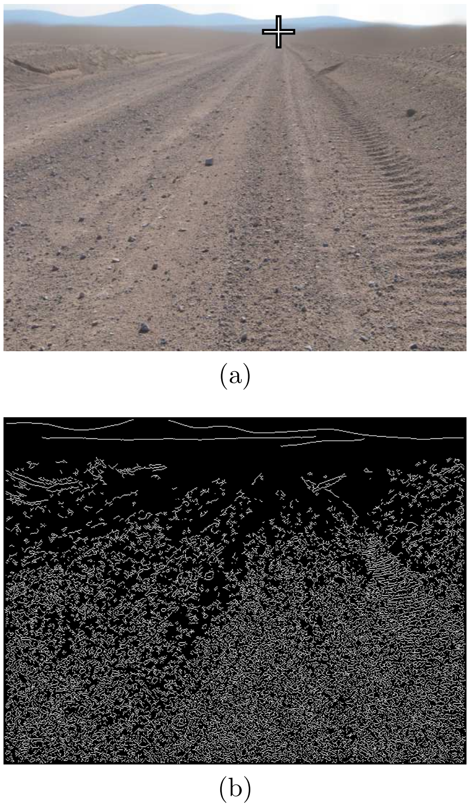
Grouping Dominant Orientations for Ill-Structured Road Following

Christopher Rasmussen

Dept. Computer & Information Sciences
University of Delaware
Newark, DE 19716

Abstract

Many rural roads lack sharp, smoothly curving edges and a homogeneous surface appearance, hampering traditional vision-based road-following methods. However, they often have strong texture cues parallel to the road direction in the form of ruts and tracks left by other vehicles. In this paper, we describe an algorithm for following ill-structured roads in which dominant texture orientations computed with multi-scale Gabor wavelet filters vote for a consensus road vanishing point location. In-plane road curvature and out-of-plane undulation are estimated in each image by tracking the vanishing point indicated by a horizontal image strip as it moves up toward the putative vanishing line. Particle filtering is also used to track the vanishing point sequence induced by road curvature from image to image. Results are shown for vanishing point localization on a variety of road scenes ranging from gravel roads to dirt trails to highways.



(a)

(b)

Figure 1: (a) Desert road from DARPA Grand Challenge example set (with vanishing point computed as in Section 2.2); (b) Canny edges of same scene

1 Introduction

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 [1, 2, 3] 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 [3, 4, 5, 6]. 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. Of course, the contrast between, for example, the road color and the background color tends to change over time, necessitating adaptation of the discriminant function [7]

Most road images can be successfully interpreted using a variant of one of the two above approaches. Nonetheless, there are some scenes that possess neither strong edges nor contrasting local characteristics. Figure 1(a) shows one such road (the cross has been added by our algorithm and is explained in Section 2.2). This image is from a set of "course examples" made available to entrants in the 2004 DARPA Grand Challenge, an autonomous cross-country driving competition [8] [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 Matlab's Canny function shows in Figure 1(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, presumably 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.

A number of researchers have used vanishing points as global constraints for road following or identification of painted features on roads (such as so-called "zebra crossings," or crosswalks) [9, 10, 11, 12]. 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 [13] 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 [14, 15].

All of the voting methods for localizing a road's vanishing point that we have identified in the literature appear to be based on a prior step of finding line segments via edge detection. Moreover, with the exception of [12] (discussed below), vanishing-point-centric algorithms appear not to deal explicitly with the issue of road curvature and/or undulation, which remove the possibility of a unique vanishing point associated with the road direction. Both of these limitations are problematic if vanishing point methods are to be applied to bumpy back-country roads like the desert scene discussed above.

In this paper we present a straightforward method for locating the road's vanishing point in such difficult scenes through texture analysis. Specifically, we replace the edge-detection step, which does not work on many such images because the road bands are too low-frequency to be detected, 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.

We also add a second important step to deal with road curvature (both in- and out-of-plane), which is the notion of tracking the vanishing points associated with differential segments of the road as they are traced from the viewer into the distance. By integrating this sequence of directions, we can recover shape information about the road ahead to aid in driving control. The method is easily extended to temporal tracking of the vanishing point over sequences of images.

A related idea was proposed for planar texture analysis using "spectral voting" in [16], but only tested on synthetic images with no curvature. Another related piece of work is the CMU RALPH road follower [17]. Their system also exploits texture within the road including tracks, drips, and lane lines, but it works by a different mechanism: for every frame, a small set of in-plane curvatures are hypothesized and one is chosen that best fits the image.

2 Methods

There are three significant components to the road following algorithm, which we describe in the following subsections. First, a dominant texture orientation is computed at every image pixel. Second, assuming a straight, planar road, all dominant orientations in the image vote for a single best road vanishing point. Third, if the road curves or undulates a series of vanishing points for each tangent direction along the road must be estimated. For a sequence of road images we must also estimate the deformation of this vanishing point contour from image to image.

2.1 Dominant orientation estimation

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 [18] 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 [19].

We experimented with several approaches to local orientation estimation, including that of [19], and observed the most qualitatively accurate results on a wide variety of road images with a bank of Gabor wavelet filters [20]. 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 standard "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 this work the testing images were captured by 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 we use a Gabor filter wavelength of $\lambda = 2^{\lfloor \log_2(w) \rfloor - 5}$, where w is the width of the input image. For 720×480 and 640×480 images, for example, this results in $\lambda = 16$ and a kernel size of 50×50.

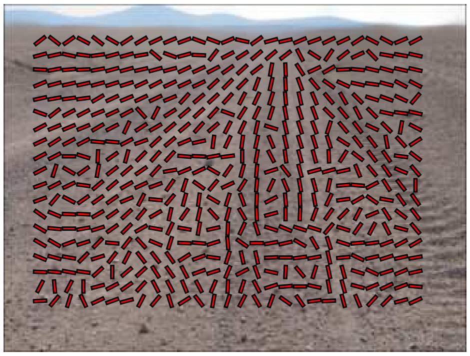


Figure 2: Dominant orientations (subsampled) for desert image

A large number of orientations (e.g., $n = 72$) were 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. Nonetheless, we found that using the convolution theorem and the FFTW Fourier transform library [21] allows dominant orientations to be obtained with adequate speed. For example, calculating $\theta(\mathbf{p})$ for every pixel in a 160×120 image with $n = 72$ takes $\sim$130 ms on a 3.0 GHz Pentium IV (the same machine is used for all speed figures in this paper).

An example of the dominant orientations computed over a sparse grid of locations is shown in Figure 2 for the desert road image.

2.2 Vanishing Point Voting

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 below). Though road vanishing points may lie outside the field of view (FOV) of the road-following camera, it is reasonable to limit our search to the area of the image itself if the following conditions are met: (1) The camera's optical axis heading and tilt are aligned with the vehicle's direction of travel and approximately level, respectively; (2) Its FOV is sufficiently wide to accomodate the maximum curvature of the road; and (3) The vehicle is roughly aligned with the road (i.e., road following is proceeding successfully).

Furthermore, we assert 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.

In the following subsections we describe methods for formulating an objective function $\textit{votes}(\mathbf{v})$ to evaluate the support of road vanishing point candidates $\mathbf{v} = (x, y)$ over a search region C roughly the size of the image itself, and how to efficiently find the global maximum of *votes*.

2.2.1 Objective function

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 line defined by $(\mathbf{p}, \theta(\mathbf{p}))$. Because our method of computing the dominant orientations has a finite angular resolution of $\frac{n}{\pi}$, uncertainty about the "true" $\theta(\mathbf{p})$ should spread this support over an angular interval. Thus, if the angle of the line joining an image pixel $\mathbf{p}$ and a vanishing point candidate $\mathbf{v}$ is $\alpha(\mathbf{p}, \mathbf{v})$, we say that $\mathbf{p}$ votes for $\mathbf{v}$ if the difference between $\alpha(\mathbf{p}, \mathbf{v})$ and $\theta(\mathbf{p})$ is within the dominant orientation estimator's angular resolution. Formally, this defines a voting function as follows:

$$\textit{vote}(\mathbf{p}, \mathbf{v}) = \begin{cases} 1 & \text{if } |\alpha(\mathbf{p}, \mathbf{v}) - \theta(\mathbf{p})| \leq \frac{n}{2\pi} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

This leads to a straightforward objective function for a given vanishing point candidate $\mathbf{v}$:

$$\textit{votes}(\mathbf{v}) = \sum_{\mathbf{p} \in R(\mathbf{v})} \textit{vote}(\mathbf{p}, \mathbf{v}) \tag{3}$$

where $R(\mathbf{v})$ defines a *voting region*. For straight, flat roads, we set $R(\mathbf{v})$ to be the entire image, minus edge pixels excluded from convolution by the kernel size, and minus pixels *above* the current candidate $\mathbf{v}$.

Only pixels below the vanishing line $\mathbf{l}$ implied by $\mathbf{v}$ are allowed to vote because support is only sought from features in the plane of the road (though in urban scenes out-of-plane building features, etc. may corroborate this decision [11]). This reduces ambiguity in the voting objective function and speeds the voting computation somewhat. A possible effect of this choice is a bias of *votes* toward higher image locations (since lower ones have fewer voters "eligible" to support them), but we have not found this to be an issue in practice.

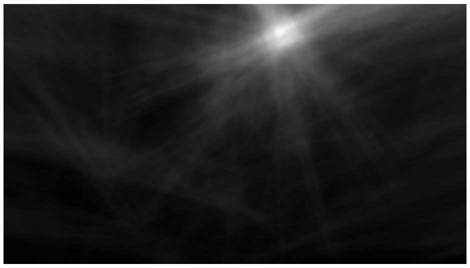


Figure 3: Votes for vanishing point candidates (top 3/4 of image in Figure 1(a))

In this work, we assume that $\mathbf{l}$ is approximately horizontal. Other researchers have inferred $\mathbf{l}$ for road and urban scenes by computing a second vanishing point obtained with either another set of parallel lines [13, 10, 16] or the cross ratio of equally spaced lines [10]. The road scenes we are considering have insufficient structure to carry out such computations.

We did not find that variations on the voting function *vote* 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. An obvious alternative criterion, the point-to-line distance between a candidate $\mathbf{v}$ and the ray defined by $\theta(\mathbf{p})$, was considered, but it performed poorly when there were few voters near the candidate. This occurred, for example, when the vanishing point was occluded by a vehicle further along the road.

An example of *votes* computed at a 1-pixel resolution over C = the top three-fourths of the image in Figure 1(a) (where the maximum vote-getter $\mathbf{v}_{max}$ is indicated with a cross) is shown in Figure 3.

2.2.2 Computational considerations

An efficient alternative to arithmetic computation of Equation 2 results from noting that the voting procedure is equivalent to rasterizing one thin isosceles "triangle of votes" per voter defined by $\mathbf{p}$, $\theta(\mathbf{p})$, and n in an additive accumulation buffer A in which each pixel is a vanishing point candidate $\mathbf{v}$. After drawing every vote triangle, the pixel in A (which represents C at a fixed resolution) with the maximum value is $\mathbf{v}_{max}$. This kind of *raster voting* can be conducted very quickly (especially on graphics hardware) vs. the slower *dot product voting* method described above. For example, voting can be carried out (given the dominant orientations) with every pixel in a 160×120 image as a voter in $\sim$35 ms even with no graphics card acceleration[1].

For dot product voting, an exhaustive search over C for the global maximum of *votes* of the type necessary to generate Figure 3 is costly and unnecessary. Rather, a hierarchical scheme is indicated in order to limit the number of evaluations of *votes* and control the precision with which $\mathbf{v}_{max}$ is localized. Because it integrates easily with the tracking methods described in the next subsection, we perform a randomized search by carrying out a few iterations of a particle filter [22] initialized to a uniform, grid-like distribution over C (with spacing $\Delta x = \Delta y = 10$). Exhaustive search adds little to the cost of raster voting, however. Overall, the vanishing point can be robustly estimated for 160×120 images with raster voting at roughly 6 fps, making vehicle control at the moderate speeds appropriate on ill-structured roads viable. For a slight accuracy penalty the frame rate can be pushed up to real-time with voter subsampling, smaller n, etc.

2.3 Tracking the Vanishing Point

Road curvature and/or non-planarity result in a set of different apparent vanishing points associated with different tangents along the section of road seen by the camera. Therefore the procedure of the preceding subsection, which assumes a unique vanishing point, must be modified to estimate the sequence of vanishing points which correspond to differential segments of road at increasing distances from the camera. Furthermore, over a sequence of images gathered as the camera moves along the road, the *vanishing point contour* thus traced for a single image must itself be tracked from frame to frame.

2.3.1 Road curvature in one image

Suppose the spine of the approaching road section that is visible to the camera is parametrically defined by a nonlinear space curve $\mathbf{x}(u)$, with increasing u indicating greater distance along the road. If $\mathbf{x}(u)$ lies entirely in one plane then the image of these vanishing points $\mathbf{v}(u)$ is a 1-D curve on the vanishing line $\mathbf{l}$. If $\mathbf{x}(u)$ is not in the plane, then the vanishing line varies with distance according to $\mathbf{l}(u)$ (still assumed to be horizontal) and $\mathbf{v}(u)$ is a 2-D curve.

We cannot directly recover $\mathbf{v}(u)$ since $\mathbf{x}(u)$ is unknown. However, u is a monotonically increasing function of the image scanline s, where $s = 0$ is the bottom row of pixels, so we can attempt to estimate the closely related curve $\mathbf{v}(s)$. This implies modifying Equation 3 to

$$votes(\mathbf{v}(s)) = \sum_{\mathbf{p} \in R(\mathbf{v}, s \pm \Delta)} vote(\mathbf{p}, \mathbf{v}) \qquad (4)$$

where $R(\mathbf{v}, s \pm \Delta)$ is now a differential horizontal *strip* of voting pixels centered on scanline s. Smaller values of the height of the strip 2Δ yield a more accurate but less precise approximation of the road tangent ($\Delta \approx 0.1h$, where h is the image height, for the results in this paper). s is iterated from 0 until $\mathbf{v}_{max}(s) \in R(\mathbf{v}, s \pm \Delta)$ (roughly the point where the strip crosses the vanishing line).

A strip-based approach to vanishing point detection for curvature estimation was also used in [12] for edge-detected road boundaries and lane lines, but with only a few non-overlapping strips. The resolution of their approach would be improved with overlapping strips, as we use.

Furthermore, we do not simply estimate a best vanishing point fit $\mathbf{v}_{max}(s)$ for every s independently by rerunning the full randomized search over C as described in the previous subsection. Rather, we *track* the vanishing point by continuing to run the particle filter with weak dynamics $p(\mathbf{v}(s) \,|\, \mathbf{v}(s-1))$ (e.g., a low-variance, circular Gaussian). This allows a more accurate estimate of $\mathbf{v}_{max}(s)$ because of the concentration of particles already in the solution area, and reduces the chance of misidentification of the vanishing point due to a false peak somewhere else in the image.

The vanishing point of each strip s implies a tangent to the image of the road curve at s. By hypothesizing an arbitrary point on the road, we can integrate this tangent function over s (i.e., with Euler steps) and thereby trace a curve or "flow line" followed by the road point. This procedure is illustrated in the next section.

2.3.2 Image sequences

As the vehicle travels down the road, the perceived road shape changes over time t according to $\mathbf{x}(u, t)$. A simplification that nonetheless works fairly well is to ignore road curvature within a single image and just track the global vanishing point $\mathbf{v}_{max}$ of the entire image from frame to frame.

However, following the analysis from the preceding subsection, we are interested how the vanishing point contour $\mathbf{v}(s, t)$ depends on its shape $\mathbf{v}(s, t-1)$ at the previous time step. A straightforward approach is to carry forward the start of the curve $\mathbf{v}(0, t-1)$ with some temporal dynamics to "seed" the tracker in the next

[1] For the raster voting results in this paper, the vote triangles are approximated by lines, and A has a bit-depth of 8, both of which can lead to some artifacts

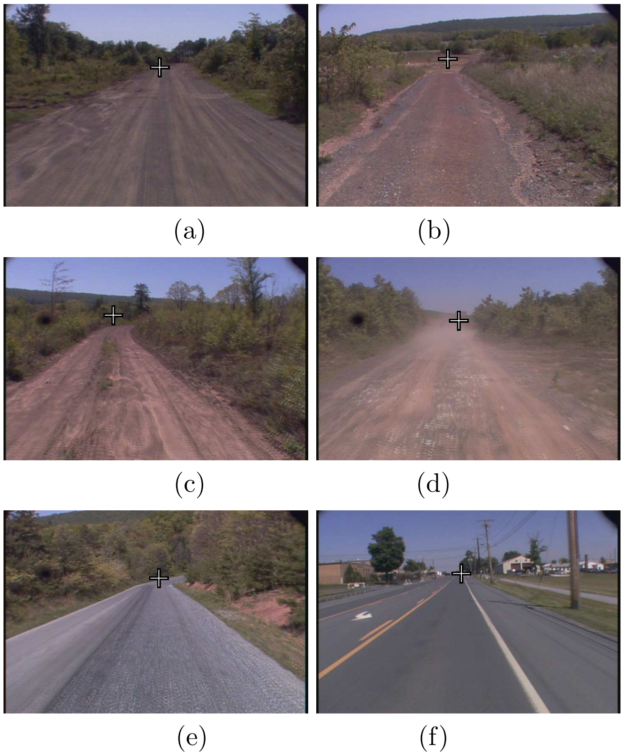


Figure 4: Computed vanishing points using the entire image as the voting region (dot product voting)

frame at $\mathbf{v}(0,t)$ and then track points along the vanishing point contour as described above. It is much less efficient and effective to track the state of the vanishing point contour as a whole by sampling entire curves because of the high dimensionality of the state space.

3 Results

Straight roads Vanishing point localization by dominant texture orientation voting worked robustly on predominantly straight roads with a wide variety of surface characteristics. Results on 720 × 480 DV camera images captured on a variety of roads at Ft. Indiantown Gap, PA (the "FITG" data) are shown in Figure 4. Here dot product voting was used, the voters were decimated by half vertically and horizontally within R, and the particle filter-based search used $\sim$600 particles. None of the dirt road examples in Figures 4(a)-(d) are truly straight or level, so some apparent discrepancies are due to these departures from the assumption. Although the algorithm was formulated especially for heavily banded roads, as Figures 4(e) and (f) show, it also works with mostly homogeneous asphalt roads by utilizing the few edges that are present in a manner similar to the edge-based methods described in the introduction.

Results on a more extensive group of scenes are shown in Figure 5. 16 illustrative images (the "Mojave" data) 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 particle-based search; the figure 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 $\sim$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 Mojave images. 16 subjects completed the study; 11 of their 256 choices (4.3%) 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).

Curved roads Examples of tracking the vanishing point for curved, non-planar roads from the FITG data are shown in Figures 6(a) and (b) (again using dot product voting). The yellow contour indicates the trace of the estimated vanishing point positions as the scanline s was incremented and the set of voters changed to points farther along the road. Horizontal movement of the vanishing point is of course proportional to left-right road curvature, and vertical movement indicates a rise or dip in the road. Thus we see that the road in Figure 6(a) is relatively level with a slight dip in the middle, which is correct. On the other hand, the vanishing point for the road in Figure 6(b) rises, indicating an approaching hill.

A more intuitive interpretation of the vanishing point motion can be obtained by simply integrating the implied tangent to the image of the road curve (the line joining any point on scanline s with its vanishing point $\mathbf{v}(s)$) for a few sample points. These are the green contours, or "flow lines," in the images in Figures 6. These curves are not meant to indicate the width of the road, for no segmentation has been carried out, but rather only to illustrate the structure of the terrain that the road ahead traverses.

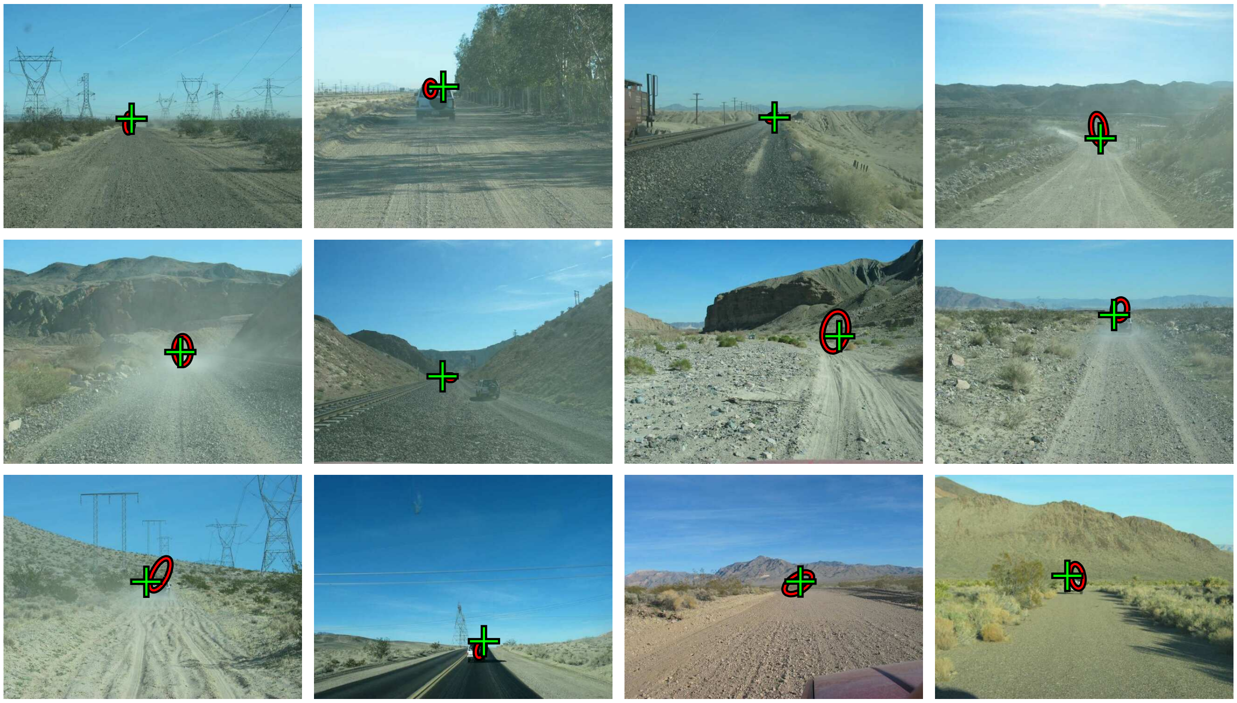


Figure 5: Computed vanishing points (raster voting) are shown as crosses, with Gaussian fits of a set of human responses marked with ellipses

Image sequences Vanishing point samples from the particle filter tracking a global vanishing point over a difficult sequence of images from the FITG data are shown in Figure 7 (the images were subsampled to 360×240 and dot product voting was used for this sequence). Near the start of the sequence, in Figure 7(a), the particles are coalescing around a consensus vanishing point after starting in a uniform distribution over the image. The cluster of particles tightly follows the road vanishing point despite a number of turns and bumps, and continues to track it even in the poorly defined area of Figure 7(d).

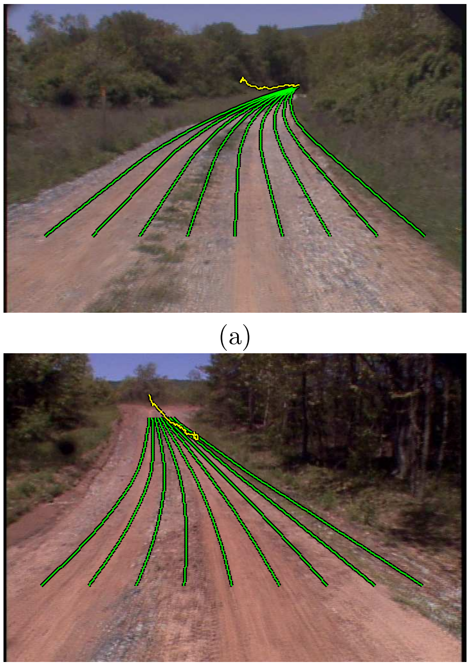
4 Conclusion

We have presented an algorithm for road following that at its most basic level relies on road texture to identify a global road vanishing point for steering control. By spatially partitioning the voting procedure into strips parallel to the vanishing line and tracking the results, we can estimate a sequence of vanishing points for curved road and recover road structure through integration. With no training, the system is robust to a variety of road surface materials and geometries, and it runs quickly enough for real vehicle control.

One algorithmic component that we have not discussed here regards road segmentation. The estimated road curvature alone does not provide information about the vehicle's lateral displacement that would allow centering–for this we need estimates of the left and right road boundaries. The ”flow lines” described above provide a powerful constraint by indicating the shapes these edges might take at hypothetical image locations. This allows the road segmentation task to be formulated as simply a 2-D search for a pair of flow lines which maximize the difference of some visual discriminant function inside the road region vs. outside it. We are currently experimenting with using this constraint with discriminant functions based on color as well as the local density of pixels that voted for $\mathbf{v}_{max}$.

5 Acknowledgments

Portions of this work were supported by a grant from the National Institute of Standards & Technology. The Mojave data was generously provided by Caltech's Grand Challenge team.



(a)

(b)

Figure 6: Tracks of vanishing point (yellow) for curved road and induced road "flow lines" (green)

References

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

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

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

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

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

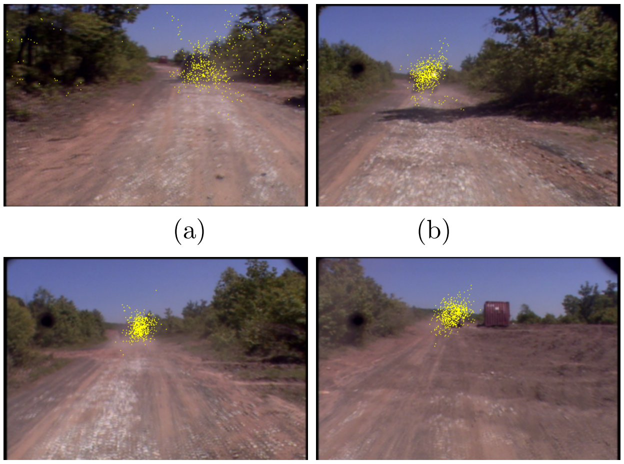
[6] J. Zhang and H. Nagel, "Texture-based segmentation of road images," in *Proc. IEEE Intelligent Vehicles Symposium*, 1994.

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

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

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

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



(a)

(b)

(c)

(d)

Figure 7: Particle filter vanishing point samples for a temporally-tracked sequence

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

[12] 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.

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

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

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

[16] E. Ribeiro and E. Hancock, "Perspective pose from spectral voting," in *Proc. Computer Vision and Pattern Recognition*, 2000.

[17] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert: Special Issue on Intelligent Systems and their Applications*, vol. 11, no. 2, pp. 19–27, 1996.

[18] 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.

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

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

[21] 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.

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