Geometrical model and image processing for lane line detection in motorways

C Niño-Rondón[1], S Castro-Casadiego[2], B Medina-Delgado [3]

[1,2,3] Universidad Francisco de Paula Santander, San José de Cúcuta, Colombia

E-mail: carlosvicentenr@ufps.edu.co

Abstract. Lane line detection is one of the most important processes in the driving process. The regions of interest refer to specific sections of the image to be filtered or transformed for information extraction. In this paper, lane line detection is performed by images processing to defined sections within a region of interest based on a triangular geometric model, where the vertices are denoted in relation to the height and width of the captured image. The image processing consists of grayscale conversion, smoothing and closing morphology filtering, edge detection and thresholding. An average time of 0.75 seconds is obtained from image acquisition to processing with lane line detection. The results obtained in both the processing and the response time measurement allow to infer the adaptability of the method for real time autonomous driving applications.

1. Introduction

The automotive industry represents one of the fastest growing sectors in the economy globally[1], especially in Latin America and the Caribbean [2]. Due to its demand, and seeking to provide solutions to the needs of its customers, several companies in the sector tend to use automation technologies that facilitate and improve the driving experience of vehicles [3]. One of the main challenges for this type of technology lies in the system's ability to adapt and offer safety in environments such as motorways, where due to the high speeds at which vehicles move and added to driver distractions, traffic accidents often occur that result in the vehicle leaving the road [4].

The lane line detection is one of the most relevant applications in computer vision processes, in which by means of image processing with techniques of recovery and transformation of the same [5] characteristics are obtained that accompanied by arrangements and geometric models allow to extract the straight and curved lines corresponding to the rail lines [6].

This document shows the lane lines detection by image processing and a region of interest by means of a triangular geometric model, to video images taken with a 16MP video capture device, located inside a circulating vehicle on the international motorway that connects to San José de Cúcuta, Colombia, with San Antonio del Táchira, Venezuela. Image processing is done in Python programming language using a Raspberry Pi 3B+ embedded board.

2. Methodology

The proposed methodology consists of 3 stages and is presented in Figure 1. The first stage concerns the video capture device location. The second stage deals with the definition of the geometric model for the search of the regions of interest (ROI) in the captured image. The third stage consists of the application of image processing techniques for the lane lines detection.

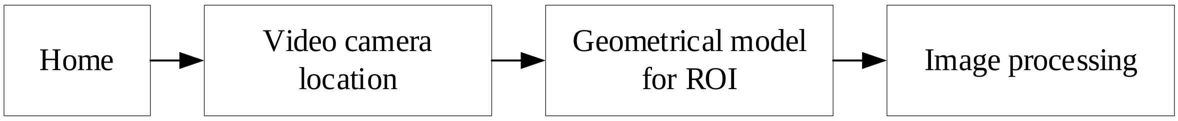

Figure 1. Methodology applied.

2.1 Video camera location

In computer vision processes, efficiency is largely dependent on both the characteristics of the video capture device and its location for image and video capture [7][8]. Therefore, it is proposed to place the video camera inside a car in order to mitigate image and environmental noise factors, potentially present at the camera location outside the vehicle [9]. Factors such as the reflection of the front glass are taken into account, as well as the line of sight of the device in order to have the greatest amount of visible field available in the subsequent search for the region of interest. Once the capture device is located, the image is acquired and the process of defining the region of interest begins.

2.2 Geometrical model for ROI

On motorways, vehicles move at high speeds and tend to move between two parallel lines referring to the lane lines [10], and at a point relatively distant from the driver, these parallel lines tend to cross by an optical effect [11]. Therefore, a geometrical model is proposed for the search of the region of interest [12], so that for the image processing only the image regions located in this region are involved and this melting point between the lane lines along which the vehicle runs is taken into account [13].

The proposed geometrical model consists of a triangular region, in which the vertices are defined as follows: vertex A has its abscissa at the midpoint of the width of the line of sight, while the ordinate is in the first third of the height of the image; vertex B has its abscissa at the midpoint of the width and its ordinate at the maximum value of the height; vertex C has its abscissa at the maximum value of the width and its ordinate at the maximum value of the height. For coding purposes and based on the toolkit available in OpenCV, the value of the height increases from top to bottom and the value of the width increases from left to right.

2.3 Image processing

As for the image processing, the original image is converted into grayscale, then a Gaussian blur filtering is performed and by morphological operation, the edges are distinguished and it is finished with a thresholding stage.

Grayscale conversion is done in the manner suggested by the NTSC, as shown in equation 1, where r refers to the colour red, g to the colour green, and b to the shade blue [14]. Once the image is converted to grayscale, the region of interest proposed by the triangular geometric model, described in the previous section, is applied to it.

$$y = (r * 0.3) + (g * 0.59) + (B * 0.11) \qquad (1)$$

On the other hand, gaussian blur filtering is done to remove noise from the image, generating a slight blur effect by combining the neighboring pixels in the image [15]. The filter uses a gaussian function, which requires a kernel for blurring. In this case, a 3x3 kernel is used, which is then averaged and shown in equation 2.

$$k = \frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad (2)$$

A morphological closure filter is also applied, combining dilation and erosion operations [16], as shown in equation 3.

$$A \bullet B = (A \oplus B) \ominus B \qquad (3)$$

Gaussian blur filtering coupled with morphological closure filtering facilitates the Canny algorithm's edge distinction process, as the first stage of this process is precisely smoothing or blur filtering [17][18]. Subsequently, the magnitude and direction of the gradient is found taking into account the derivative of the X and Y components. Similarly, because there is Non-maximum Suppression, a sweep is made between the pixels, and pixels that are not at the top of the ridge are assigned a value of 0 [19].

Finally, a thresholding process is carried out with a value of T=100, to the image pixels that are found as borders within the region of interest. In this way, only the lane lines are identified.

Once the processing of the image with the region of interest obtained by the proposed triangular geometric model is completed, the validation of the method is performed by quantifying the response time of the processing [20], since in the processes of lane line detection real time decision making is required.

3. Results

Following the proposed methodology, the results obtained in the stages required to obtain lane lines by a computer vision process using a triangular geometric region are presented below.

Figure 2 shows on the left side the X, Y, Z plane representation of the line of sight of the video capture device, taking as a reference the visible range of the camera, while the right side shows its location in the vehicle. The camera is located at half the width of the vehicle corresponding to 0.87 meters, and at an approximate height of 1.1 meters.

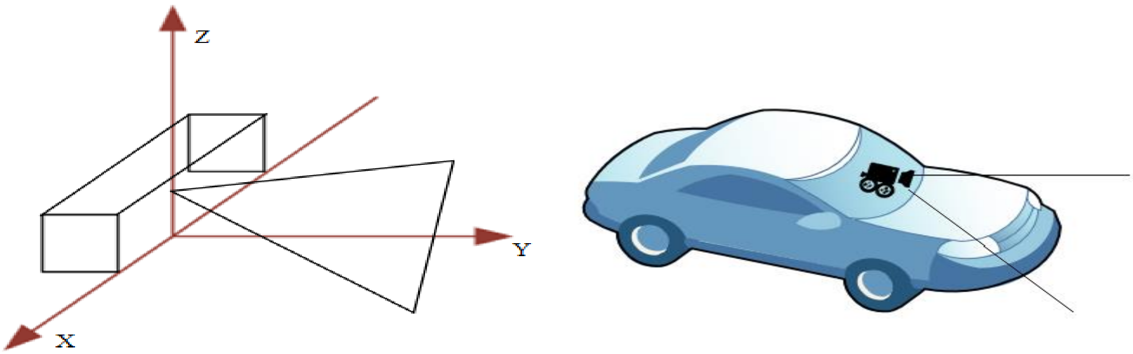

Figure 2. Representation of video camera location

Likewise shown is the shape of the triangular geometric model for the region of interest based on the size of the width and height of the image. This ensures that only regions that theoretically represent the driver's line of sight on a motorway or intercity road are included in the processing, thus reducing the machine resources required.

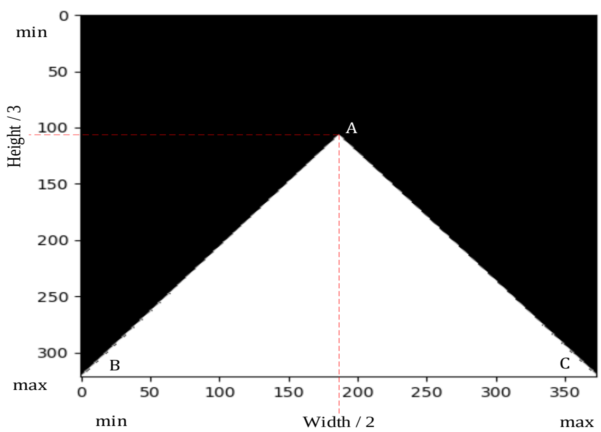

Figure 3. Proposed triangular geometric model for the definition of the region of interest

Similarly, the processing to the captured video images is presented in Figure 4. From left to right, the original image captured by the camera is presented, followed by the grayscale conversion. Likewise, the triangular geometric model is shows for the definition of the region of interest, which is later filtered by Gaussian blur and closure morphology. Finally, the detection of edges within the region of interest and the distinction of the lane lines after the thresholding process is shown. The processing yields images with fully identified lane lines, where the other objects present in the image frame and which could potentially become false positives are suppressed, thus inferring the reliability of the proposed method.

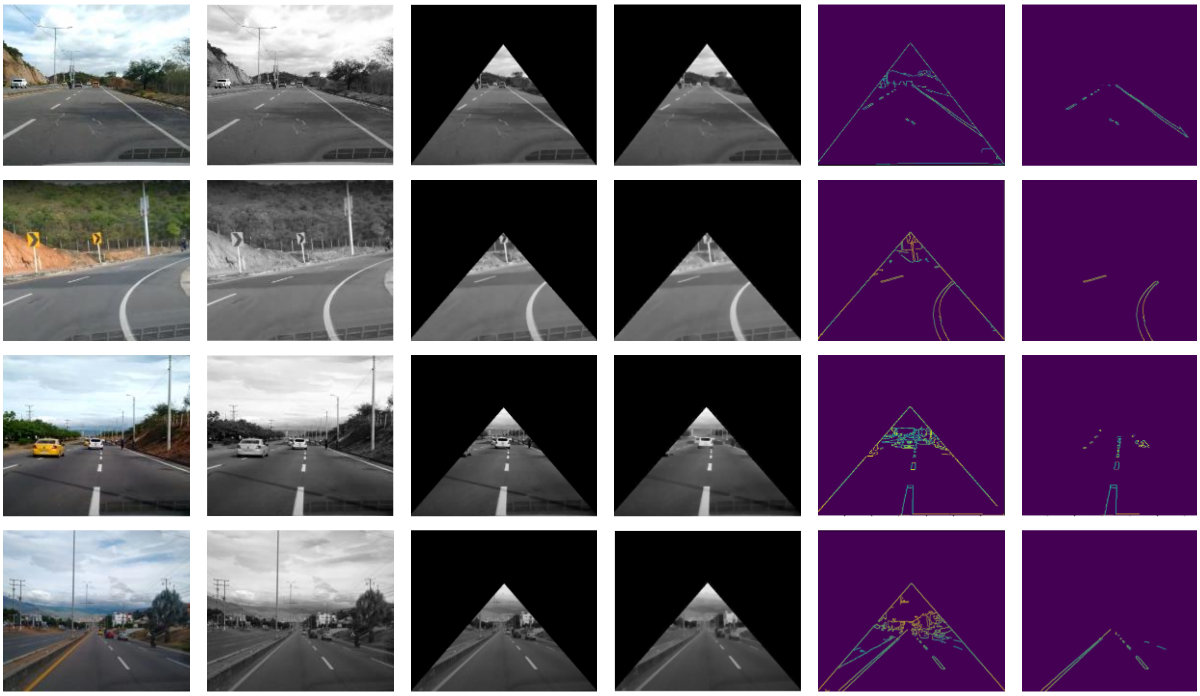

Figure 4. Image processing for lane line detection

Finally, the response time of the processing system is shown as a validation of the performance of the method used. The counted time is taken from the moment the input image is captured, until the detection of the lane lines is generated. The average response time of the proposed lane line detection system for 50 samples was 0.75 seconds. From this value and with the idea of providing real-time responses, it can be said that the system presents an optimal performance for the application.

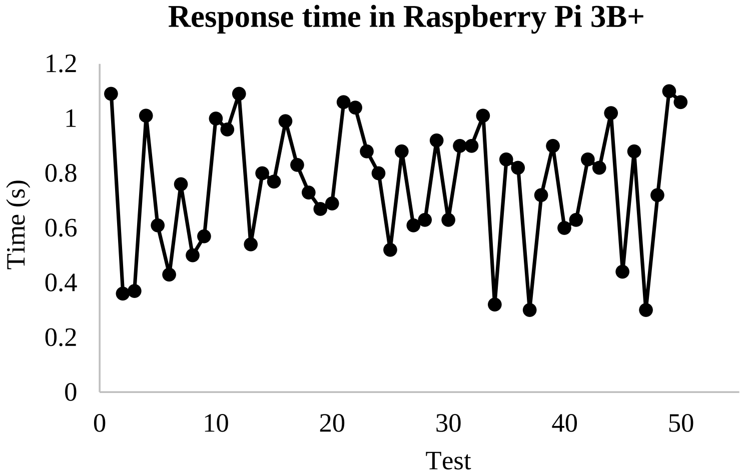

Figure 5. Response time of the proposed method in Raspberry Pi 3B+

4. Conclusions

The lane lines detection by means of a geometric triangular model as a method of definition of the region of interest is computationally efficient, because the distinction of most of the lane lines is achieved, besides that in processes of both human driving and autonomous driving, real time decision making is required, obtaining with the proposed method an average time of 0.75 seconds from the acquisition of the image to the processing for the distinction of lane lines. In the future, it is proposed to implement the method on embedded plates with greater processing capacity and memory than the Raspberry Pi 3B+, thus improving the response time in execution, as well as a stage of adaptability to possible discontinuous lane lines present in some sections of motorways or intercity roads.

References

[1] X. Zhang and F. Chen, "Lane Line Edge Detection Based on Improved Adaptive Canny Algorithm," *J. Phys. Conf. Ser.*, vol. 1549, no. 2, 2020.

[2] A. Pregonero Guzmán, "Evolución de la industria de vehículos surcoreana y desaceleración de la industria automotriz : a propósito del tlc entre Corea del Sur y Colombia .," *Divergencia*, vol. 2, no. 23, pp. 35–42, 2017.

[3] Y. Li, M. Díaz, S. Morantes, and Y. Dorati, "Vehículos autónomos : Innovación en la logística urbana Autonomous vehicles : Innovation in urban logistics," *Ric*, vol. 4, pp. 34–39, 2018.

[4] J. Yang, Z. Li, and L. Li, "Lane detection based on classification of lane geometrical model," *Int. Conf. Signal Process. Proceedings, ICSP*, vol. 2, pp. 842–846, 2012.

[5] H. Xu, L. Li, M. Fang, and L. Hu, "A method of real time and fast lane line detection," *Proc. - 8th Int. Conf. Instrum. Meas. Comput. Commun. Control. IMCCC 2018*, pp. 1665–1668, 2018.

[6] X. Du and K. K. Tan, "Comprehensive and Practical Vision System for Self-Driving Vehicle Lane-Level Localization," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2075–2088, 2016.

[7] C. V. Niño Rondón, S. A. Castro Casadiego, and B. Medina Delgado, "Caracterización para la ubicación en la captura de video aplicado a técnicas de visión artificial en la detección de personas," *Rev. Colomb. Tecnol. Av. RCTA*, vol. 2, no. 36, pp. 83–88, 2020.

[8] R. Barbuzza, J. D'amato, A. Rubiales, L. Dominguez, A. Perez, and M. Venere, "Un Método para la Sustracción de Fondo en Videos Inestables," *Mecánica Comput.*, vol. 34, no. 51, pp. 3409–3417, 2016.

[9] R. Baran, T. Rusc, and P. Fornalski, "A smart camera for the surveillance of vehicles in intelligent transportation systems," *Multimed. Tools Appl.*, vol. 75, no. 17, pp. 10471–10493, 2016.

[10] Z. Zhao, Q. Wang, and X. Li, "Deep Reinforcement Learning Based Lane Detection and Localization," *Neurocomputing*, 2020.

[11] R. Kanjee, A. K. Bachoo, and J. Carroll, "Vision-based adaptive cruise control using pattern matching," *Proc. - 2013 6th Robot. Mechatronics Conf. RobMech 2013*, no. October, pp. 93–98, 2013.

[12] X. J. An, J. Li, and E. K. Shang, "Region-of-interest generation for lane detection using high-level information," *Int. Conf. Wavelet Anal. Pattern Recognit.*, pp. 103–108, 2011.

[13] L. Dang, G. Tewolde, X. Zhang, and J. Kwon, "Reduced resolution lane detection algorithm," *2017 IEEE AFRICON Sci. Technol. Innov. Africa, AFRICON 2017*, pp. 1459–1464, 2017.

[14] G. Sánchez-Torres and J. A. Taborda-Giraldo, "Estimación automática de la medida de ocupación de playas mediante procesamiento de imágenes digitales," *TecnoLógicas*, vol. 17, no. 33, p. 21, 2014.

[15] A. Thakur, A. Kausar, and A. Iqbal, "Comparison efficacy of restoration method for space variant motion blurred images using kalman & wiener filter," *Proc. 2016 6th Int. Conf. - Cloud Syst. Big Data Eng. Conflu. 2016*, pp. 508–512, 2016.

[16] B. Ding, G. Wen, C. Ma, and X. Yang, "Target recognition in synthetic aperture radar images using binary morphological operations," *J. Appl. Remote Sens.*, vol. 10, no. 4, p. 046006, 2016.

[17] L. A. Akter and G. R. Kwon, "Integration of Contourlet Transform and Canny Edge Detector for

Brain Image Segmentation," *Int. Conf. Ubiquitous Futur. Networks, ICUFN*, vol. 2018-July, pp. 798–800, 2018.
[18] A. Amalia, D. Rachmawati, I. A. Lestari, and C. Mourisa, "Implementation of Canny and Isotropic Operator with Power Law Transformation to Identify Cervical Cancer," *J. Phys. Conf. Ser.*, vol. 979, no. 1, 2018.
[19] X. Geng, K. Chen, and X. Hu, "An improved Canny edge detection algorithm for color image," *IEEE Int. Conf. Ind. Informatics*, pp. 113–117, 2012.
[20] G. Cavanzo, M. Pérez, and F. Villavisan, "Medición de eficiencia de algoritmos de visión artificial implementados en raspberry pi y ordenador personal mediante Python," *Ingenium*, vol. 18, no. 35, pp. 105–119, 2017.