On-Vehicle and Aerial Texture Analysis for Vision-Based Desert Road Following

Christopher Rasmussen

Department of Computer & Information Sciences
University of Delaware
cer@cis.udel.edu

Thommen Korah

Department of Computer & Information Sciences
University of Delaware
tkorah@cis.udel.edu

***Abstract*— We present two components of a vision-based approach to autonomous driving on and near rural and desert roads. The first component comprises fast processing of live vehicle camera video to robustly extract linear direction and midline estimates of marginal roads. The second uses satellite imagery immediately surrounding the vehicle's GPS position to trace the road ahead for curve and corner anticipation, and to inform the vehicle planner of a nearby road when traveling cross-country. The algorithms are built upon Gabor wavelet filters for texture analysis: *on-board*, they are employed to find ruts and tracks from which the road vanishing point can be inferred via Hough-style voting, and *aerially*, they localize the edges of low-contrast road contours. Mechanisms for both modules to determine whether the vehicle is currently on- or off-road are also explained. Our system's efficacy is illustrated for several difficult datasets, including a log from one vehicle's run during the 2004 DARPA Grand Challenge.**

I. Introduction

With the running of the first DARPA Grand Challenge (DGC) robot race in 2004 and a second iteration scheduled for the fall of 2005, there is heightened interest in algorithms for autonomously following "difficult" unpaved paths and roads. Although the DGC race course may cross any kind of traversable terrain, requiring sensors and methods for general obstacle avoidance and slope analysis as studied in [1], [2], roads by design offer benign channels through otherwise more difficult areas. Nearly all of the 2004 course was narrowly constrained to a series of flat or hilly desert roads, but in a more realistic scenario path-planning algorithms will have the freedom (and responsibility) to decide whether to chain together segments of sketchy roads or set out on cross-country routes which are more direct but likely riskier.

In this paper, we are concerned with two aspects of this problem. First, at the low level we examine how to use on-board visual sensing to guide a vehicle along an unpaved road that often lacks the high-contrast edge and/or color features typically used for road following on highways, streets, or dirt roads bordered by vegetation [3], [4], [5], [6], [7], [8]. Instead, our approach [9] relies on oriented texture associated with the road such as the *banding* pattern of ruts and tire tracks left by previous vehicles, as well as road border edges or painted lines if they are present. All such oriented features, provided they are parallel to the road direction, share a common vanishing point (see [10], [11], [12], [13] for other work on vanishing point finding). This vanishing point indicates the road direction, giving an angular error suitable for input to a low-level steering control and acceleration module.

In addition to road direction, information about the vehicle's lateral displacement that allows centering is also needed. Vanishing point information provides a powerful constraint on the possible road region geometry by defining a 1-D family of possible edge lines (for straight roads) radiating below the horizon. Under the assumption of oriented texture in the road, there are often also linear features inside the road region that also "support" the vanishing point hypothesis. Grouping such texture features gives a rough idea of the extent of the road region and hence its midline.

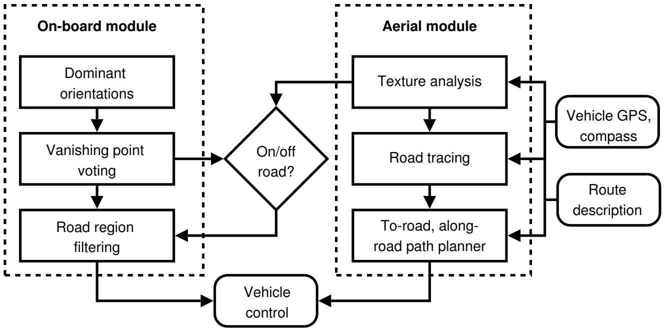


Fig. 1. System components

There are other cues such as color which are powerful aids for segmentation [14], [6], [7], [8], but in this paper we focus on what can be done with oriented texture alone. One virtue of our approach is relative insensitivity to the road surface material and lighting conditions–there is no need to tune, learn, or adapt parameters as the algorithm runs on a variety of roads.

The second problem that we address here is the mid-level one of finding nearby roads to travel on. This helps the on-board module to know when the vehicle is off-road and therefore should stop running, or when it is near a road again and should resume. Digitized/vectorized roads are not common in the rural and desert areas we want to operate in, so we use static aerial imagery to detect the roads directly with image processing. In this work we assume a reliable estimate of the vehicle's current position and the goal is to direct it along or towards the road using on-board and aerial road-following algorithms. Given the current position, the aerial view of the environment is used to plan a path of a couple of hundred meters towards the final destination. This local plan in the vicinity of the robot is used to guide the on-board road following.

A critical system state is whether the vehicle is currently on a road or not. This is simple for 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. This information is important in order to "gate" the on-board module's output to the vehicle controller to prevent spurious steering commands from being issued when there is no road to follow. It is also useful for annotating the aerial module's planned paths so that the high-level vehicle controller knows how cautious to be.

In the next two sections, we will describe the on-board module and aerial module, as well as the on/off road decision methods for each, after which we will show results demonstrating the system's capabilities and discuss extensions we are currently working on.

II. On-Vehicle Road Following

There are three significant stages to the on-vehicle road following pipeline, as shown in Fig. 1, which we describe in the following subsections. First, dominant texture orientations are computed 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 road support region, consisting of rays along which the dominant orientations agree with the chosen vanishing point, is estimated and a putative road midline extracted from it.

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. There is a considerable body of work on methods for estimating dominant orientations [15], [16]; our approach is to use a bank of Gabor wavelet filters [17]. 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:

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

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.

Based on empirical observation of performance on diverse road scenes and cameras with varying internal calibrations, we have found that a single wavelength related to the image dimensions by an *ad hoc* scaling factor gives very good results at a significant computational savings vs. multi-scale schemes. This scaling factor, which works well for image scales ranging from 320×240 to 80×60, results in $\lambda = 4$ and a kernel size of 12×12.

A fairly large number of orientations (e.g., $n = 36$ for all of the results in this paper) are necessary to achieve superior angular resolution for $\theta(\mathbf{p})$ given the voting method described in the next subsection. Performing this many convolutions per image with such large kernels is obviously computationally expensive, consuming about $4/5$ of the running time of the on-board module. Using the convolution theorem and the FFTW Fourier transform library [18] at single precision, however, allows dominant orientations to be obtained with adequate speed. For example, calculating $\theta(\mathbf{p})$ for every pixel in a 160×120 image with $n = 36$ takes ~ 55 ms on a 3.0 GHz Pentium IV (note that this is independent of λ).

Fig. 2(b) shows the calculated dominant orientations for the image in Fig. 2(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. Linear Vanishing Point Detection

For a straight road segment on planar ground, there is a unique vanishing point associated with the dominant orientations of the pixels belonging to the road. Curved segments induce a set of vanishing points (discussed in [9]), but we will satisfy ourselves with an approximation to the tangent here.

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. In [9], 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 conveniently accelerates this voting operation, but until recently only 8-bit accumulation buffers have been available, limiting "elections" to a maximum of 256 votes per candidate. When large numbers of votes are cast, such as at image resolutions of 320×240 and higher, this can lead to saturation artifacts preventing true vote totals from being recorded. In this paper all results were obtained on 160×120 images with 8-bit accumulation buffers. [1]

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], [4], [5]. Particles are initially distributed uniformly in order to coarsely localize the vanishing point. Weak dynamics $p(\mathbf{v}_t \mid \mathbf{v}_{t-1})$ (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.

With internal and external calibration of the camera, a tangent to the curvature of the road in the ground plane can be obtained from the x coordinate of the vanishing point. This road direction measurement is necessary for steering control, while the motion of the y coordinate or horizon line indicates out-of-plane undulation of the road and may be useful for braking control.

Fig. 2(c) shows the vanishing point candidate function for the image in Fig. 2(a). Its filtered maximum is indicated by the blue square in Fig. 2(a).

The "sharpness" of the vanishing point peak is 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 [13].

When the KL of $votes(\mathbf{v})$ is over a threshold (we use 0.2 in

[1] The Nvidia GeForce 6800 supports 16-bit floating point blending with an 11-bit mantissa and thus boosts the maximum to 2048 votes, enough for higher resolutions

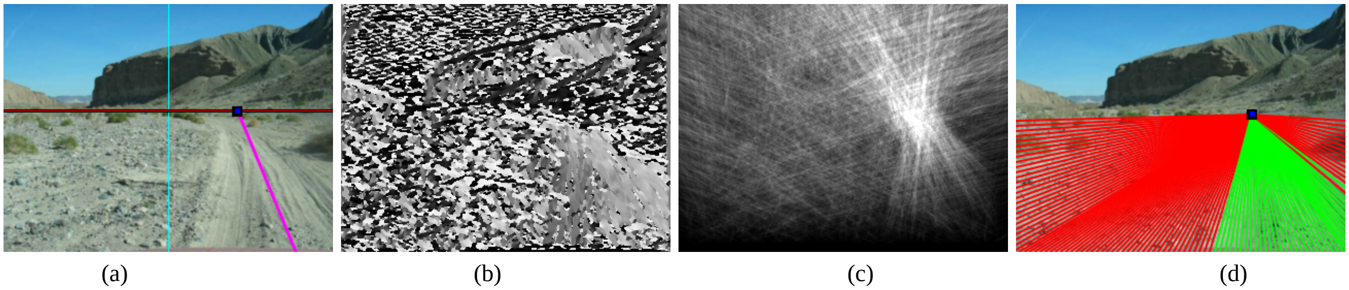


Fig. 2. (a) Captured road image with estimated vanishing point, horizon line, road midline, and image midline overlaid; (b) Dominant orientation at each pixel ($[0, \pi]$ radians → $[0, 255]$ intensity values); (c) Vote function for vanishing point (d) Road support based on thresholded mean disagreement between dominant orientations and direction to vanishing point

this paper for 160×120 images), the estimated vanishing point is considered reliable. To smooth this decision, we store a history of the last 10 seconds of results of this threshold comparison and require $2/3$ of them be over threshold for the current vanishing point information (as well as road midline computed below) to be passed on to the vehicle controller.

C. Road Support Region Segmentation

One can think of a vanishing point $\mathbf{v}$'s *support* as the set of all pixel positions $\mathbf{v}_{supp}$ whose dominant orientations cause them to vote for it. Given that for most rural road scenes the dominant orientations of much off-road texture such as vegetation, rocks, etc. are randomly and uniformly distributed with no strong points of convergence, supporters often come from inside the road region. Therefore, the road region may be roughly segmented as an angular range below the vanishing point with a higher density of supporters than in the flanking ranges.

One issue of generality is that although this model fits unpaved roads fairly well, texture on paved and painted roads is usually sparser. That is, most of the road region in the latter is homogeneous, with the oriented texture just along the road border and lane lines. We have found that explicitly trying to identify the left and right edges of the road using texture support alone is less robust than estimating its *midline* directly.

We do this by extending a set of imaginary "rays" down from $\mathbf{v}$ at regular intervals. If the ith ray $\mathbf{r}_i$'s orientation is $\phi_{\mathbf{r}_i}$ and length (clipped to the image) is $|\mathbf{r}_i|$, the mean angular discrepancy between the ray orientation and dominant orientations along it is calculated as $\mu(\mathbf{r}_i) = \sum_{(x,y)\in\mathbf{r}_i} |\phi_{\mathbf{r}_i} - \theta(x, y)|/|\mathbf{r}_i|$. Ray i is labeled as a *support ray* if its discrepancy is less than a threshold $\mu(\mathbf{r}_i) < \tau_\mu$. The support rays obtained with a threshold of 0.75 for the vanishing point of the image in Fig. 2(a) are drawn in green in Fig. 2(d) (red lines are non-support rays).

Finally, an instantaneous measurement $\hat{\mathbf{m}}_t$ of the road midline at time t is obtained as the mean road support ray (without weighting) for that image. A smoothed midline estimate is calculated with a simple filter $\mathbf{m}_t = \mathbf{m}_{t-1} + \alpha(\hat{\mathbf{m}}_t - \mathbf{m}_{t-1})$ (we used $\alpha = 0.1$ for the results in this paper). The $\mathbf{m}$ estimated from the support rays of the example image is shown in magenta in Fig. 2(a).

The camera's vertical and horizontal fields of view θ_V and θ_H, height H off the ground plane, and pitch angle ρ establish the distance D to the nearest ground point visible—along the bottom row of the image $y = h$. The x coordinate of the intersection of the midline $\mathbf{m}$ and the bottom row can thus be converted to an approximation of the vehicle's lateral displacement Δ via the formula $\Delta = D \tan[\theta_H(x/w - 0.5)]$, where w is the width of the image in pixels.

III. Aerial Road Following

There are three key components in the aerial module diagrammed in Fig. 1: texture analysis, road tracing, and planning. First, texture analysis is performed on static satellite imagery[2] in order to measure the likelihood that a road is present over the position and orientation space. The area examined is masked by RDDF corridor boundaries, if any. Second, a temporal filter is used to find a maximally likely road contour in the search region and trace along it. Road tracer performance often benefits from biasing its dynamics by the angle of the major axis of the current RDDF corridor, even for very wide segments. Finally, the aerial planner works by re-initializing the tracer at the current vehicle position $\mathbf{x}_t$ and building a fixed-length waypoint sequence along the road in front of it. As with the on-board module, the base aerial module assumes that the vehicle is on the road, seeking only to keep it there.

A. Texture Analysis

The unpaved and desert road contours in the grayscale aerial imagery used in this work are low contrast and noisy, making simple edge- or color-based methods impractical. In common with the on-board road follower, we use a bank of Gabor wavelet filters [17] to measure the likelihood that a hypothetical vehicle state $\mathbf{x}$ consisting of its position (x, y) (i.e., UTM coordinates) and direction θ (i.e., compass heading) are both on and tangent to a road. The expected range of roads widths dictates the choice of scale for the filter; once again just a single wavelength is used. Vehicle orientations are discretized to $K = 9$ equally separated angles in $[0, \pi]$. The measurement likelihood $p(\mathbf{I} \,|\, \mathbf{x})$ of a hypothetical state $\mathbf{x}$ being on the road is given by the closest-orientation Gabor filter's complex response (explained in the dominant orientation section above) at that position of the aerial image $\mathbf{I}$.

B. Road Tracing

Our method of predicting the road ahead from aerial images is most similar to JetStream [21], a particle filtering approach to spatially track edge contours that has been applied to roads. Tracking generally refers to following the state of a target over time, but for still aerial images "time" is associated only with the progressive extension of the estimated contour as in [21], [22]. In previous work [23], we showed that our formulation of the measurement likelihood seemed to work better on desert road images than JetStream's more general

[2] All aerial images in this paper were obtained from Microsoft's Terraserver with a resolution of 1 m / pixel

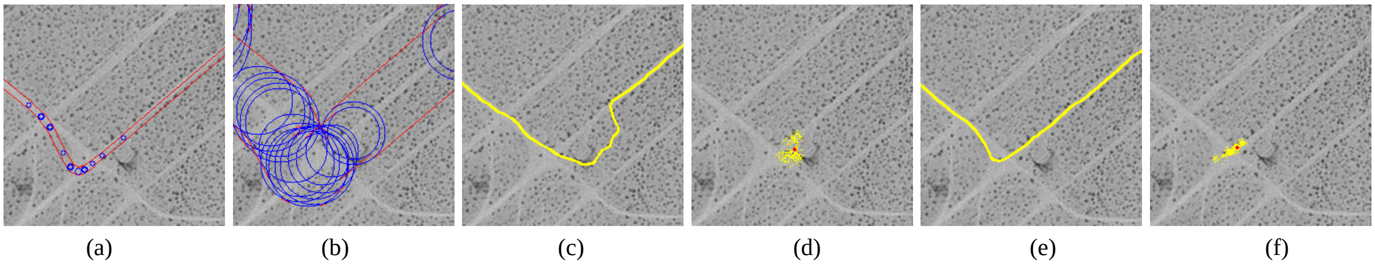


Fig. 3. Incorporating RDDF into the dynamics prior for the particles. (a) 2004 DGC corridors at a sharp turn early in the race; (b) Corridor width increased $10\times$ to test our road tracer; (c) Road trace using texture-driven dynamics only; (d) Bimodal distribution of particles at the same intersection for texture-driven dynamics; (e) Road trace using RDDF-based dynamics; (f) Unimodal distribution of particles at the intersection for (e)

approach. Moreover, in that work we introduced the idea of biasing the dynamics $p(\mathbf{x}_t \,|\, \mathbf{x}_{t-1})$ of the road tracer with the GPS waypoints of a manually-driven vehicle, which tightly corresponded to the road shape (the objective was to post-process a noisy GPS track under the assumption that the vehicle was driven on roads the whole way).

For efficiency, image processing is done only within a small floating tile (61×61 here) around the particles, allowing convolutions to be cached until a minimum fraction of the particle set leaves its confines. This allows real-time performance on large images (a second-level system of loading and unloading adjacent raw aerial image files has not yet been implemented, as all of our runs have thus far been limited to 2.5×2.5 km square areas).

If the vehicle is being driven in a race situation akin to the DARPA Grand Challenge, analogous information may be available from a *route description* consisting of a series of waypoints and maximum lateral boundary offsets (LBOs) from the line segments connecting them. The term used by DARPA for the 2004 DGC for this was the *RDDF*, which we will use here. In an RDDF, two successive waypoints along with the LBO define a *route segment*. The boundaries at the ends of each segment are demarcated by a semi-circle of radius equal to the LBO, providing a transition zone from one segment to the other.

In the 2004 DGC the RDDF LBOs were quite narrow, with a mean segment width of 3.2 m and median width of 2.4 m, and followed roads for almost the entire route. This of course obviated any road finding at all, but in general LBOs may be much wider. Even allowing substantial increases in corridor widths that necessitate finding the road within them, any vague hint of the road direction from the corridor direction is a strong cue for the road tracer dynamics. We implement this by augmenting the vehicle state $\mathbf{x}$ with a variable for the current RDDF segment m. For each particle, samples of θ are distributed around the orientation of route segment m. In case the current vehicle position is outside the corridor, the particles are initialized at a point within the closest segment by dropping a perpendicular to it.

The dynamics of each particle can further be influenced by the RDDF. All particles are subjected to forward and rotational motion $(R + N(\sigma_r), \Theta + N(\sigma_\Theta))$ where $N(\sigma)$ denotes Gaussian noise with variance σ^2. A fixed step size is used for R, but Θ can take on two different values. For a particle within a very narrow segment, Θ can be set to the orientation of this segment. This is pure corridor following, which might also be the recommended option on sections where the road likelihood measure is very low. However, on a long wide segment with curved roads, this is too restrictive. Instead, the orientation for each particle is distributed around the angle θ it was traveling in the previous time step. We term the latter as *texture-driven dynamics*, which allow particles to follow the most likely road within the segment in whatever direction it leads.

While pure texture-driven dynamics works well on smoothly curving road segments, road forks or sharp turns can result in a multi-modal posterior. Such situations are greatly mitigated by *RDDF-driven dynamics*, in which we detect segment transitions and distribute θ around the orientation of the next segment for one iteration, reverting to texture-driven dynamics within segments.

Fig. 3 illustrates the difference between these two approaches to road tracing dynamics by focusing on a particular turn from the 2004 DGC RDDF. Fig. 3(a) shows the narrowness of the original race corridors, while Fig. 3(b) shows the effect of scaling all LBOs by a factor of 10. The road traced in Fig. 3(c) using only texture-driven dynamics overshoots the left turn, resulting in a multi-modal distribution for the particles (Fig. 3(d)) with the weighted mean (shown as a red dot) falling outside the road. The particles converge onto the correct road only after several iterations. In contrast, with RDDF-driven dynamics providing a strong hint that the road is turning even if its location is unknown, the road tracer in Fig. 3(e) makes the turn almost perfectly. The particles also form a tight cluster oriented along the correct path as shown in Fig. 3(f).

As well as guiding the evolution of particles for road tracing, the measurement likelihood function above can also be used as a confidence measure for the vehicle being on the road. Given the GPS coordinates P and the orientation θ of the vehicle, we compute the mean strength of the θ-filter responses in an 11×11 meter window around P and threshold it.

C. Planning

The planner is a straightforward extension to the road tracer. Every frame, the road tracer is re-initialized by distributing particles uniformly around the vehicle's current GPS position. The tracer proceeds for a fixed number of iterations, generating a trajectory as a dense series of waypoints (up to a few hundred meters ahead). The plan inside corridors is a series of closely-spaced waypoints leading to and along the nearest road from the vehicle. These are generated by the state of the particle filter as it searches forward in the current and succeeding route corridor segments a fixed distance. When the vehicle is outside the corridor, a straight-line plan leading to the nearest point on the midline of a segment is created, after which the waypoints come from the road tracer.

This information is useful to the on-board road follower for anticipating corners, which are failure modes for the vanishing point finder. In addition to prediction, this plan also gives a sense of the direction and distance to safety for impasses such as a vehicle that is off-road with no trail visible to any on-board sensors.

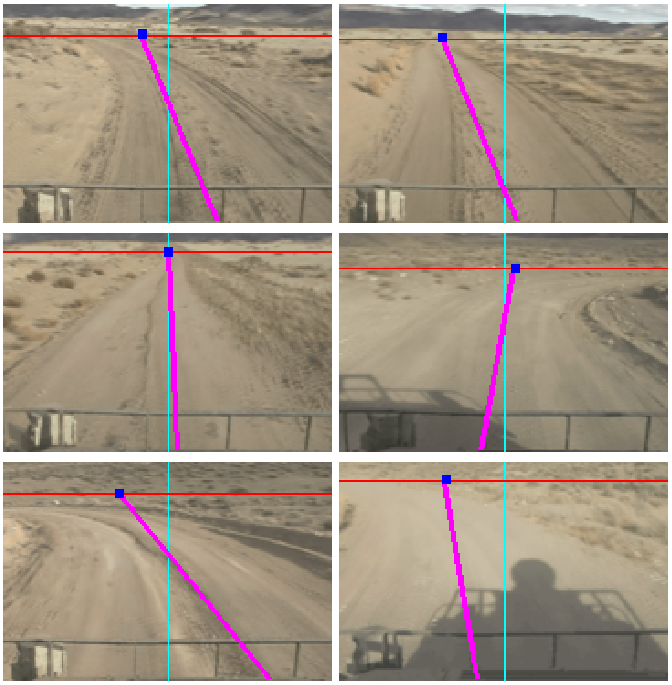


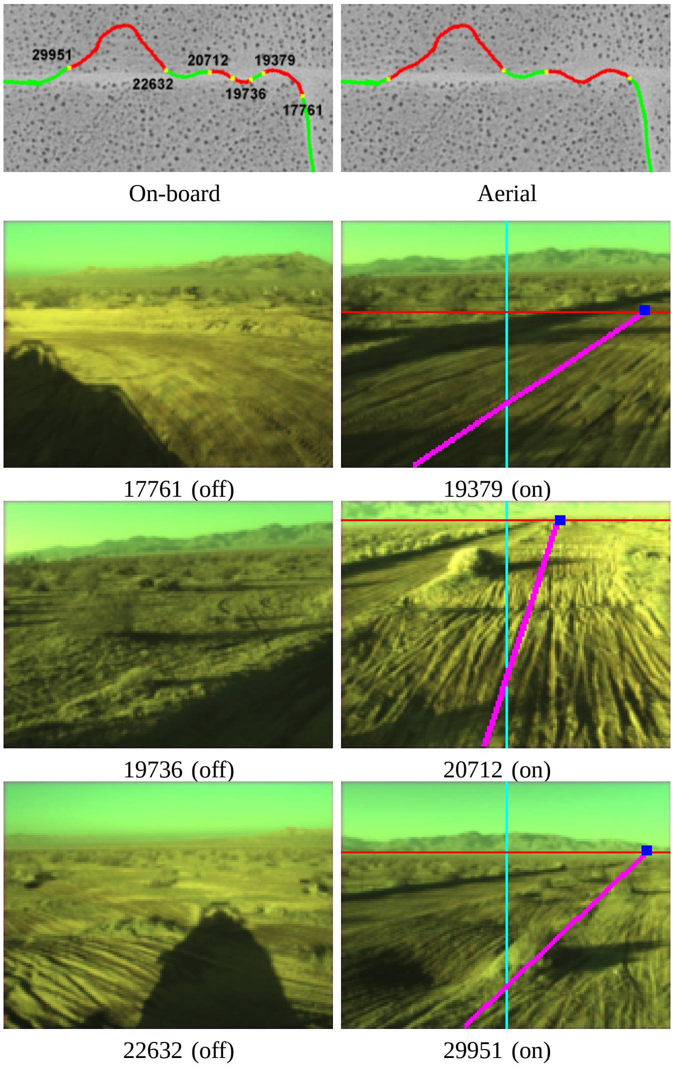
Fig. 4. On-board system's vanishing point and road midline estimates for selected frames from a 2004 DGC vehicle training run

IV. Results

Output of the on-board module for selected frames from a long training sequence collected by Carnegie-Mellon's 2004 DGC team are shown in Fig. 4. Note that the system deals fairly well with non-trivial road curvature even with its linear estimate, and is not thrown off by the outlier textures generated by the shadow of the vehicle cast on the road in front of it.

The accuracy of our on-board method for discriminating road from non-road images is shown in Fig. 5. A plot of the one vehicle's GPS trace as it negotiated an early 2004 DGC turn are shown are red and green segments in the upper-left subfigure. Green indicates that the smoothed KL divergence method described above considers the image to contain a trackable road, while red implies the opposite. Sample image captures from the numbered frames along the route are shown below, with the left column consisting of inferred off-road images and the right on-road images. In the on-road images the system's estimated vanishing point and road midline are shown.

To quantify the efficacy of the aerial road tracer, we can make use of the original 2004 DGC RDDF. The route segments given by DARPA fell almost entirely on roads and were usually narrower than the actual road. Tracing the road with such a narrow corridor width gives a dense track of locations guaranteed to be on the road. We compared this trace with another obtained by running our technique on an artificially widened (10×) corridor. This resulted in a median distance error between the two traces over the first few miles of the course of 1.3 m. It is important to note that certain segments provided by DARPA such as that in Fig. 3 are outside the road, while our technique correctly follows a road within the wide corridor if not necessarily the midline of the narrow corridor.



On-board Aerial

17761 (off) 19379 (on)

19736 (off) 20712 (on)

22632 (off) 29951 (on)

Fig. 5. Detail of on/off road decisions made by each module. Green and red segment colors indicate on and off road, respectively. Numbers stand for first frame of segment (the gross direction of travel is up and then to the left).

Fig. 6 demonstrates how our algorithm can assist in localized planning. The figures show in green the track of Caltech's 2004 DGC vehicle's GPS locations during the 2004 race. When initialized with the vehicle's current GPS position, the particle filter is run for t iterations returning a dense track $p_{0:t}$ of waypoints that are very likely to be on the road. Drawn in yellow, is this prediction of the road immediately ahead or in the vicinity of the vehicle. Using such a plan could have helped their robot avoid and recover from certain off-road situations they encountered during the race.

The bottom two images in Fig. 6 are the result of road tracing in later mountainous segments of the race course. While the dynamics of our particle filter can effectively handle various types of roads exhibiting sharp corners as well as high curvature, the likelihood function is able to constrain the particles on the road purely by analyzing the local texture in the aerial image. Even on such low quality images with poor contrast, this is a powerful cue that can assist the navigation modules, for both localization and planning.

As shown in the upper-right subfigure of Fig. 5, a simple on/off road confidence measure based on thresholding the likelihood function and passed through a smoothing filter discriminates the positions and orientations where the vehicle is clearly off or not pointing down the road. There is remarkable agreement between the on-board and aerial methods considering how different they are. Much of the

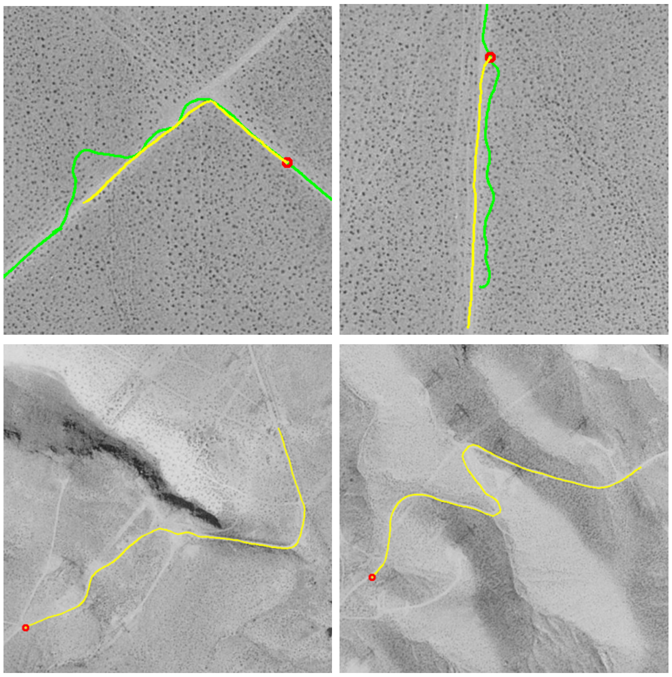


Fig. 6. Top: GPS plot of a 2004 DGC vehicle's route (green) on raceday and predicted path (yellow) from a fixed location (red dot) at different sections of the course. The predicted distance is approximately 325 meters with the particle filter run for 75 iterations. The original corridor width for each segment is widened by a factor of 10. Bottom: Road tracing for 175 iterations in a high altitude segment of the 2004 course with the corridor width (not shown) widened by a factor of 10.

difference can be ascribed to the fact that the on-board decision component assesses the ground ahead of the vehicle, while the aerial mechanism decides on the basis of the image directly surrounding the vehicle, causing a "lag" for the aerial decision.

V. Conclusion

We have presented a system for road following on desert and unpaved road that relies on road texture, analyzed both from an on-board camera and satellite imagery, 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, as well as extracting the road midline using residual information from the voting process. The aerial component finds and traces roads near the current vehicle position, providing a medium-term plan and an alternative to pure cross-country obstacle avoidance. Moreover, both pieces of the system analyze their own performance and automatically turn off when the vehicle is not near a road.

We are currently investigating two lines of extension to this work. The on-board module's road midline estimate can be somewhat biased by shadows cast on one side of the road and other appearance asymmetries. To validate and improve the segmentation, we have begun incorporating information from a cab-mounted laser range-finder that, when calibrated with the camera, can provide road width estimates that are unaffected by lighting conditions. Secondly, the aerial module currently only finds a single road near the vehicle. 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

Portions of this work were supported by a grant from the National Institute of Standards & Technology. DGC data was generously provided by Caltech.

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

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

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

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

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

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

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

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

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

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

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

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

[15] 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.

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

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

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

[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] P. Perez, A. Blake, and M. Gangnet, "Jetstream: Probabilistic contour extraction with particles," in *Proc. Int. Conf. Computer Vision*, 2001, pp. 524–531.

[22] 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.

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