RoadCompass: Following Rural Roads with Vision + Ladar Using Vanishing Point Tracking

Christopher Rasmussen
University of Delaware
cer@cis.udel.edu

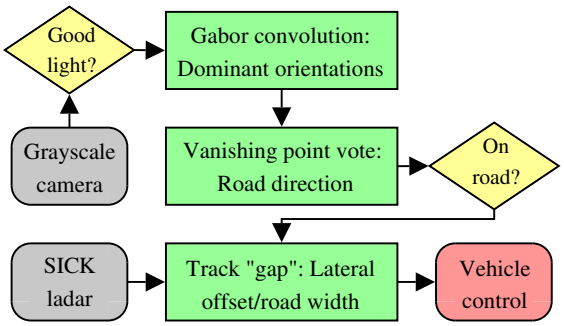
***Abstract*— We present a vision- and ladar-based approach to autonomous driving on rural and desert roads that has been tested extensively in a closed-loop system. The vision component uses Gabor wavelet filters for texture analysis to find ruts and tracks from which the road vanishing point can be inferred via Hough-style voting, yielding a direction estimate for steering control. The ladar component projects detected obstacles along the road direction onto the plane of the front of the vehicle and tracks the 1-D obstacle "gap" presumed due to the road to yield a lateral offset estimate. Several image- and state-based tests to detect failure conditions such as off-road poses (i.e., there is no road to follow) and poor lighting due to sun glare or distracting shadows are also explained. The system's efficacy is demonstrated with analysis of diverse logged data including from the 2005 DARPA Grand Challenge, as well as tests with full control of a vehicle over 15 km of difficult roads at up to 37 km/h with no waypoints.**

I. Introduction

The running of the DARPA Grand Challenge Event (GCE) desert robot races in March, 2004 and October, 2005 [1] heightened interest in the development of perceptual algorithms for autonomous ground vehicles to follow "difficult" unpaved paths and roads. Though there were some off-road portions, nearly all of the 2004 and 2005 GCE courses, as specified by a series of GPS waypoints, were along dirt roads. The recent DARPA Urban Challenge in November, 2007 took place almost entirely on roads, and vehicles were given sparser waypoints, making a visual navigation capability increasingly paramount. In this paper, we describe a perceptual module for following marginal as well as paved roads *without waypoints* that uses one grayscale camera in conjunction with a SICK ladar to rapidly obtain an estimate of the oncoming road area and transmit appropriate steering and throttle commands to the vehicle controller.

The module, which we call *RoadCompass* (in the spirit of [2], [3]) because it seeks to infer the vehicle's orientation relative to the road ahead based on the vanishing point structure of the image, was developed to operate on board Team Caltech's 2005 GCE vehicle "Alice," pictured in Figure I(a). As such, *RoadCompass* was part of a larger system developed by many others which comprises modules for higher-level navigation, structural map maintenance, off-road steering, direct control of the vehicle, hardware fault monitoring, and so on—the complete system is described in [4]. In the 2005 GCE, the ultimate action taken by Alice in any given situation was a complex function of its state and sensor inputs that is explained in [4], with

(a)



(b)

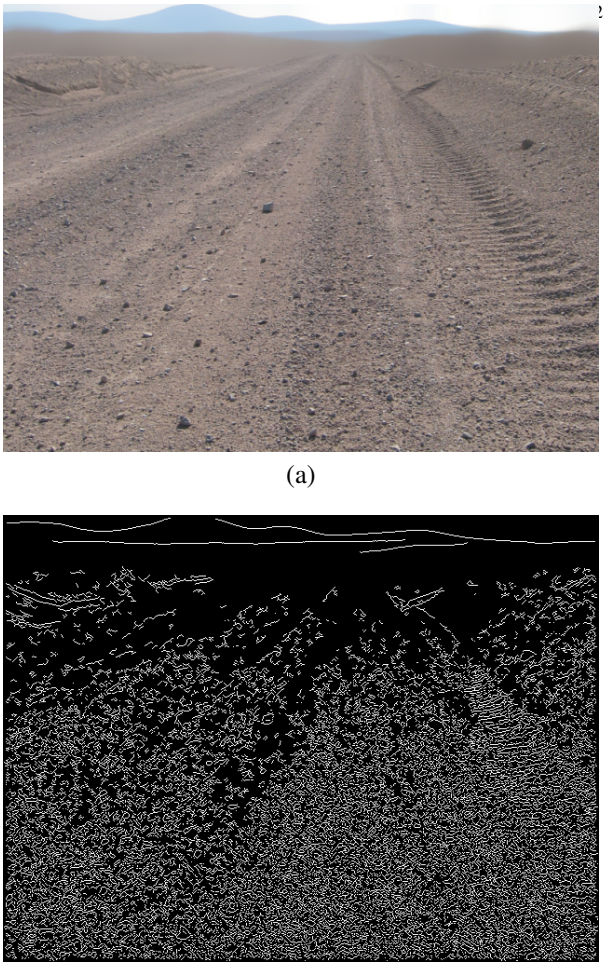
Fig. 1. (a) Sample sensor configuration on "Alice," a fully-actuated Ford E350 with a modified suspension and 4 wheel drive. SICK ladars are mounted on the bumper and roof; road following camera is one of stereo pair over windshield; (b) Organization of *RoadCompass* module components. The green boxes are the three primary perceptual steps necessary to build a model of the road region discussed in Sections II-A, II-B, and II-C. The two yellow diamonds are tests for failure conditions covered in Section III.

RoadCompass effectively constituting only one opinion among many. However, pre-race tests with simplified versions of the vehicle controller demonstrated *RoadCompass*'s ability to successfully control Alice in a closed-loop fashion over long distances. *RoadCompass*'s system components are diagrammed in Figure I(b).

Many complementary strategies for visual road following have been developed based on certain assumptions about the characteristics of the road scene. For example, edge-based methods such as those described in [5], [6], [7] are often used in relatively urban areas to identify lane lines or road borders, which are fit to a model of the road curvature, width, and so on. These algorithms typically work best on well-engineered roads such as highways which are paved and/or painted, resulting in a wealth of high-contrast contours suited for edge detection. Another popular set of methods for road tracking are region-based [8], [7], [9], [10], [11]. These approaches use characteristics such as color or texture measured over local neighborhoods in order to formulate and threshold on a likelihood that pixels belong to the road area vs. the background. When there is a good contrast for the cue chosen, there is no need for the presence of sharp or unbroken edges, which aids these methods on unpaved rural roads.

The original motivation for this work was our empirical finding that many desert roads, being unpaved and colored similarly to the surroundings, present difficulties for these approaches because they possess neither strong edges nor contrasting local characteristics. Figure 2(a) shows one such road. This image is from a set of "course examples" made available to entrants in the 2004 GCE (DARPA air-brushed part of the image near the horizon to obscure location-identifying features). There is no color difference between the road surface and off-road areas and no strong edges delimiting it, as the output of the Matlab `canny` edge detection function shows in Figure 2(b). The one characteristic that seems to define the road is texture, but not in a locally measurable sense, because there are pebbles, shadows, and stripes everywhere. Rather, one seems to apprehend the road easily because of its overall *banding* pattern. This banding, due to ruts and tire tracks left by previous vehicles driven by humans who knew the way, is aligned with the road direction and thus most apparent because of the strong grouping cue imposed by its *vanishing point*. The percept of the vanishing point is reinforced by other oriented texture such as road border intensity edges or painted lines if they are present, and thus is an almost invariant feature of road images taken from the driver's perspective regardless of road width or surface properties.

A number of researchers have used vanishing points as global constraints for road following or identification



(a)

(b)

Fig. 2. The difficulty of color segmentation or edge detection for road following: (a) Desert road from 2004 DARPA Grand Challenge example set; (b) Canny edges of same scene.

of painted features on roads (such as so-called "zebra crossings," or crosswalks) [12], [13], [14], [15]. Broadly, the key to the approach is to use a voting procedure like a Hough transform on edge-detected line segments to find points where many intersect. Peaks in the voting function are good candidates for vanishing points. This is sometimes called a "cascaded" Hough transform [16] because the lines themselves may have first been identified via a Hough transform. Similar grouping strategies have also been investigated outside the context of roads, such as in urban and indoor environments rich in straight lines, in conjunction with a more general analysis of repeated elements and patterns viewed under perspective [17], [18].

All of the voting methods for localizing a road's vanishing point cited above are based on a prior step of finding line segments via edge detection. This is

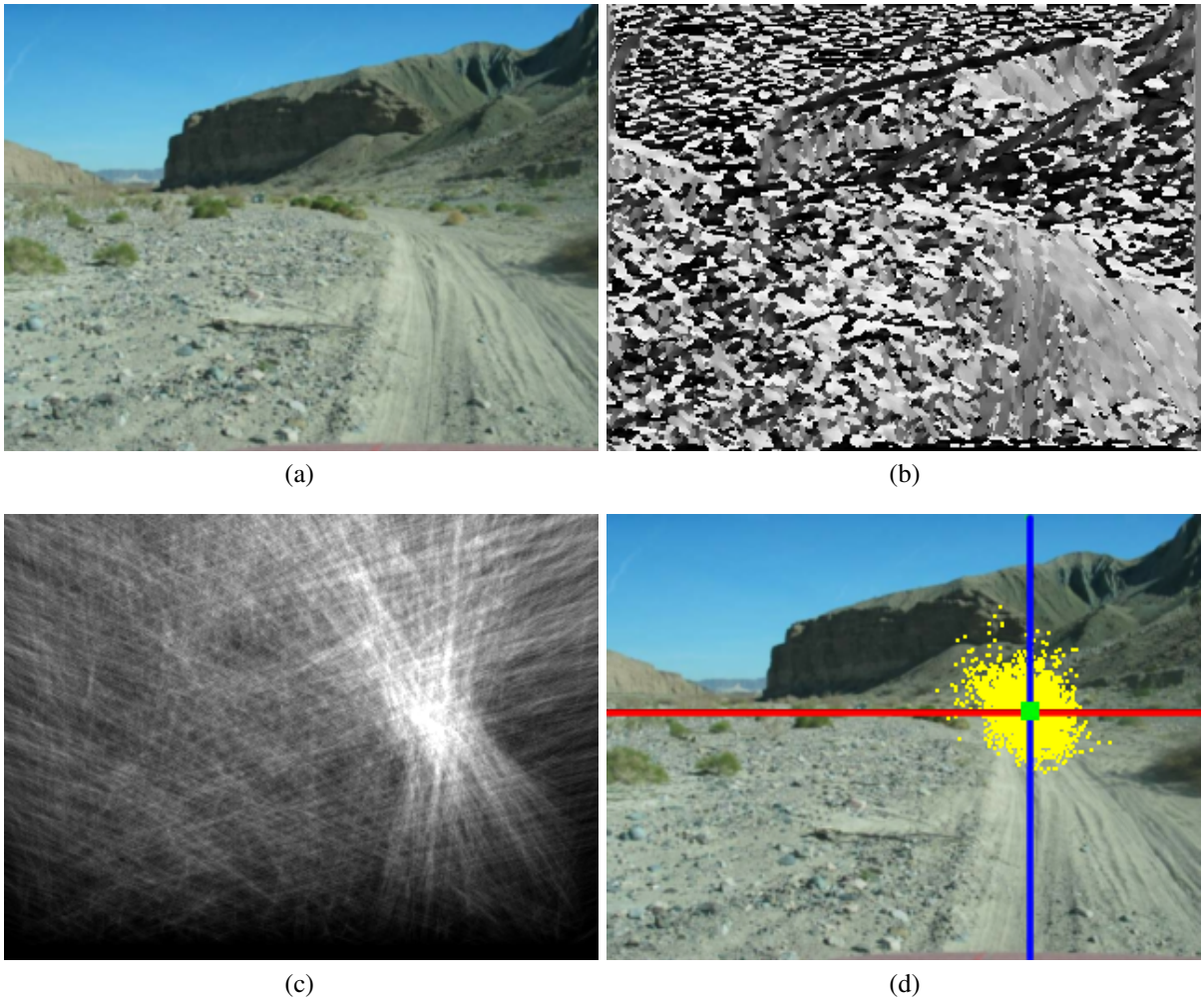


Fig. 3. Steps of vanishing point localization: (a) Captured road image; (b) Dominant orientation at each pixel ($[0, \pi]$ radians $\rightarrow$ $[0, 255]$ intensity values); (c) Vote function for vanishing point over image region; (d) Distribution of particles and estimated vanishing point location with horizon line indicated.

unworkable for many desert road scenes because the road bands are too low-frequency to be explicitly detected. The "Manhattan world" approach outlined in [2], [19] does not require an edge detection step, but is not suitable for real-time applications such as autonomous driving. In contrast, *RoadCompass* uses a fast and straightforward method (introduced by us in [20] and extended in [21]) for locating the road's vanishing point in such difficult scenes through texture analysis. Specifically, we replace edge detection with estimates of the *dominant orientation* at each location in the image. These suffice to conduct voting in a similar fashion and find a vanishing point. Figure 3 shows the steps of *RoadCompass*'s output on a sample image.

For a straight road segment on planar ground, there is a unique vanishing point associated with the road. Its horizontal image position indicates the road direction, and its vertical position marks the horizon line of the road plane. The significance of the estimated road direction, of course, is that the difference between it and the vehicle's current direction of travel implies the steering adjustment necessary to stay on the road. However, as the vanishing point is only the image projection of a tangent to the road curve, there is no unique vanishing point for curving or undulating road segments. More detailed texture analysis to extract road curvature information is possible (and discussed in [20]), but *RoadCompass* demonstrates that it is unnecessary for robust road following at moderate speeds.

In addition to the image-derived road direction information, a second key part of *RoadCompass* is extraction of the vehicle's *lateral offset* from the road midline. Alignment of vehicle direction with the road direction does not assure that the vehicle is on the road—only that

is is moving parallel to it. Without a corrective centering impulse, the vehicle may drift off of a straight road over time or cut off or overshoot curves. Image-based road segmentation is one possible approach here [9], [11], and we have experimented with several methods using the vanishing point as a shape constraint [22], [23]. However, we ultimately found them insufficiently robust due to sensitivity to the road surface material and lighting conditions (particularly strong shadows).

Rather, *RoadCompass* uses an instantaneous map of the obstacles ahead derived from a SICK LMS ladar to estimate the lateral offset of the road midline relative to the vehicle. Considering a 1-D slice through the map orthogonal to the road direction, the road region corresponds to an obstacle-free *gap* between the left and right road edges. Our strategy is to try to locate and track this gap so that the vehicle can center itself while decreasing directional error. Like our vanishing point method, this approach has no need to tune, learn, or adapt parameters as the algorithm runs on a variety of roads. In flat areas where the gap is indistinct because only visual appearance demarcates the road, drifting due to imprecise centering is of minimal concern assuming similar on- and off-road material properties (i.e., both dirt or both sand).

Trying to extract the road direction directly from ladar data rather than camera images, while tempting, is frequently difficult for single scans due to the sparsity of data provided by one or two SICKs in an outdoor, desert environment (see, for example, the scenes in Figure 7). However, we have had some success since the 2005 GCE using a RANSAC-like approach with temporal filtering [24], especially in urban scenes. For the 2005 GCE, we chose to use instantaneous gap estimates instead of temporally integrating ladar data before estimation because of a lack of confidence in the accuracy of Alice's state estimation. Indeed, it was a registration error in Alice's integrated map (not part of *RoadCompass*) due to a GPS glitch that was the primary cause of its race-ending crash [4]. Obviously, if a reliable, high-precision state estimate is available, then combining ladar scans over time and processing this fused representation as described in [25], [26] is superior at least for road width and offset estimation.

Finally, a critical road-follower state is whether it currently "sees" a road or not and thus should be trusted. The 2004 and 2005 GCE courses went through several roadless areas, though in actuality these were on dry lake beds where tracks left by previous autonomous and course official vehicles created virtual roads. Leaving the road in more difficult terrain would necessitate falling back to generalized obstacle avoidance and slope analysis techniques as studied, for example, in [27], [28].

The decision mechanism for making such a switch is the last component of *RoadCompass*, along with several methods for detecting other failure situations related to poor lighting conditions such as sun glare, distracting shadows, and overall darkness that may prevent the module from recognizing a road even if it is on one. Of course the road locations of most North American and European highways and urban streets, as well as many rural roads, have been digitized in vector form, but digital maps may not cover every obscure desert path we want to follow. *RoadCompass* thus includes an image-based function to discriminate scenes that contain a strong forward vanishing point—assumed to belong to a road—from those that do not. We believe that this provides an additional argument for combining imagery with ladar data: this decision function, described in Section III, is significantly more reliable than would be one based solely on ladar data.

In the next two sections, we will detail the steps of how *RoadCompass* estimates road shape, controls the vehicle, and detects failure conditions. For each of these components we will present results supporting and clarifying the techniques used. A concluding "Integrated Tests" section demonstrates the system's capabilities as a whole by analyzing performance over several extended closed-loop test runs and during the 2005 GCE itself, both running live on Caltech's vehicle and on logged data from another team that finished the course.

II. Road Shape Estimation

There are three significant stages to road shape estimation which we describe in the following subsections. First, a texture analysis is performed by computing dominant texture orientations over the current image. Second, a linear approximation to the road direction θ_{road} is measured by having all dominant orientations in the image vote for a single best road vanishing point. Finally, the vehicle's lateral offset δ_{road} from the road center and the road width w_{road} are estimated from ladar data. Two methods of controlling Alice with these parameters are also covered in a fourth subsection.

A. Dominant Orientations

The *dominant orientation* $\theta(\mathbf{p})$ at pixel $\mathbf{p} = (x, y)$ of an image is the direction that describes the strongest local parallel structure or texture flow. This is of course a scale-dependent measure. As we will explain in more detail in the next subsection, precise estimates of the dominant orientations are crucial in order to obtain sharp peaks in the voting objective function and hence accurately localize the vanishing point. There is a considerable body of work on estimating dominant orientations. For example, we may apply a bank of multi-scale, oriented filters such as steerable filters [29]

and analyze the maximum responses. Another approach is to generate a Gaussian pyramid of the image, use principal components analysis on the set of gradients within a small window to obtain a consensus direction at each scale, and then interpolate [30].

We experimented with several approaches to local orientation estimation, including that of [30], and observed the most qualitatively accurate results on a wide variety of road images with a bank of Gabor wavelet filters [31]. Gabor wavelet filters essentially perform a Gaussian-windowed Fourier analysis on the image via convolution with a set of kernels parametrized by orientation θ, wavelength λ, and odd or even phase. To generate a $k \times k$ Gabor kernel (we use $k = \lfloor \frac{10\lambda}{\pi} \rfloor$), we calculate:

$$\hat{g}_{odd}(x, y, \theta, \lambda) = \exp[-\frac{1}{8\sigma^2}(4a^2 + b^2)] \sin(2\pi a/\lambda) \quad (1)$$

where $x = y = 0$ is the kernel center, $a = x\cos\theta + y\sin\theta$, $b = -x\sin\theta + y\cos\theta$, $\sigma = \frac{k}{9}$, and the "sin" changes to "cos" for $\hat{g}_{even}$. The actual convolution kernel g is then obtained by subtracting $\hat{g}$'s DC component (i.e., mean value) from itself and normalizing the result so that g's L^2 norm is 1.

To best characterize local texture properties including step and roof edge elements at an image pixel $\mathbf{I}(x, y)$, we examine the *complex response* of the Gabor filter given by $\mathbf{I}_{complex}(x, y) = (g_{odd} * \mathbf{I})(x, y)^2 + (g_{even} * \mathbf{I})(x, y)^2$ for a set of n evenly spaced Gabor filter orientations. The dominant orientation $\theta(x, y)$ is chosen as the filter orientation which elicits the maximum complex response at that location.

With *a priori* knowledge of the distribution of actual (3-D) road texture wavelengths λ_{road}, the camera focal length, and the pitch or tilt angle of the camera with respect to the ground plane, the distribution of perceived road texture wavelengths in the image λ_{image} could be established. This information would allow a principled choice of a range of filter wavelengths to run at each image location and weights for combining them, with larger-scale filters being applied toward the bottom of the image and finer filters used closer to the horizon line.

However, in the early stages of this work the testing images came from a variety of uncalibrated cameras mounted with unknown height and tilt. Furthermore, the data contain a number of significant departures from the planar ground assumption. Based on empirical observation of performance using 4 octave-separated wavelengths both independently and in combination, we found that a single wavelength related to the image dimensions by an *ad hoc* scaling factor gave good results at a significant computational savings vs. multi-scale schemes. Thus, for all of the results in this paper except where otherwise noted, the image has been scaled via an image pyramid down to 80×60 resolution, the number of Gabor orientations used is $n = 36$, and a single wavelength $\lambda = 4$ resulting in a kernel size of 12×12 is used. The FFTW Fourier transform library [32] at single precision is used to calculate dominant orientations speedily, taking ~ 55 ms on a 3.0 GHz Pentium IV for a 160×120 image.

Figure 3(b) shows the calculated dominant orientations for the image in Figure 3(a). Gray level intensities proportional to an estimated angle from 0 to 180 degrees (in 36 discrete steps) are shown. Observe that most parallel structure is in the dirt road on the right.

B. Vanishing Point Detection

A central assumption of *RoadCompass* is that for most road scenes, especially rural ones, the vanishing point due to the road is the only one in the image. In rural scenes, there is very little other coherent parallel structure besides that due to the road. The dominant orientations of much off-road texture such as vegetation, rocks, etc. are randomly and uniformly distributed with no strong points of convergence. Even in urban scenes with non-road parallel structure, such texture is predominantly horizontal and vertical, and hence the associated vanishing points are located well outside the image. Thus we can search for a single vanishing point located approximately in the camera's field of view. With the camera's internal and external calibration the x component of the vanishing point can be converted to the road direction θ_{road} (calibration procedures are detailed in Section IV-A).

The possible vanishing points for an image pixel $\mathbf{p}$ with dominant orientation $\theta(\mathbf{p})$ are all of the points (x, y) along the ray defined by $\mathbf{r_p} = (\mathbf{p}, \theta(\mathbf{p}))$. Intuitively, the best estimate for the vanishing point $\mathbf{v}_{max}$ is that point lying on or near the most such dominant orientation rays (see [16], [15], [12], [19] for recent work on vanishing point finding). In [20], we formulated an objective function $votes(\mathbf{v})$ to evaluate the support of road vanishing point candidates $\mathbf{v}$ over a search region C roughly the size of the image itself.

In the manner of a Hough transform, an efficient and relatively accurate (given enough orientations) voting scheme, which we call *raster voting*, is to draw a "ray of votes" $\mathbf{r_p}$ per voter in an additive accumulation buffer A in which each pixel is a vanishing point candidate $\mathbf{v}$. After rendering every vote ray, the pixel in A (which represents C at a fixed resolution) with the maximum value is $\mathbf{v}_{max}$. Graphics hardware accelerates this voting operation, though 8-bit accumulation buffers limit "elections" to a maximum of 256 votes per candidate, which

Fig. 4. *RoadCompass*'s output on a diverse set of road scenes, with computed vanishing points shown as crosses. For comparison, Gaussian fits of a number of human responses (see text) are marked with ellipses.

is quite enough for the resolution of C that we use[1]. We did not find that variations on the voting function such as weighting $\mathbf{p}$'s vote by the strength of the filter response at $\theta(\mathbf{p})$ or the anisotropy of the filter response over all angles at $\mathbf{p}$ improved the accuracy of vanishing point localization.

The raw maximum of $votes(\mathbf{v})$ is noisy, and since the vanishing point shifts only slightly between frames as the vehicle moves, we smooth the estimate using a particle filter [33]. Particles are initially distributed uniformly in order to coarsely localize the vanishing point. Random-walk dynamics $p(\mathbf{v}_t \,|\, \mathbf{v}_{t-1})$ (e.g., a low-variance, circular Gaussian on the vanishing point position) then limit the search region to track the vanishing point closely, reducing the chance of misidentification due to a false peak elsewhere in the image. The averaging effect of filtering also mitigates saturation by returning the middle of a region of saturated votes as the max, which generally correlates with where the unsaturated maximum would be. We have not found that diminishing lag by adding the image velocity of the vanishing point to the state improves system performance appreciably, as the system runs at 30+ fps and generally tracks the road vanishing point tightly as long as it is in view. By definition, track of the vanishing point is lost if it leaves the search region C, such as at 90-degree turns and on very curvy roads. In this case the particle filter's particles automatically disperse in a kind of search until the vanishing point re-enters C and is reacquired. An example of this occurring at a sharp turn is shown in Figure 19(a2).

Figure 3(c) shows the vanishing point candidate function for the image in Figure 3(a). A cluster of the highest vote totals is at the apex of the road region to the right. The current particle distribution and its weighted mean (i.e., the estimated vanishing point location) are shown in Figure 3(d). Results for tracking the road vanishing point over a sample sequence of images in a different location are shown in the top row of Figure 7. A vertical cyan line marks the vehicle direction in these and some other figures to clarify that when there is a yaw offset between the camera and vehicle coordinate systems the image projection of the vehicle direction does not correspond to the image center.

To gain confidence in the accuracy of the vanishing point estimation method in a variety of situations, we conducted an experiment comparing the algorithm's

[1]Larger numbers of votes being cast at C resolutions much more than 80×60 led to saturation artifacts in 8-bit buffers, preventing true vote totals from being recorded. Generations of graphics cards from the NVIDIA GeForce 6800 onward have supported higher-precision blending, removing this issue, but these were not available on Alice at the time of the 2005 GCE. This, plus the need for real-time dominant orientation calculation, drove the original choice of the 80×60 scale rather than any fundamental algorithmic considerations.

outputs to human perceptions [20]. 16 illustrative images were chosen from a large set of high-resolution digital photographs taken on a scouting trip along a possible Grand Challenge route in the Southern California desert. The algorithm was run on resampled 320×240 versions of the images using raster voting and several iterations of particle-based search; Figure 4 shows the computed $\mathbf{v}_{max}$ for 12 of the 16 images with a green cross. To assess the algorithm's performance vs. human perception of the vanishing point location, we invited ~ 30 members of the UD computer science department to participate in a web-based study. Subjects were given a short definition of road vanishing points, shown two different example images with the vanishing point marked, and asked to click where they thought the vanishing point was in 640×480 versions of each of the 16 images. 16 subjects completed the study; 11 (4.3%) of their choices were manually removed as obvious misclick outliers. The figure indicates the distribution of human choices with red 3σ error ellipses, most of which were fairly tight. The mean (median) positional difference at the 320×240 scale between our algorithm's estimates and the human choices was 7.8 pixels horizontally (5.3) and 8.0 pixels vertically (4.6).

C. Ladar-based Lateral Offset Estimation

Given θ_{road} from the visual vanishing point tracker ($\theta_{road} = 0$ is straight ahead), we next compute a *maximum a posteriori* estimate of the lateral offset δ_{road} of the gap in the *obstacle map* in front of the vehicle that corresponds to the road. Figure 5 diagrams the steps of the lateral offset calculation from an overhead perspective for a sample scene (frame 075 of the sequence in Figure 7). We call the line defined in vehicle coordinates by (θ_{road}, δ_{road}) the *road centerline*.

First, an obstacle map (plotted at the top of Figure 5(b) as a set of red points) is constructed by transforming all ladar hit points over all registered ladars to vehicle coordinates and filtering them with a "danger" criterion. The criterion is a large absolute elevation difference from the height of the bottom of the vehicle's tires (we use 0.5 m as a threshold). Any such point is treated as evidence of a possible hazard that cannot simply be driven over: either a too-tall positive obstacle or a too-deep negative obstacle. The primary SICK ladar used to populate the obstacle map is mounted on the bumper about 0.5 m above the ground and pitched horizontally, as shown in Figure I(a). We experimented with a secondary ladar mounted on the roof (also shown in Figure I(a)) about 2 m above the ground and pitched down about 5-10 degrees to detect negative obstacles, but it was not used in any of this paper's results because of pitch calibration issues.

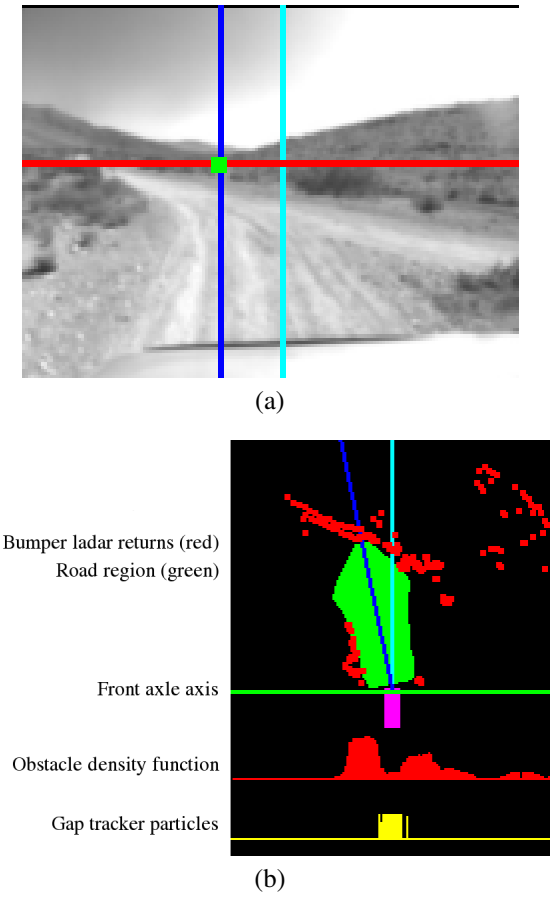


Fig. 5. Road region estimation process: (a) Camera image of sample scene with vanishing point estimate overlaid; (b) Steps of ladar gap tracker. Bumper ladar-identified obstacles (red points) are projected along θ_{road} (direction of diagonal blue line) onto the front axle axis of the vehicle (green line). The estimated obstacle density in this projection is graphed in red below, with the particle filter particle distribution shown in yellow below that. δ_{road} is indicated by the intersection of the centerline with the front axle axis—here very close to 0.

To find the road-induced gap in the obstacle map, obstacles are projected via parallel projection along the vanishing-point-based estimate of θ_{road} (the diagonal blue line in Figure 5(b)) onto the axis defined by the front axle of the vehicle. Intuitively, we seek a segment along this axis that is at least as wide as the vehicle width $w_{vehicle}$, has minimal projected obstacle density, and is close to the vehicle's current center. Under the assumption that the vehicle is currently on the road, the purpose of favoring gap hypotheses in its immediate vicinity is to avoid swerving toward attractively empty areas beyond the road boundary.

The obstacle density $D(x)$ is assessed at a particular lateral offset x along the axle axis ($x = 0$ marks the vehicle center) by summing the number of projected obstacle points $\mathbf{p}$ in a roughly road-width-

sized neighborhood around x. We count nearby obstacles more than distant ones toward the steering decision by exponentially weighting obstacle points by their z distance from the axle axis $\mathbf{p}_z$, yielding the formula $D(x) = \sum_{\mathbf{p} \in [x-\Delta, x+\Delta]} e^{-\lambda \mathbf{p}_z}$, where we use the values $\Delta = w_{vehicle}$ and $\lambda = 0.05$. The projected obstacle density for the sample scene is graphed in red in Figure 5(b) under the projection axis—note the two humps corresponding to the berms and vegetation that flank each side of the road.

In practice, the projected obstacle density is not computed at regular intervals as in the figure, but rather sampled randomly with a particle filter [33]: the *gap tracker*. Each particle i's 1-D state x_i ($n = 100$ particles are used with a sampling variance of $\sigma^2 = 0.1$) represents a hypothesis about the road gap location; its likelihood is measured as:

$$p(x_i) = \begin{cases} e^{-\gamma D(x_i)} & \text{if } -\Delta \leq x_i \leq \Delta \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $\gamma = 0.1$. Gap hypotheses more than a vehicle width (the value of Δ) from the vehicle center are assigned 0 likelihoods in order to prevent swerves as mentioned above. This likelihood function produces correct behavior when the actual road width is less than 2Δ, and when the road width is more than 2Δ the estimated centerline will tend to be near one side of the road. Though not the true road centerline in such cases, it is still suitable for safe driving.

The distribution of gap tracker particles for our example is graphed in yellow in Figure 5(b) beneath the projected obstacle density. More likely particles are drawn as taller vertical lines. The gap estimate derived from the particle distribution via a weighted sum is shown by the intersection of the blue centerline with the axle axis. In the figure's example, the lateral offset estimate δ_{road} is close to 0, indicating a belief that the vehicle is well centered. Note the importance of the road direction: if obstacles were projected straight down onto the axle axis without regard for it, the lateral offset would be biased in a curved road segment to force the vehicle toward the inside of the curve, a dangerous inaccuracy.

The road centerline can be augmented with road width estimates at n discrete points along it to obtain the *road region*. At the ith point along the centerline ($n = 25$ with an interval of 2 m between points here), the road width w^i_{road} is inferred as the diameter of the smallest circle that contains k points in the obstacle map. We chose $k = 3$ empirically to somewhat mitigate ladar sensor noise, and limit the maximum circle radius to 5 m when there are $< k$ obstacles in the vicinity. The road region terminates at the mth point ($m < n$) if $w^m_{road} < w_{vehicle}$. For speed of rasterization, rather than calculating the union of all such circles we use a polygonal region with vertices defined by the endpoints of the circle diameters orthogonal to the centerline. A sample road region calculated in this fashion is drawn in green in Figure 5(b).

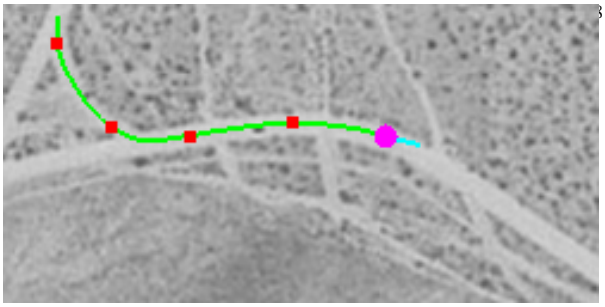


Fig. 6. Satellite photograph of the road segment driven in Figure 7. The GPS history of the vehicle position is overlaid in green. Frame 000 was captured at the position marked by the upper-left red dot; frame 300 was captured at the lower-right purple dot.

An example of *RoadCompass* running passively while the vehicle is under manual control is shown in Figure 7 (Figure 6 is a zoomed 1 m / pixel aerial photograph of the area where the vehicle was driven from left to right with frame locations marked). Note the shortening of the road region as the turns are entered in frames 075 and 225, and its lengthening as the road ahead straightens in frames 150 and 300. The estimated lateral offset shows the vehicle to be right of center in frame 000 and left of center in frame 150. In the camera images, *RoadCompass* can be seen to anticipate both the left and the right turn, and the vision-based θ_{road} estimates agree well with the obstacle map.

A phenomenon not shown in these data but worth discussing is what happens when Alice hits a bump or dip at speed. In such situations the vehicle may pitch down slightly, causing the bumper ladar to "see" the ground plane itself as a line of obstacles stretching across the road. This causes the road region to be erroneously shortened, but only briefly and without negative consequence. Because the obstacle map is not integrated over multiple frames no phantom obstacles remain after the vehicle suspension rights itself.

D. Vehicle Control

We have experimented with two methods of controlling Alice with the parameters computed above. The first, *trajectory following*, connects *RoadCompass* directly to Alice's low-level controller, which attempts to reach a series of closely-spaced UTM waypoints with a specified velocity at each one (see Section

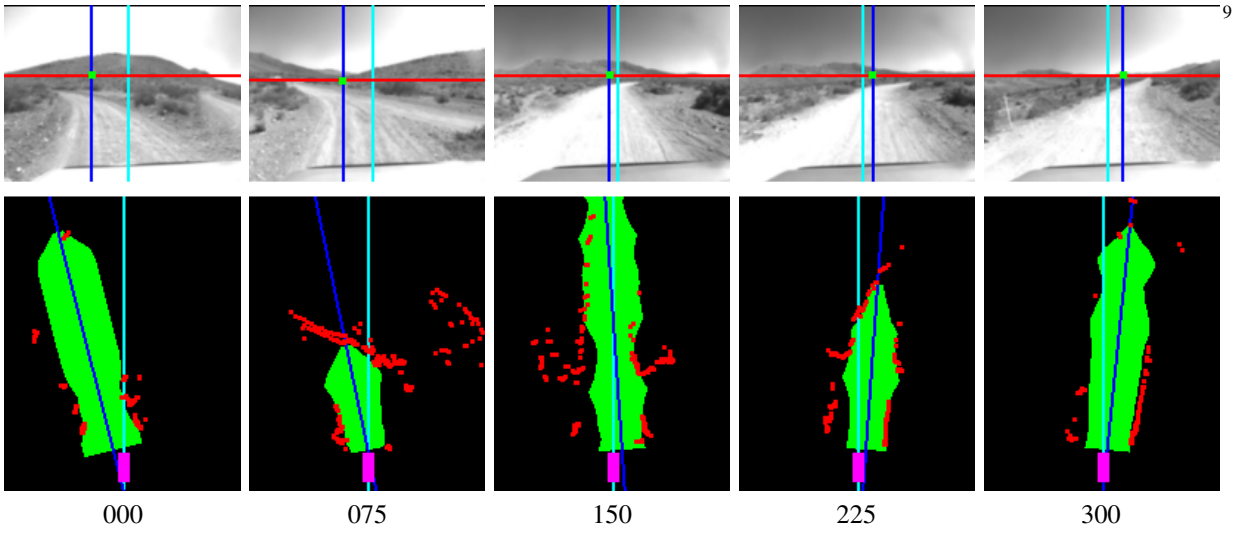


Fig. 7. Estimated road shape from manual driving around an S-turn. Illustrative data 75 frames apart (about 2.5 second intervals) are shown, with the calculated vanishing points overlaid on the camera images and the road offset and width calculations shown below. The scale of the obstacle map images is 40 m wide and 50 m in front of the vehicle (which is about 2 m wide and 5 m long).

3.4 of [4] for details). Under this method, the road centerline defined by θ_{road} and δ_{road} directly parametrizes a linear trajectory relative to the current vehicle position. As these values change the trajectory is refreshed, allowing the vehicle to smoothly go around curves while maintaining clearance from the road edges, since only a fraction of the old waypoints are reached before new goals are set. Caps on the speed targets are set before the run and reduced in proportion to the magnitude of θ_{road} to cause slowing around curves. In practice, the vehicle-centric trajectory coordinates are converted to global coordinates by adding the vehicle's current GPS- and INS-derived estimate of its UTM coordinates.

The second control method, which we call *cost painting*, allows *RoadCompass* to interact with Alice as one of a set of weighted opinions about how to steer (other factors include obstacle detections, RDDF boundaries[2], and so on). The chief difference here is that *RoadCompass* paints its estimated road region into a joint *cost map* which Alice's overarching planner (Section 5.1 of [4]) generates trajectories from. When *RoadCompass* is the only input to the map, the planned trajectories are virtually the same as those of the method described above.

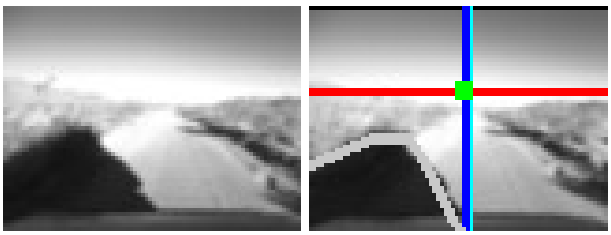
III. Failure Detection

There are a number of visual situations that can cause the vanishing point estimator described above to fail. These fall into two general categories: (1) poor lighting conditions, and (2) non-road images. In the first case, because of darkness, strong shadows, or sun glare, it may be difficult or impossible to accurately infer the road vanishing point. The second failure condition, non-road images, occurs when the vehicle comes to a 90-degree turn, starts off the road, or leaves the road—for example, when driven manually, or because the GCE route description requires it—and there is no vanishing point or obstacle gap to find.

When either type of situation is encountered, if *RoadCompass* is in trajectory following mode (i.e., in complete control of the vehicle) it simply halts for safety. If it is in cost painting mode (i.e., operating as a part of the complete Alice system) it signals the other modules that it cannot offer a confident road shape estimate. When failure conditions are determined to end, *RoadCompass* restarts with its particle filters initialized in uniform distributions in order to refind the vanishing point and obstacle gap *de novo*.

Because we knew that *RoadCompass* would run the 2005 GCE in the daytime in cost painting mode along with many other modules, we set our failure thresholds relatively low in the belief that other parts of the system would pick up the slack should it go off-line. Had *RoadCompass* been intended to always have sole control of the vehicle, we would of course have "hardened" its ability to operate in adverse visual conditions and find roads not visible directly in front of it. In [23] we reported results on an aerial image-based road tracker and detection module which would have helped substantially in this regard. However, this component was

[2]The DARPA *RDDF* (Route Data Description File) used for the 2005 GCE specifies a sequence of waypoints to visit and the widths of "corridors" between them that the vehicle must stay within. For the 2007 Urban Challenge information about intersection locations was added and it was called the *RNDF* (Route Network Description File).



(a) (b)

Fig. 8. Shadow failure conditions: (a) A problematic image with the vehicle shadow cast onto the road in front of it; (b) Same image with outline of predicted vehicle shadow location in image overlaid.

not deemed necessary for the final configuration of *RoadCompass* and was not used in any of this paper's results.

A. Darkness

A number of Alice's other modules are completely ladar-based and thus can operate at night. *RoadCompass*'s vanishing point detection does not work, of course, with no light and performs poorly in twilight when contrast is too low for reliable dominant orientation estimation. To test for such conditions, we use a simple astronomical calculation of the sun's altitude angle β_{alt} above the horizon (using code from [34]) and see if it is over a threshold. This calculation requires the current date and time and the vehicle's UTM coordinates. Such a sun-position-based *darkness test* effectively sets a daily window of "operating hours" for *RoadCompass* regardless of where and when it is running.

Based on empirical observation over many days of testing in terrain where distant hills raised the effective horizon, we set the threshold to 5 degrees, which is about 30 minutes (at the latitude where testing was conducted and the 2005 GCE was run) from the end of so-called "civil twilight" defined by $\beta_{alt} = -6$ degrees. We found this approach to be satisfactory for the conditions that we tested in, but an image-based test could be superior by also detecting darkness caused by tunnels, shadows cast by nearby steep terrain, and cloudiness or other inclement atmospheric conditions. The *no road test* described below offers some coverage of such situations.

B. Shadows

RoadCompass does not have problems with strong shadows cast by power towers, trees, and other tall objects near the road *per se*[3]. It depends on their angle: shadows cast *across* the road tend not to matter because the dominant orientations induced by their edges vote for vanishing points well outside the search region C. Examples of this can be seen in the second column, first row of Figure 4 and in the frames at km 20, 40, 70, 75, and 185 of Figure 20. Rather, it is shadows that are cast nearly *along* the road, contradicting our assumption that the road vanishing point is the only strong one in C, which are problematic. Fortunately, these are quite sparse in desert environments and thus usually outliers in the dominant orientation voting process as the vehicle passes by; empirically, they have not been a problem during extensive testing.

The most pernicious situation occurs when the sun is behind the vehicle and low enough to cast the vehicle's *own shadow* far along the road in front of it, as in Figure 8(a). When this happens, the edges of the triangular region of the vehicle shadow can form two converging lines with their own vanishing point. If the shadow edges are strong enough and the actual road vanishing point is close enough to the phantom vanishing point, the vanishing point tracker may be distracted and "hop" from the road peak in $votes(\mathbf{v})$ to the vehicle shadow peak. When this happens, *RoadCompass* will direct the vehicle to follow its own shadow until and unless the vanishing point tracker hops back to the road peak, possibly leading to a road departure.

One possible approach to mitigating this problem is to try to explicitly detect and remove such shadows in the image (as, for example, in [35] and [36]) as a pre-processing step before dominant orientations are calculated. We have not attempted to implement such a technique, since we deemed it a less reliable and more computationally-expensive option than an astronomical prediction of whether the vehicle shadow is "dangerously" close to the tracked road vanishing point. Furthermore, we do not try to explicitly decide whether the vanishing point tracker has actually hopped from one peak to the other, but instead simply threshold on the angular distance between the predicted shadow tip and the road vanishing point.

As with the *darkness test* above, we calculate the sun altitude β_{alt}, and also the sun azimuth or compass direction β_{azi}. The latter variable is converted to a vehicle-centric value $\hat{\beta}_{azi}$ using the vehicle's heading so that $\hat{\beta}_{azi} = 0$ is always straight ahead. A similar conversion of β_{alt} using Alice's pitch angle could be done to improve shadow prediction on up- and down-hill grades, but we did not actually turn on this option for the race and thus assumed approximately level, planar ground. The first part of the *shadow test* is similar to the *darkness test*: is the sun low enough to cast a vehicle shadow that will be long enough to be anywhere near the horizon—the y component of the road vanishing point—

[3]Shadows from distant terrain features like hills and mountains have very soft edges, and thus are more important as causes of global illumination changes.

in the image? Empirically for our camera calibration, we set this to be true when $\beta_{alt} \leq 15$ degrees. The second part of the *shadow test* is whether the lateral distance between the predicted shadow tip and the road direction is small enough that the tracker might be distracted. We determined this to be true when $|\hat{\beta}_{azi} - \theta_{road} - 180| \leq 30$ degrees. Finally, because the vehicle heading estimates are somewhat noisy, we temporally filter both parts of the *shadow test* such that both must have been true in at least 3 of the last 10 frames for *RoadCompass* to stop outputting road shape estimates.

For visualization purposes, we can predict the shadow region in the image by modeling the 3-D shape of Alice as a rectangular box and projecting its 8 corners onto the ground plane via parallel projection from the direction $(\beta_{alt}, \hat{\beta}_{azi})$. The convex hull of the projected points defines a bounding polygon for the shadow. With the camera's internal and external calibration, we can then backproject this region into the image and see how well it agrees with the actual shadow. An example vehicle shadow prediction is outlined in gray in Figure 8(b).

This sun-position-based *shadow test* is likely much more conservative than a good image-based one would be since it does not take weather or terrain into account. If conditions are cloudy, shadows may be weakened enough that no sun position will cause problems. Also, if the ground is not planar, the shadow edges may not be visible (e.g., they could be cast over the edge of a cliff) or non-straight (e.g., distorted by bumpy vegetation or rocks) and hence not capable of forming a distracting vanishing point.

C. Sun glare

Almost the opposite problem of the shadow issue described above occurs when the sun is low and in front of the vehicle such that it is visible to the road-following camera. Such a sun position causes many pixels on the camera CCD to saturate, and often results in a vertical striped *blooming* pattern such as that exhibited by the image in Figure 9(a). When this occurs, the stripe introduces very strong spurious edges that bias the vote function and induce a phantom vanishing point along it. If not detected or corrected for, *RoadCompass* could erroneously follow the sun instead of the road.

We do not use an astronomical approach for the *glare test* for two reasons. First, since glare can occur for larger values of β_{alt} than the shadow issue, a similarly conservative solution like shutting down when the sun is close to being visible would result in too few hours of operation for *RoadCompass*. Moreover, the simple presence of the sun in the image does not always cause distracting blooming. Figure 9(c), for example, shows how many sky pixels may be saturated without affecting the appearance of the ground texture. Second and most importantly, the alternative of an image-based test is much more feasible because detecting saturated pixels is trivial compared to detecting shadow pixels.

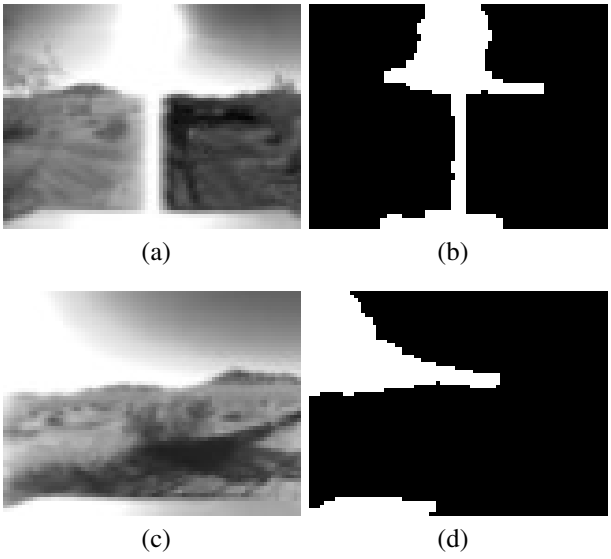


Fig. 9. Sun glare failure conditions: (a) Vertical-stripe blooming saturation caused by direct viewing of sun; we recognize such conditions and turn off road following during them; (b) Mask of saturated pixels in (a)'s image after dilation; (c) About two seconds later the sun is leaving the image as the vehicle turns. The sky is very saturated, but the vertical stripe is not there as shown in (d)'s saturated pixel mask of image (c). *RoadCompass* accepts such images.

The first step of our approach is to find all saturated pixels in the current image and perform a single dilation using a 3×3 structuring element. The result after this step is shown for the example images in Figures 9(b) and (d). Simply counting saturated pixels is not sufficient, as Figure 9(d) has more total saturation than Figure 9(b) (17.2% of the image vs. 16.1% after dilation), yet it is not the problem image of the two. Instead, we look explicitly for a vertical stripe by checking whether any *column* of pixels in the image is nearly all (more than 80%) saturated. Figure 9(b)'s worst column is 100% saturated, whereas Figure 9(d)'s worst is 55%. Finally, this instantaneous measurement is then temporally filtered like the *shadow test*, with at least 3 of the last 10 images required to exhibit the stripe for the *glare test* to be true.

D. No road

The final failure condition that we detect is the *no road test*, for when the vehicle cannot see a road ahead to continue following. This may happen if the road dead-ends or peters out, if a T-intersection or right-angled turn is approached and the continuation of the road is outside the camera field of view, or if the vehicle should for any reason leave an otherwise normal

road. This is the most important failure test because traveling off-road inadvertently is the worst thing that *RoadCompass* could allow to happen, and the thing that the other tests are supposed to prevent. If they do not work, an accurate *no road test* provides a second level of security to prevent a mishap. Put another way, the darkness, shadow, and glare tests address the most common *causes* of *RoadCompass* failing; the *no road test* actually senses *when* it has failed, regardless of the reason.

An obvious approach to this test which we discussed briefly in the introduction is to use a digital road map and the vehicle's GPS-derived UTM coordinates to check whether the vehicle has strayed from a road via some threshold on distance. There are two reasons why we did not use this approach. First, available digital road maps have the best coverage in urban and developed areas and considerably less in the desert areas where testing was conducted and the GCE was expected to occur. We did spot checks of the free U.S. Census Bureau's TIGER/Line database (2004 2nd edition) and found it to have totally inadequate coverage of our testing areas in the Mojave desert of Southern California. The commercial map products of companies such as NAVTEQ and TeleAtlas are more complete; checking Google Maps when it was based on NAVTEQ, we found very good coverage for all roads in our major testing area. Because the cost of licensing such a database was very high, however, we did not pursue this option further. A related option, which we studied in [37], [23] and has been a subject of research by many others (e.g. [38], [39], [40]), is automatic extraction of road networks from satellite imagery. As mentioned at the beginning of Section III, we did not use this module in *RoadCompass*'s final configuration.

The second reason we did not use a map-and-GPS approach to the *no road test* is the imprecision of the vehicle's position estimates. Traveling on dirt roads often only a few meters wider than the vehicle, small lateral positional errors could put the vehicle outside of the road when it really is not (or vice versa), and occasional losses of GPS lock could temporarily blind the system. An image-based test would be much more sensitive to small position and heading changes, and immune to interruptions in GPS coverage.

A system for classifying images as containing a road or not could be trained from examples, but through previous experience with classifier-based road segmentation [10] we felt that variability in road material, illumination, and weather would make achieving true robustness difficult. Instead, we derived a definition of road-containing images from *RoadCompass*'s vanishing point finder that has turned out to be very general and reliable. The intuitive idea of the approach is that a

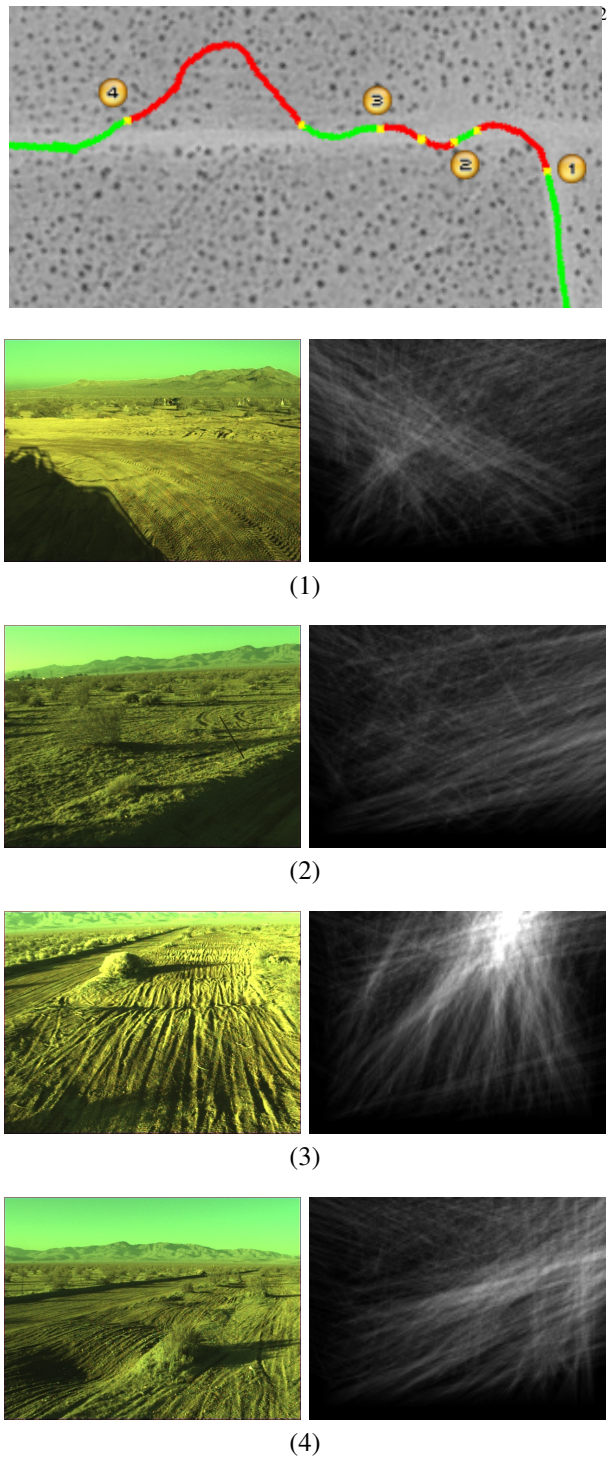


Fig. 10. *No road test* results for a segment of Team Caltech's 2004 GCE route (direction of travel is up and to the left). In the top figure red segments are "no road" and green are "road" (the image covers about 300 m by 150 m). The bottom four images were captured at the numbered locations in the top image; their corresponding vote functions are shown at right. Images 1 and 2 begin no road segments, while 3 and 4 begin road segments.

road image easily trackable by our algorithm must have a strong vanishing point, so we test the vote function $votes(\mathbf{v})$ to see if it has a strong peak. As we asserted at the beginning of Section II-B, rural scenes without roads tend to have a more random distribution of dominant orientations, and thus should have a flatter, more uniformly distributed vote function. Thus, the sharpness of the peak in the vote function should serve as an indicator of the reliability of any estimate that we derive from it.

One way to assess the peakedness of the vote function is to compute the Kullback-Leibler (KL) divergence [41] between it and a uniform distribution of the vote totals.[4] The KL divergence is a pseudo-metric for measuring the dissimilarity between two probability distributions p and q. It is defined for discrete p, q in one form as

$$KL(p, q) = -\sum_k p_k \ln \frac{q_k}{p_k}$$

Here the probability distributions are over the 256 possible vote totals at each pixel for an 8-bit, saturating accumulation buffer, so $p_k = \frac{1}{256}$ and q_k is taken from the vote function.

Low KL values are obtained when many different vote totals are observed in the candidate region, while high values are measured with bunching of vote totals at either the high or low end. We decide that an input image is "road-like" when the KL of $votes(\mathbf{v})$ is over a threshold and the estimated vanishing point can be considered reliable. Because of how we treated zero-probability vote counts q_k in an early implementation (not using proper smoothing techniques [42]), computed KL values were not necessarily non-negative. This resulted in a shifted range for the *road confidence* that did not appear to affect the accuracy of the method, and explains our empirically chosen threshold -0.1. We have kept this shifted scale for consistency of comparison over several years of logged values.

To temporally smooth this decision, we examine the last 100 results of the road confidence threshold comparison (about 3.3 seconds worth of images) and require 50% of them to be over threshold for the estimated road geometry to be passed on to the vehicle controller.

Some results demonstrating the efficacy of this criterion are shown in Figure 10. These are taken from Caltech's 2004 GCE logs near the start of the course. The vehicle was driven on, off, and around a road for several hundred meters, crossing the road several times. The output of the *no road test* is graphed over the vehicle's track with red indicating off-road segments and green on-road. Note that the test is not strictly about where the vehicle currently is, but rather what it sees immediately ahead. Thus the end of the road at the T-intersection is anticipated, and the road is not re-recognized until the 90-degree turn is completed.

[4] An approach using the likelihood ratio was used to make a similar decision about whether a scene had vanishing points or not in [19].

IV. Integrated Tests

We present results for extended runs of *RoadCompass* in three autonomous robotic scenarios. First, we describe how *RoadCompass* was able to successfully drive Caltech's vehicle in pre-GCE testing as the sole controller over 15+ km, with the test only terminated due to unrelated hardware issues. Second, we analyze *RoadCompass*'s performance as a module running live on Caltech's vehicle Alice during the 2005 GCE. Finally, since Alice only covered about 6% of the 212 km course length before being disabled, we ran *RoadCompass* on logged data from Carnegie-Mellon University's (CMU) GCE 2005 entry H1ghlander [26], which completed the entire course. Data for CMU's other vehicle Sandstorm was also available, but not used because its only camera stopped logging around the 200 km mark, just before the difficult Beer Bottle pass section.

A. Pre-GCE testing

RoadCompass has been tested in autonomous mode in a number of situations. In a culminating experiment that took place on the morning of August 4, 2005 in the Stoddard Valley Off-Highway Vehicle (OHV) Area in Southern California, *RoadCompass* commanded Alice's motion directly using the trajectory following method of Section II-D. Modulo several pauses to adjust the maximum vehicle speed, this control scheme safely and smoothly guided the vehicle along two challenging desert road segments. The first segment, whose start and finish are marked with the ① and ② icons respectively in Figure 11)(a), was a 12.1 km-long curving road which gained about 140 m in elevation over its length. The mean speed (including pauses) over the approximately 45-minute run was 16.0 km/h, with maximum sustained speeds of up to 37 km/h.

The second segment, marked with the ③ and ④ icons in the figure, was about 3.4 km in length and included an approach to, ascent, and descent of a 50 m-high hill with a maximum grade of 10%. The apex of the hill is marked with a "+" icon. Manual throttle control was used by the safety driver on the descent because the trajectory follower was not pitch-aware in setting its speed limits. Figures 11)(b) and (c) show views of the approach to the hill and back toward the start of the segment from the summit, respectively. The mean speed over the roughly 15-minute run was 14.0 km/h.

On/off road testing was not used for either run—*RoadCompass* assumed it was on a road at all times. The road segment between icons ② and ③ was driven

(a)

(b)

(c)

Fig. 11. (a) Traces of Stoddard Valley OHV autonomous trajectory follower runs. Map covers a 14.3 km by 7.4 km area. (b) View toward top of hill marked by "+" icon in map; (c) View from summit of hill back toward approach.

manually because it was insufficiently interesting for autonomous testing given hardware-related time constraints (the CPU rack cooling system was malfunctioning that day). Because of drop-outs in the GPS logs of the runs, the vehicle's position had to be manually interpolated to create the figure. The start and stop of each segment is precise, however.

In subsequent experiments, we controlled the vehicle less directly, through the *cost painting* method described in Section II-D. Several more autonomous kilometers were logged on different desert roads in this fashion with *RoadCompass* exercising primary control, and more still with it as part of a larger ensemble of "opinion providers." One key attribute of this method of vehicle control is that higher-level constraints such as GCE course boundaries can override found roads. For example, a sample course visiting all three vertices of the triangle of roads in Figure 11(a) is untraversable in full by the trajectory following method because it requires a turn at each vertex off of the main road onto a secondary road, and *RoadCompass*'s tendency is to stay on whatever road it starts on. Using an RDDF module in conjunction with cost painting forces the vehicle to take a particular choice at a fork regardless of *RoadCompass*'s opinion.

For these experiments and the 2005 GCE, the internal calibration of Alice's road-following camera was obtained using standard techniques [43]. The external calibrations of the ladar-to-vehicle and camera-to-vehicle transformations were performed by hand by

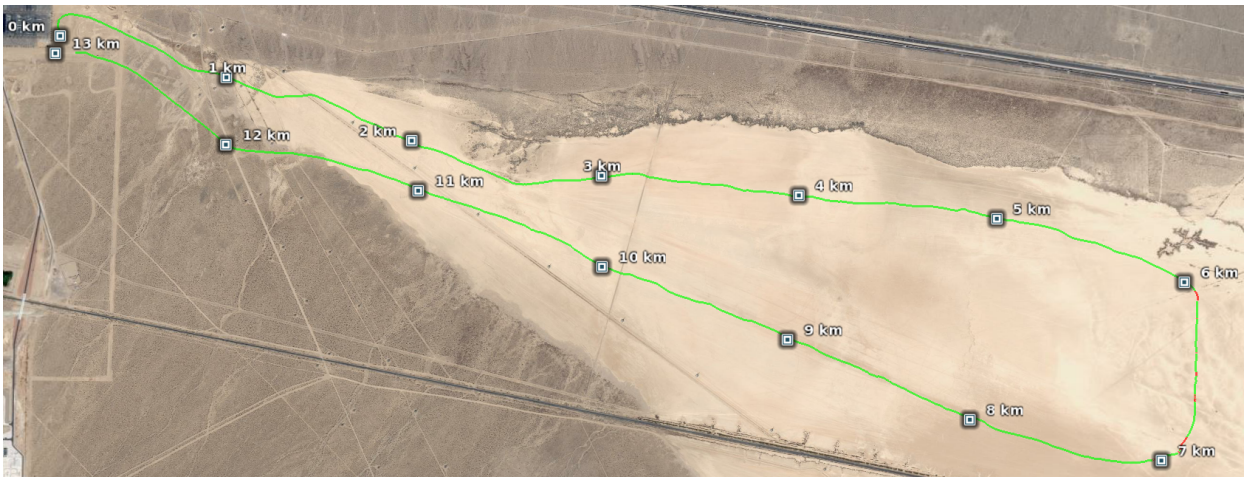


Fig. 12. GPS log of Alice's GCE 2005 race. Markers at 1 km intervals show the directionality of the course. Green locations were classified as road and red as non-road (with a -0.1 confidence threshold). The area shown is roughly 6.2 km by 2.4 km; north is to the right (image courtesy of Google Earth)

Caltech team members. The yaw component of the camera-to-vehicle transformation was then refined using the following self-calibration procedure. First, *RoadCompass* was run on a sequence of images captured while Alice was driven straight down a straight road. Assuming *RoadCompass* to be an unbiased estimator of road direction, the difference between the median x coordinate of *RoadCompass*'s tracked road vanishing point and half the camera's horizontal resolution would be 0 if the camera's optical axis was directly aligned with the longitudinal direction of vehicle motion. This method yielded a correction of about 2 degrees in the camera yaw estimate.

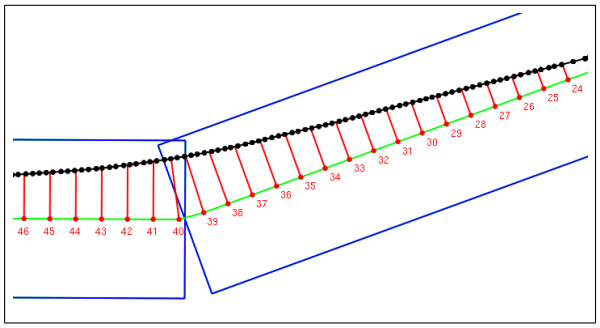


Fig. 13. How vehicle trajectories are converted to distances along course (detail of Caltech data). The raw GPS trajectory is shown in black and the RDDF corridor boundaries in blue. The corridor midline is a green line with red numbered meter markers.

B. 2005 GCE: Alice

The 2005 GCE began on the morning of October 8 in Primm, NV. Based on inspection of logged images, Alice started moving at 8:53:05 am and was paused by DARPA at 9:24:59 am after crashing into a barrier. Over its 31:54 on the course, Alice covered approximately 12.99 km at an average speed of 24.4 km/h. The portion of the course covered by Alice is pictured in Figure 12.

The proximate cause of the crash, described in detail in [4], was a GPS problem experienced as Alice passed under power lines. For the GPS-based statistics that follow, we treat Alice's position estimates as accurate until the 12.90 km course marker.

The sun altitude was calculated at 26.5 degrees when Alice started moving and 31.8 degrees when she stopped, so no darkness or shadow failures occurred, as these are only possible with sun altitudes less than 5 or 15 degrees, respectively.

In order to assess the accuracy of *RoadCompass*'s road direction estimates relative to the baseline performance of each vehicle, we calculated a linear trajectory consisting of waypoints along the RDDF corridor midline at 1 m intervals over the entire 2005 GCE course.

For each such "meter marker", we derive a nominal course heading linearly from the preceding and succeeding marker positions. Each meter marker is corresponded to a vehicle's state variables by intersecting a line through it and orthogonal to the course heading with the vehicle's GPS trajectory, as shown in Figure 13. Position, heading, and other state parameters at that point are linearly interpolated from the bracketing logged values. For efficiency as well as correctness when the course loops, the intersection test is only done on a window of the GPS trajectory $\pm$ a few meters from the last found intersection point.

Let the *heading difference* be the signed difference between the course heading at the corresponding meter

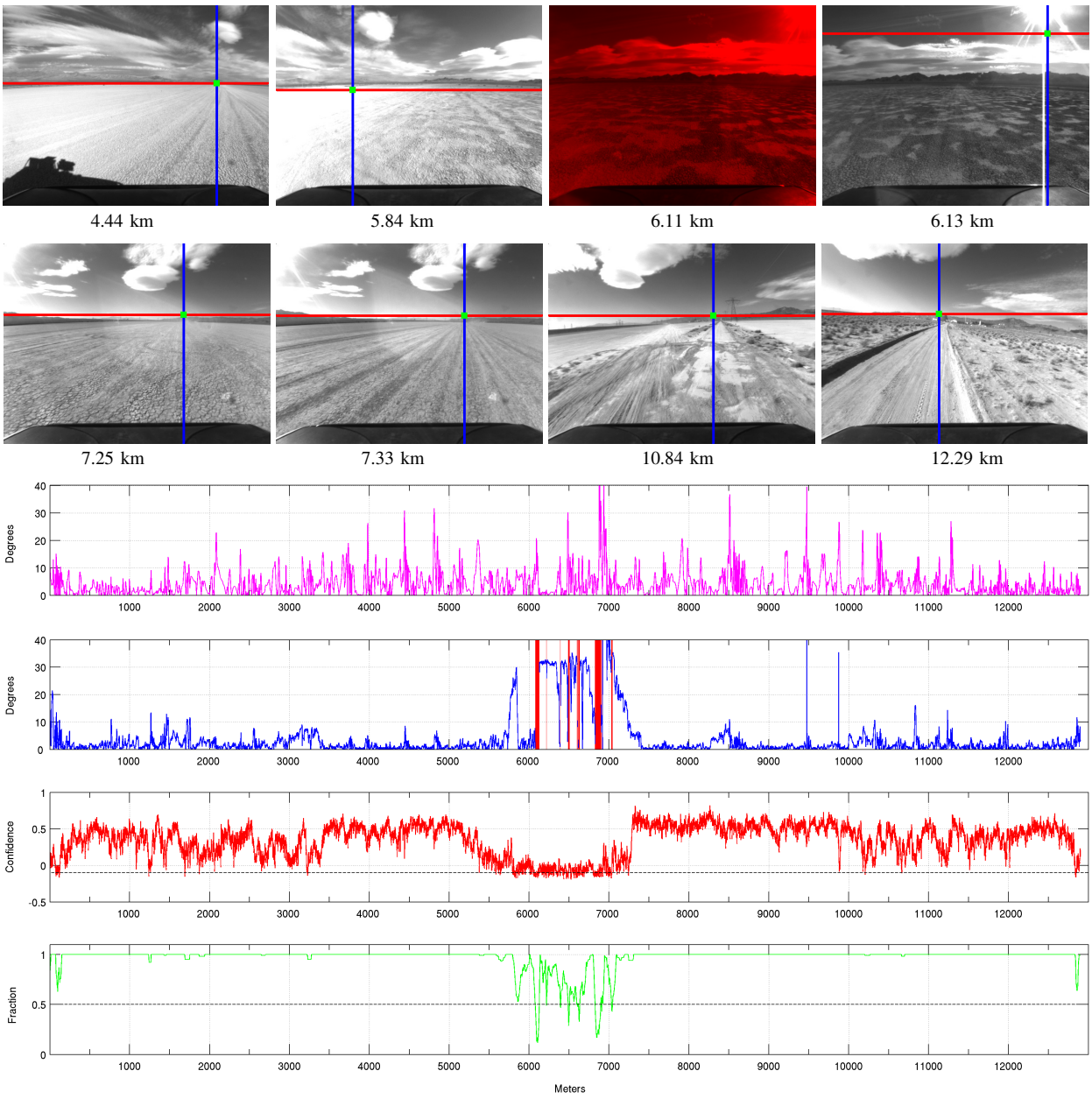


Fig. 14. *RoadCompass* on Alice during GCE 2005. Images at top are discussed in the text. **Top (magenta) graph**: Absolute difference between Alice's heading and RDDF-indicated course direction at each meter marker. **Upper middle (blue) graph**: Absolute difference between Alice's *RoadCompass*-corrected heading and course direction. Sections classified as non-road are shaded red. **Lower middle (red) graph**: KL divergence-derived road confidence for current image. Values over -0.1 (dashed line) are considered high confidence. **Bottom (green) graph**: Percentage of last 100 images with high road confidence. If under 50% (dashed line), current image is classified as non-road.

marker and the actual vehicle heading, with negative values indicating that the vehicle should turn left to correct and positive that it should turn right. Over Alice's 12.90 km on the course, the median absolute value of the heading difference (plotted in the first graph of Figure 14), or *median absolute heading difference* (MAHD), was 2.93 degrees (mean $\mu = 4.20$ and standard deviation $\sigma = 4.34$ after eliminating several spurious values). The *median absolute lateral offset* (MALO) between Alice's position and the corresponding point on the course midline was 3.65 m. While this MAHD sounds small, qualitatively Alice swerved back and forth across the course midline quite a bit, though it stayed within the RDDF corridor boundaries until the crash. For comparison, over the same section of the course H1ghlander followed a much straighter course, with a

MAHD of 0.54 degrees ($\mu = 0.84$, $\sigma = 1.06$) and a MALO of 1.10 m.

Although *RoadCompass* was running on Alice during GCE 2005 and contributing to a combined cost map used for planning, as best the author can tell from personal communications with other members of the Caltech team, its role on race day was essentially limited to raising Alice's speed limit when a road was detected. Thus, we assume that Alice's heading was not correlated with *RoadCompass*'s output, making the following analysis valid.

RoadCompass classified 98.4% of the meter markers over Alice's 12.90 km as "on road" (red bars in the second plot of Figure 14 indicate off-road classifications). While most of that section of the course was on a dry lake bed and not actually a road, it was often heavily marked with tire tracks of vehicles that had previously driven the course, and thus had mostly strong vanishing point (VP) structure.

RoadCompass's road direction estimate θ_{road}, explained in Section II-B, is relative to the vehicle's current direction and can be thought of as an estimated heading difference as defined above. For all meter markers classified as on-road, then, a direct measure of θ_{road}'s correctness is how well it tracks the heading difference. This is captured by a *RoadCompass*-corrected heading difference (CHD) defined as the logged heading difference minus θ_{road}. The median of the CHD (plotted in the second graph of Figure 14) is 1.25 degrees, a 42.7% improvement on Alice's logged heading.

A serious issue that reduced *RoadCompass*'s performance between 6.1 and 7.2 km along the course was the presence of undetected sun glare conditions, causing a run of erroneous θ_{road} estimates. An example is shown in the 6.13 km image of Figure 14. From our analysis, it appears that the glare was not detected by the method of Section III-C because the sun-induced column of saturated pixels which the method looks for was in fact rarely fully saturated, but still bright enough (especially with little other oriented texture along this dry lake section of the course) to attract the vanishing point finder.

Such partially-saturated bright columns were never observed during pre-race testing. We have several hypotheses about why this new phenomenon occurred. First, the race day road-following camera was changed from the one used during pre-race testing at the last minute and may have had different CCD blooming characteristics. Second, the sun's particular altitude, which placed the solar disk partially outside the image and thus somewhat attenuated its strength, may not have been encountered during testing.

In fact, the general rise in the *RoadCompass*-corrected heading error between about 5.75 and 7.25 km

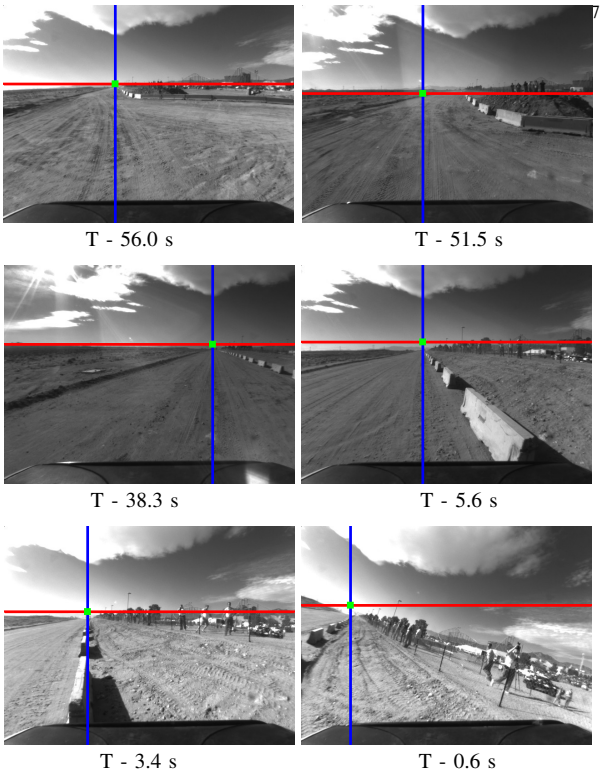


Fig. 15. Samples of *RoadCompass*'s output in the last minute before Alice crashed and was e-stop paused by DARPA. The time beneath each image indicates how many seconds remain until the e-stop

corresponds almost exactly to the northernmost edge of the loop the course takes around the lakebed as shown in Figure 12 . Tire tracks from previous vehicles are clearly visible along the northbound (< 5.75 km) and southbound (> 7.25 km) legs of the loop on what appears to be softer soil, but hardly at all during the west-to-east crossing through the lakebed center, which seems to have very hard-packed ground. When the sun glare does not confuse the on/off road classifier on this section, one can see from the third and fourth graphs of Figure 14 that road confidence is at a sustained low compared to the rest of Alice's run.

Here we briefly annotate the images of Figure 14 showing illustrative moments in *RoadCompass*'s performance:

- 4.44 km: Alice heads left of course direction (error spike on graph); *RoadCompass* correctly sees tracks to right. CHD local maximum at about the same place is probably due to tracking lag as Alice subsequently corrects
- 5.84 km: About the peak of the spike in the CHD corresponding to a hard-caked section of lakebed with fewer visible tracks and thus little VP structure. Also having exposure issues as vehicle

turns toward sun–ground on left side is overexposed. Note how road confidence dips almost to the threshold here
- 6.11 km: Very hard-caked surface–no visible VP structure classified as off road. Sun is about to enter image as vehicle continues turning right
- 6.13 km: Sun is inside image, saturating column below it. This makes *RoadCompass* see VP structure and track it incorrectly
- 7.25 km: Last low road confidence at end of lakebed crossing. The estimated road direction here does correspond to the tire tracks, but the RDDF does not—it is merging at a diagonal
- 7.33 km: Rebound in road confidence as course now parallels existing tracks
- 10.84 km: *RoadCompass* is briefly distracted by road crossing to right, causing rise in CHD
- 12.29 km: On more conventional section of road easily tracked by *RoadCompass* as Alice swerves right onto berm briefly (note tilted horizon)

It is also worth noting that *RoadCompass* performed nominally up to the moment that Alice crashed. Since this occurred after the 12.90 km mark and reliable GPS logs are not available, we document *RoadCompass*'s output by time remaining until DARPA e-stopped the vehicle in Figure 15.

C. 2005 GCE: H1ghlander

To run *RoadCompass* on H1ghlander's logged data, we used its sensors that most nearly matched those used on Alice. These were a DV camera mounted on the passenger-side roof and the lower of two centered, bumper-mounted SICK ladars, all visible in Figure 16 (the camera is in a cylindrical white housing).

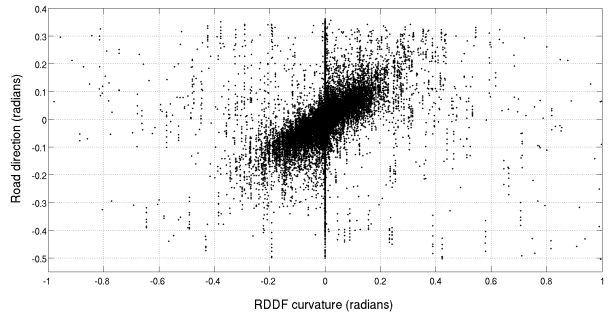
H1ghlander's camera was only used by CMU for documentation purposes, and was not explicitly synchronized to the ladar and state data. However, we were able to infer precise timestamps for image frames by correlating motion in the video with GPS and ladar motion and assuming a standard NTSC framerate. Furthermore, no explicit internal or external calibration was available for the camera. We estimated from the data the two minimal parameters necessary to convert *RoadCompass*'s vanishing point in pixels to a road direction in radians: the camera's horizontal field of view and its yaw relative to the vehicle. The camera's horizontal field of view was estimated by relating the pixel width of a rectangular tunnel mouth imaged at 114.5 km along the course to ladar measurements of its width from the same position. The camera yaw was estimated using the technique described in Section IV-A on all of H1ghlander's 2005 GCE data. A robust method such as this was considered reasonable since from the RDDF the vast majority of the course consisted of straight road segments.

Fig. 16. CMU H1ghlander vehicle during 2005 GCE

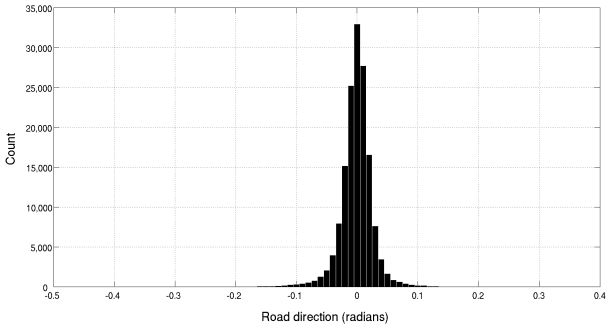
Since H1ghlander's DV camera had a resolution of 720×480 vs. 640×480 for Alice's camera and was yawed significantly to the left, we cropped the leftmost 80 pixels from each H1ghlander image before scaling to *RoadCompass*'s 80×60 processing resolution. After cropping, our estimate of the camera's horizontal field of view was 54.5 degrees. The effective yaw of the cropped images was 4.5 degrees to the left. It should be noted that H1ghlander's hood occupies a non-trivial portion of the lower image, blocking road texture, but this did not seem to diminish performance significantly.

Based on close inspection of its state and camera logs, H1ghlander ran a very clean race over the 212 km from start to finish, staying close to the course midline and heading as it averaged 29.3 km/h not including pauses. Over the entire course H1ghlander's median absolute heading deviation as defined above was 0.63 degrees ($\mu = 0.95$, $\sigma = 1.18$). Nearly all of this is attributable to high curvature segments on which our simple linear method for calculating RDDF headings is a poor approximation rather than any error by CMU's vehicle. H1ghlander's primary difficulties, described in [26], were with velocity control due to engine problems and are not relevant to this analysis.

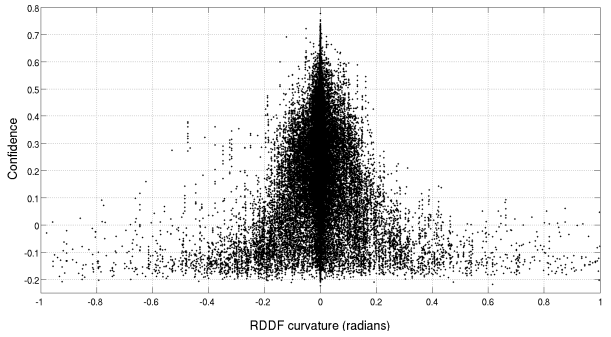
Because of H1ghlander's already-tiny MAHD, we cannot assert as with Alice that *RoadCompass*'s road direction estimates could have been used for course corrections to improve performance. However, based on a several full runs on H1ghlander's logs with virtually the same parameters as used for Alice, there is strong evidence that *RoadCompass* could reliably predict the upcoming road direction in order to steer along almost the entire course *without an RDDF*. Furthermore, *RoadCompass*'s no road test did very well at identifying problem segments such as sharp turns, tunnels, and dry lake beds that lacked strong vanishing point structure.



(a)



(b)



(c)

Fig. 17. 2005 GCE H1ghlander data: (a) Plot of *RoadCompass* tracked road direction vs. RDDF curvature (for images classified as road and sampled at 1 m intervals over entire course); (b) Histogram of road direction estimates for straight road segments (i.e., RDDF curvature of 0); (c) Plot of *RoadCompass* road confidence vs. RDDF curvature

Figure 17 shows several views of *RoadCompass*'s output over one such run to support this assertion. Figure 17(a) plots the road direction θ_{road} estimate (for images classified as road according to Section III-D) vs. the ground truth local road direction (which we call "RDDF curvature"), at every one of the 212,017 meter markers along the entire 2005 GCE RDDF. The RDDF curvature κ is obtained by subtracting the RDDF heading at meter marker i from the heading ahead at marker $i + \Delta$ ($\Delta = 10$ m is a compromise value on how far ahead the camera is looking and which pixels are doing the bulk of the voting for the vanishing point). Under the assumption that H1ghlander is tracking the RDDF closely (which we assert above) and that the RDDF tracks roads closely, values of 0 mean the road is perfectly straight, increasing positive values correlate with stronger rightward curvature, and similarly with negative values. This plot shows a strong, if noisy, zero-centered linear correlation between θ_{road} and the upcoming RDDF heading, with a robustly-estimated slope of the distribution very close to 1. This is what we want: κ-degree bends in the road should yield κ-degree values for θ_{road}.

The data are very sparse for RDDF curvatures outside of $[-0.3, 0.3]$ radians, making it difficult to ascertain how θ_{road}'s variance increases with κ. With the vanishing point candidate region C the same size as the image, for this camera θ_{road} cannot be estimated outside the range $[-0.55, 0.4]$ radians regardless.

Figure 17(b) gives a snapshot of the variance of θ_{road} over a huge variety of different scenes (see Figure 20 for a sample). It is a histogram of θ_{road} over all meter markers for which the road ahead was "very straight" ($|\kappa| \leq 0.001$ radians, $\Delta = 25$ m, bin width 0.01 radians). There were 151,437 of these, or about 71% of all meter markers along the course. This figure is essentially a 1-D "slice" through the probability density function implied by Figure 17(a) which corresponds to the vertical line at $\kappa = 0$. The mean and median are nearly zero (to within 0.01 degrees), and the standard deviation is 1.76 degrees. Thus, for straight RDDF segments it classified as road, *RoadCompass* found the correct road direction to within about ± 5 degrees over 99% of the time.

Figure 17(c) plots RDDF curvature vs. the KL-divergence road confidence measure on which the no road test is based for every image corresponding to a meter marker. It shows that there is a strong positive correlation between RDDF segment straightness and road confidence, evidence that the no road test is indeed diagnostic for sharp turns for which the road direction is out of view. Filtering out such problematic scenes reduces the error in θ_{road}. Using the same road confidence threshold of -0.1 from Alice's run, *RoadCompass* classified 97.5% of the meter markers over H1ghlander's entire run as "on road"[5] Raising the road confidence threshold to -0.05 (based on our analysis of the Caltech results), which excludes more images, changed this to value to 95.2%.

Figure 18 maps H1ghlander's GPS log over the

[5]As the θ_{road} estimates and the various failure detection methods all involve temporal filtering, it should be noted that *RoadCompass* was actually run on all 797,395 captured images between the start and final end of H1ghlander's motion; the meter marker images are simply a state-corresponded subset of these.

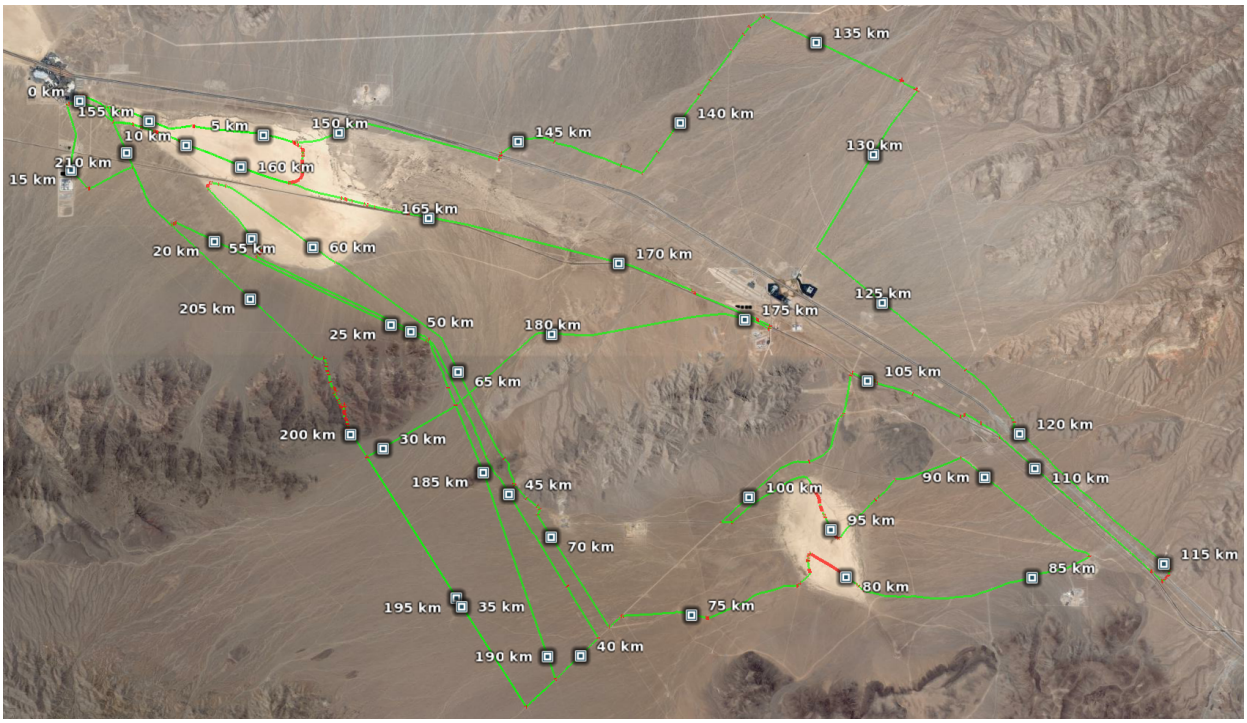


Fig. 18. GPS log of H1ghlander's GCE 2005 race. Locations classified by *RoadCompass* as road are plotted in green and non-road are in red (confidence threshold -0.05). Markers at 5 km intervals show the directionality of the course. The area shown is roughly 31.8 km by 18.3 km; north is to the right (image courtesy of Google Earth)

entire 2005 GCE course on a satellite photograph of the area. Meter markers are colored by their road/non-road classification from the run using the -0.05 confidence threshold. Although the resolution is too low for reproduction of fine detail across the course, it should be apparent that many red "no road" classifications are correlated with sharp turns and corners, as Figure 17(c) implies. Details of a representative approach to and departure from a corner just before 73 km are shown in Figure 19(a).

There were a number of other situations besides high road curvatures that resulted in no road decisions. As discussed in the Caltech section above, hard portions of dry lake beds that did not show tracks well were typically classified as non-road (and softer areas were not). This can be most clearly seen in Figure 18 between 6 km and 7 km and for two crossings of another lake bed around 80 km and 95 km. The three tunnels on the course were also classified as non-road as H1ghlander approached them. Details of the last tunnel just before 146 km are shown in Figure 19(b).

The Beer Bottle pass section of the 2005 GCE course between 200 km and 203 km was universally agreed to be the most technically demanding section for all finishers [25], [26], and it was no different for *RoadCompass*. Very high road curvature, tall rock walls abutting the road, and difficult illumination conditions combined to make this the most challenging segment of this course. *RoadCompass* classified a significant fraction of the pass as non-road as seen in Figure 18, most likely due to high curvatures that made relatively little of the oncoming road visible. Left turns were worse because the cliff on the right blocked visibility, whereas right turns had a drop-off on the left leaving more of the road ahead visible, and this was reflected in the pattern of no road classifications. Details of an early section of the pass are shown in Figure 19(c). The images for 200.27 km and 200.29 km in particular illustrate how in many respects curve approaches were like sharp turns elsewhere on the course. Nonetheless, *RoadCompass* obtained accurate road direction estimates along many of the straighter sections of the pass.

At times, however, *RoadCompass* output erroneously high horizon estimates on approaching cliff walls and underestimated the curvature of approaching left turns (e.g., the images at 200.23 and 200.25 km) before switching to non-road classifications. This may have been partially due to the temporal filtering in the no road classifier—there is a lag associated with the accumulation of evidence that was especially pronounced here as the scene was changing so rapidly.

Illumination conditions may also have played a role

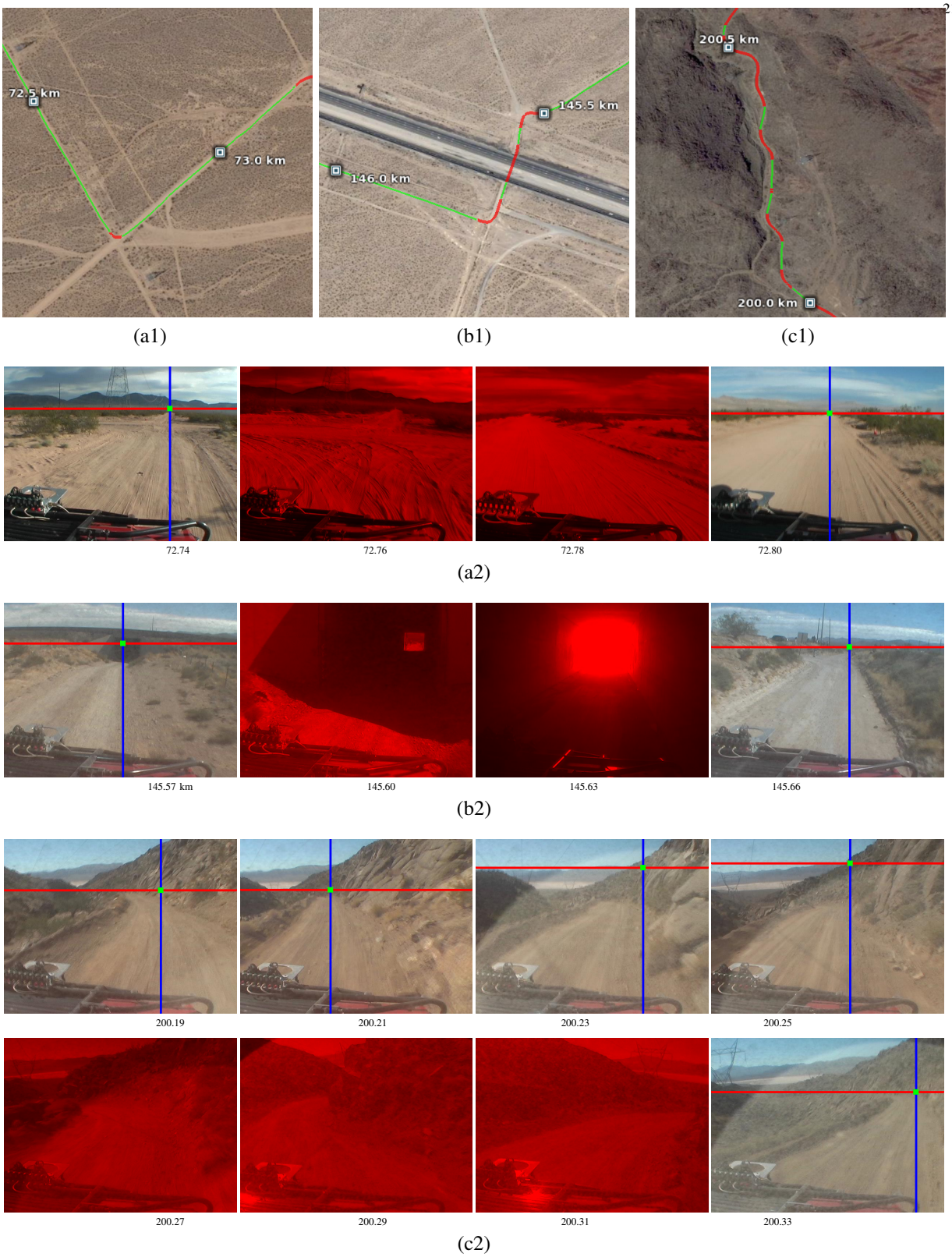


Fig. 19. Details of *RoadCompass*'s output on H1ghlander: (a1-c1) are aerial pictures of areas seen from ground in (a2-c2) sequences. (a) Making a hard left turn ([72.7, 72.8] km); (b) Going through a tunnel ([145.57, 145.66 km]); (c) Beginning of Beer Bottle pass ([200.19, 200.33] km)

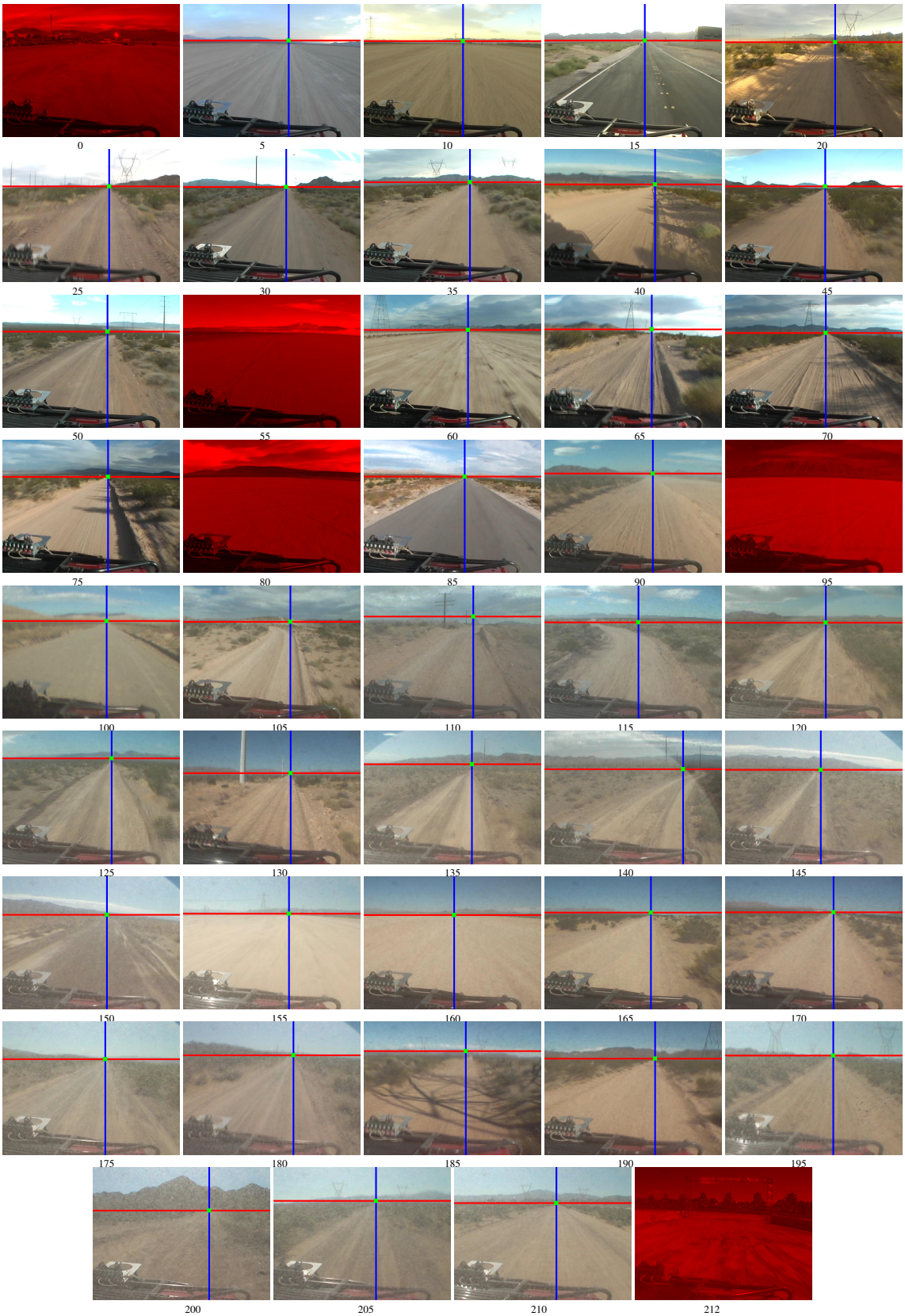


Fig. 20. *RoadCompass*'s output on H1ghlander at 5 km intervals, with km marker number noted below each image (compare to the map in Figure 18). The road confidence threshold used here is -0.05.

in degrading performance. Dust build-up on the camera lens as H1ghlander approached Beer Bottle pass was immense, as the 200 km image in Figure 20 shows. Moreover, for certain headings the late-afternoon sun cast a semi-circular shadow of the camera's housing on the lens that may have biased dominant orientation estimates (e.g., the 200.23, 200.25, 200.27, and 200.33 km images in Figure 19(c2)).

As a final set of images summarizing *RoadCompass*'s performance on the CMU H1ghlander 2005 GCE data, we present a digest of its output along the entire course in Figure 20. The images are samples from fixed 5 km intervals in order to show the range of different terrain and road types encountered, as well as the progression of illumination conditions over the course of the day.

V. Conclusion

We have presented a system for road following on desert and unpaved roads that relies on road texture analyzed from an on-board camera and ladar-based structural information to robustly identify and track the road. The on-board component recovers the road vanishing point in near real-time for many kinds of surface materials with no tuning, and it analyzes its own performance and automatically turns off when the vehicle is not near a road. Based on the experiments run live and on logged data, this is the first system capable of making steering decisions for a vehicle over such a wide range of road types based primarily on vision with no previous training.

Despite the emphasis on desert scenarios here, *RoadCompass*'s algorithmic foundations are applicable to urban imagery such as that encountered by entrants in the 2007 DARPA Urban Challenge, which took place in Victorville, CA on November 3, 2007. Figure 21 shows some results from running *RoadCompass* on image sequences captured by Stanford's robot Junior during qualifying runs. Figures 21(A1-A2) are from NQE Area A, where the ability of vehicles to merge with live traffic on a loop course was tested. A1 shows the road direction estimate along a straight segment, while A2 shows a no road classification at a T intersection. Figures 21(C1-C3) are from NQE Area C, where vehicles' ability to obey precedence rules at four-way stops and to make U-turns when the road was blocked were tested. C1 shows *RoadCompass*'s correct road direction estimate as a blockage is approached in near-glare conditions, while C2 shows *RoadCompass* seeing no road midway through a U-turn. In C3 Junior is driving back the other way around a curve and with shadows across the road.

Versus desert environments, urban roads exhibit several important differences that bear on the effectiveness of an unmodified *RoadCompass*. First, paved roads have less road-oriented texture (mostly just curbs and lane lines), so the vanishing point estimate tends to depend on fewer votes. This allows any cast shadow textures to have a proportionally greater bias on the estimate, and there is generally a higher density of shadows in urban settings due to trees and buildings. While Figures 21(C3) and other results in this paper demonstrate that shadows are mostly harmless, to harden *RoadCompass* against heavily-shadowed areas we would certainly experiment with explicit shadow removal as described in Section III-B. Another factor that would necessitate modifications to *RoadCompass* for effective urban operations is the frequency of intersections and 90-degree turns, which both cause the system to lose its sense of direction. These could be prevented by using an omnidirectional or active-pointed camera. Finally, another vehicle just in front of ours may occlude a large portion of the road ahead. We would want to detect these [44], [45] and either mask them out or switch to a following mode.

There are many other improvements possible to move *RoadCompass* toward a more reliable, standalone system. Because of its development as part of a larger system for Caltech's vehicle, for example, we have not really addressed speed control. Situations requiring sophisticated speed control in addition to steering include not only turns, downhills and uphills, and small (and not-so-small) bumps and dips, but also include changes in the material properties of the road: e.g., sand, gravel, mud, puddles, ice, snow. The algorithm as it stands has no idea that the middle of the road is anything but homogeneous and benign. [46] presents some interesting recent work on this subject.

The core of the *RoadCompass* algorithm could also be upgraded in several regards, including going to a higher resolution for vanishing point voting for additional precision. It would also be interesting to work on a night-operation capability: because *RoadCompass* does no color classification or the like it is very insensitive to lighting conditions, yet our initial experiments trying to run the algorithm with the vehicle's headlights on encountered problems. These are likely fixable with some adjustments to the camera hardware and some software parameters.

Finally, a major area of ongoing work is incorporating aerial imagery and digital elevation data into longer-range planning and anticipation. We have begun promising preliminary work using skeletonization and watershed image processing techniques to extract a road network in the vicinity of the vehicle, offering more choices to the vehicle and possibly graph-based path-planning.

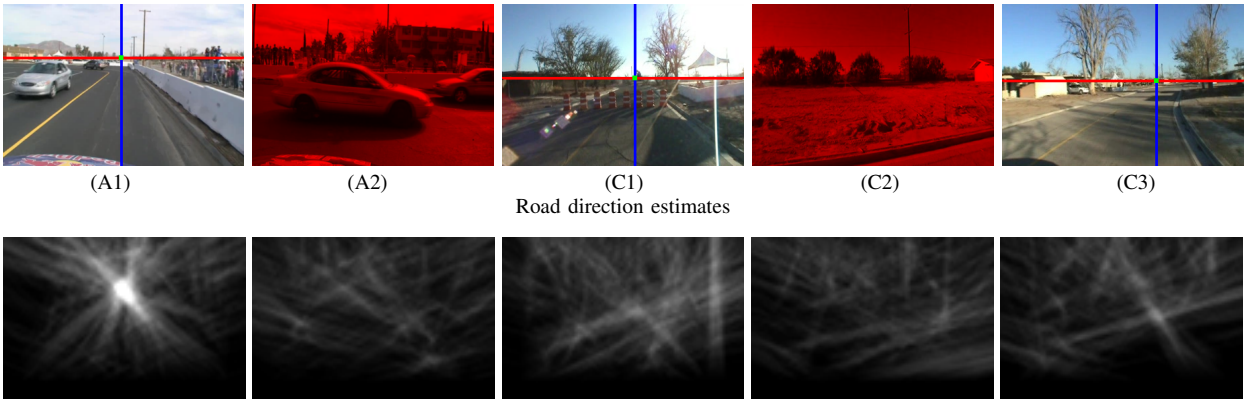


Fig. 21. Results of running *RoadCompass* on image sequences from Stanford's 2007 DARPA Urban Challenge qualifying runs. The first two images are from the merging test in NQE Area A; the last three are from the roadblock test in NQE Area C as Junior (C1) approaches a blockage, (C2) is in the middle of a U-turn away from it, and (C3) subsequently drives away (images courtesy of the Stanford Racing team).

ACKNOWLEDGMENTS

Many thanks to Richard Murray and the Caltech Grand Challenge team for their generous provision of access to and data from Alice and its predecessor Bob over several years. Jeremy Gillula, Gustav Lindstrom, Kevin Duncklee, and Dan Soudek provided invaluable assistance in interfacing the road-following code with the vehicle controller, planner, and ladars.

Thank you very much as well to William "Red" Whittaker of Carnegie-Mellon University for graciously granting permission to use data from Red Team's 2005 GCE entries and to Joshua Anhalt for rounding it all up.

REFERENCES

[1] Defense Advanced Research Projects Agency (DARPA), "DARPA Grand Challenge," Available at `http://www.darpa.mil/grandchallenge`. Accessed July 22, 2003.

[2] J. Coughlan and A. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proc. Int. Conf. Computer Vision*, 1999.

[3] J. Kosecka and W. Zhang, "Video compass," in *Proc. European Conf. Computer Vision*, 2002.

[4] L. Cremean, T. Foote, J. Gillula, G. Hines, D. Kogan, K. Kriechbaum, J. Lamb, J. Leibs, L. Lindzey, C. Rasmussen, A. Stewart, J. Burdick, and R. Murray, "Alice: An information-rich autonomous system for high speed desert navigation," *Journal of Field Robotics*, vol. 23, no. 9, 2006.

[5] C. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.

[6] B. Southall and C. Taylor, "Stochastic road shape estimation," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 205–212.

[7] N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.

[8] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006.

[9] J. Crisman and C. Thorpe, "UNSCARF, a color vision system for the detection of unstructured roads," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1991, pp. 2496–2501.

[10] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.

[11] J. Zhang and H. Nagel, "Texture-based segmentation of road images," in *Proc. IEEE Intelligent Vehicles Symposium*, 1994.

[12] S. Se and M. Brady, "Vision-based detection of staircases," in *Proc. Asian Conf. Computer Vision*, 2000, pp. 535–540.

[13] S. Se, "Zebra-crossing detection for the partially sighted," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, pp. 211–217.

[14] N. Simond and P. Rives, "Homography from a vanishing point in urban scenes," in *Proc. Int. Conf. Intelligent Robots and Systems*, 2003.

[15] A. Polk and R. Jain, "A parallel architecture for curvature-based road scene classication," in *Vision-Based Vehicle Guidance*, I. Masaki, Ed., pp. 284–299. Springer, 1992.

[16] T. Tuytelaars, M. Proesmans, and L. Van Gool, "The cascaded Hough transform," in *Proc. IEEE Int. Conf. on Image Processing*, 1998.

[17] A. Turina, T. Tuytelaars, and L. Van Gool, "Efficient grouping under perspective skew," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001.

[18] F. Schaffalitzky and A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing*, vol. 9, no. 18, pp. 647–658, 2000.

[19] J. Coughlan and A. Yuille, "Manhattan world: Orientation and outlier detection by bayesian inference," *Neural Computation*, vol. 15, no. 5, pp. 1063–088, 2003.

[20] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[21] C. Rasmussen, "A Hybrid Vision + Ladar Rural Road Follower," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2006.

[22] C. Rasmussen, "Texture-based vanishing point voting for road shape estimation," in *Proc. British Machine Vision Conference*, 2004.

[23] C. Rasmussen and T. Korah, "On-vehicle and aerial texture analysis for vision-based desert road following," in *IEEE Workshop on Machine Vision for Intelligent Vehicles*, 2005.

[24] C. Rasmussen and W. Ulrich, "Adaptive road image segmentation from ladar-derived labels," in *Workshop on the DARPA Grand Challenge at Robotics: Science and Systems*, 2006.

[25] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau,

C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley, the robot that won the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, 2006.

[26] C. Urmson, J. Anhalt, D. Bartz, M. Clark, T. Galatali, A. Gutierrez, S. Harbaugh, J. Johnston, H. Kato, P. Koon, W. Messner, N. Miller, A. Mosher, K. Peterson, C. Ragusa, D. Ray, B. Smith, J. Snider, S. Spiker, J. Struble, J. Ziglar, and W. Whittaker, "A robust approach to high-speed navigation for unrehearsed desert terrain," *Journal of Field Robotics*, vol. 23, no. 8, pp. 467–508, 2006.

[27] A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen, "Real-time, multi-perspective perception for unmanned ground vehicles," in *AUVSI*, 2003.

[28] K. Kluge and M. Morgenthaler, "Multi-horizon reactive and deliberative path planning for autonomous cross-country navigation," in *SPIE Defense and Security Symposium*, 2004.

[29] W. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891–906, 1991.

[30] X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *Asilomar Conference on Signals, Systems and Computers*, 2002.

[31] T. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[32] M. Frigo and S. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*, 1998, vol. 3, pp. 1381–1384.

[33] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. European Conf. Computer Vision*, 1996, pp. 343–356.

[34] Square One Research, "Solar position calculator," Available at `www.sq1.com/solar/solar-position.html`. Accessed August, 2005.

[35] M. Ollis and A. Stentz, "Vision-based perception for an autonomous harvester," in *Proc. Int. Conf. Intelligent Robots and Systems*, 1997, pp. 1838–1844.

[36] G. Finlayson, S. Hordley, C. Lu, and M. Drew, "On the removal of shadows from images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, 2006.

[37] T. Korah and C. Rasmussen, "Probabilistic contour extraction with model switching for vehicle localization," in *Proc. IEEE Intelligent Vehicles Symposium*, 2004.

[38] A. Yuille and J. Coughlan, "Fundamental limits of Bayesian inference: Order parameters and phase transitions for road tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 160–173, 2000.

[39] T. Géraud and J. Mouret, "Fast road network extraction in satellite images using mathematical morphology and markov random fields," *EURASIP Journal on Applied Signal Processing*, , no. 16, pp. 2503–2514, 2004.

[40] U. Bacher and H. Mayer, "Automatic road extraction from multispectral high resolution satellite images," in *ISPRS Workshop CMRT: Object Extraction for 3D City Models, Road Databases and Traffic Monitoring*, 2005.

[41] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.

[42] S. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, pp. 359–394, 1999.

[43] J. Bouguet, "Camera Calibration Toolbox for Matlab," Available at `www.vision.caltech.edu/bouguetj/calib_doc/`. Accessed May 11, 2001.

[44] F. Dellaert, D. Pomerleau, and C. Thorpe, "Model-based car tracking integrated with a road-follower," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1998.

[45] S. Avidan, "Subset selection for efficient SVM tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 85–92.

[46] D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," in *Proceedings of the Conference on Uncertainty in AI*, 2006.