Vanishing Point: a Visual Road-Detection Program for a DARPA Grand Challenge Vehicle

Danko Antolovic
Pervasive Technology Labs, Indiana University
Alex Leykin
Department of Computer Science, Indiana University
Steven D. Johnson
Department of Computer Science, Indiana University

Abstract

This report describes a visual approach to road detection. A field of dominant trend lines is established over the image, by convolution with Gabor filters; the trend lines are then grouped into "visual segments," providing evidence for macroscopic linear features. Sets of these macroscopic features are evaluated against a list of heuristic criteria, to determine their likelihood of representing a road in the image. This procedure was implemented as software, for use by the Indy Robot Racing Team, a participant in the DARPA Grand Challenge 2005. It is capable of road detection at rates appropriate for driving at moderate speeds on dirt roads (ca. 5 Hz on a 3.4 GHz processor).

Introduction

Road detection in a video stream, at frame rates relevant to realistic driving speeds, has received some attention recently [1, 2], much of it due to the technological challenge put forward by the Defense Advanced Research Projects Agency (DARPA) in the years 2004-05 [3]. The challenge was to build an autonomous vehicle capable of traversing more than 100 miles of unpaved desert roads as rapidly as possible.

This is not a very general problem in computer vision, nor a very general cognitive problem, since the target (the road) is typically a prominent and fairly simple object in the vision field. The major issues are detecting the road reliably, and at an adequate frame rate.
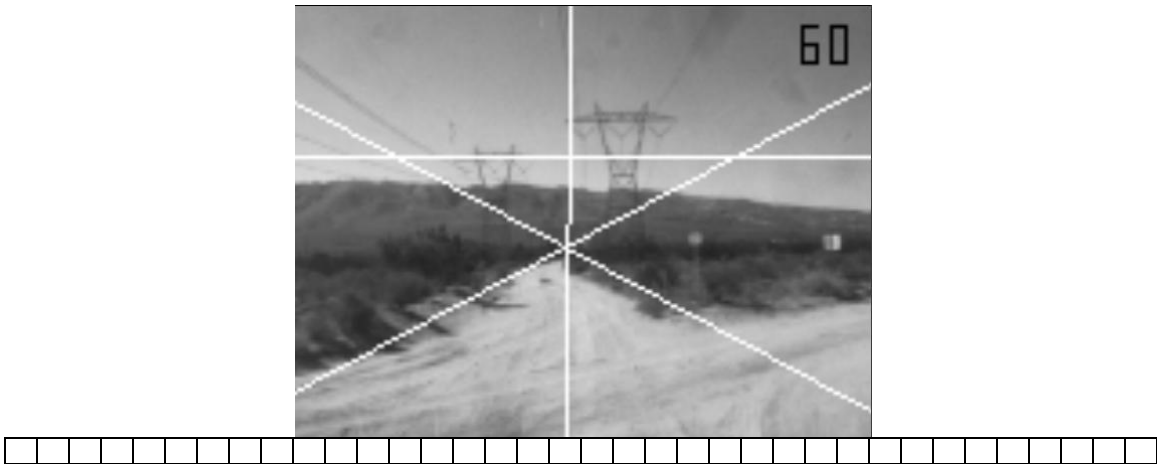

Figure 1. Elements of the road-finding output: road edges, center of the road, skyline and the confidence number, on the scale 0 - 100. Also drawn, but not displayed by the program, is the extended baseline of the image, divided into bins.

Trend Lines and Gabor Filters

Our method works with simple 8-bit grayscale, which we find adequately rich in information for road finding. The fundamental visual element is the trend line or "directionality" of the image data, in the neighborhood of a pixel. We measure this directionality as the magnitude of the convolution of a pixel's neighborhood with a Gabor filter.

Gabor filters are directional wavelet-type filters, or masks [4, 5]. They consist of a plane wave, in any direction in the image plane, modulated by a Gaussian around a point. The Gaussian localizes the response and guarantees that the convolution integral converges, and the plane wave affords a non-zero convolution over discontinuities that are perpendicular to the wave's direction. In other directions, the periodicity of the wave brings the convolution close to zero.

The phase of the wave, relative to a feature at the origin, is best accounted for by a sine/cosine combination of filters: cosine responds to even-parity features, sine responds to odd-parity features, and the other responses are vector combinations of these two. We are interested only in the magnitude of the response.

The common formulas for the Gabor filter are:

$$G_C(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos(\omega x')$$

$$G_S(x, y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\sin(\omega x')$$

where the wave propagates along the x' coordinate, which is rotated in some arbitrary direction relative to the image coordinates x and y; ω is the circular frequency of the wave, $\omega = 2\pi / \lambda$. Parameter σ is the half-width of the Gaussian, and defines roughly the size scale on which discontinuities are detected. Parameter γ is the eccentricity of the Gaussian: if $\gamma \neq 1$, the Gaussian is elongated in the direction of the wave, or perpendicular to it.

Repeated convolutions of the same mask with every pixel's neighborhood can be accelerated by convolving in the Fourier space, where the convolution of two functions is expressed as the product of their Fourier transforms. Intuitively, one can think of the same mask being translated across the image. In Fourier space, translations become multiplications by a phase factor, and so it is possible to transform the image and the mask once, multiply the transforms, then perform the inverse transform to obtain the value of the convolution at every pixel of the image at once [6].

In our program, the image is convolved with Gabor filters in a (large) number of evenly spaced directions, from zero to 180 degrees, and the direction with the strongest response is retained as the local direction field (trend line) of the image. Naturally, this calculation is by far the most computationally intensive part of the algorithm. We used the well-known Fast Fourier Transform library FFTW [7], the Windows version [8], in a single-precision, two-dimensional variant.

Theory of the Fourier transform is described in countless books and articles. For an account of its application to image processing, see [9]; a deeper mathematical description can be found in e.g. [10].

Segmentation

This term generally denotes partitioning an image into macroscopic features, i.e. grouping the pixels into image "segments" which can be treated as logical units. In this case, we are interested in the presence of objects that are roughly linear, and of substantial size. Such objects are likely to be part of the desired road, either as edges or as tracks/grooves along the road.

While the calculation of the trend lines is relatively abstract and detached from the image content, the segmentation step already allows some specialization, guided by the procedure's ultimate objective. The

driver of a terrestrial vehicle is primarily interested in features on the ground, in relatively close proximity to the vehicle. Rather than account for all linear features, we seek only those that intersect the lower edge of the image, extended in both directions (the image baseline), as shown in Figure 1.

Segmentation consists of two steps. First, the baseline is divided into (many) intervals of equal length (bins). The local direction field is scanned, the intersection of each trend line with a baseline bin is calculated, and the line strength is accumulated in a table indexed by baseline bins and direction angles. In short, evidence for linear features is accumulated in the (k,l) parameter space.

[Note: at this point the reader may ask whether this is just the Hough transform. In fact, this algorithm is simpler than Hough's, since the direction at each potential feature point is already known. The standard Hough transform [11] works with point locations only, and accumulates evidence for *all* lines that the point could belong to, regardless of the inclination. Also, Hough transform does not apply to grayscale, so the brightness changes must be first translated into solid lines by edge detection or thresholding, followed usually by thinning. None of these preliminaries are necessary in the Gabor mask approach.]

As a further simplification, the accumulator table is flattened along the angle coordinate. For every baseline bin, the strongest response is selected among the angles: these responses form a histogram-like structure along the baseline, with the provision that the angles at adjacent points are unrelated, and are recorded separately. This simplification was introduced to speed up the calculation, but is not fundamental to the algorithm.

Second step is the interpretation of the baseline histogram. This histogram is expected to be multimodal (to have many peaks), and the peaks can be reasonably interpreted as intersections of the linear features with the baseline.
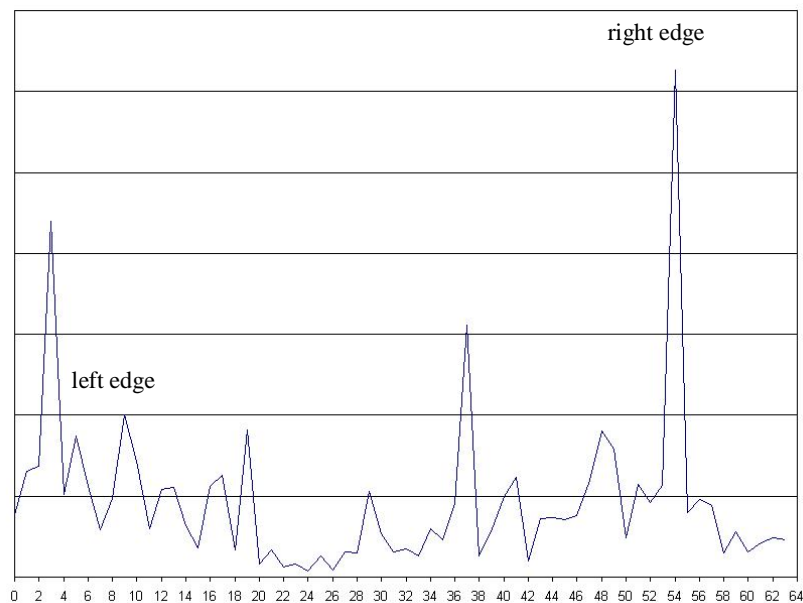

Figure 2: Baseline histogram of the scene shown in Fig. 1. Baseline is divided into 64 bins, and the peak heights are the accumulated strengths of Gabor responses, on a relative scale. Note that the left edge shows a less prominent peak than some of the surrounding features (see the section on feature interpretation)

Unfortunately, merely finding maxima in a histogram can be deceptive, since visually rich images often produce multiple sharp peaks which are difficult to interpret (see Fig. 2). We use an iterative algorithm, which breaks up the histogram into segments and cyclically readjusts the boundaries and mean values of the segments [6, 12]. In the authors' experience, this algorithm is quite successful at grouping histogram peaks

into physically meaningful features.

Feature Interpretation

The segmentation step yields a list of linear features which intersect the baseline. These features are indicated by an accumulation of roughly collinear trend lines, but the pixels of the feature need not be contiguous in the image. This can give rise to some spurious features, but it also makes the road detection resistant to multiple small breaks in the road edges, which are very common.

Finally, the decision has to be made: what in the collection of linear features represents the road? For simplicity, we define the road as a pair of its outer boundaries (edges), but the road could also be represented by a collection of converging tracks and edges, or something similar. If the edges define the road, which pair of features should be selected to represent the edges of the road?

We emphasize that there is no general, context-free criterion for this selection. For example, one cannot simply select two most prominent features, because telephone poles, for example, yield strong linear features which have nothing to do with the road. Similarly, looking for intersecting lines (i.e. vanishing points) can be misleading because the road may curve in the distance, mountain slopes and the horizon create spurious intersection points etc. Road elements are recognized as such because their relationship to each other, and to other elements of the image, falls within certain common-sense constraints.

In order to check all these geometric constraints explicitly, one would have to build and maintain an unmanageably large decision tree. The right answer to this type of problem is an adaptive system, which could be guided by a human driver until it establishes an internal representation of what a valid road looks like. For reasons of expediency, we take a simplified approach instead.

We maintain a list of heuristic criteria (static from the point of view of the algorithm), which represents some human knowledge about roads. Each criterion carries an increment value, which, again heuristically, represents the criterion's relative importance. For example, an edge leaning to the right from the vertical is viewed more favorably if it intersects the left side of the baseline than if it is located to the right. Some criteria carry an infinite negative increment (the show-stoppers). For example, a pair of diverging edges is too unlikely to be considered at all.

Every pair of features is tested against this list of criteria, and is assigned a quality function (QF), which is the accumulation of increments from all the criteria that the pair passed. The pair with the largest QF is selected to represent the road, and the value of the QF (as the fraction of the largest possible QF) is output as the measure of confidence in the decision made.

Auxiliary Criteria

It turned out that the linear features alone weren't enough to detect the road reliably, at least in the driving situations that we were interested in. First, power lines in the sky contribute linear features that can sometimes satisfy the criteria for road edges. This led to adding a simple skyline detection, which roughly separates the uniformly bright sky above from the darker ground below. Features above the skyline are reliably eliminated.

Second, in a sufficiently rich image, it is often possible to find spurious edge pairs which happen to have a reasonably high QF. For example, the algorithm would sometimes confidently find a road in an impassable field of boulders. We built an additional bit of context into the decision-making: the road has to be of reasonably uniform brightness, and reasonably different (darker or brighter) from the rest of the ground.

[Note: this phenomenon of "seeing things" is common to vision systems. For example, humans are good at discerning faces in meaningless, but sufficiently rich, patterns. The way to avoid this kind of confusion is to find appropriate contextual information, which can separate hallucinations from reality.]

Finally, it turns out that there is short-term variability in the video stream, as the appearance of the features

changes from frame to frame. The program maintains a buffer which holds the results of the analysis of the previous few frames, and a QF increment is added to those edge pairs which are close to the previous results. If something was a road a moment ago, it is likely still a road.

Results

The road-finding program runs on a 3.4 GHz Hewlett-Packard laptop, and receives the video stream from a Dragonfly camera made by the Point Grey Company [13]. Camera output is spatially downsampled to 160 by 120 pixels, and color is converted to grayscale before processing. Processing frequency is ca. 5 frames per second, and is essentially independent of the image complexity. This is due to the fact that the computationally intensive steps, the FFTs and the line accumulation scan, depend only on the frame size, not on the image content.

We tested the vision system on dirt roads in the high desert north of San Bernardino range, in southern California. During the field tests the software also collected road footage, which can be played back into it, in lieu of camera input.

Understandably, the algorithm is best at detecting relatively straight roads. It is reasonably immune to extraneous objects such as trees, telephone poles and power-line pylons, shadows across the road and mountain slopes in the distance. Its confidence number decreases when the extraneous clutter increases; in our estimation, the confidence above 50% is acceptable.

References

[1] Christopher Rasmussen, "Texture-Based Vanishing Point Voting for Road Shape Estimation," British Machine Vision Conference, 2004, http://www.bmva.ac.uk/bmvc/2004/papers/paper_261.pdf

[2] A. Broggi, A. Fascioli, "Artificial Vision in Extreme Environments for Snowcat Tracks Detection", IEEE Transactions on Intelligent Transportation Systems, 3 (3), 2002.

[3] http://www.darpa.mil/grandchallenge/index.html

[4] http://www.cs.rug.nl/~imaging/simplecell.html

[5] T. Lee. Image representation using 2D Gabor wavelets. IEEE Trans. Pattern Analysis and Machine Intelligence, 18(10):959–971, 1996.

[6] M. Sonka, V. Hlavac, R. Boyle, "Image Processing, Analysis and Machine Vision," (Chapman & Hall, 1993)

[7] http://www.fftw.org/

[8] http://www.ece.cmu.edu/~franzf/fftw.org/

[9] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, "Computer Graphics, Principles and Practice," (Addison-Wesley, 1991)

[10] F.W. Byron, R.W. Fuller, "Mathematics of Classical and Quantum Physics," (Dover, 1992)

[11] E.R. Davies, "Machine Vision," (Academic Press, 1990).

[12] Ridler and Calvard, "Picture Thresholding Using an Iterative Selection Method," IEEE Transactions on Systems, Man and Cybernetics, 8(8):630-632, 1978.

[13] http://www.ptgrey.com/