FPGA Based Real-Time Lane Detection and Tracking Implementation

I. El hajjouji, A. El mourabit, Z. Asrih, S. Mars and B. Bernoussi,
Laboratory of Information and Communication Technologies,
National School of Applied Sciences, UAE University
Tangier, Morocco
eismail89@gmail.com

***Abstract*— In this article, we present a system for robust lane detection and tracking. The proposed algorithm has two main parts. The first one is for lane detection using the Sobel operator with an adaptive threshold and the Hough transform, and the second part deals with lane tracking using the Kalman filter. To meet real time requirements, we use gradient directions of edge stage to simplify CORDIC algorithm. The obtained results show that the proposed algorithm can reliably detect and track lanes in different illumination condition.**

Keywords—FPGA, Sobel, Adaptive Threshold, CORDIC, Hough, Kalman Filter.

I. INTRODUCTION

Active driver assistance system (ADAS) is presented today as an efficient way to increase the road safety. Recent trend and developments in several technologies (micro-sensor, embedded electronics...) and computer vision plays an important role in the development of these systems to allow big ability to perform embedded intensive calculations with low power, more intelligence and low cost. Several examples in the literature [1] [2] report applications such lane departure prevention, detection and recognition of road signs, pedestrian detection, parking-aid, automatic cruise control, automatic switching on/off beams, collision risk warning, etc.

One main problem of the road scene is the visibility and the dependence to the illumination environment conditions: sunny, foggy, rainy, cloudy, etc. In such condition camera could give noisy frames which may generate false alarm to the driver or misinterpret a dangerous situation.

At a first approach, ADA system has to satisfy at least to the three following conditions:

1/ Real-time: it is the most important constraint, for an embedded functions, since ADAS are going to be integrated into more complex intelligent vehicles systems which consume a lot of computing resources.

2/ Results have to be accurate, especially for segmentation function which is the first processing bloc. Any false detection will affect and erroneous the final result. Segmentation have to be done independently of the illumination road scene [3].

3/ The algorithm has to be able to manage with different illumination scenarios such as: sunny, cloudy, rainy...

This work is supported by MESRSFC and CNRST, Morocco

In this article, we present a robust real-time system able to detect and track reliably lanes under different illumination conditions. The conditions cited before are implemented through a modified version of Sobel detector and by combining Hough transform (HT) with the Kalman filter (KF). The paper is organized as follows: Sections II presents the conventional SHT. The proposed method is described in section III. The Results are reported in the section IV.

II. STANDARD HOUGH TRANSFORM

Standard Hough transformation (SHT) is widely used in the literature as a way to find different sharps in the image, it is as a better choice for lane detection where parameters number is minimal. The advantage of Hough transform is that the pixel lying on one line need not all be contiguous. Therefore, it is very useful for detecting lines with short breaks in them due to noise or partially occluded by objects. Following steps are used for standard detecting lines with SHT:

The operation of the SHT is based on the use of a parametric space, called Hough space of two dimensions (ρ, θ). The SHT is implemented from the equation (1):

$$\rho = x.\cos(\theta) + y.\sin(\theta) \quad (1)$$

θ is the orientation of the normal vector, ρ is the distance from the origin.

The following steps are used to implement a detecting line with SHT:

1) Obtaining the binary image by using an edge detector.

2) Mapping of edge points to the Hough space and storage in an accumulator.

3) Interpretation of an accumulator to obtain infinite length lines. The interpretation is done by thresholding and possibly other constraints.

4) Conversion of infinite lines to finite lines.

Therefore, SHT is not only computation demanding but also memory-demanding. For a given point-fully sequential algorithm implementation has complexity $\theta(N^2 x \theta_{max})$ where θ_{max} is the number of quatizations in θ space and represents the accuracy desired for a NxN image. For example, if the minimal step of θ is 1°, there are 360 times of the

978-1-4673-8469-8/16/$31.00 ©2016 IEEE

multiplication operation and 180 times of the addition operation, just for one edge point.

In the next sections we present an adaptation of edge and HT operators. The overall goal is to reduce calculus and FPGA complexity.

III. Proposed Algorithm

Fig. 1 presents the adopted approach. Edge detection is adapted to meet the required constraint: Robust detection even if changing lighting conditions in the image with optimal FPGA resource. HT is implemented by CORDIC algorithm to meet hard constrains real time.

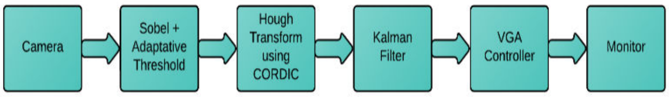


Fig. 1: Proposed Lane Detection and Tracking Chain

A. *Proposed Edge Detection*

To deal with the constraints reported above, the following algorithm was implemented:

1. Convolution with Sobel vertical and horizontal kernels [4]
2. Gradient calculation at each pixel G
3. Convolution to generate Mean gradient Gmean
4. Thresholding the difference image G - Gmean

$$\begin{cases} EM = 1 & if \quad G > G_{mean} \\ EM = 0 & else \end{cases} \tag{2}$$

Applying the some approach of dynamic threshold to Canny detector require implementation of complex task of fine tuning of two thresholds. The proposed Sobel detector with adaptive thresholding can be an alternative to Canny detector especially where the available resources are limited.

To enhance speed processing, horizontal and vertical gradients are calculated in parallel, and the total gradient is obtained as the sum of the two gradients absolute values. FPGA implementation of multiplication by 2 is replaced by a right shift operation. The absolute value operator is implemented using comparators to compare the two gradient directions and to decide the range of gradient phase (4). Gradient and phase range are inputs of the next HT stages. Phase range is used in the next stage to reduce the rotation number of HT. (see Fig. 2).

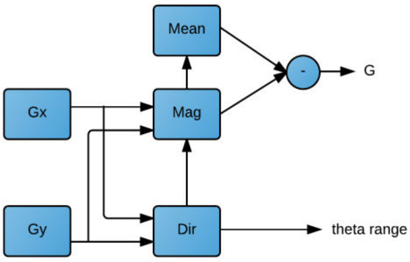


Fig. 2: Gradient and Phase Calculus

B. *Lane Detection using Hough Transform*

The generation sinus and cosines of trigonometric functions is one of the main barriers to implementation of the SHT in hardware [4]. One technique is to employ LUTs [5] [6], but on the FPGA implementation intended herein, LUTs would have a severe impact on on-chip memory usage, given that memory is also required for vote arrays. Another possibility is distributed arithmetic. The Coordinate Rotation Digital Computer offers an elegant way of trigonometric calculus implementation. Thanks to the iterative approximation of an angle by a rotation, the algorithm use only a number of addition, shift and comparison operations, making CORDIC-based algorithm remains attractive for application where real time hard constrains are critical.

$$\begin{cases} x_{i+1} = K_i[x_i - y_i d_i 2^{-i}] \\ y_{i+1} = K_i[y_i - x_i d_i 2^{-i}] \end{cases} \tag{3}$$

$\arctan(2^{-i})$ is approximated by 2^{-i}, $K_i = \cos(\arctan(2^{-i}))$, and $d_i = \pm 1$.

The CORDIC algorithm works in two modes, rotation and vector mode. In the vector mode, we initialize the Cartesian coordinates x and y of the vector to calculate the module and the phase of the vector. In the rotation mode, we start by initializing the rotational angle to calculate simultaneously the corresponding sinus and the cosine values.

The pipeline implementation of the edge operator and HT is shown in Fig. 3. We start from a resulting binary image from a proposed Sobel edge detector. θ is obtained through a CORDIC vector mode, sine and cosines are obtained through a rotation CORDIC mode. In order to reduce the number of iterations, we restrict the range of each set of rotations according to the absolute values of G_x and G_y and the resulting sign of their comparison as reported in (4) which improve the accuracy of θ, without gain-rotations number trade-off. We note that this simplification of HT optimizes latency and power consumption since the number of elementary operation is reduced.

$$\begin{cases} 0 < \theta < \pi/4 & if\ |G_x| > |G_y|\ and\ G_x > 0 \\ \pi/4 < \theta < \pi/2 & if\ |G_x| > |G_y|\ and\ G_x > 0 \\ \pi/2 < \theta < 3\pi/4 & if\ |G_x| > |G_y|\ and\ G_x < 0 \\ 3\pi/4 < \theta < \pi & if\ |G_x| > |G_y|\ and\ G_x < 0 \end{cases} \quad (4)$$

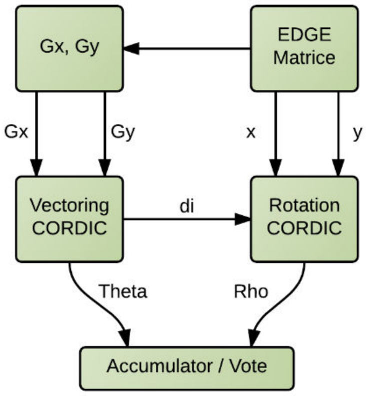


Fig. 3: CORDIC HT principle

Fig. 4 shows the circuit of the HT using CORDIC bloc.

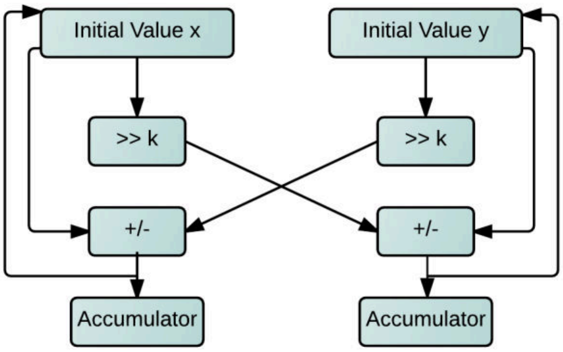


Fig. 4: CORDIC Pipeline Architecture

C. *Lane Tracking using Kalman Filter*

The Kalman filter is a very powerful method of estimation based on a set of mathematical equations. It is able to estimate and correct the state of a system, and it is applied to track previously detected lanes. The Kalman filter operates in four stages:

1/ Process Equation:

$$x_k = A_k.x_{k-1} + B_k.u_k + w_{k-1}$$

A_k : is the transition matrix between x_k and x_{k+1}. B_k is the control input model which is applied to the control vector u_k. w_{k-1} is the Gaussian process noise.

2/ Measurement Equation:

$$z_k = H_k.x_k + v_k$$

z_k : is the measurement observed at a time k and $k+1$. H_k : is the measurement matrix. v_k is the Gaussian measurement noise.

3/ Time Update Equation (Prediction):

$$\hat{x}_k^- = A_k.\hat{x}_{k-1}^- + B_k.u_k$$

$$P_k^- = A_k.P_{k-1}.A_k^T + Q_k$$

$\hat{x}_k^-$: is the prior estimate. P_k^- : is the prior error covariance.

4/ Measurement Update Equation:

$$K_k = P_k^-.H_k^T.R_k^{-1}$$

$$K_k = P_k^-.H_k^T.(H_k.P_k^-.H_k^T + R_k)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k.(z_k - H_k.\hat{x}_k^-)$$

$$P_k = (I - K_k.H_k).P_k^-$$

K_k : is the Kalman Gain.

R_k : is the measurement noise covariance matrix.

P_k : is the estimation matrix of a posterior error covariance.

$\hat{x}_k$: is the prerequisite state estimation

I : is the identity matrix with appropriate dimensions.

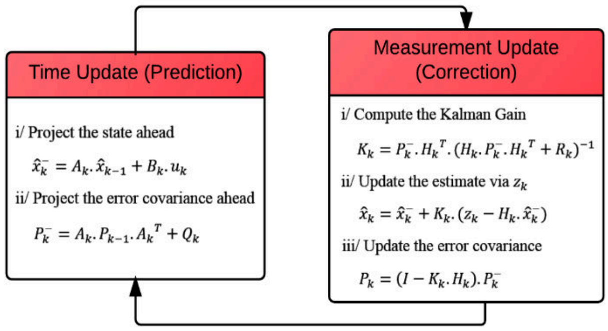


Fig. 5: Descripton of Kalman filter's update as a predictor-correct

IV. Implementation and Results

The set-up test of our algorithm is shown in Fig.6. The video input feed required by our project is generated by a real time digital camera in standard NTSC video format with a digital camera Nikon COOLPIX P510 connected to the ALTERA DE2 Board via the NTSC connector. The board is connected to a PC via rs232 interface for test and comparison purpose. Sample videos were used to test the reliability of our algorithm. Each image recorded by the camera is sent to the FPGA is stored in SRAM memory. The module is designed around a dedicated FSM. For the proper functioning of our design, and in order to avoid problems of synchronization, we use two clocks 27 MHz and 50 MHz. Table. 1 shows the operating frequencies of individual design modules. The main clock source of the system is the external 27 MHz crystal oscillator. The 50 MHz and 200 MHz clocks are generated internally by a phase locked loop (PLL) from the input clock. The signal transmission between two clock domains is processed using the double clock FIFO buffers and synchronization steps to avoid synchronization problems. Implementation occupies 45% of the Cyclone II EP2C35F672C resources. In terms of latency, timing analysis shows that Software the maximum operating frequency (Fmax) was 115 MHz. The standard analog encoding of NTSC video requires a processing speed of 25 frames per second, which proves that the real time constraint for this standard is given.

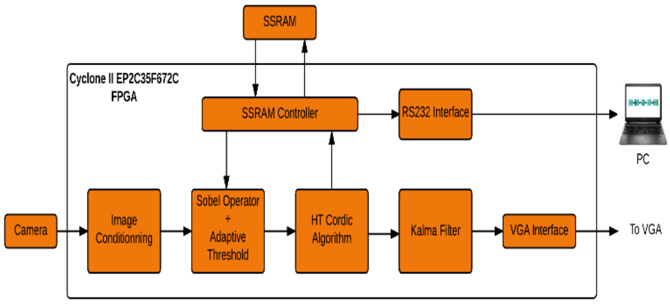


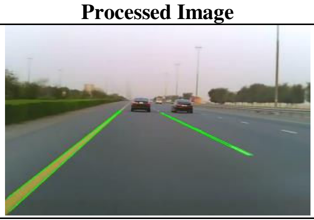
Fig. 6: Test setup setting

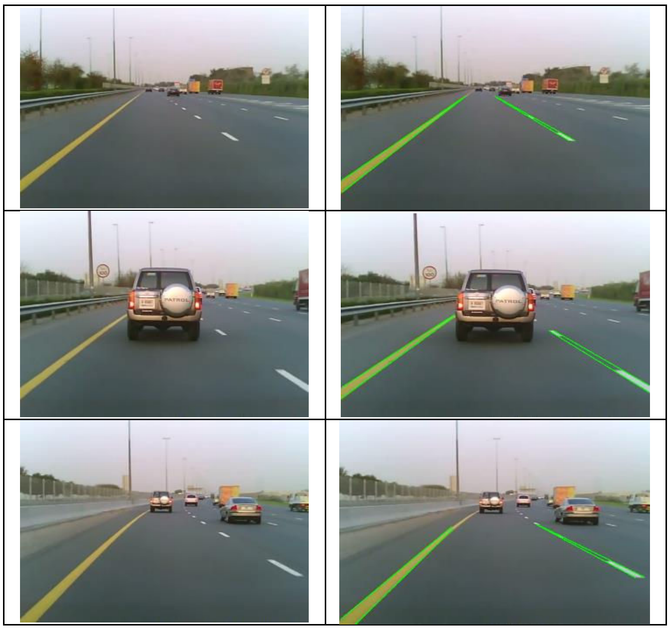
TABLE I. FPGA RESOURCES UTILIZATION

Logic Utilization	Used	Available	Utilization
Total logic elements	14,945	33,216	45%
Total combinational functions	14,945	33,216	45%
Total memory bits (Kbit)	1,555	3,888	40%
Embedded 18 x 18 multipliers	15	266	6%
Total PLLs	2	4	50%

Table. 2 shows the proposed system results on multiple images. One can see that the proposed algorithm can reliably detect and track lane.

TABLE II. SHOWS THE TRACKING RESULTS

Original Image	Processed Image
	



V. CONCLUSION

In this article, we have implemented a robust algorithm to detect and track lane, whatever are the conditions of visibility. It is a system based on the Hough Transform combined with the Kalman Filter. Segmentation is based on the Sobel edge detector with an adaptive threshold. It is implemented using the CORDIC algorithm to improve the performance and thus the detection efficiency. The detected lines are used to predict the next model. This model is followed by using the Kalman Filter.

REFERENCES

[1] Jack Greenhalgh and Majid Mirmehdi, "Real-Time Detection and Recognition of Road Traffic Signs", IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 13, NO. 4, pp. 1498-1506, Dec. 2012.

[2] P-R. Possa, S-A. Mahmoudi, N. Harb, C. Valderrama, "A Multi-Resolution FPGA-Based Architecture for Real-Time Edge and Corner Detection", IEEE TRANSACTIONS ON COMPUTERS, VOL. 63, NO. 10, pp. 2376-2388 Oct. 2014.

[3] Jia H., Hui R. Jinfeng G., Wei H. A lane detection method for lane departure warning system, International Conference Optoelectronics and Image Processing (ICOIP), 2010, 28-31.

[4] H. Maître, m« Un Panorama de la Transformée de Hough, Traitement du Signal 2" (4) (1985).

[5] Xiangjing A, Erke Shang, Jinze Song, Jian Li, Hangen He. "Real-time lane departure warning system based on a single FPGA", EURASIP Journal on Image and Video Processing 2013.

[6] S-M. Karabernou, F. Terranti, "Real-time FPGA implementation of Hough Transform using gradient and CORDIC algorithm", *Image and* Vision Computing, Elsevier, 2005, 23 (11), pp. 1009-1017.

[7] Vijay G., Shashikant L., "An improved lane departure method for advanced driver assistance system", International Conference on Computing, Communication and Applications (ICCCA), 2012, 1 – 5.

[8] Stephan S., Sarath K., Alen A., Gamini D., "Robust lane detection in urban environments", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 123-128.

[9] Seonyoung L., Haengsmeon S., Kyungwon M. "Implementation of lane detection system using optimized Hough transform circuit, *2010 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS),* 406-409.

[10] Xiangjing A, Erke Shang, Jinze Song, Jian Li, Hangen He. "Real-time lane departure warning system based on a single FPGA", EURASIP Journal on Image and Video Processing 2013.

[11] Jingjing Zhang, Liang Liu, Binhai Wang, Xiguang Chen, Qian Wang, Tianru Zheng. "High Speed Automatic Power Line Detection and Trasking for a UAV-Based Ispection". International Conference on Industrial Control and Electronics Engineering 2012.

[12] G, Welch and G, Bishop,"An Introduction to the Kalman Filter, "Department of Computer Science, University of North Carolina at Chapel Hill Chapel Hill, NC 27599-3175.

[13] Y. Li, A. Iqbal, and N.R. Gans. "Multiple Lane boundary detection using a combination of Lowel-level image features". In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, pp 1682-1687. IEEE, 2014.

[14] A. Borkar, M. Hayes, and M.T. Smith. "Robust lane detection and tracking with ransac and Kalman Filter", Proceedings of the IEEE International Conference on Image Processing, pp 3261-3264, 2009.