

A Hybrid Vision + Ladar Rural Road Follower

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” 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 full control of a vehicle over 10+ miles of difficult roads at up to 25 mph, as well as analysis of logged data in diverse situations.

I. INTRODUCTION

With the running of the DARPA Grand Challenge (DGC) robot races in March, 2004 and October, 2005, there has been a heightened interest in algorithms for autonomously following “difficult” unpaved paths and roads. Nearly all of the 2004 and 2005 courses were narrowly constrained to a series of flat or hilly desert roads, many of which were unpaved and offered little color contrast to the surrounding sandy environment. The DGC courses also went through roadless areas, requiring sensors and methods for general obstacle avoidance and slope analysis as studied in [1], [2], and thus require some decision mechanism as to whether a vehicle is on- or off-road.

In this paper, we describe a perceptual module for following marginal roads that uses a monocular, grayscale camera in conjunction with a SICK LMS ladars to rapidly obtain an estimate of oncoming road structure and transmit appropriate steering and throttle commands to the vehicle controller. The module, which we will call VP_FOLLOW, was developed to operate as part of the overall system for the 2005 DGC team of Caltech, and functions related to higher-level navigation, structural map update, off-road steering, direct control of the vehicle, hardware fault monitoring, etc. reside in modules developed by other team members. In the 2005 DGC, the ultimate action taken by the vehicle in any given situation was a complex function of its state and sensor inputs that is beyond the scope of this paper to describe; the full system is covered in [3]. Thus, we focus here on the workings of VP_FOLLOW in isolation and in conjunction with two simplified versions of the vehicle controller which essentially allow it to completely control the vehicle.

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 [4], [5], [6] are often used 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 [6], [7], [8], [9]. 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 tends to make these methods more appropriate for unpaved rural roads.

We have found empirically that many desert road scenes present difficulties for these approaches, as for example a dirt road through a dry environment is delineated by neither strong sharp edges nor contrasting local color or texture characteristics (e.g., Figure 1(a)). One cue that often identifies such roads unambiguously, however, is their overall *banding* pattern. This banding, often 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.

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 (see Figure 1(d) for an example of VP_FOLLOW’s output on the previously referenced image). The significance of the former, of course, is that the difference between it and the vehicle’s current direction of travel yields an angular error suitable for input to a low-level steering control and acceleration module. As the vanishing point is associated with the road’s tangent, there is no unique vanishing point for curving or undulating road segments [10]. One empirical result we show here is that a constantly updated “mean” vanishing point tangent to approximately the middle of the curve ahead—that is, only a fast-changing linear shape estimate—suffices (along with a lateral offset control discussed below) to negotiate non-trivially curved roads at moderate vehicle speeds.

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

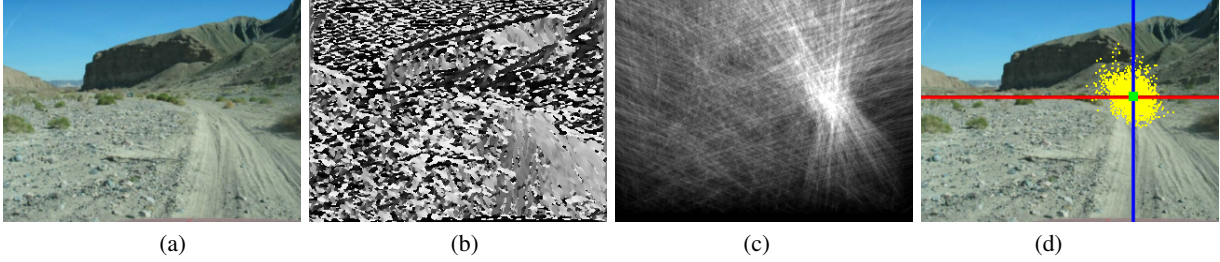


Fig. 1. (a) Captured road image; (b) Dominant orientation at each pixel ($[0, \pi]$ radians $\rightarrow [0, 255]$ intensity values); (c) Vote function for vanishing point; (d) Distribution of particles and estimated vanishing point location with horizon line indicated

[7], [9], and we have experimented with several methods using the vanishing point as a powerful shape constraint [11], [12], but ultimately have found them insufficiently robust. Rather, VP_FOLLOW uses a simple obstacle map derived from SICK LMS ladars to estimate the lateral offset of the road midline relative to the vehicle. By tracking the 1-D “gap” between obstacles (positive and/or negative) on the left and right sides of the road, the vehicle can center itself while decreasing directional error. This approach is insensitive to the road surface material and lighting conditions, and there is no need to tune, learn, or adapt parameters as the algorithm runs on a variety of roads. In flat areas without obstacles to constrain lateral offset (i.e., only visual appearance demarcates the road), any drifting is of less concern than elsewhere because by definition the off-road area is less hazardous.

A critical system state is whether the vehicle currently sees a road and thus its “advice” should be followed or else no road is visible and the vehicle controller should ignore spurious steering commands. The final component of VP_FOLLOW is a suite of methods for detecting failure situations in order to shut itself off as well as to restart itself when visual conditions warrant. The first failure situation we examine is whether the vehicle is currently on a road or not. This is a simple procedure for a GPS-equipped vehicle on highways and urban streets which have been digitized in vector form, but the unnamed access roads and desert paths we are interested in exploiting are mostly unmapped. VP_FOLLOW thus includes an image-based function to discriminate images that contain a strong vanishing point—assumed to belong to a road—from those that do not. Other failure detection functions relate to poor lighting conditions such as sun glare, distracting shadows, and darkness that may prevent the module from recognizing a road even if it is on one.

In the next sections, we will describe the steps of how VP_FOLLOW estimates road shape and introduce its failure detection methods, after which we will show results demonstrating the system’s capabilities and discuss extensions we are currently working on.

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 is measured by having all dominant

orientations in the image vote for a single best road vanishing point. Finally, the vehicle’s lateral offset from the road center is estimated from ladar data.

A. Dominant Orientations

The *dominant orientation* $\theta(\mathbf{p})$ of an image at pixel $\mathbf{p} = (x, y)$ is the direction that describes the strongest local parallel structure or texture flow. We estimate $\theta(\mathbf{p})$ by convolving the image with a bank of Gabor wavelet filters [13] 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:

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

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.

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 [14] 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 1(b) shows the calculated dominant orientations for the image in Figure 1(a). Gray level intensities proportional to an estimated angle from 0 to 180 (in 36 discrete steps) are shown. Observe that most parallel structure is in the dirt road on the right.

B. Vanishing Point Detection

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 $\mathbf{r}_{\mathbf{p}} = (\mathbf{p}, \theta(\mathbf{p}))$. Intuitively, the best estimate for

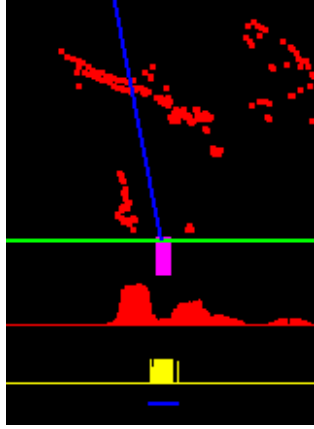


Fig. 2. Steps of ladar “gap” tracker for lateral offset estimation. Ladar-identified obstacles (red points) are projected along road direction (diagonal blue line) onto axis defined by front axle (green line) of vehicle (purple rectangle). The estimated obstacle density in this projection is graphed in red below, with the particle filter distribution and likelihoods shown in yellow below that, and the weighted sum of the particles indicating the gap estimated shown as a blue line segment at the bottom of the figure. [Source image is shown in Figure 6(c); one ladar mounted on the bumper parallel to the ground was used here]

the vanishing point \mathbf{v}_{max} is that point lying on or near the most such dominant orientation rays (see [15], [16], [17], [18] for recent work on vanishing point finding). In [10], 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. 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 is quite enough for our image resolution.

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 [19], [5], [6]. Particles are initially distributed uniformly in order to coarsely localize the vanishing point. Weak dynamics $p(\mathbf{v}_t | \mathbf{v}_{t-1})$ (e.g., a low-variance, circular Gaussian) 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. Finally, 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.

Figure 1(c) shows the vanishing point candidate function for the image in Figure 1(a). The current particle distribution and their weighted mean (i.e., the estimated vanishing point location) are shown in Figure 1(d). The vertical cyan line in Figure 1(d) indicates the vehicle direction. When the road following camera is perfectly aligned with the vehicle direction,

it is midway across the image. In cases where there is some yaw offset between the camera and vehicle coordinate systems, such as with the data in Figure 6, the vehicle direction line may be off the image center.

C. Ladar-based Lateral Offset Estimation

Given the road direction from the visual vanishing point tracker, our goal is to find a *maximum a posteriori* estimate of the lateral offset of a “gap” in the obstacle field in front of the vehicle that corresponds to the road. The obstacle field is defined as the set of ladar hit points (over all registered ladars) in vehicle coordinates that pass a “danger” test. Here the danger test simply consists of having an absolute elevation difference from the bottom of the vehicle’s tires of ≥ 0.5 m.

Figure 2 diagrams the steps of the lateral offset calculation from an overhead perspective with the vehicle (shown as a purple rectangle) traveling up the page. At the top of the figure is the obstacle field graphed as a set of red points (the source image corresponding to this data is Figure 6(c)). Obstacle points are projected via parallel projection along the road direction (the blue diagonal line) onto the line defined by the front axle of the vehicle (indicated by a green line in the figure).

We seek the segment—aka the road “gap”—with the lowest 1-D obstacle density along this line that is both wider than the vehicle width w and in the vehicle’s neighborhood under the assumption that it is currently on the road. The weight of each obstacle point falls off exponentially with its distance from the vehicle so that nearby obstacles count considerably more. The projected obstacle density is graphed in red in Figure 2 under the projection axis—note the two humps corresponding to the berms and vegetation that flank each side of the road.

A particle filter, the “gap tracker,” is used to estimate the obstacle density online. Each particle ($n = 100$ here) consists of a hypothetical $2w$ -wide gap randomly chosen no further than w from the vehicle’s current edges. The purpose of limiting gap hypotheses to the immediate vicinity of the vehicle is to avoid seeing attractively empty areas outside the road but far away and causing the vehicle to swerve abruptly toward them. The likelihood of each particle is inversely proportional to the obstacle density within it. The distribution of gap particle centers for this data is graphed in yellow beneath the projected obstacle density, with more likely particles drawn with longer vertical lines. Finally, the most likely gap estimate derived from the particle distribution is shown as a blue $2w$ -wide segment at the bottom of the figure.

The center of this segment determines where VP_FOLLOW thinks the nominal road centerline (the blue line in the figure) intersects its projection axis, which is treated as the front of the vehicle. The road centerline functions as a linear trajectory for the vehicle controller to follow. Its constantly changing direction and lateral translation are what allow the vehicle to go around turns while maintaining clearance from the road edges.

The road centerline can be further augmented with width estimates at discrete distances along it. These are obtained by expanding circles centered at 2 m intervals along the trajectory until $k = 3$ obstacle points are enclosed within them. The maximum circle radius is limited here to 5 m.

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) non-road images, and (2) poor lighting conditions. In the first case, the vehicle may leave the road—for example, when driven manually, or because the DGC route description compels it to—and VP_FOLLOW must recognize that it should no longer output a road centerline to follow (in the larger DGC system, other modules would continue to offer their opinions of what to do). In the second case, the vehicle may still be on a road, but because of darkness, sun glare, or confusing shadows it may not be able to accurately infer the road vanishing point. Again, we would like to recognize such eventualities and gate the road follower’s output.

a) On-off road inference: 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.

The “sharpness” of the vanishing point peak in the vote function over C , which is generally only slightly larger than the input image, is intuitively an indicator of the reliability of the estimate. There are a number of ways to measure this, but we have found that the Kullback-Leibler (KL) divergence [20] between the vote function and a uniform distribution of the vote totals (256 possible values for 8-bit accumulation buffers) correlates well with this intuition. 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. A similar approach using the likelihood ratio was used to decide whether a scene had vanishing points or not in [18].

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. To smooth this decision, we store a history of the last 5 seconds of results of this threshold comparison and require 1/2 of them to be over threshold for the estimated road geometry to be passed on to the vehicle controller.

There is not space in this paper to examine this criterion in depth, but some results demonstrating its efficacy are shown in Figure 3. A vehicle was driven on, off, and around a road for several hundred meters, crossing the road several times. The output of the KL on-off 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.

b) Sun glare: Although the situation is less common, vanishing point estimation can be fooled by spurious edges caused by saturated pixels blooming on the camera CCD

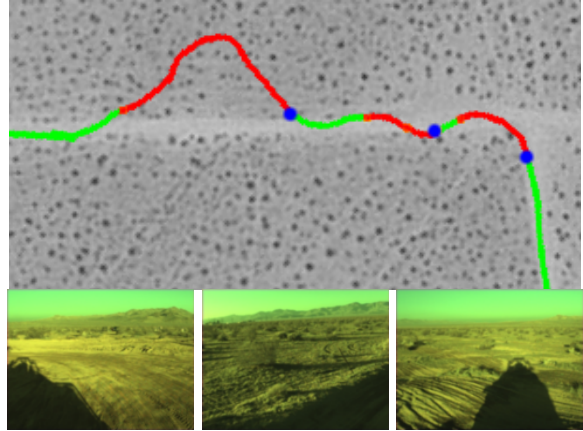


Fig. 3. On/off road decisions for an arbitrarily-driven segment (direction of travel is up and to the left). Top figure is 300 m by 150 m; red segments are where images were classified as “off road”; green are “on road”. Bottom images correspond to first frame of each off-road segment (marked by blue circles).

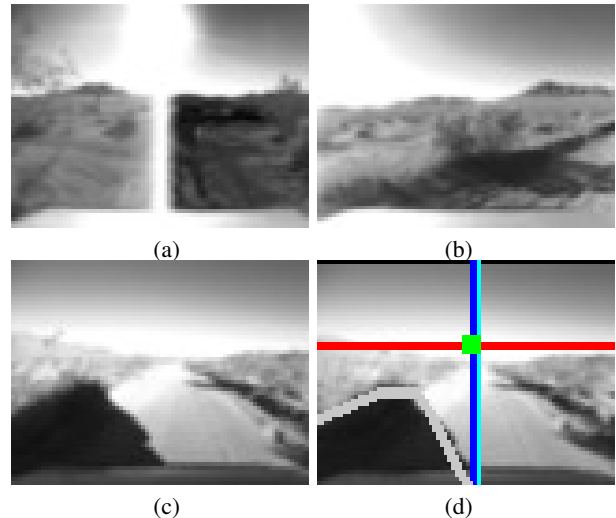


Fig. 4. (a) Blooming saturation caused by direct viewing of sun; we recognize such conditions and turn off road following during them; (b) A few frames later the sun is just out of view. The sky is still saturated, but the vertical stripe is not there. VP_FOLLOW accepts such images; (c) A problematic image with the vehicle shadow cast into the road in front of it; (d) Same image as (c) with outline of predicted shadow location in image overlaid.

from the sun being directly visible in the image. An example is shown in Figure 4(a). VP_FOLLOW uses a straightforward image processing procedure to recognize such occurrences. This consists of finding saturated pixels in the current image, doing a dilation, and then checking for whether any column is nearly all (more than 0.8 of them) saturated. This instantaneous test is then temporally filtered. Simply computing the fractional area of an image that is saturated does not work, as the sky is often saturated without significantly affecting the contrast of ground pixels, as in Figure 4(b).



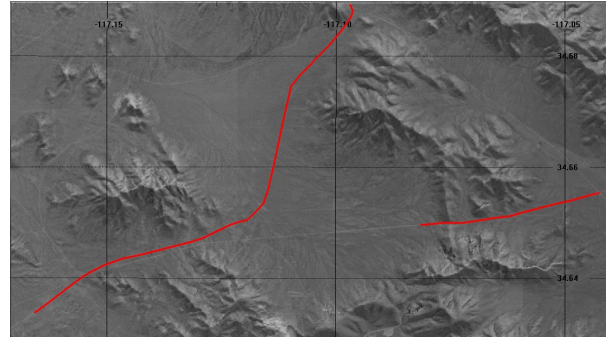
Fig. 5. Robotic vehicle used for testing in this paper. SICK LMS ladars are mounted on the bumper and roof; road following camera is one of stereo pair over windshield.

c) Shadows and darkness: There is a rare situation in which the assumption that the road vanishing point is the only strong vanishing point in the image is contradicted. It 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 (see Figure 4(c)). If the actual road vanishing point is close enough to the point of the triangular shadow, the two may be conflated, leading to a road departure. Detecting and removing shadows in images is still an open topic [21], but we have found that a calculation of the local sun angle from the vehicle state (northing, easting, heading, pitch, date, and time), plus rough knowledge of the vehicle’s 3-D shape and an assumption of planar ground, suffices for predicting the vehicle shadow in the image—e.g., Figure 4(d). In practice, we simply threshold on the azimuthal angle between the predicted tip of the shadow and the current vanishing estimate. If they are too close, VP_FOLLOW’s output is considered untrustworthy and it is disregarded. A corollary computation for failure due to darkness that does not require image processing is the local sun elevation angle. If it is too low, we assume that it is too dark for VP_FOLLOW to operate.

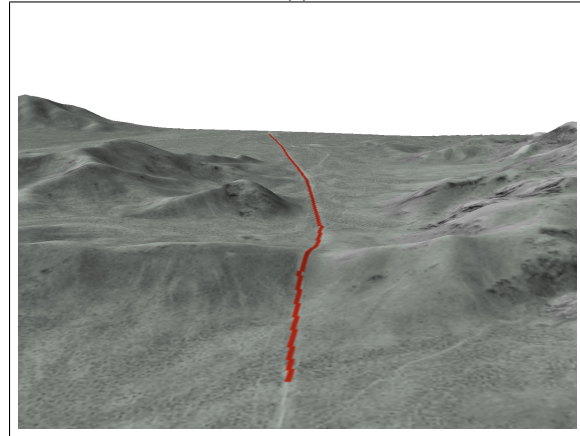
IV. RESULTS

We will first show results for various components of the system, then present results for the entire system running autonomously. For all of results given here, the vehicle used is a fully-actuated Ford E350 with a modified suspension and 4 wheel drive. Only a single bumper-mounted SICK LMS ladar was used for lateral offset and road width estimation. The vehicle is pictured in Figure 5.

A sample of VP_FOLLOW running passively while the vehicle is under manual control is shown in Figure 6. The leftmost image is a detail of an aerial photograph of the test area in which the vehicle is driven from right to left through an S-turn. The five pairs of camera images and ladar obstacle maps to the right correspond to the vehicle locations marked with red dots in the aerial image. The green road regions are based on the estimated road centerline trajectories and the discrete width estimates, with the lookahead distance of the road clipped when a width posting is $\leq 2m$. Note the



(a)



(b)

Fig. 7. (a) Approximate traces of autonomous trajectory follower runs with latitude, longitude markings overlaid. Map covers about an 11 km by 6.5 km area. (b) GPS trace of trajectory follower run over hill (lower right segment from (a)) overlaid on terrain model rendered from digital elevation model, aerial imagery.

shortening of the road region as the turns are entered in Figures 6(c) and (e), and its lengthening as the road straightens in Figure 6(d) and (f). In the camera images, VP_FOLLOW can be seen to smoothly anticipate both the left and the right turn.

VP_FOLLOW has been tested in autonomous mode in a number of situations. In one experiment, the road centerline was directly used by the vehicle as a trajectory to follow. Modulo several pauses to adjust the maximum vehicle speed, the vehicle was able to follow a long curving road in gently sloped terrain quite smoothly (the diagonal leg in Figure 7 using only the single camera and a bumper ladar. A second segment was driven that approached, ascended, and descended a significant hill (the horizontal segment in Figure 7 and 3-D rendering) under full autonomous control save manual throttle control on the descent (the vehicle was not pitch-aware in setting its speed limits).

In more recent experiments, we have controlled the vehicle less directly, through painting of “costs” in a 2-D map that a planning module uses to generate nonlinear trajectories. The details of this are beyond the scope of this paper, but suffice it to say that several more autonomous miles have been logged on different desert roads in this fashion with VP_FOLLOW exercising primary control, and more still with

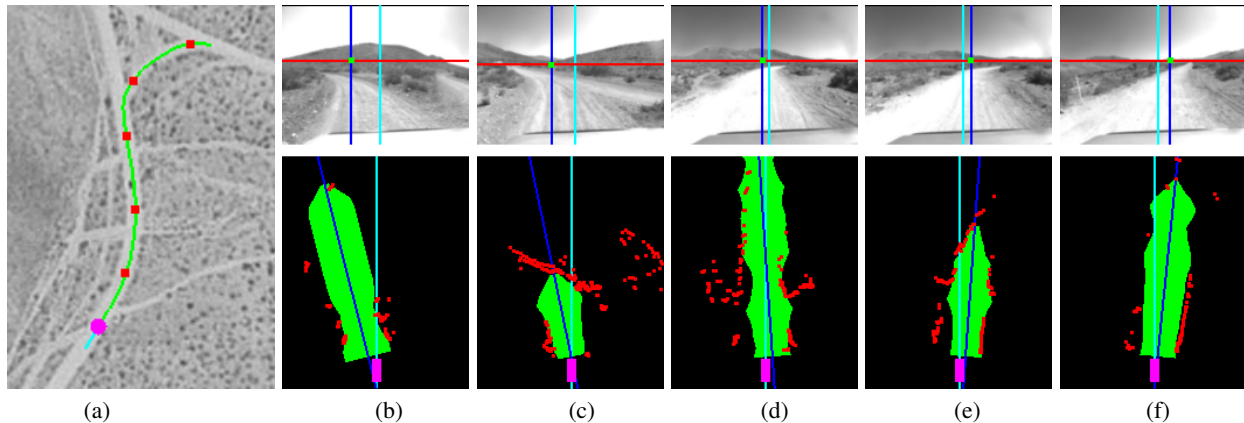


Fig. 6. Estimated road shape from manual driving around an S-turn. The leftmost image is a satellite photograph of the testing area with the vehicle track overlaid. The scale of the obstacle field images is 40 m wide, 50 m high (vehicle is 2 m wide, 5 m long)

it as part of a larger ensemble of “opinion providers.” One positive aspect of this less-direct method of vehicle control is that higher-level matters such as DGC course boundaries can override found roads. For example, a sample course visiting all three vertices of the triangle of roads in Figure 7 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 VP_FOLLOW’s tendency is to stay on whatever road it starts on.

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.

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.

ACKNOWLEDGMENTS

Many thanks to Team Caltech for their generous provision of vehicle access and data.

REFERENCES

- [1] 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.
- [2] K. Kluge and M. Morgenthaler, “Multi-horizon reactive and deliberative path planning for autonomous cross-country navigation,” in *SPIE Defense and Security Symposium*, 2004.
- [3] L. Cremean, T. Foote, J. Gillula, G. Hines, D. Kogan, K. Kriebbaum, J. Lamb, J. Leibs, L. Lindzey, A. Stewart, J. Burdick, and R. Murray, “Alice: An information-rich autonomous system for high speed desert navigation,” Submitted to *Journal of Field Robotics*, 2006.
- [4] C. Taylor, J. Malik, and J. Weber, “A real-time approach to stereopsis and lane-finding,” in *Proc. IEEE Intelligent Vehicles Symposium*, 1996.
- [5] B. Southall and C. Taylor, “Stochastic road shape estimation,” in *Proc. Int. Conf. Computer Vision*, 2001, pp. 205–212.
- [6] N. Apostoloff and A. Zelinsky, “Robust vision based lane tracking using multiple cues and particle filtering,” in *Proc. IEEE Intelligent Vehicles Symposium*, 2003.
- [7] 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.
- [8] C. Rasmussen, “Combining laser range, color, and texture cues for autonomous road following,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2002.
- [9] J. Zhang and H. Nagel, “Texture-based segmentation of road images,” in *Proc. IEEE Intelligent Vehicles Symposium*, 1994.
- [10] C. Rasmussen, “Grouping dominant orientations for ill-structured road following,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.
- [11] C. Rasmussen, “Texture-based vanishing point voting for road shape estimation,” in *Proc. British Machine Vision Conf.*, 2004.
- [12] 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.
- [13] T. Lee, “Image representation using 2D Gabor wavelets,” *IEEE Trans. PAMI*, vol. 18, no. 10, pp. 959–971, 1996.
- [14] M. Frigo and S. Johnson, “FFTW: An adaptive software architecture for the FFT,” in *Proc. 1998 IEEE Intl. Conf. Acoustics, Speech, & Sig. Proc.*, 1998, vol. 3, pp. 1381–1384.
- [15] T. Tuytelaars, M. Proesmans, and L. Van Gool, “The cascaded Hough transform,” in *Proc. IEEE Int. Conf. on Image Processing*, 1998.
- [16] A. Polk and R. Jain, “A parallel architecture for curvature-based road scene classification,” in *Vision-Based Vehicle Guidance*, I. Masaki, Ed., pp. 284–299. Springer, 1992.
- [17] S. Se and M. Brady, “Vision-based detection of staircases,” in *Proc. Asian Conf. Computer Vision*, 2000, pp. 535–540.
- [18] J. Coughlan and A. Yuille, “Manhattan world: Orientation and outlier detection by bayesian inference,” *Neural Computation*, vol. 15, no. 5, pp. 1063–088, 2003.
- [19] M. Isard and A. Blake, “Contour tracking by stochastic propagation of conditional density,” in *Proc. European Conf. Computer Vision*, 1996, pp. 343–356.
- [20] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, 1991.
- [21] M. Ollis and A. Stentz, “Vision-based perception for an autonomous harvester,” in *Proc. Int. Conf. Intelligent Robots and Systems*, 1997, pp. 1838–1844.