Bulletin of the *Transilvania* University of Braşov • Vol. 8 (57) No. 2 - 2015
Series I: Engineering Sciences

DYNAMIC ROAD STRUCTURE ESTIMATION

S.M. GRIGORESCU[1] G. MĂCEŞANU[1]

Abstract: *The paper presents a road structure estimation system capable of robustly segmenting and tracking lanes perceived using a camera mounted on a vehicle. The algorithm is based on a camera calibration procedure that is able to calculate in real time de roll, pitch and yaw of the vehicle. Further, the method is able to compute not only the ego-lanes, on which the car is driving, but also the neighbouring lanes. The approach is evaluated in two different driving scenarios, that is, highway and inner city roads driving.*

Key words: *lane detection, advanced driver assistance systems, camera calibration.*

1. Introduction

The field of image processing techniques, for both complex traffic scene analysis and safe vehicle driving [1], [5], gains more and more attention mainly because of the autonomous vehicle driving goal [3]. In many cases, the algorithms are focused on resolving parts of the main goal (e.g. recognising traffic signs, detecting and classifying lane markings, lane departure warning, recognising vehicles etc.).

In this paper a road structure estimation systems [6] is presented, namely the design of a lane detector that can be used in various automotive scenarios, such as driver assistance and autonomous driving.

2. Computational Road Model

One of the main goals of a road structure estimation algorithm is to determine the model of the road [2], shown in Figure 1. Mathematically, the road model can be represented as a 5-element vector:

$$[\varphi, \theta, p, w, c]^{\mathrm{T}}, \tag{1}$$

where φ [rad] is the pitch angle and it measures the rotation around the X axis, θ [rad] is the yaw angle and it measures the rotation around the Y axis, p [m] is the lateral offset between the middle of the ego-track and the middle of the ego-vehicle, w [m] is the track width, defined as the distance between the left and right lines of a lane and c is the curvature of the track.

First of all, a correspondence between real world (X, Y, Z) coordinates and image (u, v) coordinates needs to be determined.

From the two similar right triangles that have a red hypotenuse, blue and dashed black sides the following Equation can be deduced:

$$\frac{(v + f_v \sin\varphi)\cos\varphi}{(f_v - v\tan\varphi)\cos\varphi} = \frac{h_0}{l - l_0}. \tag{2}$$

[1] Automation and Information Technology Dept., *Transilvania* University of Braşov.

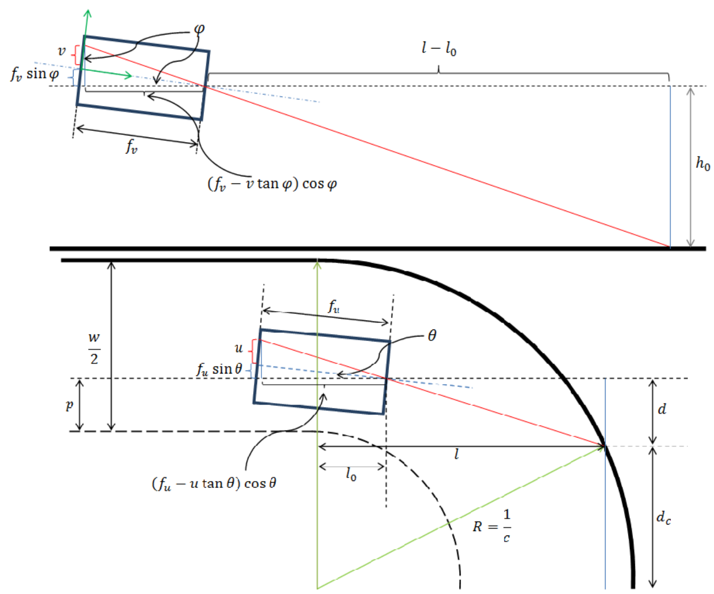

Fig. 1. *The projection of the road onto the image domain*

Since the pitch angle φ is considered small, the following simplifying assumptions can be made:

$$v + f_v \sin\varphi \approx v + f_v\varphi, \quad f_v - v\tan\varphi \approx f_v. \tag{3}$$

Thus, a simple vertical coordinate transformation is obtained:

$$v = f_v\left[\frac{h_0}{l - l_0} - \varphi\right]. \tag{4}$$

For the horizontal projection, the 3D lateral distance is computed:

$$d = R - d_c - \frac{w}{2} + p, \quad d_c = \sqrt{R^2 - l^2} = R\sqrt{1 - \left(\frac{l}{R}\right)^2}. \tag{5}$$

Under the assumption that the look-ahead distance is much smaller than the radius of curvature of the lane $l << R$, the square root could be approximated by:

$$\sqrt{1 - \left(\frac{l}{R}\right)^2} \approx R - \frac{1}{2}cl^2. \tag{6}$$

Therefore the approximate lateral distance is:

$$d = \frac{1}{2}cl^2 - \frac{w}{2} + p. \tag{7}$$

Next, the two similar right triangles in the image that have red hypotenuses and blue opposite sides are considered, obtaining:

$$\frac{(u + f_u \sin\theta)\cos\theta}{(f_u - u\tan\theta)\cos\theta} = \frac{d}{l - l_0}. \tag{8}$$

Again, similarly to the vertical projection case, the yaw angle θ is considered to be small, leading to:

$$\frac{u}{f_u} + \theta = \frac{d}{l - l_0}, \quad u = -f_u\left[-\frac{d}{l - l_0} + \theta\right]. \tag{9}$$

Thus:

$$u = -f_u \left[-\frac{\frac{1}{2}cl^2 + p - \frac{w}{2}}{l - l_0} + \theta \right]. \quad (10)$$

3. On-line Camera Calibration

The calibration process [4] aims at determining the parameters of the camera that are varying during a driving scenario. These are camera orientation angles yaw, pitch and roll. For this process, only visual information is used. Different approaches are used to determine the roll angle than the pitch and yaw angle.

3.1. Roll Angle Estimation

In the first step, the acquired image is searched for gradient features that can be tracked. The gradient is computed only along the *X* image axis since is the most affected when the camera rolls. By thresholding the value of the gradient for each image point only strong features are identified. Further, the orientation of the gradient is determined as follows:

$$orientation = \frac{\frac{s_F\, d_R\, g_y}{g_x} + 90 d_R\, d_S}{d_S}, \quad (11)$$

where, s_F is the slope factor; d_R is the resolution of the direction; d_S is the gain used to scale the orientation and g_x and g_y are the magnitudes of the gradient. The slope factor is computed as follows:

$$s_F = d_S \cdot \frac{180}{\pi}. \quad (12)$$

Next, a histogram of the gradient orientation is constructed. If a large amount of features will point in a certain direction, then the histogram will have a distinctive global maximum. Using this, the roll angle is computed as follows:

$$roll = \frac{\pi}{180}\left(90 - \frac{g_M}{d_S}\right), \quad (13)$$

where, g_M is the global maximum obtain from the histogram of the gradient orientation.

3.2. Pitch and Yaw Estimation

For this purpose the same features used for roll estimation are again used within an optical flow approach. Across frames, the optical flow describes the movement of the features in the scene. The vectors obtained are further intersected in order to create a cloud of intersecting points. The mass centre of these points defines the *focus of expansion* (FOE) which is a point in the horizon. For a more accurate position of this point, around him a region of interest is defined and inside this the FOE point is again determined.

The yaw and pitch angles are determined using the coordinates of FOE point and the coordinate of the camera centre point (C_p), which for an image with the resolution 640x480 is approx. 320x240.

From this model, the equations used to calculate the yaw and pitch angles out of FOE coordinates are the following:

$$yaw = \frac{x_{FOE} - x_{Cp}}{f}, \qquad pitch = \frac{y_{FOE} - y_{Cp}}{f}, \quad (14)$$

where, x_{FOE} and y_{FOE} are the x and y coordinates of the FOE point, f is the focal distance and x_{Cp} and y_{Cp} are the x and y coordinates of the camera central point.

4. Road Features Detection and Tracking

The algorithm aims at estimating the structure of the road, more specifically, detecting the current lane on which the vehicle is driving (ego-track) and the neighbouring left and right lanes, classifying the respective lane markings (e.g. dashed short, solid etc.), detecting the vehicles present in traffic and estimating the distance between the ego-vehicle and the other traffic participants.

4.1. Ego and Neighbouring Lanes Detection

The lane detection algorithm can be conceptually divided into two main components: feature extraction, in which the image features (e.g. gradient) are computed, and model fitting, where the previously detected features are fitted to the road model and the model is adapted accordingly.

The process of detecting lane markings is divided into two stages: the global search of lane markings (detection stage) and the local search (update stage), both stages being governed by the road model $[\varphi, \theta, p, w, c]^T$, meaning that the feature extraction is performed on the predicted road structure.

The detection (initialization) stage has at its core two detection algorithms: a Hough-line based one and an Inverse Perspective Mapping (IPM) one. The two methods are used alternatively, based on the previous fitting likelihood. The consecutive bad fitting likelihoods are counted, and if the resulted number is above a threshold value, the current detection method is switched with the other one, as illustrated in Figure 2.

The Hough-line detection algorithm uses the gradient image in order to obtain the initial lane hypotheses, by evaluating the segmented pixels collinearity through the Hough transform. The left and right lines are calculated using the length and slope of the determined hypothesis.

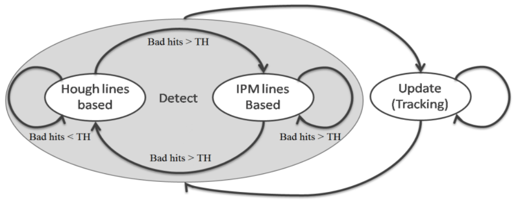

Fig. 2. *The detection switcher algorithm. If the number of bad detections is greater than a threshold (TH), the currently used method is switched with the other one. If the detection process succeeds with a high enough confidence, the update phase is started*

Using the found horizon line, a filtering process takes place. The lines are grouped successively, two at a time, and their intersection point is found. If the intersection point is too far from the horizon line, the current pair is eliminated. The remaining rightmost and leftmost lines, having the correct slope and oriented in the correct direction are the best candidates for the ego-track right and left lines.

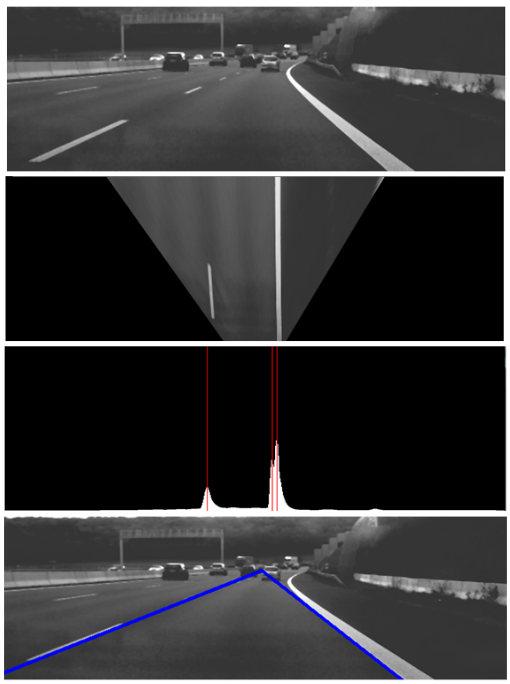

Fig. 3. *The Inverse Perspective Mapping Algorithm*

The IPM algorithm consists of mapping the original image to an aerial view (bird's eye view), as shown in Figure 3, using the Inverse Perspective Mapping transform. The perspective view distorts the actual image of the road, such that the lines that were initially convergent into the vanishing point become parallel.

5. Experimental Results

5.1. Experimental Setup

For evaluation purposes, a uEye camera was used to acquire visual information from the driving scene. The camera was mounted on the windshield just below the rear mirror. The configured resolution was 640x480. The lane detection algorithm was ran on an Core I7 platform with 4 GB of RAM.

In order to test the proposed algorithm two scenarios were considered: highway and national roads. The first scenario aimed at testing the accuracy and the speed of the approach whereas in the second scenario robustness against noise and faded lane markings were analysed.

5.2. Performance Metrics

The evaluation of the lane detection procedure has been made with respect to the Euclidean distance between the 2D points defining the ego track estimated model and the 2D points of the ego track annotated using a special tool. Since the main goal of the proposed algorithm is to determine the road model with respect to the ego track, this distance can be considered to be a fitting measure. The evaluation procedure is applied only on the ego track.

The final metric is obtained by summing the Euclidean distance between each 2D point of the estimated ego track model and the closest to it annotated point. In order to avoid bad sparse 2D points, the distance is thresholded to a distance equal to one courter of the ego track width, the point not fulfilling this constrained being eliminated from the metrics. The summed distances are further normalized against ½ of the ego track width, the final fitting metric being defined in the interval [0, 1].

A given frame is considered to be good (a hit) if the fitting likelihood is grater then 0.87, respectively bad (miss) if below.

Due to the good condition of the lane markings and also to the lack of tight curves, highway scenarios are an ideal scene where to test the lane marking algorithm. Table 1 describes the precision and recall of the algorithm in different conditions of a highway scenario.

Highway detection precision Table 1

Scene	Frames	Recall	Precision
Highway 1	880	0.9215	0.9188
Highway 2	774	0.9411	0.9345
Highway 3	13283	0.9113	0.9056
Highway 4	3451	0.9279	0.9266

The highway 1 and 2 scenarios from Table 1 were taken in a sunny day, on a road with 3 lanes per way and light traffic. The sun felled from the front, respectively, behind of the vehicle. The 3rd and 4th scenarios were taken in a cloudy day, on a road with 2 lanes per ways and a medium-hard traffic.

The recall consists in the ratio between good fitted frames and the total number of frames. The precision was considered to be the sum of all good frames fitting metric divided by the total number of good frames.

Detecting lanes in inner roads is the most challenging for the proposed apparatus due to the high curves and the incoming traffic from the opposite way. Statistics obtained during such scenarios can be seen in Table 2.

Table 2

Inner roads detection precision

Scene	Frames	Recall	Precision
Road 1	1089	0.8735	0.8931
Road 2	5671	0.8211	0.8899
Road 3	9113	0.7161	0.8713

5. Conclusions and Future Work

The paper presented a robust lane detection algorithm that can be used in different driver assistance and autonomous driving scenarios.

As future work, the authors plan to enhance the approach by using a Bayesian inference method to selecting and validating hypotheses during lane detection phase, together with an improved method for extracting lane marking measurements. Additional to his, a further task will be to improve the computation time through parallel computational devices such as GPUs, or FPGAs.

References

1. Aly, M.: *Real time Detection of Lane Markers in Urban Streets.* In: Int. IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, 2008, p. 7-12.
2. Dickmanns, E.D., Mysliwetz, B.: *Recursive 3-D Road and Relative Ego-State Recognition.* In: IEEE Trans. on Pattern Analysis and Machine Intelligence **14** (1992) No. 2, p. 199-213.
3. Hillel, A.B., Lerner, R., Levi, D., Raz, G.: *Recent Progress in Road and Lane Detection: A Survey.* In: Machine Vision and Applications (2012), p. 1-19.
4. Marita, T., Oniga, F., Nedevschi, S., Graf, T., Schmidt, R.: *Camera Calibration Method for Far Range Stereovision Sensors Used in Vehicles*. In: Int. IEEE Intelligent Vehicles Symposium, Tokyo, Japan, June 2006, p. 356-363.
5. Mathias, M., Timofte, R., Benenson, R., Van Gool, L.: *Traffic Sign Recognition - How Far Are We From the Solution?* In: IEEE International Joint Conference on Neural Networks, Dallas, USA, 2013, p. 1-8.
6. McCall, J.C., Trivedi, M.M.: *Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation.* In: IEEE Transactions on Intelligent Transportation Systems **7** (2006) No. 1, p. 20-37.