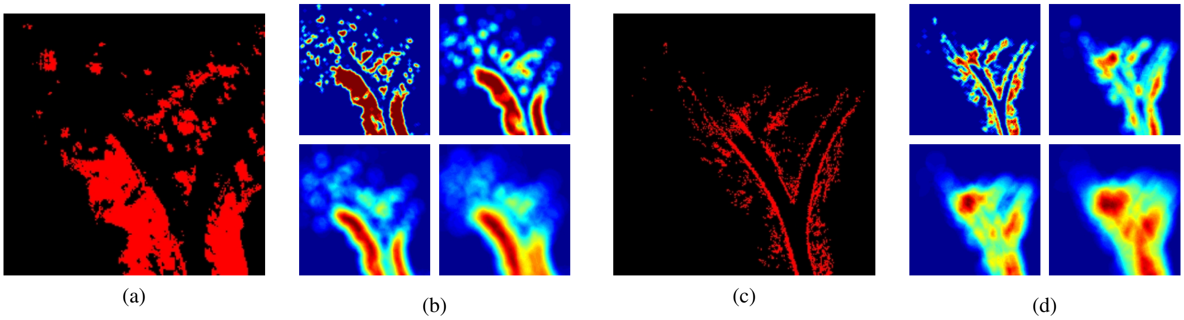
V. Objective Function

The obstacle and feature maps generated from LIDAR data allow the road estimation system to evaluate potential road curves based on the road polynomial defined in Equation 1. In order to determine the quality of the fit of a potential road curve, we have developed a fast recursive convolution engine that will evaluate the likelihood of the curve's fit to the data. The objective function that we have defined makes use of both 2D convolution filters as well as a distance transform.

A. *Fast Convolutions*

Terrain roughness and obstacle density are good indications of the presence and quality of a potential road shape. These features are computed using convolutions of the incoming laser data. If we define obstacle density as the sum of the number of obstacles inside the proposal road and terrain roughness as a scaled sum of the number of edge detections within the road, then the computation of the score of the proposed road particle involves the convolution of the road shape with a binary map containing the relevant statistics (obstacle or edge detection).

These convolutions are computationally intense, requiring subdivision of the road shape into cells followed by the comparison of the subdivided shape with the feature map. This bears a resemblance to the configuration space (C-space) expansion problem that is often considered in path planning [24]. In practice, the C-space expansion problem



(a) (b) (c) (d)

Fig. 4: Obstacle and curb maps are thresholded before being passed to the convolution filters. (a) Obstacle map, binarized. (b) Obstacle map expanded to 2.6, 6.1, 10.1, and 14.6 meters resp. (c) Curb map, binarized. (d) Curb map expanded to 2.6, 6.1, 10.1, and 14.6 meters resp.

is solved by approximating the extent of the robot with a disk centered on the robots pose. The map can then be transformed by first expanding all obstacles by a disk with radius equal to the size of the robot. Planning then occurs assuming the robot is a point. The C-space expansion can be performed efficiently using convolution of the map with a disk-shaped kernel.

Unfortunately, roads and trails vary significantly in width, so the tracking case is not as easily solved. In order to handle the varying widths, the features maps are C-space expanded by the full set of possible road widths and the score of a road shape with a given width is the sum along the center-line of the road in the map that corresponds to the road width. These widths range from 2.2 meters to 16.2 meters at half meter spacings for a total of 15 convolutions per feature per frame. For a 10 Hz cycle rate, this requires 300 convolutions per second with kernels ranging from four to thirty-two cells square. If the convolutions were completely recomputed each cycle, then this would clearly be computationally infeasible.

In order to generate the convolution maps in real-time, it is necessary to recursively compute the maps and update only portions of the map that have changed. Suppose that we have two binary maps M_1 and M_2 such that M_1 and M_2 differ in only one pixel. The convolution of M_1 with any kernel differs from the convolution of M_2 with the same kernel only by the value of that kernel in the area of the difference. In other words, by the linearity of convolutions, the maps can be updated by convolving the difference of the two maps and adding (or subtracting) to get the final result.

To enable random access, the maps are stored as hash tables where the hash function is simply the global location of a cell modulo the size of the map. Hash collisions are resolved by overwriting old data with new. As new data arrives, new points are convolved with the kernel and the result of the convolution is written into the hash map. As a precaution, a maximum number of updates per cycle is computed and if the number of required updates is greater than this maximum, the updates are randomly rejected to ensure that the estimator continues to operate at its required rate.

The final score of a proposed road is the sum over sample points, $x_s^i \in X_i$, in the roughness map, M_r, and obstacle map, M_o, indexed by the width of the road, w^i:

$$c_o(X_i) = \sum_{s=0}^{S} M_o(x_s^i, w^i) \tag{5}$$

$$c_r(X_i) = \sum_{s=0}^{S} M_r(x_s^i, w^i) \tag{6}$$

Figure 4 illustrates the signal response from an example obstacle and curb point map taken from the same location. The maps are thresholded to create binary versions before the convolutions are applied to them.

B. Distance Transforms

In order to draw the road shape towards curbs, we incorporate an attractive function that is computed as minus the square of the distance between the edge of interest and the proposal road shape. In order to compute this value, the curb points are inserted into a 2D map and a Distance Transform [9] of the map is computed. This technique is common in image-based particle filters, see e.g. [18]. An example transform is shown in Figure 5.

In order to compute the objective, the expected location of the road edge at S sample points is computed by offsetting the sample point $x_s^i \in X_i$ by half the width of the road times the normal to the direction of the road. This position is evaluated in the distance map, D. The edge of the road is expected to lie along a curb or berm in many cases, but breaks in the road edge are common on both urban roads and dirt roads. In order to prevent the estimator from following local, sudden changes in the road, the overall score for the road shape is computed as the average distance to the closest 70% of the samples:

$$c_{cl}(X_i) = \sum_{s \in closest(S_l)} D(x_s^i - o_e^i) \tag{7}$$

$$c_{cr}(X_i) = \sum_{s \in closest(S_r)} D(x_s^i + o_e^i) \tag{8}$$

Where o_e^i is the offset between the road center-line and the expected curb location, and S_l and S_r are the set of left and right edge point samples.

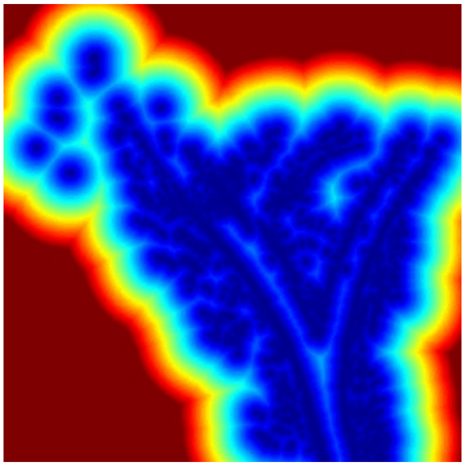


Fig. 5: Distance transform applied to the binary curb map shown in Figure 4. The image is thresholded at a distance of 10 meters to enhance the local detail in the map.

C. Objective Function

The resulting objective function balances the roughness and obstacle density of the road against the distance between the proposal density and the road edges. As a result, the tracker converges on the least "costly" road, where cost is defined as the sum of obstacle density and terrain roughness that best matches the positive indications of the road.

The final observation likelihood is the exponential of a weighted sum of the individual objectives outlined above. Let $\vec{\boldsymbol{\theta}}$ be the vector of weights used to mix the objectives, $C(X)$ be the scalar value of the objective function, and $\vec{\mathbf{C}}(X)$ be a vector containing the individual objectives:

$$\vec{\mathbf{C}}(X) = [c_o(X), c_r(X), c_{cl}(X), c_{cr}(X)]^T \tag{9}$$

$$C(X_{i_t}) = \vec{\boldsymbol{\theta}}^{\boldsymbol{T}} \vec{\mathbf{C}}(X_{i_t}) \tag{10}$$

Where X_{i_t} is the *ith* proposed road state at time t.

VI. State Estimation

Particle filters are a class of Bayesian filter that can track non-Gaussian probability distributions. These filters represent the state space as a set of weighted point masses appropriately called "particles." At each time step, the particles are updated using a deterministic process model and then perturbed to account for process noise. The particles are then re-weighted as a function of the current observation and resampled if necessary to improve numerical stability. For a more thorough description of particle filtering, please see [2].

Several techniques exist for producing a final estimate including the robust mean or mode estimation. Because of the non-parametric representation, particle filters make few assumptions about the measurement and process models. As a result, the particle filter has been used successfully to solve several very challenging tracking problems including curve tracking in dense clutter [12] and highway road tracking [23].

Each particle stores a state vector representation of the road shape defined in Equation 1. The elements of the state vector store the parameters necessary to reproduce the road shape. The state vector is represented as: $[Y, \phi, C_0, C_1, W]$ where Y and ϕ are the origin and direction of the curve under the vehicle in a vehicle-centric coordinate frame (with $X = 0$), C_0 and C_1 are the curvature and rate of curvature respectively, and W is the road width. Figure 6 shows a snapshot of the particles while the robot is tracking a path.

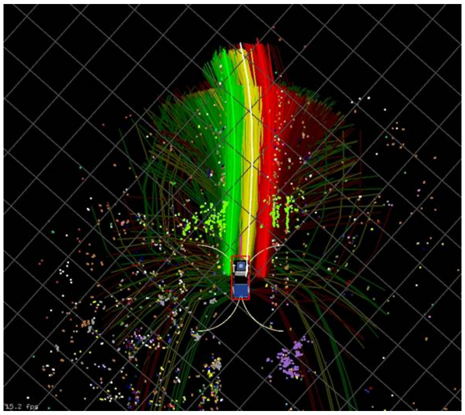


Fig. 6: Example particle density while the robot is tracking the shape of a dirt road. The green, yellow, and red lines are the left edges, centers, and right edges of the road shape estimates, respectively. Only curb points are shown in this image.

A. State Transition Model

The deterministic state evolution equations are as follows:

$$\begin{bmatrix} Y_{t+1} \\ \phi_{t+1} \\ C_{0_{t+1}} \\ C_{1_{t+1}} \end{bmatrix} = \begin{bmatrix} 1 & \Delta s & \frac{(\Delta s)^2}{2} & \frac{(\Delta s)^3}{6} \\ 0 & 1 & \Delta s & \frac{(\Delta s)^2}{2} \\ 0 & 0 & 1 & \Delta s \\ 0 & 0 & 0 & 0.99 \end{bmatrix} \begin{bmatrix} Y_t \\ \phi_t \\ C_{0_t} \\ C_{1_t} \end{bmatrix} + \begin{bmatrix} 0 \\ -\Delta\phi_t \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

Where Δs is the displacement of the vehicle along the curve of the trajectory from time step t to $t+1$. The C_1 update term embodies the assumption that the road will tend to straighten out over time which helps to bias the filter away from highly curving estimates.

B. Observation Model

The observation density is quite general and can be any function which produces a valid probability. Clearly, the choice of observation model has a drastic effect on the performance of the filter and we have found it to be one of the most important elements when designing a filter of this class.

Many possible observation models could be appropriate for road tracking. We chose to model the observation as an

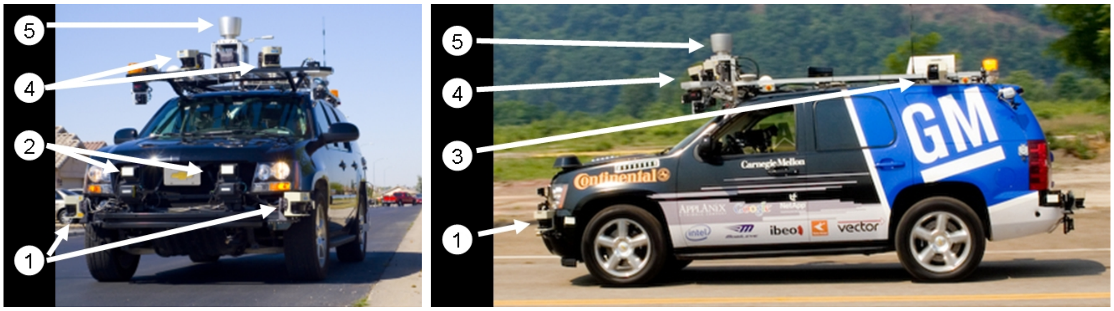


Fig. 7: An image of the test vehicle with complete sensor suite. 1. Laser range finder (SICK LMS 180) oriented to detect other vehicles, 2. RADAR, 3. Laser range finder (SICK LMS 90) oriented to detect lane markings. 4. Laser range finder (SICK LMS 90) oriented to detect obstacles and rough terrain, 5. 3D laser range finder (Velodyne) for long range obstacle and curb detection.

exponential density,

$$p(x|z) \propto e^{-C(x,z)} \tag{12}$$

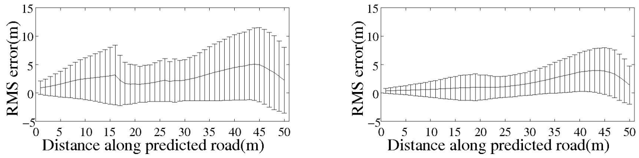
where $C(x,z)$ is a weighted sum over a set of representative features. This model has several attractive qualities including its relative simplicity in tuning.

VII. Experimental Results

Our tracker is implemented on "Boss", the Carnegie Mellon University DARPA Urban Challenge race vehicle. Boss is a 2007 Chevy Tahoe modified with a drive-by-wire system that controls the steering, brake, and accelerator. The vehicle is outfitted with a wide array of LIDARs and RADARs for detecting obstacles, moving traffic, and the detecting the road. The sensors relevant to road finding include: five single axis scanning SICK LIDARs mounted on the roof (3 forward-facing and two aimed on the left and right sides of the vehicle, respectively) and two-axis 360^o scanning Velodyne LIDAR also mounted on the roof. See Figure 7 for images of the sensor placements.

In this implementation, data for the wavelet feature detector comes from the LIDAR units attached to the vehicle. All the SICK data, and 25 (out of 64 total) lasers from the Velodyne provide short-, medium-, and long-range geometric data. This results in approximately 518,000 LIDAR returns a second, providing data up to 70 meters away. The curb detection algorithm presented processes this data in real time to provide curb features for higher level fusion. In contrast, the obstacle maps use all of the data from the Velodyne which is more than a million returns a second. The road shape estimator is designed to generate updates to the lane geometry in areas where the road is marked as sparse. These updates are generated at 10 Hz.

The road shape estimator was evaluated in two different domains. The first domain was a paved road with small curbs and few obstacles. The second was an off-road trail with a significant number of bushes and trees delimiting the edges of the traversable area. To determine the quality of the tracker, the root mean square error between the centerline of the road and the estimated center of the road was measured. The centerline of the road was measured by hand driving a high accuracy GPS unit down the center of the road.



(a) On-road error

(b) Trail error

Fig. 8: Error as a function of distance along the detected road for (a) on-road and (b) off-road. The errorbars show one standard deviation from the mean error.

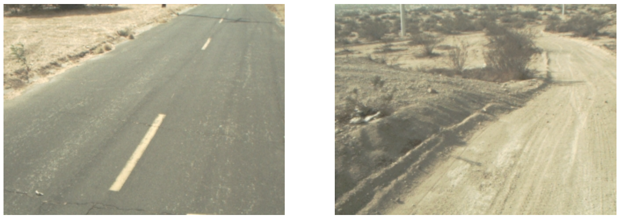
A. On Road

On-road performance was measured on a poorly marked asphalt road with low curbs at the Castle Air Force Base in California. The curbs are, in some sections of the course obscured by leaves. A section of the road is bounded on one side by a chain-link fence, which the tracker uses as a cue. Figure 8a shows error on-road as a function of lookahead.

B. Off Road

Off-road performance was measured on power-line access trails in Victorville, California near the race site. The course is a desert trail with berms on either side and bushes near the trail. The trail is reasonably well-maintained, but does have significant ruts, washboard, and changes in elevation. Figure 8b shows error as a function of lookahead.

It is interesting to note that the on-road error is larger than the off-road error. This is due to the fact that the off-road course is more well-defined by geometric features than the on-road course. Figure 9 shows a typical section of road from the on- and off-road datasets. These images show the differences in the features available in the two sets. Whereas the off-road test set is well-defined by the geometric features surrounding the road, the on-road test set is defined only by the curb and a slight increase in roughness due to grass.



(a) A representative image from the on-road test set

(b) A representative image from the off-road test set

Fig. 9: Images from (a) the on-road test set and (b) the off-road test set. The on-road test set has low (5-8 cm) curbs. The off-road test set has significantly more geometric information - berms are pronounced, bushes surround the road, and the road edge is visible. The texture off of the road in the on-road test set is slightly greater than on the road. The on-road test set is also a wider road than the off-road test set.

VIII. Conclusions

This paper describes a road detection system for an autonomous vehicle which fuses data from multiple LIDAR to identify geometric features that define the boundaries of a roadway. We have presented a method for using a wavelet transform for identifying features such as curbs, berms, and shoulders. A fast convolution and distance transform algorithm have also been presented which are used to evaluate the quality of fit of a given road shape to the data. A particle filter estimator with a combination of uniform and deterministic resampling is used to search the likelihood space to optimize the parameters. This algorithm has bee successfully used by an autonomous vehicle to navigate both paved roads and unpaved trails.

Acknowledgments

This work would not have been possible without the dedicated efforts of the Tartan Racing team and the generous support of our sponsors including General Motors, Caterpillar, and Continental. This work was further supported by DARPA under contract HR0011-06-C-0142.

References

[1] Nicholas Apostoloff. Vision based lane tracking using multiple cues and particle filtering. Master's thesis, Australian National University, February 2005.

[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

[3] Romuald Aufrere, Roland Chapuis, and Frederic Chausse. A fast and robust vision based road following algorithm. In *Proceedings of IEEE Intelligent Vehicles Symposium*, 2000.

[4] Romuald Aufrere, Christoph Mertz, and Charles Thorpe. Multiple sensor fusion for detecting location of curbs, walls, and barriers. In *Proceedings of IEEE Intelligent Vehicles Symposium*, 2003.

[5] C.K. Chui, editor. *An Introduction to Wavelets*. Academic Press, San Diego, 1992.

[6] Lars B. Cremean and Richard M. Murray. Model-based estimation of off-highway road geometry using single-axis ladar and inertial sensing. In *Proceedings of IEEE International Conference on Robotics and Automation*, May 2006.

[7] Ingrid Daubechies. *Ten Lectures on Wavelets (CBMS-NSF Regional Conference Series in Applied Mathematics)*. Soc for Industrial & Applied Math, December 1992.

[8] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), February 1992.

[9] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. Technical Report TR2004-1963, Cornell, September 2004.

[10] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69:331–371, 1910.

[11] T. Hong, T. Chang, C. Rasmussen, and M. Shneier. Road detection and tracking for autonomous mobile robots. In *Proceedings of SPIE Aerosense Conference*, volume 4715, Orlando, Florida, April 2002.

[12] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[13] Karl Kluge. Extracting road curvature and orientation from image edge points without perceptual grouping into features. In *Proceedings of Intelligent Vehicles Symposium*, 1994.

[14] Karl Kluge and Chuck Thorpe. The yarf system for vision-based road following. *Mathematical and Computer Modelling*, 22(4-7):213–233, August 1995.

[15] Raphael Labayrade, Jerome Douret, and Dider Aubert. A multi-model lane detector that handles road singularities. In *Proceedings of IEEE Transportation Systems Conference*, September 2006.

[16] Bing Ma, Sridhar Lakshmanan, and Alfred O. Hero. Simultaneous detection of lane and pavement boundaries using model-based multisensor fusion. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):135–147, September 2000.

[17] Stephane Mallat and Sifen Zhong. Characterization of signals from multiscale edges. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(7):710–732, 1992.

[18] P. Perez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of IEEE Special Issue on Sequential State Estimation*, 92(3):495–513, March 2004.

[19] Dean Pomerleau. Neural network based autonomous navigation. In Charles Thorpe, editor, *Vision and Navigation: The CMU Navlab*, pages 83–92. Kluwer Academic Publishers, 1990.

[20] Dean Pomerleau. Ralph: rapidly adapting lateral position handler. In *Proceedings of the Intelligent Vehicles 1995 Symposium*, pages 506–511, Detroit, MI, September 1995.

[21] Christopher Rasmussen. A hybrid vision + ladar rural road follower. In *ICRA*, pages 156–161, 2006.

[22] Ming-Yu Shih and Din-Chang Tseng. A wavelet-based multiresolution edge detection and tracking. *Image Vision Comput.*, 23(4):441–451, 2005.

[23] Ben Southall and C. J. Taylor. Stochastic road shape estimation. In *Proceedings of the International Conference on Computer Vision*, volume 1, pages 205–212, Vancouver, BC, Canada, July 2001.

[24] Anthony (Tony) Stentz and Martial Hebert. A complete navigation system for goal acquisition in unknown environments. In *Proceedings 1995 IEEE/RSJ International Conference On Intelligent Robotic Systems (IROS '95)*, volume 1, pages 425 – 432, August 1995.

[25] Yu Tianhong, Wang Rongben, Jin Lisheng, Chu Jiangwei, and Guo Lie. Lane mark segmentation method based on maximum entropy. In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, September 2005.

[26] W.S. Wijesoma, K.R.S. Kodagoda, , and A. P. Balasuriya. Road boundary detection and tracking using ladar. *IEEE Transactions on Robotics and Automation*, 20(3), June 2004.