Fast Roadway Detection using Car Cabin Video Camera

Daria Krokhina[a], Veniamin Blinov[b], Sergey Gladilin[b], Ivan Tarhanov[c], Vassili Postnikov[a,c]
[a]National University of Science and Technology MISiS,
[b]Institute for Information Transmission Problems, [c]Cognitive Technologies

ABSTRACT

We describe a fast method for road detection in images from a vehicle cabin camera. Straight section of roadway is detected using Fast Hough Transform and the method of dynamic programming. We assume that location of horizon line in the image and the road pattern are known. The developed method is fast enough to detect the roadway on each frame of the video stream in real time and may be further accelerated by the use of tracking.

Keywords: Image processing, visual intellect, autonomous driving, roadway detection, fast Hough transform.

1. INTRODUCTION

Visual road following is one of the basic problems of autonomous driving. A robust roadway detection and tracking algorithm are core elements of self-driving systems. As any other core elements, they must be as fast and reliable as possible.

Existing methods fall into three categories: edge-based [1-5], region-based [6,7] and other models (e.g. feature-based) [8]. Sometimes authors fuse methods from different classes to improve accuracy and robustness [8].

Region-based methods exploit color or pattern similarity to distinguish road and non-road areas. This is achieved by solving a classification problem either by using simple filters [8] or by means of neural networks [7]. Methods of this class are relatively robust, yet need a careful tuning of parameters (learning a neural network in the latter case).

Compared to their region-based analogs, edge-based methods are more universal for they do not require such careful parameter tuning. The main idea is to highlight edges on the image (e.g. with Canny or DoG filter) and find those corresponding to the road bounds. If the bounds are straight, one can apply Hough transform. In other cases, more crafty methods are utilized [9, 10].

The last group of methods exploits specific features of the road, e.g. special road markings, guardrails, trolleybus wires etc. These methods are usually application-specific.

In the present paper we improve the existing edge-based methods by applying Fast Hough Transform, introduced in [11,12]. The non-accelerated variant of Hough transform is slow and researchers are forced to use its limited version [1, 2], or apply the transform once per several frames [3, 4] to avoid scaling issues. Exploiting the full power of the Fast Hough Transform promotes the methods of this class to real-time ones, keeping all their strengths in the same time.

Another difference from conventional schemes is that the latter use the existence of vanishing point as a constraint for road detection. Identification of the vanishing point can be based on voting procedure (based on Hough transform [1-5] or gradient directions [13]) and provides a good tracking invariant. However, in some cases the information of the horizon line position can be tracked more effectively than the vanishing point. That is why we shall confine ourselves to the case when the location of the horizon is available as an input along with the image.

2. PROBLEM FORMULATION

The cross-correlation technique is a method to get the displacement information of two consecutive images by comparing the similarity of a pair of image signals [6, 7, 8]. The traditional cross-correlation method was showed in Fig.1 and the cross-correlation coefficient was defined as formula

The properties of an image captured by a car camera can vary in many different ways. Weather, time of day, terrain, road quality affect the output image. It would be too presumptuously to start making an algorithm that is effective in any

Eighth International Conference on Machine Vision, edited by Antanas Verikas,
Petia Radeva, Dmitry Nikolaev, Proc. of SPIE Vol. 9875, 98751F · © 2015 SPIE
CCC code: 0277-786X/15/$18 · doi: 10.1117/12.2229617

situation, so we shall restrict ourselves to a specific case. We shall imply that the input to our algorithms contains: (1) an image with a straight road going forward; the road has clear edges; (2) position of the line of horizon on that image; (3) number of visible (clearly separated) road lanes or other straight lines that define the road. Re-formulating this demands in a more mathematical manner, we can say that we consider a road as a part of the earth's surface bounded by two (perhaps discontinuous) straight lines intersecting at the vanishing point (VP) on the horizon. The road can contain other lines passing through the vanishing point:

1. Road markings, separating the lanes on a multilane road, see fig. 1a;
2. Lines formed by the road construction elements: beams, rails, road fence or concrete slabs edges, see fig. 1b;
3. Off-road wheel tracks, see fig. 1c.

These lines divide the road into "lanes". The number of "lanes" and their widths form a pattern of this road. The latter can be obtained experimentally from external sources (e.g. taken from a satellite map), or detected directly from the video stream by a separate algorithm.

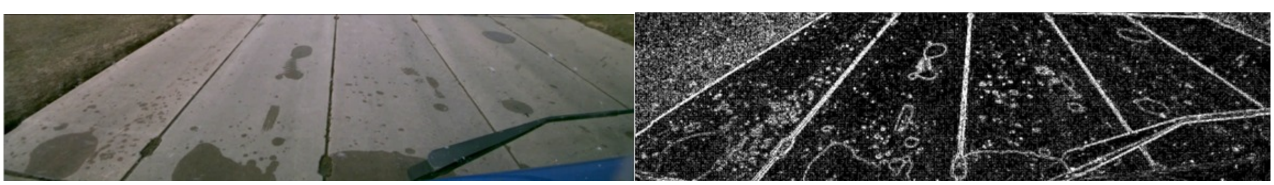
a b c

Figure 1. Types of lines on a road: (a) paint markings, (b) construction elements edges, (c) wheel tracks.

Using the introduced notion we can now formulate the requirements of the algorithm more accurately. The input data are: a photo image from vehicle cabin, the horizon location in it and a pattern of the road. The algorithm finds road edges and a set of lines on the road in accordance with the pattern.

3. THE ALGORITHM

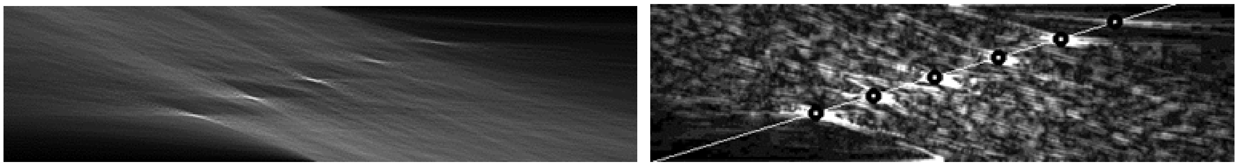
On the first step we convert the input image to grayscale and pass through a DoG (Difference of Gaussians) filter. Such preprocessing highlights the edges and prepares the image for a Fast Hough transform. This procedure is illustrated in fig. 2.



a b

Figure 2. Source image (a) before and (b) after DoG filter.

The second stage of the algorithm is Hough transform. We generate a map of integral intensities of lines in the input image. The key idea is to use a fast version of Hough transform to reduce the complexity from $O(w^2h)$ to $O(wh \log h)$ (w and h correspond to the input image width and height respectively). Such crucial performance gap is the main reason this algorithm can be applied in real time. Instead of the classical HT parameterization (ρ,ϕ), fast Hough transform uses the space (x_0,shift) [12]. In this case the images of lines intersecting at vanishing point are the points on a straight line. Fig. 3a illustrates the output of the transform: straight lines of the input image (fig. 2b) appear as bright points lying on a straight line which is the image of the vanishing point, fig. 3b.



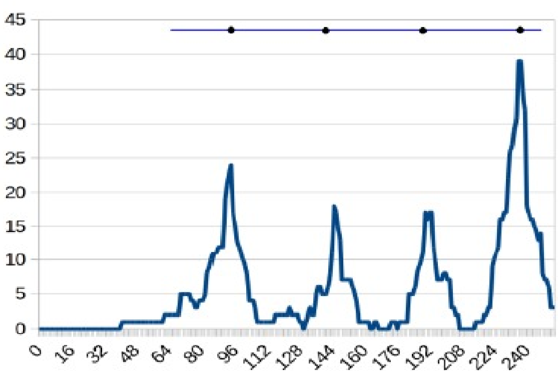
a b

Figure 3. (a) Hough-image; (b) DoG-filtered Hough-image.
The straight line corresponds to the vanishing point.

A search for a vanishing point constitutes the third step of the algorithm. The fast Hough transform described in [12] gives the output in coordinates (*a; b*), corresponding to the slope-intercept lines form $y = ax + b$. In contrast, the conventional "slow" Hough transform outputs in polar plane (r, ϕ), implying $y = -ctg(\phi)x + r/sin(\phi)$ lines in the input image. In the latter case the image of point (x_0, y_0) (as a set of all lines containing this point) is a set of (r, ϕ) that satisfy $r = x_0 \cos(\phi) + y_0 \sin(\phi)$, i.e. a sine curve. The fast version of the transform maps the same point to a straight line $a = -b/x_0 + y_0/x_0$. This means that we can now apply another Hough transform to identify the VP.

In practice, an attempt to find a vanishing point in the whole image gives multiple answers and is not robust in general. To avoid these problems we restrict the search to a single line – the horizon. Beside the robustness, this restriction accelerates the search – it takes *O(wh)* to perform this procedure and find the vanishing line (VL), i.e. a line in Hough space that corresponds to the vanishing point. This estimate is due to linear relation between the input image size and the size of corresponding Hough space.

As we have identified the vanishing point we can proceed to the next step. The goal of this step is to find road edges and edges of lanes. Here we need two objects as an input: the intensities of pixels along the vanishing line and the road pattern. A sample of pixels intensities distribution along VL is presented in fig. 4.



The line with bold points on the top represents a road pattern.

Figure 4. The dependence of pixel intensities on the x-coordinate in the image.

At first glance, the demand to have a road pattern as an input may seem artificial. However, it is a natural way to improve algorithm robustness. Firstly, the information about the pattern of current road can be extracted from a satellite map or from another source. Secondly, even if the pattern is unknown, we can use a pattern of two lines and the road width and it will work (though less precisely).

Proposed algorithm searches the set of X-coordinates, sum of intensities in which is maximal, and the distances between which conform with the road pattern. Such a problem can easily be solved by a simple convolution method. However, we allow X-coordinates to deviate from the ideal road pattern up to some threshold ε. This makes the algorithm more robust, but also computationally much harder. Exhaustive search using a functional that allows deviations of X-coordinats from the road pattern is not computationally feasible. To speed up the process without loss of precision we use dynamic programming.

4. EXPERIMENT

The developed algorithm has been tested on different types of roads (fig. 5).

To estimate the error of roadway detection on the frame we performed a three-dimensional reconstruction of the scene. Let A_0A - found roadway edge line and B_0B - ideal (marked by human) roadway edge line. We calculated the angle between the planes FA_0A and FB_0B (fig. 6a), where F is the optical center of the video frame.

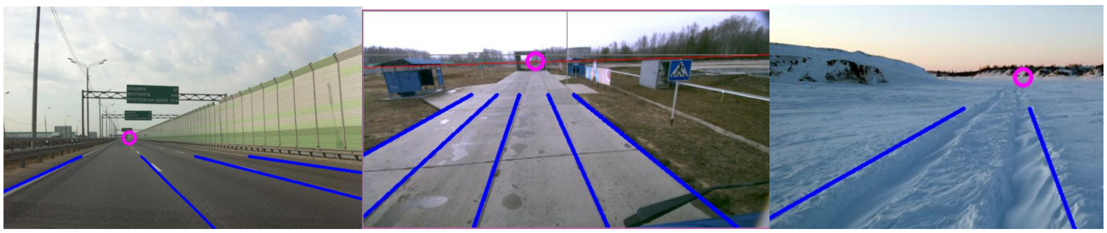
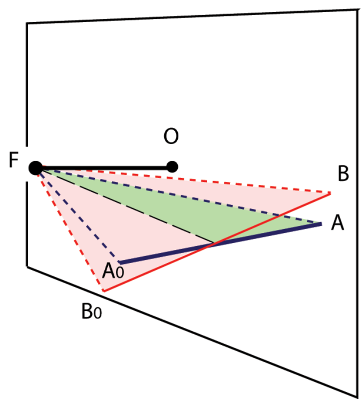
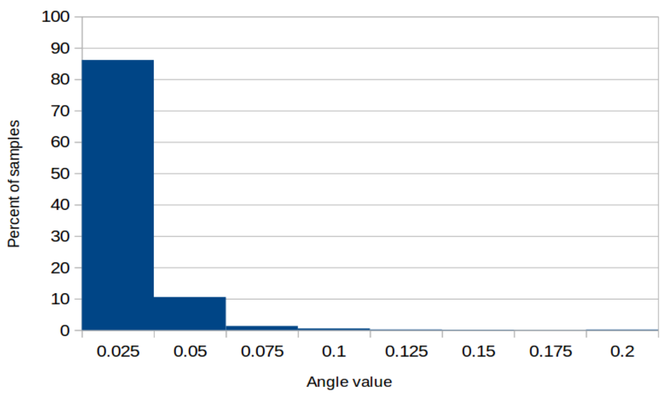


Figure 5. Recognition results



a



b

Figure 6. a. Illustration of two planes involved in error calculation. FO – the focal length of the camera (FO is normal to the frame plane), B0B - real road position, A0A - road position found by the algorithm.

b. Distribution of error angle (in radians) based on 5974 sample images

To estimate the quality of the developed algorithm we applied it to several series of videos from cabin of a car moving on a road of concrete slabs. For each of 5974 frames we measure the angle as described above. The resulting histogram is presented in fig. 6b.

This algorithm has been implemented on a prototype of an autonomous-driving car. For a straight section of road of concrete slabs the algorithm demonstrated a good recognition quality and required about 30ms per frame on Core i7 2.4 GHz without using GPU.

5. CONCLUSIONS AND FURTHER STUDY

Edge-based road detection algorithms are robust and demonstrate good accuracy. Their main drawback is low speed caused by the utilization of Hough transform (HT). We have demonstrated that the utilization of fast Hough transform (FHT) instead makes such algorithms real-time fast. Moreover, FHT has the property of duality, meaning a point on the source image corresponds to a line in the Hough space. This property makes dealing with fast Hough space more essential compared to the ordinary Hough space, where a point corresponds to a sine curve.

Using road patterns as an additional information makes the algorithm more robust.

Further improvement of the algorithm may include providing tracking of lines and filtering (e.g. Kalman) to avoid outbursts. Another possible upgrade is a fusion with a region-based recognition method, that could make the algorithm more robust in case the road is turning. Despite of the fact, that the algorithm is designed to work with straight road segments, the Hough transform has an intrinsic quality estimation ability, which can be used to detect bad cases and force the system to switch to another method (see e.g. [14, 15]).

ACKNOWLEDGEMENT

Applied scientific research is supported by Ministry of Education and Science of Russian Federation (project № RFMEFI57914X0011)

REFERENCES

[1] Maire F., Bigdeli A. Obstacle-free range determination for rail track maintenance vehicles. Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on. IEEE, 2010.

[2] Waxman A.M. et al. A visual navigation system for autonomous land vehicles. IEEE Journal of Robotics and Automation, 3(2), 124-141, 1987.

[3] Fardi B., Wanielik G. Hough transformation based approach for road border detection in infrared images. Intelligent Vehicles Symposium, IEEE, 2004.

[4] Mattern N., Schubert R., Wanielik G. Image landmark based positioning in road safety applications using high accurate maps. Position, Location and Navigation Symposium, IEEE/ION, 2008.

[5] Tuytelaars T., Proesmans M., Van Gool L. The cascaded Hough transform as support for grouping and finding vanishing points and lines. Algebraic Frames for the Perception-Action Cycle. Pp. 278-289, Springer Berlin Heidelberg, 1997.

[6] Zhang J., Nagel H.-H. Texture-based segmentation of road images. Proceedings of the Intelligent Vehicles' 94 Symposium, IEEE, 1994.

[7] Stoklasa R., Matula P. Road Detection Using Similarity Search. 2nd International Conference on Robotics in Education, 2011.

[8] Kluge K., Thorpe C.E. Explicit models for robot road following. Vision and Navigation. Springer US, 25-38, 1990.

[9] M. Shahed Nejhum. Implementation of B-snakes //university of Florida URL: https://www.cise.ufl.edu/~smshahed/ImgSeg05Report.pdf.

[10] Ballard D.H. Generalizing the Hough transform to detect arbitrary shapes. Pattern recognition, 13(2), 111-122, 1981.

[11] Brady, M.L. 1998. A fast discrete approximation algorithm for the Radon transform. In SIAM J. Computing, V. 27(1), 107-119.

[12] Nikolaev D.P. et al. Hough transform: underestimated tool in the computer vision field. Proceedings of the 22th European Conference on Modelling and Simulation, 2008.

[13] Kong H., Audibert J.-Y., Ponce J. Vanishing point detection for road detection. IEEE Conference on Computer Vision and Pattern Recognition, United States, pp.96-103, 2009.

[14] A.Grigoryev, T.Khanipov, D.Nikolaev., Determination of axle count for vehicle recognition and classification, 8th Open German-Russian Workshop «Pattern Recognition and Image Understanding»: Workshop proceedings. — Nizhny Novgorod, 2011. — p. 89–91.

[15] A. Grigoryev, D. Bocharov, A. P. Terekhin, and D. P. Nikolaev, "Vision-based vehicle wheel detector and axle counter," in 29th European Conference on Modelling and Simulation, ECMS 2015, Albena (Varna), Bulgaria, May 26-29, 2015. Proceedings. V. M. Mladenov, G. Spasov, P. Georgieva, and G. Petrova, eds., 521–526, European Council for Modeling and Simulation (2015).