# Autonomous Vehicle And Real Time Road Lanes Detection And Tracking

Farid Bounini, Denis Gingras
LIV – Université de Sherbrooke
Sherbrooke, Canada
*farid.bounini@usherbrooke.ca*
*denis.gingras@usherbrooke.ca*

Vincent Lapointe, Herve Pollart
Opal-RT Technologies Inc,
Montreal, Canada
*vincent.lapointe@opal-rt.com*
*herve.pollart@opal-rt.com*

***Abstract*— Advanced Driving Assistant Systems, intelligent and autonomous vehicles are promising solutions to enhance road safety, traffic issues and passengers' comfort. Such applications require advanced computer vision algorithms that demand powerful computers with high-speed processing capabilities. Keeping intelligent vehicles on the road until its destination, in some cases, remains a great challenge, particularly when driving at high speeds. The first principle task is robust navigation, which is often based on system vision to acquire RGB images of the road for more advanced processing. The second task is the vehicle's dynamic controller according to its position, speed and direction. This paper presents an accurate and efficient road boundaries and painted lines' detection algorithm for intelligent and autonomous vehicle. It combines Hough Transform to initialize the algorithm at each time needed, and Canny edges' detector, least-square method and Kalman filter to minimize the adaptive region of interest, predict the future road boundaries' location and lines parameters. The scenarios are simulated on the Pro-SiVIC simulator provided by Civitec, which is a realistic simulator of vehicles' dynamics, road infrastructures, and sensors behaviors, and OPAL-RT product dedicated for real time processing and parallel computing.***

***Keywords*— Intelligent Transportation System; ADAS; image processing; Road lane's detection; line tracking; Hough transform; least-square; Kalman filter; fuzzy logic.***

## I. INTRODUCTION

One of the enabling technologies that deeply contribute to an Advanced Driving Assistance System (ADAS) proliferation is the low-cost embedded MEMS ( Micro-Electro-Mechanical Systems) sensors, which are becoming cheaper compared with last decades. Affordable memory and important computing resources contribute moreover to the advances in ADAS and intelligent vehicles. These systems allow to enhance road safety and resolve some of the traffic issues. One of the main challenges for self-driving vehicles is related to navigation issues in uncertain or a non-static environment. Nevertheless, artificial intelligence and computer vision offer potential solutions for autonomous vehicles' navigation in an unstructured environment, scene analysis, classification, etc. One of the solutions is a system vision that could be based, either on one monocular camera with morphological image processing operator **[1-3]**, fusing road geometry and B-Snakes **[4]**, or several cameras for advanced processing like inter-objects distance estimation and 3D objects reconstruction **[5,6]**. However, road lane detection is still a difficult task for intelligent vehicles. This is essentially due to the huge amount of data that should be processed in real time. The algorithms should recover the road state and uncertainties issues, shadows, vehicles' vibration, sensors' noise, etc. To overcome such constraints; new methods should be fused with the traditional image processing algorithms to improve them and to better target the region of interest and to minimize the computational cost for the real-time applications.

This paper presents a simulation architecture for autonomous vehicles and real-time road lane detection and tracking. The paper involves five sections: the first part is dedicated to related work on road boundaries and painted lines detection. In the second section, a description of the main scenario that was built on Pro-SiVIC **[11, 12]**— a simulator of infrastructure, vehicles' dynamic, and embedded sensors.— is presented. The third part deals essentially with road lanes detection algorithms by applying Canny edge detector and Hough Transform to initialize the proposed method that stands on the adaptive region of interest, least-square method and Kalman filter to predict the next position of the road boundaries and lines' parameters. The fourth section describes the control strategy based on road lane center and fuzzy logic. This is followed by fifth section, which provides some simulation results on road lanes tracking, and then the paper is concluded.

## II. SIMULATION SCENARIO

### *1. Simulators used*

The scenarios are mainly implemented on two different platforms in order to simulate the road lanes detection system. The first platform is called Pro-SiVIC, which is a real-time simulator of road, environment, vehicle dynamic, and embedded sensors (Inertial Navigation System INS, odometers, Light Detection And Ranging LiDAR, camera, etc.) and provided by Civitec. It allows to develop multi-vehicles scenarios for real-time simulations. The second platform is OPAL-RT product, which is Real Time Laboratory (RTLAB) to separate, compile, and prepare the mathematical model — initially built on Simulink/Matlab, C/C++, etc.— for parallel computing. RTLAB and Pro-SiVIC combination forms a good platform for ADAS, intelligent/autonomous and collaborative vehicles simulation and validation. The reader can find more details on those platforms in **[7]** and **[8]**.

The main scenario consists of generating a virtual environment, set here to the standard Pro-SiVIC trajectory called "Horse Ring". The circuit includes road, traffic barriers, landscape, etc. Two cars are used: the black one is equipped with an INS attached to the car chassis to measure the vehicle's acceleration and orientation, and a camera aligned with the car on its top directed towards the road's scene, and to take pictures each 40ms. To detect the road lanes, the RGB (Red, Green, Blue) images are converted to gray one. The Canny edge detector is applied to the grey frame to perform "Hough Transform" on the edged images in order to detect the lines which match road boundaries and painted lines. This initializes the proposed algorithm that consists of selecting the right region of interest — a rectangle with a center that corresponds to the mean of each road boundaries, width W, and height H.— that fits with the last measurement and their KF parameters estimation.

### 2. Select a region of interest corresponding to road part of the image

Onboard camera, usually fixed at the front of a car and protected behind the windshield, takes N frames per second. The frame itself contains an amount of data, which includes information about the road boundaries and the white lines. Due to this reason, a region of interest (ROI), that holds road's major information, is carefully selected to mitigate the computational cost like shown in figure 1 (green rectangle). Another solution is to minimize the size of the ROI corresponding to each road boundary and line. This could be done after determining the road lines' position, then a ROI is defined and associated to each line. This procedure allows the algorithm to better target the lines and minimize the image-processing time, more details are given in the next sections.

The next step, after choosing the ROI, is RGB image conversion to a grayscale. The main objective is to generate an image with one layer rather than three like RGB one does, and without compromising the original road lines information. The concept saves more computational power for further processing data like searching and tracking the adequate road lanes in the grayscale frame.

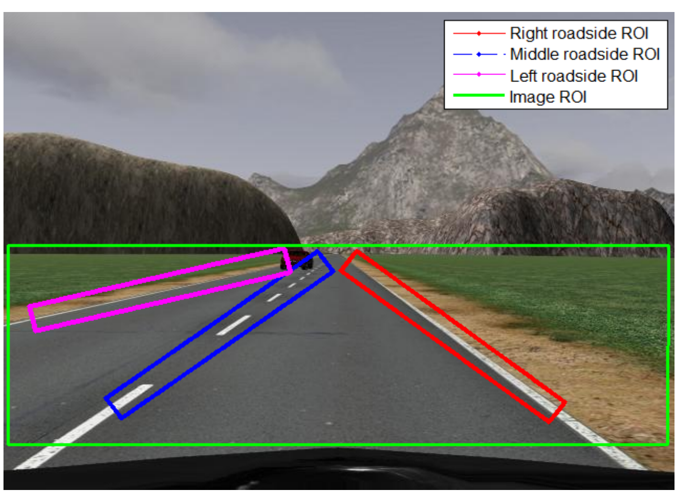


Figure 1: Region of interest section

## III. Road Lanes Detection

### 1. Edges detection

Road boundaries detection is the most difficult and important task that an intelligent vehicle should perform. They are often defined by the sharp contrast between the road bitumen and any non-pavement surface or painted lines **[1]**. The pixels, with high gradient values, are the image edges, where some of them correspond to road boundaries coordinates. To determine those edges, there are several edge detectors that could work like Sobel, Prewitt, zero cross, Roberts, Laplacian of Gaussian, and Canny. Nevertheless, Canny edge detector is still the one which provides the best edged frames **[1]**, and it is globally described by two steps **[2]**:

a— Smoothing the ROI by convolving a Gaussian mask with the image.

b— Getting the region of high contrast with a 2D first derivative operator.

In this work, a Canny edges' detector is used to determine frames' edges, and essentially the road boundaries and painted lines as shown in figure 2. In order to keep the autonomous vehicle on the road, the frame's edges should be classified to hyperboles **[1]** or to straight lines **[9]** according to the relative vehicle's position and direction — to predict the future road lanes parameters— and to the road geometry.

### 2. Edges and painted lines color

The color detection method is an additional but an unnecessary step. It could be used to check the road lanes' edges and distinct them from the image's background. It makes some classification, like determining either road lanes edges are surrounded by bitumen or not, which is often dark, gray or black. The main problem is the threshold determination that should be adaptive to handle the different RGB image contrast and brightness. One solution is to construct, for each frame, a histogram in order to calibrate the threshold, which allows to look for any white or yellow marks on the road.

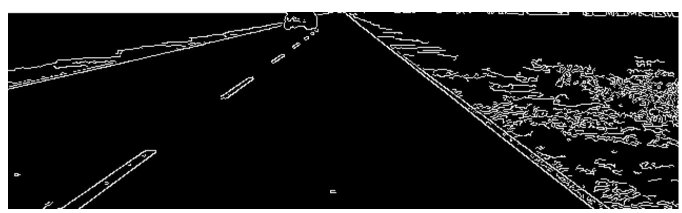

Figure 2: Canny edges detector results

### 3. Hough transform and line detection

High-speed process and computational resources improvement permit an important advance in computer vision, allowing several complex operations on each image for further advanced applications like building a 3D model from 2Ds frames, inter-objects distance estimation, etc. That is conditioned by searching and classification of multiple lines in an edged image. So to overcome the particular time-consuming issues, one should appeal powerful and efficient algorithms that detect different lines in an edged frame. One of the most effective methods is Hough transform (HT) **[9]**, which is nominated to its inventor Paul Hough that patented it in 1962 **[10]**. The main advantage of HT, inasmuch as HT is invariant to pixels' position in a frame, is its robustness to noise and occlusions, that means the algorithm is still valid to estimate noisy and dashed edges **[11]**. It transforms a set of frame pixels, points, in the Cartesian space to another space known as Hough space over some parameter space **[11]**.

Some limitations could be taken regarding the camera viewport, road geometry, or modify HT as proposed in **[1]** by discarding any horizontal line, restricting lines' slope in the left and the right side at $\pm\alpha_{\max}$. Another option is to partition the ROI to $M$ sections: each section is tied to road boundary or a painted line.

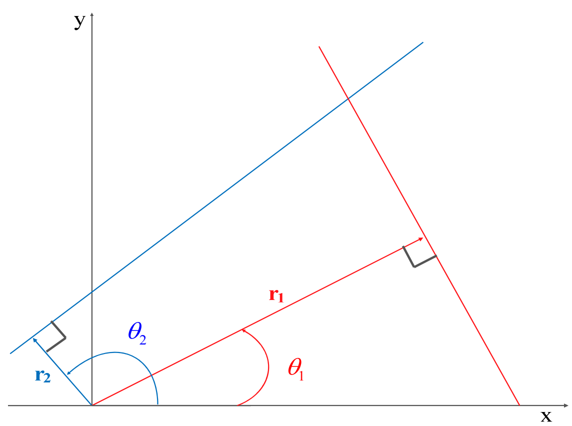


Figure 3: Lines representation in Hough space

$$r_i = x\cos\theta_i + y\sin\theta_i \quad (1)$$

Where: $r_i \in \left]0, \sqrt{rows^2 + cols^2}\right[$ and $\theta_i \in \left]0, \pi\right[$

## IV. Adaptive Region of Interest and Polynomial Approximation

Since Hough-Transform is computationally expensive, which makes it not feasible for real-time simulation or embedded implementations. That is due to its line clustering in the high-resolution images with an accurate numerical processing **[9]**. To meet those requirements and overcome the HT gaps, vibrations, and high-speed constraints, first we propose to call HT to detect and reconstruct road lanes' boundaries. This initializes our method, which is stood on adaptive ROI dedicated to each road boundary and painted lines as shown in figure 1 (red, blue and magenta rectangles). The second step is to find a mathematical model, polynomial model (2), that fits the road boundary edges.

$$y = p_n x^n + p_{n-1} x^{n-1} + \cdots + p_0 \quad (2)$$

This model consist to look for a polynomial of the $n^{th}$ order that fits better the road lines counter, which is represented by a linear system of form $X * P = Y$.

*X, Y* and *P* are x counters' coordinate vector, y counters' coordinate vector, and polynomial parameters vector respectively.

$$\begin{pmatrix} x^n & x^{n-1} & \cdots & 1 \end{pmatrix}\begin{pmatrix} p_n & p_{n-1} & \cdots & p_1 \end{pmatrix}^T = y \quad (3)$$

In order to find $P$ values, we need at least a set of $n$ equations formed by each edges pixel, which gives:

$$\begin{pmatrix} x_1^n & x_1^{n-1} & \ldots & 1 \\ x_2^n & x_2^{n-1} & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ x_n^n & x_n^{n-1} & \ldots & 1 \end{pmatrix}\begin{pmatrix} p_n \\ p_{n-1} \\ \vdots \\ p_1 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (4)$$

The new X matrix is called a Vandermonde matrix.

### 1. *Least-square method.*

The least-square (LS) is a method for fitting a set of measured data with a mathematical model of the type $y = f(p, x)$. Where $\{(y_i, x_i), i = 1, \cdots, n\}$ is a set of measured points that could be represented in the Cartesian space, $p$ is a vector of parameters which minimizes the sum of the squares of the difference between the observed data and the fitted mathematical model as expressed by (5); often called *residual* or *measurement error*. In order to solve and determine the vector of parameters that fits better the mathematical model, one should take care about the main constraint on the number of equations — which matches better the measured set of points— has to be greater than the number of unknown variables.

$$J = \min\sum_{i=1}^{n}\varepsilon_i^2 = \min\sum_{i=1}^{n}\left(y_i - f\left(p, x_i\right)\right)^2 \quad (5)$$

In case of ordinary least-square and linear road lines representation, the polynomial model order is set to $n = 1$ and (2) becomes $y = p_1 x + p_0 = ax + b$. This model fits more our work in terms of real-time road lanes detection: lines parameters determination. The problem is reduced to estimate two parameters $\hat{a}$ and $\hat{b}$ corresponding to lines' slope and their intersection with Y-coordinate respectively.

$$\hat{a} = \frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)}{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)} \text{ and } \hat{b} = \overline{y} - \hat{a}\overline{x}$$

Where $(\overline{x}, \overline{y}) = \left( \frac{1}{n} \sum_{i=1}^{n} x_i, \frac{1}{n} \sum_{i=1}^{n} y_i \right)$ is the center of gravity of the measured set of points.

LS method could be adapted to estimate more complex models for hyperbole shapes, that usually used to model sharp corners.

### 2. *Kalman filter road lanes prediction and tracking*

Road irregularity causes vehicles' vibrations, and when these vibrations are associated to vehicles' high speed, they shift and blur the road boundaries on the acquired images. This creates a blurred array that compromises the edge detection methods based on the image textures. To overcome such constraints, a Kalman filter is used to filter the measured road lanes' edges and predict the future lines' parameters and the regions of interest position, on each frame, corresponding to each line on the road $\hat{x}_k^{i\,T} = \left( \hat{a}_k^i \quad \hat{b}_k^i \quad \hat{\overline{x}}_k^i \quad \hat{\overline{y}}_k^i \right)^T$.

This reduces the ROI size and the number of image preprocessing operations. Where $\hat{a}_k^i$ and $\hat{b}_k^i$ are the i$^{th}$ line parameters, $\hat{\overline{x}}_k^i$ and $\hat{\overline{y}}_k^i$ center of the region of interest corresponding to i$^{th}$ line (a rectangle: defined by its center $(\hat{\overline{x}}_k^i, \hat{\overline{y}}_k^i)$, width *W*, height *H*, and a declination $\alpha_k^i$ which coincides to the i$^{th}$ line declination as shown in figure 1) at step $k$. The linear time invariant (LTI) model is given by (6).

$$\begin{cases} x_{k+1}^i = F^i x_k^i + w_k^i \\ y_k^i = H^i x_k^i + v_k^i \end{cases} \tag{6}$$

$F^i$ and $H^i$ are transition/state and measurement matrices respectively. $w_k^i$ and $v_k^i$ are independent Gaussian noises corresponding to unmodeled errors and exogenous perturbations, and measurement noise respectively.

$w_k^i \approx (0, Q_k^i)$ , $v_k^i \approx (0, R_k^i)$ : $E[w_k^i w_k^{i\,T}] = Q_k^i > 0$ , $E[v_k^i v_k^{i\,T}] = R_k^i > 0$ , $E\left[v_k^i w_k^{i\,T}\right] = E\left[w_k^i v_k^{i\,T}\right] = 0$ and $P_k^i = E\left[\left(x_k^i - \hat{x}_k^i\right)\left(x_k^i - \hat{x}_k^i\right)^T\right]$ are covariance matrixes at step $k$. The Kalman filter algorithm stands on two steps:

1— prediction step: to estimate the future state vector according to the mathematical model and the last update step results.

$$\begin{cases} \hat{x}_{k+1|k}^i = F^i \hat{x}_{k|k}^i \\ P_{k+1|k}^i = F^i P_{k|k} F^{iT} + Q_k^i \\ P_{k+1|k}^i = E\left[\left(x_k^i - \hat{x}_{k+1|k}^i\right)\left(x_k^i - \hat{x}_{k+1|k}^i\right)^T\right] \end{cases} \tag{7}$$

2— update step: allows to take in account the available measurements and to update the observation gain and the covariance matrix.

$$\begin{cases} \hat{x}_{k|k}^i = \hat{x}_{k|k-1}^i + K_k^i (z_k^i - H^i \hat{x}_{k|k-1}^i) \\ K_k^i = P_{k|k-1}^i H^{iT} (H^i P_{k|k-1}^i H^{iT} + R_k^i)^{-1} \\ P_{k|k}^i = P_{k|k-1}^i - K_k^i H^i P_{k|k-1}^i \\ P_{k|k}^i = E\left[\left(x_k^i - \hat{x}_{k|k}^i\right)\left(x_k^i - \hat{x}_{k|k}^i\right)^T\right] \end{cases} \tag{8}$$

$z_k^i$ : the vector of the output measurements.

The following diagram summarizes the main steps of the proposed method to meet the real time applications.

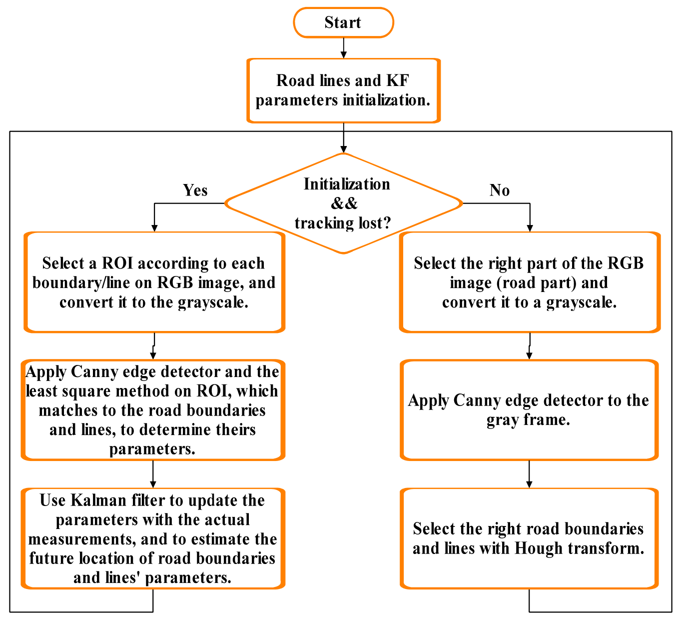


Figure 4: Road lanes detection and tracking diagram

## V. Preliminary Results

As described in section II, the road lanes' detection and tracking scenarios are designed on Pro-SiVIC. The environment and the test track are the "*Horse Ring*". Two vehicles are built on the test track: the red one is equipped with a camera, INS, and programmed to follow the center of the right road lane at 70 km/h in the opposite direction of the black vehicle, which is also equipped by a vision system as the red vehicle does, which acquires 25 RGB frames per second of the size $480 \times 640$. Due to the vehicles' dynamic, some restrictions are made on the maximum steering angle which is set to $\theta_{\max} = 60^\circ$ degrees in each direction (left and right), and the maximum of acceleration is set to $a_{\max} = 10 ms^{-2}$.

To control the vehicle, we use three simple fuzzy logic laws that depend upon the vehicle's position versus the right road

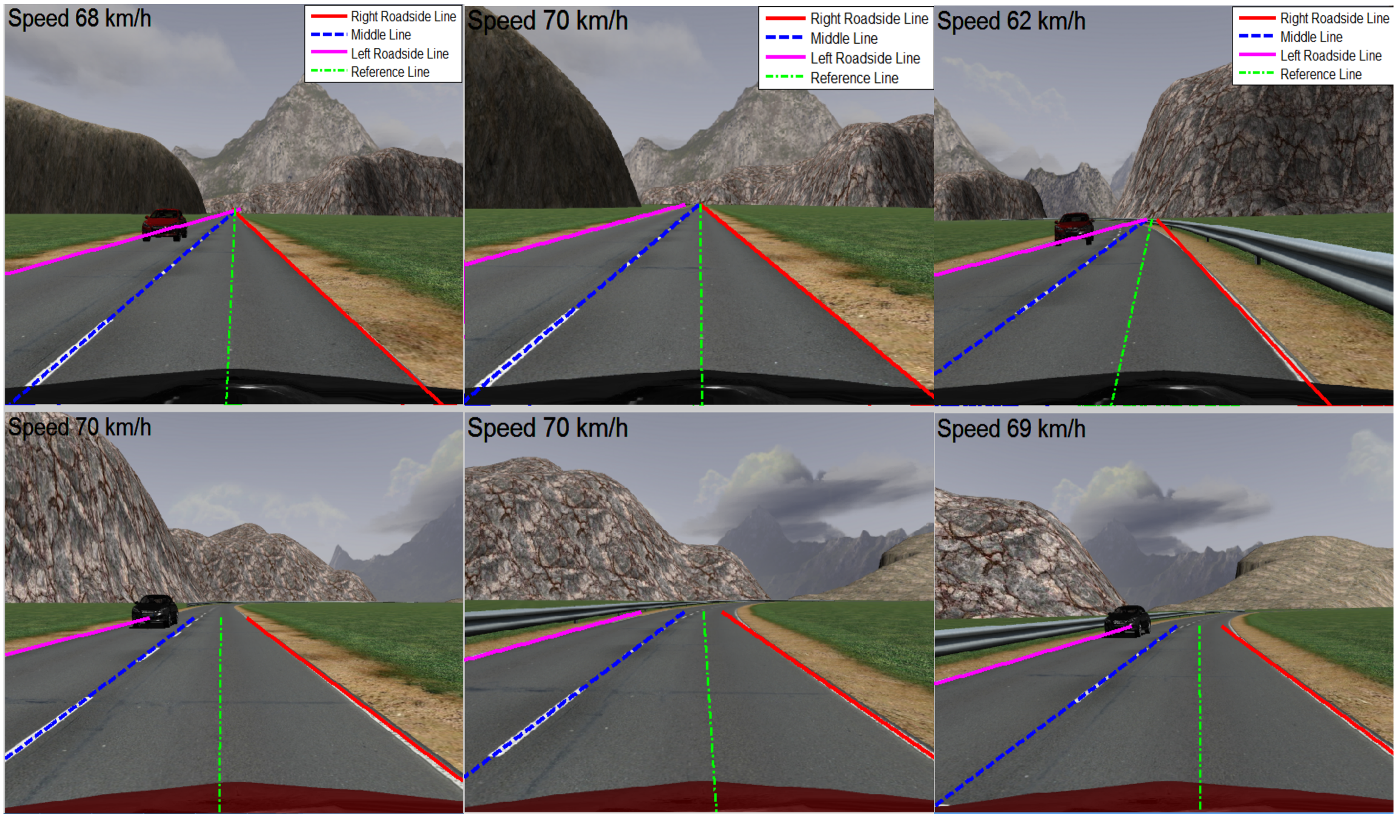


Figure 5: Real time road line detection and road lanes tracking, the algorithms tested with two cars in straight lanes and in corners.

lane's center. Once the road lanes are detected with the proposed method described just above (figure 4), then the center of the right road lane is estimated, represented by the green line in figure 5. According to vehicle's position across the road, the fuzzy controller handles the vehicle's steering by generating and applying a steering reference angle to vehicle's steering direction to keep and track the estimated road lane' center. The same laws are carried out by the speed controller that compares the actual speed — thanks to the vehicles' kinematic model and the embedded inertial navigation system (INS) **[7]** — with the speed reference. Given the speed tracking error and the exogenous perturbations, the speed controller applies an acceleration command signal over the vehicle's wheels to control and set the vehicle's speed to the reference signal.

The real-time simulation results confirm the efficiency and the robustness of the proposed method for real-time road line detection even at high speed and with discontinuous and dashed road lines. The black car runs along the test track where the right line is getting thinner and thinner until it gets completely vanished. Although the roadside line vanishes, the vehicle maintained and tracked the road lane center. The embedded algorithm successfully overcame the scenarios constraints, dissolving lines, by distinguishing the different road textures. The steering fuzzy controller is limited in corners to 70 km/h: If the speed is higher, the autonomous vehicle is no longer able to keep driving over the road, particularly in sharp turns. This is due, mainly, to the approximated vehicles' kinematic model errors, exogenous and unmodeled perturbations. To overcome such constraints, one should focus on the vehicle's mathematical model and lateral vehicle's controllers.

## VI. CONCLUTION

This work deals with a road boundaries and painted lines detection for intelligent and autonomous vehicles. The purpose of the paper is to overcome the different constraints that could overwhelm the real-time road lanes detection and tracking. Those constraints are essentially due to the huge amount of data that should be processed in case of high-resolution images, vehicle's speed, vibrations, etc. The preliminary results show that the proposed method is not only robust against exogenous perturbations and different constraints, but good enough to control the vehicle with a simple couple of fuzzy logic laws. The vehicle's controller limitations arise for a maximum speed of 70 km/h in sharp turns. Such issues will be overcome in the future work by proposing a new accurate vehicle's dynamic model controller, and fusing road lanes' parameters with the vehicle's dynamic to better predict the road boundaries, in particular cases: completely obscured boundaries and lines by other vehicles, road intersections/junction at grade.

## VII. Acknowledgement

This work is part of COOPERCOM, a 3-year international research project (Canada-France). The authors would like to thank the National Science and Engineering Research Council (NSERC) of Canada and the Agence nationale de la recherche (ANR) in France for supporting the project.


## VIII. References


[1] Assidiq, A.A.M. ; Khalifa, O.O. ; Islam, R. ; Khan, S. "*Real time lane detection for autonomous vehicles*", International Conference on Computer and Communication Engineering, 2008. ICCCE 2008. Kuala Lumpur, 13-15 May 2008.

[2] Kristijan Ma˘cek, Brian Williams, Sascha Kolski," *A lane Detection vision module for driver assistance*", EPFL, 1015 Lausanne, 2002, pp. 1-6.

[3] Xiaodong Miao, Shunming Li, Huan Shen, " *On-Board Lane Detection System For Intelligent Vehicle Based On Monocular Vision* ", international journal on smart sensing and intelligent systems, vol. 5, no. 4, december 2012

[4] Teoh, E.K. ; Dinggang Shen. " *Lane detection using B-snake*", International Conference on Information Intelligence and Systems, 1999. Proceedings. 1999.

[5] Nunez, P.; Drews Jr, P.; Rocha, R. and Dias, J. "*Data fusion calibration for a 3d laser range finder and a camera using inertial data*" in Proc. of 4th European Conf. on Mobile Robots (ECMR'2009), Sep. 2009, pp. 31–36.

[6] Aliakbarpour, H.; Nez, P.; Prado, J.; Khoshhal, K. and Dias, J. "*An efficient algorithm for extrinsic calibration between a 3d laser range finder and a stereo camera for surveillance.*" in 14th International Conference on Advanced Robotics (ICAR 2009), 2009.

[7] Bounini, F. ; Gingras, D. ; Lapointe, V. ; Gruyer, D. "*Real-time simulator of collaborative autonomous vehicles*", International Conference on Advances in Computing, Communications and Informatics ICACCI, Greater Noida, INDIA, 2014 .

[8] Bounini, F. ; Gingras, D. ; Lapointe, V. ; Gruyer, D. "*Poster: Real-time simulator of collaborative autonomous vehicles*", International Conference on Advances in Computing, Communications and Informatics ICACCI, 2014, Greater Noida, INDIA, 2014.

[9] Cappelletti, M. ; Macii, D. "*A RANSAC-based fast road lanes detection algorithm for high-speed wheeled vehicles*", Instrumentation and Measurement Technology Conference (I2MTC, Binjiang, China), 10-12 May 2011.

[10] Ghassan H. Karin A. and Rafeef G. "*Project Report for the Computer Vision Course Automatic Line Detection* " Image Analysis Group Department of Signals and Systems Chalmers University of Technology September 1999.

[11] John, B. M. and Donald, N. "*Application of the Hough Transform to Lane detection and Following on High Speed Roads*", signal &system Group, Department of Computer Science, National University of Ireland.

[12] web site http://www.civitec.com/

[13] web site http://www.livic.ifsttar.fr/.