Multi-Sensor Lane Finding in Urban Road Networks

Albert S. Huang[†]   David Moore[†]   Matthew Antone[⋆]   Edwin Olson[†]   Seth Teller[†]

[†]MIT Computer Science and Artificial Intelligence Laboratory   [⋆]BAE Systems Advanced Information Technologies
Cambridge, MA 02139   Burlington, MA 01803
{ashuang,dcm,eolson,teller}@mit.edu   matthew.antone@baesystems.com

Abstract— This paper describes a system for detecting and estimating the properties of multiple travel lanes in an urban road network from calibrated video imagery and laser range data acquired by a moving vehicle. The system operates in several stages on multiple processors, fusing detected road markings, obstacles, and curbs into a stable non-parametric estimate of nearby travel lanes. The system incorporates elements of a provided piecewise-linear road network as a weak prior.

Our method is notable in several respects: it estimates multiple travel lanes; it fuses asynchronous, heterogeneous sensor streams; it handles high-curvature roads; and it makes no assumption about the position or orientation of the vehicle with respect to the road.

We analyze the system's performance in the context of the 2007 DARPA Urban Challenge. With five cameras and thirteen lidars, it was incorporated into a closed-loop controller to successfully guide an autonomous vehicle through a 90 km urban course at speeds up to 40 km/h amidst moving traffic.

Fig. 1. Our system uses many asynchronous heterogeneous sensor streams to detect road paint and road edges (yellow) and estimate the centerlines of multiple travel lanes (cyan).

I. Introduction

The road systems of developed countries include millions of kilometers of paved roads, of which a large fraction include painted lane boundaries separating travel lanes from each other or from the road shoulder. For human drivers, these markings form important perceptual cues, making driving both safer and more time-efficient [12]. In mist, heavy fog or when a driver is blinded by the headlights of an oncoming car, lane markings may be the principal or only cue enabling the driver to advance safely. Moreover, public-safety officials use the number, color and type of lane markings to encode spatially-varying traffic rules, for example no-passing regions, opportunities for left turns across oncoming traffic, regions in which one may (or may not) change lanes, and preferred paths through complex intersections.

Even the most optimistic deployment scenario for autonomous vehicles assume the presence of massive numbers of human drivers for the next several decades. Given the centrality of lane markings to public safety, it is clear that they will continue to be maintained indefinitely. Thus autonomous vehicle researchers, as they design self-driving cars, may assume that lane markings will be encountered with significant frequency.

We define the lane finding problem as divining, from live sensor data and (when available) prior information, the presence of one or more travel lanes in the vehicle's vicinity, and the semantic, topological, and geometric properties of each lane. By semantic properties, we mean the lane's travel sense and the color (white, yellow) and type (single, double, solid, dashed) of each of its boundaries. By topological properties, we mean the connectivity of multiple lanes in regions where lanes merge, split, terminate, or start. The term geometric properties is used to denote the centerline location and lateral extent of the lane. This paper focuses on detecting lanes where they exist, and the determination of geometric information for each detected lane (Figure 1). We infer semantic and topological information in a limited sense, by matching detected lanes to edges in an annotated input digraph representing the road network.

Aspects of the lane finding problem have been studied for decades in the context of autonomous land vehicle development [5, 17] and driver-assistance technologies [8, 1, 2]. McCall and Trivedi provide an excellent survey [11]. Lane finding systems intended to support autonomous operation have typically focused on highway driving [5, 17], where roads have low-curvature and prominent lane markings, rather than on urban environments. Previous autonomous driving systems have exhibited limited autonomy in the sense that they required a human driver to "stage" the vehicle into a valid lane before enabling autonomous operation, and to take control whenever the system could not handle the required task, for example during highway entrance or exit maneuvers [17].

Driver-assistance technologies, by contrast, are intended as continuous adjuncts to human driving; one common class of such systems, lane departure warning (LDW) systems, is designed to alert the human driver to an imminent (unsignaled) lane departure [13, 10, 15]. These systems typically assume

that a vehicle is in a highway driving situation and that a human driver is controlling the vehicle correctly, or nearly so. Highways exhibit lower curvature than lower-speed roads, and do not contain intersections. In vehicles with LDW systems, the human driver is responsible for selecting an appropriate travel lane, is assumed to spend the majority of driving time within such a lane, is responsible for identifying possible alternative travel lanes, and only occasionally changes into such a lane. Because LDW systems are essentially limited to providing cues that assist the driver in staying within the current lane, achieving fully automatic lane detection and tracking is not simply a matter of porting an LDW system into the front end of an autonomous vehicle.

Clearly, in order to exhibit safe, human-like driving, an autonomous vehicle must have good awareness of all other nearby travel lanes. In contrast to prior lane-keeping and LDW systems, this paper presents a lane finding system suitable for guiding a fully autonomous land vehicle through an urban road network. In particular, our system is distinct from previous efforts in several respects: it attempts to detect and classify all observable lanes, rather than just the single lane occupied by the vehicle; it operates in the presence of complex road geometry, static hazards and obstacles, and moving vehicles; and it uses prior information (in the form of a topological road network with sparse geometric information) when available.

The apparent difficulty of matching human performance on sensing and perception tasks has led some researchers to investigate the use of augmenting roadways with a *physical infrastructure* amenable to autonomous driving, such as magnetic markers embedded under the surface of the road [18]. While this approach has been demonstrated in limited settings, it has yet to achieve widespread adoption and faces a number of drawbacks. First, the cost of updating and maintaining hundreds of thousands of miles of roadway is highly prohibitive. Second, the danger of autonomous vehicles perceiving and acting upon a different infrastructure than human drivers do (magnets vs. visible markings) becomes very real when one is modified and the other is not, whether through accident, delay, or malicious behavior.

Advances in computer networking and data storage technology in recent years have brought the possibility of a *data infrastructure* within reach. In addition to semantic and topological information, such an infrastructure might also contain fine-grained road maps registered in a global reference frame; advocates of these maps argue that they could be used to guide autonomous vehicles. We propose that a data infrastructure is useful for topological information and sparse geometry, but reject relying upon it for dense geometric information.

While easier to maintain than a physical infrastructure, a data infrastructure with fine-grained road maps might still become "stale" with respect to actual visual road markings. Even for human drivers, mapping staleness, errors, and incompleteness have already been implicated in accidents in which drivers trusted too closely their satellite navigation systems, literally favoring them over the information from their own senses [3, 16]. Static fine-grained maps are clearly
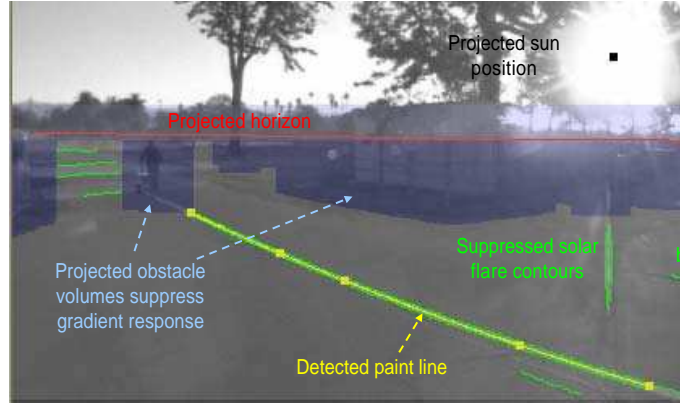

Fig. 2. Use of absolute camera calibration to project real-world quantities into the image.

not sufficient for safe driving; to operate safely, in our view, an autonomous vehicle must be able to use local sensors to perceive and understand the environment.

The primary contributions of this paper are:

- A method for estimating multiple lanes of travel in a typical urban road network using only information from local sensors;
- A method for fusing these estimates with a weak prior, such as that derived from a topological road map with sparse metrical information;
- Methods for using monocular cameras to detect road markings; and
- Multi-sensor fusion algorithms combining information from video and lidar sensors.

We also describe our method's failure modes, and describe possible directions for future work.

II. APPROACH

Our approach to lane finding involves three stages. In the first stage, the system detects and localizes painted road markings in each video frame, using lidar data to reduce the false positive detection rate. A second stage processes the road paint detections along with lidar-detected curbs [6] to estimate the centerlines of nearby travel lanes. Finally, the detected centerlines output by the second stage are filtered, tracked, and fused with a weak prior to produce one or more non-parametric lane outputs.

Separation of the three stages provides simplicity, modularity, and scalability. Specifically, we are able to experiment with each stage independently of the others and easily substitute different algorithms for each stage. For example, we experimented with and ultimately used two separate algorithms in parallel for detecting road paint, both of which are described below. By introducing sensor-independent abstractions of environmental features, we are able to scale to many heterogeneous sensors.

A. Absolute Camera Calibration

Our road-paint detection algorithms assume that GPS and IMU navigation data are available of sufficient quality to
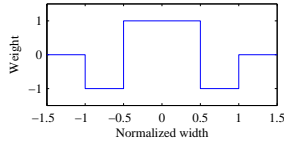
Fig. 3. The shape of the one-dimensional kernel used for matching road paint. By applying this kernel horizontally we detect vertical lines and vice versa. The kernel is scaled to the expected width of a line marking at a given image row and sampled according to the pixel grid.

(a) Original Image

(b) Filtered Image
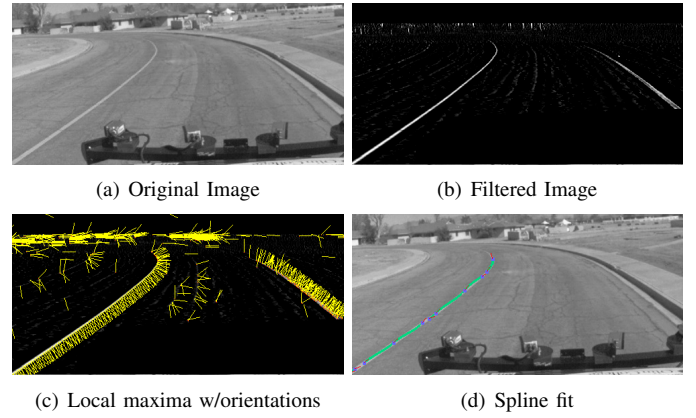
(c) Local maxima w/orientations

(d) Spline fit

Fig. 4. Our first road paint detector: (a) The original image is (b) convolved with a matched filter at each row (horizontal filter shown here). (c) Local maxima in the filter response are enumerated and their dominant orientations computed. The figure depicts orientation by drawing the perpendiculars to each maximum. (d) Nearby maxima are connected into cubic hermite splines.

correct for short-term variations in vehicle heading, pitch, and roll during image processing. In addition, the intrinsic (focal length, center, and distortion) and extrinsic (vehicle-relative pose) parameters of the cameras have been calibrated ahead of time. This "absolute calibration" allows preprocessing of the images in several ways (Figure 2):

- The horizon line is projected into each image frame. Only pixel rows below this line are considered for further processing.
- Our lidar-based obstacle detector supplies real-time information about the locations of obstructions in the vicinity of the vehicle [6]. These obstacles are projected into the image and their extents masked out as part of the paint detection algorithms, an important step in reducing false positives.
- The inertial data allows us to project the expected location of the ground plane into the image, providing a useful prior for the paint-detection algorithms.
- False paint detections caused by lens flare can be detected and rejected. Precise knowledge of the date, time, and Earth-relative vehicle pose allow computation of the solar ephemeris; line estimates that point toward the sun in image coordinates are then removed.

B. Road Paint Detection using Matched Filters

This section describes the first of two vision algorithms we use for detecting painted lines on the road. For each camera, we run a dedicated process that detects road paint and outputs a list of candidate line markings for each frame. These candidates are expressed as cubic hermite splines, which have the convenient property that the spline passes through each of the control points. Each frame is considered independently from the others; cross-frame tracking techniques could be used to improve the result.

The first step in our image processing pipeline is to construct matched one-dimensional filters, tuned to the expected width of a painted line marking at each row of the image. We consider two types of lines: Those that extend roughly away from the car towards the horizon and those that run transverse to the line of sight. The former is detected by a horizontal kernel; the latter by a vertical kernel. In both cases, each row of the image has its own kernel as computed by the projection of the expected ground plane into the image and the nominal painted line widths that such projection would imply. The shape of the kernel is shown in Figure 3.

The kernel is sampled according to the pixel grid at each row, then convolved with that row to produce the output of the matched filter. As shown in Figure 4, this operation successfully discards most of the clutter in the scene and produces a strong response along line-like features. This is done separately for the vertical and horizontal kernels, giving two output images. We then compute a list of local maxima of the filter responses and a principal direction of the line at each maximum. This direction is computed using the dominant eigenvector of the Hessian in a small window around each maximum.

The system next connects nearby maxima into splines that represent continuous line markings. To do so, it randomly selects 100 "seed" maxima near the bottom of the image from the list of all maxima. For each seed, we set it as the first control point in a cubic hermite spline, and consider an annulus around the seed of radius 50 pixels and width 10 pixels. Each of the maxima within this annulus becomes a candidate for the spline's second control point and is assigned a score. The score is computed by sampling along the spline's length the value of the distance transform function of the list of maxima. The highest scored maximum is saved as the second spline control point. If no maximum has a score above a certain threshold, we reject the whole spline. We continue to "grow" the spline in the same fashion by considering additional annuli of successive control points and finish the spline when adding another control point results in a poor score. The maxima near finished splines are removed from the maxima list so that the same lines are not re-detected. After finishing searching the 100 seeds, the algorithm is complete. The splines are inverse-perspective mapped, intersected with the ground plane, discretized into piecewise linear curves, and transmitted for further processing by the centerline estimator.

C. Road Paint Detection using Symmetric Contours

A second road paint detection mechanism employed in our system relies on more traditional low-level image processing. In order to maximize frame throughput, and thus reduce the time between successive inputs to the lane fusion and tracking components, we designed the module to utilize fairly simple and easily-vectorized image operations.

The central observation behind this detector is that image features of interest – namely lines corresponding to road paint – typically consist of well-defined, elongated, continuous regions that are brighter than their surround. This characterization encompasses solid and dashed lane markings, stop lines, crosswalks, white and yellow paint on road pavements of various types, and markings seen through cast shadows across the road surface. Thus, our strategy is to first detect the potential boundaries of road paint using spatial gradient operators, and then estimate the desired line centers by searching for boundaries that enclose a brighter region; that is, boundary pairs which are proximal and roughly parallel in real-world space and whose local gradients point toward each other (Figure 5).
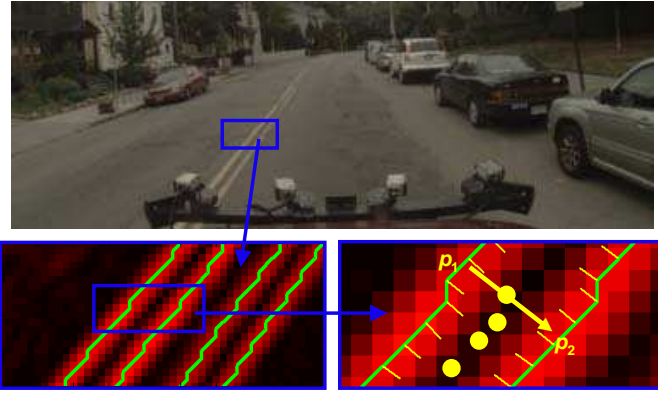

Fig. 5.   Progression from original image through smoothed gradients (red), border contours (green), and symmetric contour pairs (yellow) to form a paint line candidate.

Three steps constitute the contour-based road line detector: low-level image processing to detect raw features; contour extraction to produce initial line candidates; and contour post-processing for smoothing and false alarm reduction. The first step applies local lowpass and derivative operators to produce the noise-suppressed direction and magnitude of the raw (grayscale) image's spatial gradients. The gradient magnitude is thresholded, and non-maximal suppression is performed in the gradient direction to produce a sparse feature mask.

Next, a connected components algorithm iteratively walks the feature mask to generate smooth contours of ordered points, broken at discontinuities in location and gradient direction. This results in a new image whose pixel values indicate the identities and positions of the detected contours, which in turn represent candidate road paint boundaries. In order to localize the centerlines between these boundaries, a second iterative walk is applied. At each boundary pixel p_i

(traversed in contour order), the algorithm extends a virtual line in the direction of the gradient until it meets another contour at p_j. If the gradient of the second contour points in the opposite direction, then the midpoint between p_i and p_j is added to a growing centerline curve (Figure 2).

This step connects many short paint fragments, producing a smaller number of longer centerline candidates. The gradient constraint insures that each candidate is brighter than its surround. Since this candidate set may be corrupted by small line fragments and outliers, a series of higher-level post-processing operations is performed. We enforce global smoothness and curvature constraints by fitting parabolas to the curves and recursively breaking them at points of high deviation or spatial gaps. We then remove all curves shorter than a given threshold length (in pixels) to produce the final road paint lines. As with the first road paint detection algorithm, these are inverse-perspective mapped and intersected with the ground plane before further processing.

D. Lane Centerline Estimation

The second stage of lane finding estimates the geometry of nearby lanes using a weighted set of recent road paint and curb detections, both of which are represented as piecewise linear curves. To simplify the process, we estimate only lane centerlines, which we model as locally parabolic segments. While urban roads are not designed to be parabolic, this representation is generally accurate for stretches of road that lie within sensor range.

Lanes centerlines are estimated in two steps. First, a centerline evidence image D is constructed, where the value each pixel $D(\mathbf{p})$ of the image corresponds to the evidence that a point $\mathbf{p} = [p_x, p_y]$ in the local coordinate frame lies on the center of a lane. Second, parabolic segments are fit to the ridges in D and evaluated as lane centerline candidates.

1) Centerline Evidence Image: To construct D, road paint and curb detections are used to increase or decrease the values of pixels in the image, and are weighted according to their age (older detections are given less weight). The value of D at a pixel corresponding to the point $\mathbf{p}$ is computed as the weighted sum of the influences of each road paint and curb detection d_i at the point $\mathbf{p}$:

$$D(\mathbf{p}) = \sum_i e^{-a(d_i)\lambda} g(d_i, \mathbf{p})$$

where $a(d_i)$ denotes how much time has passed since d_i was received, λ is a decay constant, and $g(d_i, \mathbf{p})$ is the influence of d_i at $\mathbf{p}$. We chose $\lambda = 0.7$.

Before describing how the influence is determined, we make three observations. First, a lane is more likely to be centered $\frac{1}{2}$ lane width from a strip of road paint or a curb. Second, 88% of federally managed lanes in the U.S. are between 3.05 m and 3.66 m wide [14]. Third, a curb gives us different information about the presence of a lane than does road paint. From these observations and the characteristics of our road paint and curb detectors, we define two functions $f_{rp}(x)$ and $f_{cb}(x)$, where

x is the Euclidean distance from d_i to $\mathbf{p}$:

$$f_{rp}(x) = -e^{-\frac{x^2}{0.42}} + e^{-\frac{(x-1.83)^2}{0.14}} \quad (1)$$

$$f_{cb}(x) = -e^{-\frac{x^2}{0.42}}. \quad (2)$$

The functions f_{rp} and f_{cb} are intermediate functions used to compute the influence of road paint and curb detections, respectively, on D. f_{rp} is chosen to have a minimum at $x = 0$, and a maximum at one half lane width (1.83 m). f_{cb} is always negative, indicating that curb detections are used only to decrease the evidence for a lane centerline. We elected this policy due to our curb detector's occasional detection of curb-like features where no curbs were present. Let $\mathbf{c}$ indicate the closest point on d_i to $\mathbf{p}$. The actual influence of a detection is computed as:

$$g(d_i, \mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{c} \text{ is an endpoint of } d_i, \\ & \text{else} \\ f_{rp}(||\mathbf{p} - \mathbf{c}||) & \text{if } d_i \text{ is road paint, else} \\ f_{cb}(||\mathbf{p} - \mathbf{c}||) & \text{if } d_i \text{ is a curb} \end{cases}$$

This last condition is introduced because road paint and curbs are only observed in small sections. The effect is that a detection influences only those centerline evidence values immediately next to the detection, and not in front of or behind it.

In practice, D can be initialized once and incrementally updated by adding the influences of newly received detections and applying an exponential time decay at each update. Additionally, we improve the system's ability to detect lanes with dashed boundaries by injecting imaginary road paint detections connecting two separate road paint detections when they are physical proximate and collinear.

2) Parabola Fitting: Once the centerline evidence image D has been constructed, the set R of ridge points is identified by scanning D for points that are local maxima along either a row or a column, and also above a minimum threshold. Next, a random sample consensus (RANSAC) algorithm [7] is used to fit parabolic segments to the ridge points. At each RANSAC iteration, three ridge points are randomly selected for a three-point parabola fit. The directrix of the parabola is chosen to be the first principle component of the three points.

To determine the set of inliers for a parabola, we first compute its conic coefficient matrix $\mathbf{C}$ [9], and define the set of candidate inliers L to contain the ridge points within some algebraic distance α of $\mathbf{C}$.

$$L = \{\mathbf{p} \in R : \mathbf{p}^T \mathbf{C} \mathbf{p} < \alpha\}$$

For our experiments, we chose $\alpha = 1$. The parabola is then re-fit once to L using a linear least squares method, and a new set of candidate inliers is computed. Next, the candidate inliers are partitioned into connected components, where a ridge point is connected to all neighboring ridge points within a 1 m radius. The set of ridge points in the largest component is chosen as the set of actual inliers for the parabola. The purpose of this partitioning step is to ensure that a parabola cannot be fit across multiple ridges, and requires that an entire identified

Fig. 6. The second stage of our system constructs a centerline evidence image. Lane centerline candidates (blue) are identified by fitting parabolic segments to the ridges of the image. Front-center camera is shown in top left for context.

ridge be connected. Finally, a score for the entire parabola is computed.

$$s = \sum_{\mathbf{p} \in L} \frac{1}{1 + \mathbf{p}^T \mathbf{C} \mathbf{p}}$$

The contribution of an inlier to the total parabola score is inversely related to the inlier's algebraic distance, with each inlier contributing a minimum amount to the score. The overall result is that parabolas with many very good inliers have the greatest score. If the score of a parabola is below some threshold, then it is discarded. Experimentation with different values resulted in us choosing a score threshold of 140.

After a number of RANSAC iterations (we found 200 to be sufficient), the parabola with greatest score is selected as a candidate lane centerline. Its inliers are removed from the set of ridge points, and all remaining parabolas are re-fit and re-scored using this reduced set of ridge points. The next best-scoring parabola is chosen, and this process is repeated to produce at most 5 candidate lane centerlines (Figure 6). Each candidate lane centerline is then discretized as a piecewise linear curve and transmitted to the lane tracker for further processing.

E. Lane Tracking

The primary purpose of the lane tracker is to maintain a stateful, smoothly time-varying estimate of the nearby lanes of travel. To do so, it uses both the candidate lane centerlines produced by the centerline estimator and an a-priori estimate derived from a road map.

In the context of the Urban Challenge, the road map was known as the Route Network Description File (RNDF). The RNDF can roughly be thought of as a directed graph, where each node is a waypoint in the center of a lane, and edges represent intersections and lanes of travel. Waypoints are given as GPS coordinates, can be separated by arbitrary distances, and a simple linear interpolation of connected waypoints may go off road, through trees and houses. For the purposes of our system, the RNDF was treated as a strong prior on the

(a) Two RNDF-derived lane centerline priors

(b) Candidate lane centerlines estimated from sensor data

(c) Filtered and tracked lane centerlines

Fig. 7. (a) The RNDF provides weak a-priori lane centerline estimates (white) that may go off-road, through trees and bushes. (b) On-board sensors are used to detect obstacles, road paint, and curbs, which are in turn used to estimate lanes of travel, modeled as parabolic segments (blue). (c) The sensor-derived estimates are then filtered, tracked, and fused with the RNDF priors.

number and type of lanes, and a weak prior on their position and geometry.

As our vehicle travels, it constructs and maintains representations of all portions of all lanes within a fixed radius of 75 m. The centerline of each lane is modeled as a piecewise linear curve, with control points spaced approximately every 2 m. Each control point is given a scalar confidence value indicating the certainty of the lane tracker's estimate at that point. The lane tracker decays the confidence of a control point as the vehicle travels, and increases it either by detecting proximity to an RNDF waypoint or by updating control points with centerline estimates produced from the second stage.

As centerline candidates are generated, the lane tracker attempts to match each candidate with a tracked lane. If a matching is successful, then the candidate is used to update the lane estimate. To determine if a candidate c is a good match for a tracked lane l, the longest segment s_c of the candidate is identified such that every point on s_c is within some maximum distance τ to l. We then define the match score $m(c, l)$ as:

$$m(c, l) = \int_{s_c} 1 + \frac{\tau - d(s_c(x), l)}{\tau} dx$$

where $d(\mathbf{p}, l)$ is the distance from a point $\mathbf{p}$ to the lane l. Intuitively, if s_c is sufficiently long and close to this estimate, then it is considered a good match. We choose the matching function to rely only on the closest segment of the candidate, and not on the entire candidate, based on the premise that as the vehicle travels, the portions of a lane that it observes vary smoothly over time, and previously unobserved portions should not adversely affect the matching as long as sufficient overlap is observed elsewhere.

Once a centerline candidate has been matched to a tracked lane, it is used to update the lane estimates by mapping control points on the tracked lane to the centerline candidate, with an exponential moving average applied for temporal smoothing. At each update, the confidence values of control points updated from a matching are increased, and others are decreased. If the confidence value of a control point decreases below some threshold, then its position is discarded and recomputed as a linear interpolation of its closest surrounding confident control points. Figure 7 illustrates this process.

III. URBAN CHALLENGE RESULTS

Often, the most difficult part of evaluating a lane detection and tracking system for autonomous vehicle operation lies in finding a suitable test environment. Legal, financial, and logistical constraints prove to be a significant hurdle in this process. We were fortunate to have the opportunity to conduct an extensive test in the 2007 DARPA Urban Challenge, which provided a large-scale real world environment with a wide variety of roads. Both the type and quality of roads varied significantly across the race, from well-marked urban streets, to steep unpaved dirt roads, to a 1.6 km stretch of highway. Throughout the duration of the race, approximately 50 human-driven and autonomous vehicles were simultaneously active, thus providing realistic traffic scenarios.

Our most significant result is that our lane detection and tracking system successfully guided our vehicle through a 90 km course in a single day, at speeds up to 40 km/h, with an average speed of 16 km/h. A post-race inspection of our log files revealed that at no time did our vehicle have a lane centerline estimate more than half a lane width off of the actual lane centerline, and at no time did it unintentionally enter or exit a lane of travel. In saying this, we note that the output of the lane tracking system was used directly to guide the navigation and motion planning systems. Specifically, if the lane tracking system produced an incorrect estimate, our vehicle would have traveled along that estimate, possibly into an oncoming traffic lane or off-road.

The first question that arises from these statements is that of determining how much our system relied on perceptually-derived lane estimates, and how much it relied on the prior knowledge of the road as given in the RNDF. To answer this, we examine the distance the vehicle traveled with high confidence visually-derived lane estimates, excluding control points where high confidence is a result of proximity to an RNDF waypoint.

Visual range (m)	Distance traveled (km)	
≤ 0	30.3	(34.8%)
$1-10$	10.8	(12.4%)
$11-20$	24.6	(28.2%)
$21-30$	15.7	(18.0%)
$31-40$	4.2	(4.8%)
$41-50$	1.3	(1.5%)
>50	0.2	(0.2%)

TABLE I

DISTANCE TRAVELED WITH HIGH-CONFIDENCE VISUAL ESTIMATES IN
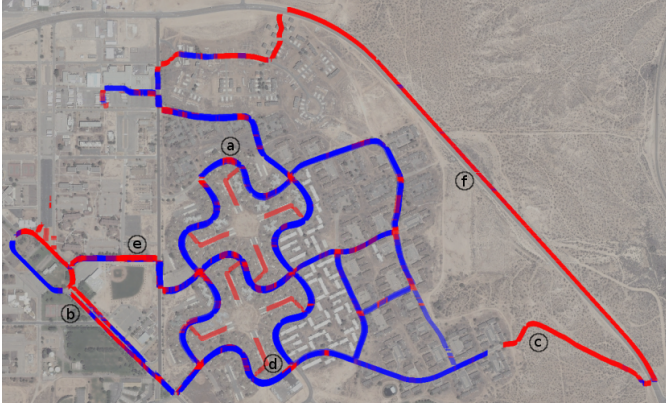CURRENT LANE OF TRAVEL.

Fig. 8. Aerial view of the Urban Challenge race course in Victorville, CA. Autonomously traversed roads are colored blue in areas where the lane tracking system reported high confidence, and red in areas of low confidence. Some low-confidence cases are expected, such as at intersections and areas with no clear lane markings. Failure modes occurring at the circled letters are described in Fig. 9.

Fig. 9. Common failure cases. The most common failure was in areas with strong tree shadows, as seen in (a) and (b). Dirt roads, and those with faint or no road paint (c-e) were also common. In (f), a very wide lane and widely spaced dashed markings were a challenge due to our strong prior on lane width. In each of these failures, the system reported no confidence in its visual estimates.

At a given instance, our system can either have no confidence in its visual estimates of the current lane of travel, or confidence out to a certain distance a in front of the vehicle. If the vehicle then travels d meters while maintaining the same confidence in its visual estimates, then we say that the system had a high-confidence estimate a meters in front of the vehicle for d meters of travel. Computing a for all 90 km of the race allows us to answer the question of how far out our system could typically see. This information is summarized in Table I. From this, we can see that our vehicle maintained high confidence visual estimates to some forward distance for 56.8 km, or 65.2% of the total distance traveled. In the remaining portion, the lane tracker relied on an interpolation of its last high confidence estimates.

A second way of assessing the system's performance is by examining its estimates as a function of location within the course. Figure 8 shows an aerial view of areas visited by our vehicle, colored according to whether or not the vehicle had a high confidence estimate at a given point. We note that our system had high confidence lane estimates throughout the majority of the high-curvature and urban portions of the course. Some of the low-confidence cases are expected, such as when the vehicle is traveling through intersections and roads with no discernible lane boundaries. In other cases, our system

was unable to obtain a high confidence estimate whereas a human would have little trouble doing so.

Images from our logged camera images at typical failure cases are shown in Figure 9, and the locations at which these failures occurred are marked in Figure 8. A common failure mode was an inability to detect road paint in the presence of dramatic lighting variation such as that caused by cast tree shadows. However, we note that in all of these cases our system reported no confidence in its estimates and did not falsely estimate the presence of a lane.

Another significant failure occurred on the eastern part of the course, with a 0.5 km dirt road followed by a 1.6 km stretch of highway. Our vehicle traversed this path four times, for a total of 8.4 km. The highway was an unexpected failure, and the travel lane happened to be very wide. Its width did not fit the 3.66 m prior in the centerline estimator, which had trouble constructing a stable centerline evidence image. In addition, the dashed lane markings on the highway were spaced much further apart than dashed lane markings are on typical urban roads.

The final common failure mode occurred in areas with faint or no road paint, such as the dirt road and roads with well defined curbs but no paint markings. Since our system uses road paint as its primary information source, in the absence

of road paint it is no surprise that no lane estimate ensues. Other environmental cues such as color and texture may be more useful [4].

The output of our system is used for high-speed motion planning; thus we would like for its estimates to remain relatively stable. Specifically, we desire that once the system produces a high confidence estimate, that the estimate does not change significantly. To assess the suitability of our system for this purpose, we can compute a *stability ratio* that measures how much its high confidence lane estimates change over time in the transverse direction.

Consider a circle of radius r centered at the current position of the rear axle. We can find the intersection $\mathbf{p}_0$ of this circle with the current lane estimate that extends ahead of the vehicle. When the lane estimate is updated at the next time step (10 Hz in this case) we can compute $\mathbf{p}_1$, the intersection of the same circle with the new lane estimate. We define the stability ratio as:

$$R = \frac{||\mathbf{p}_0 - \mathbf{p}_1||}{d_v} \tag{3}$$

where d_v is distance traveled by our vehicle in that time step.

Intuitively, the stability ratio is the ratio of the transverse movement of the lane estimate to the distance traveled by the car in that time, for some r. We can also compute an *average stability ratio* for some r by averaging the stability ratios for every time step of the vehicle's trip through the course (Figure 10). From this figure, we see that the average stability ratio remains small and relatively constant, but still nonzero, indicating that high-confidence lane estimates can be expected to shift slightly as the vehicle makes forward progress.
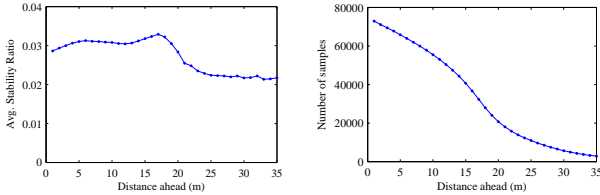

Fig. 10. (Left) The average stability ratio. (Right) The number of samples used to compute the stability ratio varies with r, as only control points with visually-derived high-confidence are used.

IV. Conclusion and Future Work

Our system attempts to extend the scope of lane detection and tracking for autonomous vehicles to the urban environment. We have presented a modular, scalable, perception-centric lane detection and tracking system that fuses asynchronous heterogeneous sensor streams with a weak prior to estimate multiple travel lanes in real-time. The system makes no assumptions about the position or orientation of the vehicle with respect to the road, enabling it to operate when changing lanes, at intersections, and when exiting driveways and parking lots. The vehicle using our system was, to our knowledge, the only vehicle in the final stage of the DARPA Urban Challenge to employ vision-based lane finding.

Despite these advances, the method is not yet suitable for real-world deployment. As with most vision-based systems, it is susceptible to strong lighting variations such as cast shadows. To address this, we are investigating the use of lidar intensity data for detecting road paint. Typical road paint has high infrared reflectivity and our preliminary lidar experiments are promising. Our highway experience in the race demonstrates the need to handle lanes with greater variance in width, which could be accomplished by first estimating lane width and then generating the centerline evidence image accordingly. Finally, since many roads do not use paint as boundary markers, we are extending our method to incorporate other environmental cues.

V. Acknowledgments

References

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

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

[3] CNN. Doh! man follows GPS onto train tracks – when train coming. http://www.cnn.com/2008/US/01/03/gps.traincrash.ap/index.html, Jan. 2008.

[4] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, Aug. 2006.

[5] Ernst Dickmanns and Birger Mysliwetz. Recursive 3-d road and ego-state recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):199–213, Feb. 1992.

[6] John Leonard et al. A perception driven autonomous vehicle. *Journal of Field Robotics*, (to appear), 2008.

[7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] Luke Fletcher and Alexander Zelinsky. Context Sensitive Driver Assistance based on Gaze - Road Scene Correlation. In *International Symposium on Experimental Robotics*, Rio, Brazil, Jul. 2006.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2001.

[10] Iteris. http://www.iteris.com.

[11] Joel C. McCall and Mohan M. Trivedi. Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transport Systems*, 7(1):20–37, Mar. 2006.

[12] Ted R. Miller. Benefit-cost analysis of lane marking. *Public Roads*, 56(4):153–163, March 1993.

[13] Mobileye. http://www.mobileye.com.

[14] U.S. Department of Transportation, Federal Highway Administration, Office of Information Management. *Highway Statistics 2005*. U.S. Government Printing Office, Washington, D. C., 2005.

[15] D. Pomerleau and T. Jochem. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert*, 11(2):19–27, Apr. 1996.

[16] New York Times S. Lyall. Turn back. exit village. truck shortcut hitting barrier. http://www.nytimes.com/2007/12/04/world/europe/04gps.html, Dec. 2007.

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

[18] Wei-Bin Zhang. A roadway information system for vehicle guidance/control. In *Vehicle Navigation and Information Systems*, volume 2, Oct. 1991.