FPGA Implement of a Vision Based Lane Departure Warning System

Yu Ren Lin and Yi Feng Su

***Abstract*—**Using vision based solution in intelligent vehicle application often needs large memory to handle video stream and image process which increase complexity of hardware and software. In this paper, we present a FPGA implement of a vision based lane departure warning system. By taking frame of videos, the line gradient of line is estimated and the lane marks are found. By analysis the position of lane mark, departure of vehicle will be detected in time. This idea has been implemented in Xilinx Spartan6 FPGA. The lane departure warning system used 39% logic resources and no memory of the device. The average availability is 92.5%. The frame rate is more than 30 frames per second (fps).

***Keywords*—**lane departure warning system; image; FPGA

I. Introduction

In recent year, as high growth of population of vehicles, the traffic accidents also become more and more serious. Most occurrences of the accidents results from the distracted driving and inattentive driving. When driver becomes tired or distracted, vehicle might move out of its lane. In order to help the drivers avoid danger as much as possible, “Lane Departure Warning System” is developed. When vehicle begins to move out of its lane on free way and arterial roads, the system will warn a driver.

The technologies of finding the lane marks are needed by the lane departure warning system. Compared with other technologies, a vision based LDW system is a human decision-make like solution to avoid the accident caused by lane departure with low cost and high reliability. In such systems, one camera is installed in the interior of the vehicle, and the environment is sensed to supply useful information.

. There are several LDW system have been developed relying on a variety of technologies[1-9]. B.Yu, W.Zhang and Y.Cai [1] used a Gaussian template to remove the dirty spots in the image and dynamical threshold choosing to find lane marks.TLC and CCP method are used to make lane departure decision. But TLC and CCP methods need too many parameters to work out the result. The system is not easy to realize in FPGA. Cl´audio and Christan[2] proposed a lane departure warning system that estimates lane orientation through linear parabolic model. P.Y.Hsiao, C.W.Yeh, S.S.Huang and L.C.Fu[3] proposed an embedded ARM-based real-time LDWS. 1-d Gaussian smoother and a global edge detector are adopted to reduce noise effects in the image and lane departure decision is based on spatial and temporal mechanisms. But the ARM-based systems are more expensive than the system which is implemented in ASIC.

W.Zhu, F.Liu, Z.Li, X.Wang and S.Zhang[4] proposed an algorithm to chooses a common curved lane parameter model which can describe both straight and curved lanes. N.M.Enache[5] used composite Lyapunov Functions, polydral-like invariant sets and linear matrix inequality(LMI) method to implement the lane-departure avoidance system. M.J Jeng[6] proposed a LDWS which is implemented in hardware and software. The hardware is implemented by FPGA. In software design, the global edge detection is able to transfer the gray level image to binary pattern and show the edge of object and then finding out the lane marks by peak finding and grouping, edge connecting, lane segment combination, lane boundaries selection. Because the lane marks are extracted based on color information, H.Y.Cheng[7] proposed a method to utilize size, shape, and motion information to find the lane mark out. A.AM.Assidiq[8] proposed a method to detect the lane mark by Canny edge detector and Hough Transform. J. F. Liu[9] used gradient method to find out the both lane marks.

Using vision solution often requires large memory to handle video data stream, image processing, which increase the hardware and software complexity. However, some of these functions can not be implemented by DSP. Besides, using DSP increases greatly the system cost. Therefore, FPGAs are instead of DSP. Broadly speaking, a FPGA implement can offer more functionality and be a lower-cost solution than DSP design[10].

The approach of this paper is extension of research by J. F. Liu[9] ,and organized into 5 sections. The basic processing step is introduced in section II. The FPGA architecture is introduced in section III. Section IV introduce the experimental results and implement results.

II. ALGORITHM OF LANE DETECTION

The proposed lane marks detection algorithm is based on gradient method. All steps in the algorithm is shown as Fig 1.

Yu Ren Lin is with the R&D Division, Automotive Research and Testing Center, Lugnag; Chunghua County, Taiwan (e-mail:ericlin@artc.org.tw).

Yi Feng Su is with the R&D Division, Automotive Research and Testing Center, Lugnag; Chunghua County, Taiwan (e-mail:efo@artc.org.tw).

.

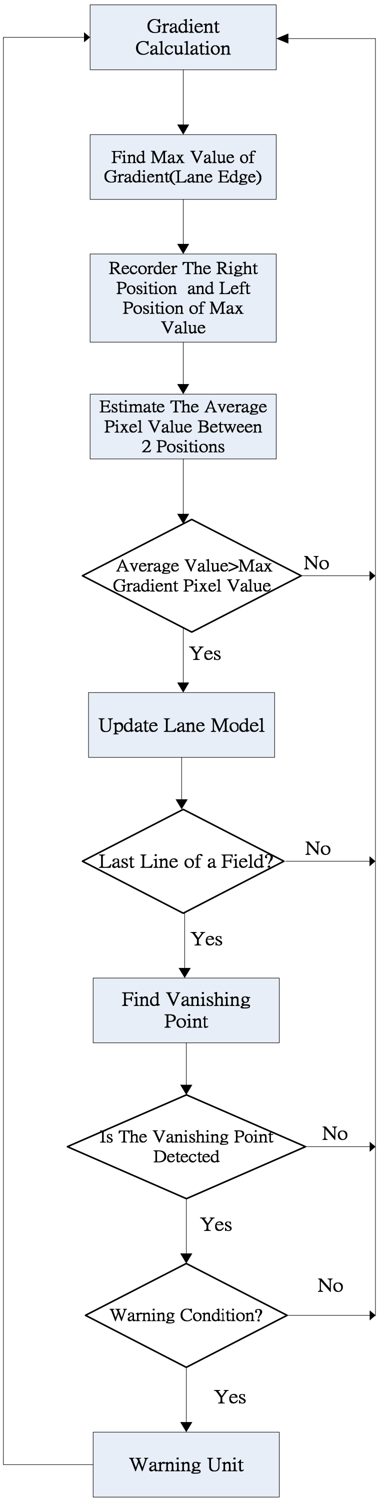

Gradient Calculation
Find Max Value of Gradient(Lane Edge)
Recorder The Right Position and Left Position of Max Value
Estimate The Average Pixel Value Between 2 Positions
Average Value>Max Gradient Pixel Value
No
Yes
Update Lane Model
Last Line of a Field?
No
Yes
Find Vanishing Point
Is The Vanishing Point Detected
No
Yes
Warning Condition?
No
Yes
Warning Unit

Figure 1 Algorithmic process of lane detection

First, the region of interested will be defined. Next, the gradient of line is calculated. The line gradient is estimated by Equation (1):

$$G_i = -P_{i-1} + 2 \times P_i - P_{i+1} \quad (1)$$

Wherein, Gi is the gradient value of position i, Pi is the pixel value in the interested region of image. After estimating the line gradient, the max gradient value of right side and left side of a lane can be found. If the distance between right position and left position is equal to or larger than the threshold value, these two positions and the pixel value of the positions will be saved. Then the average pixel value of the region between right position and left position will be estimated. If the average value is larger than the pixel value of two positions, the edge of a lane mark will be defined.

After finding the edge of one lane mark, the middle position of a lane will also be found. Then the middle position of lane marks will be used to estimate the lane model by least square method. According to equation (2) and (3), the equations of right and left lane marks will be defined.

$$a = \frac{\sum_{i=1}^{n} u_i v_i - \sum_{i=1}^{n} u_i \sum_{i=1}^{n} v_i}{n \sum_{i=1}^{n} u_i^2 - (\sum_{i=1}^{n} u_i)^2} \quad (2)$$

$$b = \frac{\sum_{i=1}^{n} u_i^2 \sum_{i=1}^{n} v_i - \sum_{i=1}^{n} u_i \sum_{i=1}^{n} u_i v_i}{n \sum_{i=1}^{n} u_i^2 - (\sum_{i=1}^{n} u_i)^2} \quad (3)$$

Wherein, ui is the middle position of a lane, vi is the row position and n is the amount of detected middle positions. The lane model can be defined as equation (4) and equation (5).

$$y = a_r x + b_r \quad (4)$$

$$y = a_l x + b_l \quad (5)$$

Then the abscissa of vanishing point y and the ordinate of vanishing point x can be found by equation (6) and equation (7), ar is the slope of right lane, br is its constant, al is the slope of left lane, and bl is its constant.

$$x = \frac{a_r - a_l}{b_l - b_r} \quad (6)$$

$$y = \frac{a_r(a_r - a_l)}{b_l - b_r} + b_r \quad (7)$$

As show as figure 2, when both lane marks are detected, the vanishing point will be in the detecting region.

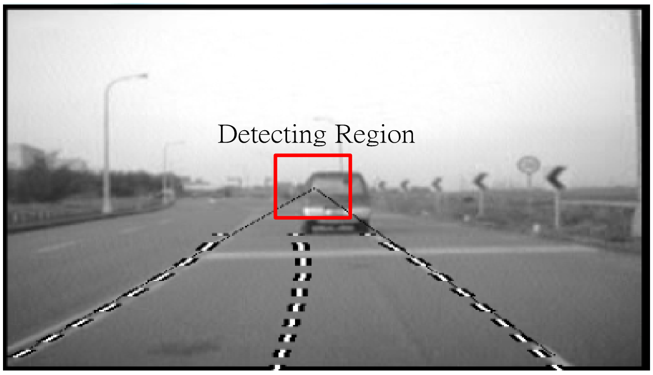


Figure 2 Software result of lane detection

The lane departure offset estimation method is as show as figure 3, the slope of lane can be found by equation (4) and equation (5). Then the distance between the vehicle and the lane ΔX can found by equation (8) and (9).

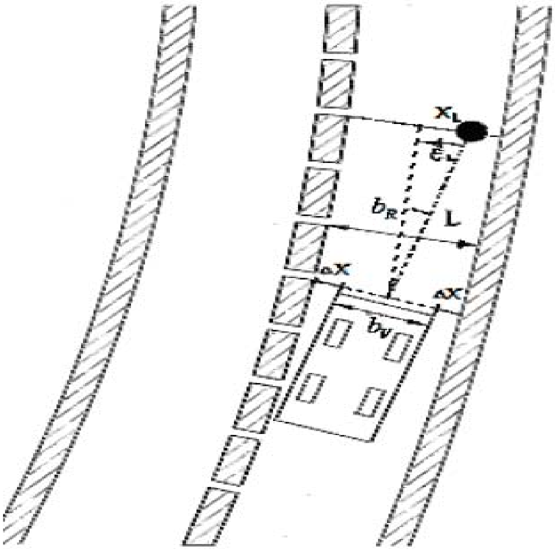


Figure 3 Method of Lane Departure Estimation

$$\Delta X_L = \frac{b_R}{2} - (\frac{b_V}{2} + X_L) \qquad (8)$$

$$\Delta X_R = \frac{b_R}{2} - (\frac{b_V}{2} - X_L) \qquad (9)$$

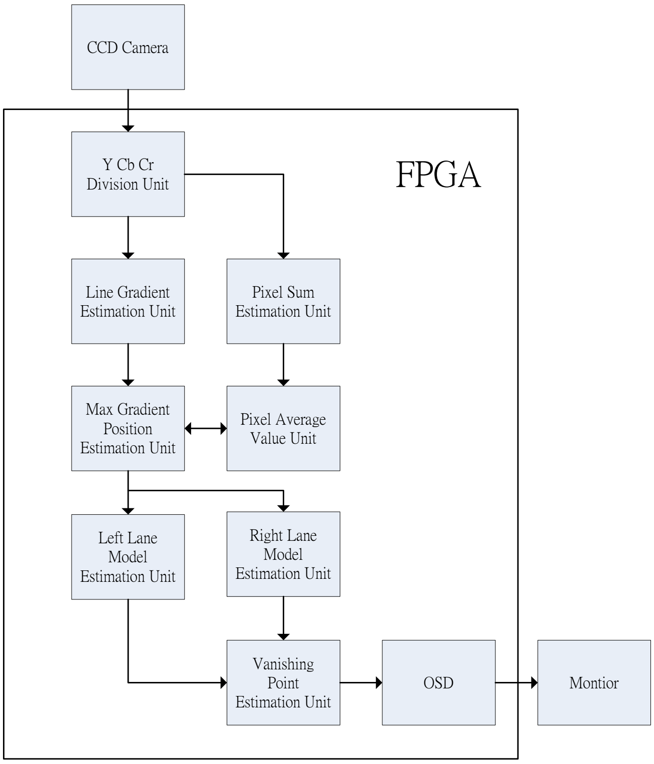
If the distance ΔX is larger than the threshold, the vehicle moves out of the lane.

III. FPGA IMPLEMENT OF LANE DETECTION ALGORITHM

The block diagram of proposed lane detection algorithm is shown in Fig. 4, which consists of five main parts: a line gradient estimation unit, pixel sum and pixel average estimation unit, a max gradient position estimation unit, right lane and left lane model estimation unit, and vanishing point estimation unit.

First, the gradient of a line is calculated by line gradient estimation unit. The max gradient position estimation unit then finds out the max value of gradient and its positions. Next, the pixel sum estimation unit and pixel average estimation unit estimate the sum and the average of pixel value which is between right max gradient position and left max gradient position. If the average larger than pixel value of right side and left side, the boundary of a lane is found.

After finding the boundary, the middle position of a lane mark is found. The middle position is used to find lane model by the left lane model estimation unit and the right lane model estimation unit. Then the abscissa of vanishing point y and the ordinate of vanishing point x will be calculated by vanishing point estimation unit. The offset ΔX is also estimated by vanishing point estimation unit. When vanishing point is in the detecting region and the ΔX is equal or greater than the alert limit, the system will alert the user.



CCD Camera
FPGA
Y Cb Cr Division Unit
Line Gradient Estimation Unit
Pixel Sum Estimation Unit
Max Gradient Position Estimation Unit
Pixel Average Value Unit
Left Lane Model Estimation Unit
Right Lane Model Estimation Unit
Vanishing Point Estimation Unit
OSD
Montior

Figure 4 Block diagram of proposed lane detection algorithm

A. *Line Gradient Estimation Unit*

The architecture of the line gradient estimation unit is shown as figure 5.

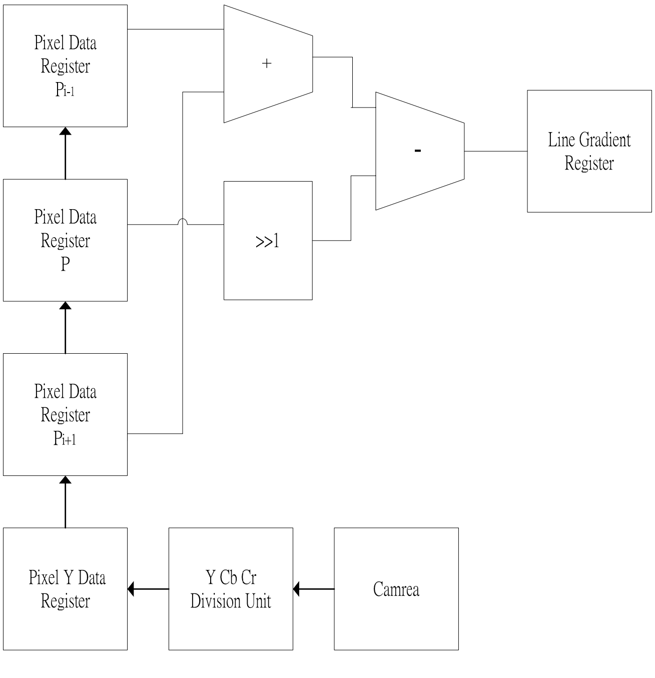


Figure 5 Architecture of the line gradient estimation unit

In order to reduce the area of memory, all data is estimated in real time. First, the Y Cb Cr division unit outputs Y data to the pixel data register Pi+1. Next, the pixel data which is in the pixel register Pi+1 will shift to register Pi and new pixel data from Y data register will be stored by register Pi+1. Pixel data register Pi-1 will receive pixel data and line gradient will be estimated until next clock cycle. After calculating the line gradient, these data will be stored in line gradient register.

B. Max Gradient Position Estimation Unit

The max gradient position estimation unit finds out the both side positions where the max line gradient is in. The architecture of this unit is as shown as figure 6.

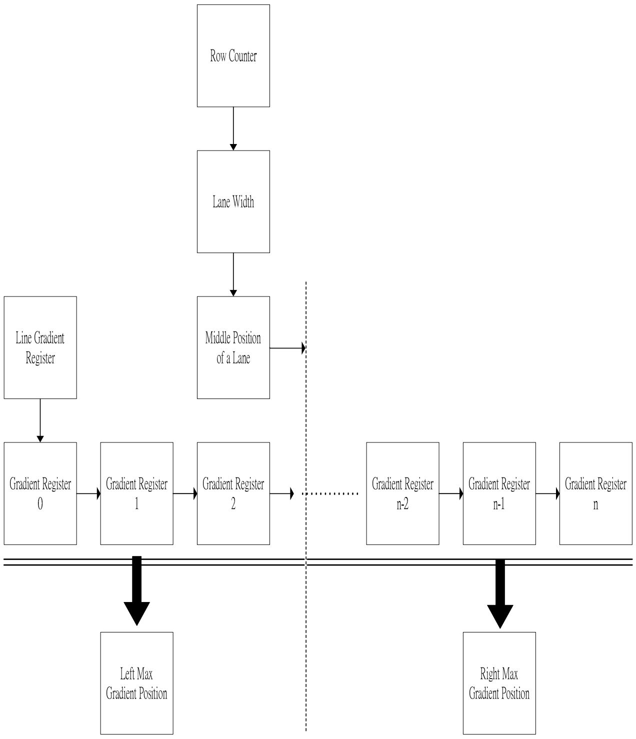


Figure 6 Architecture of the max gradient position estimation unit

In this unit, the line gradient will shift from one gradient register to next register, the left max gradient and the right max gradient will be found. One register can represent one position. For example, if max gradient is in the gradient register 4, the max gradient position is 5. According to the lane width, the distance between right max gradient position and left max gradient position can be defined. One look up table is used to define the lane width.

C. Pixel sum and pixel average unit

In order to make sure the distance between right max gradient position and left max gradient position are the boundary of a lane mark, pixel average is needed. If the average larger than the pixel value which is in the right position and left position, the boundary of a lane mark can be found. The architecture of pixel sum and average unit is shown as figure 7.

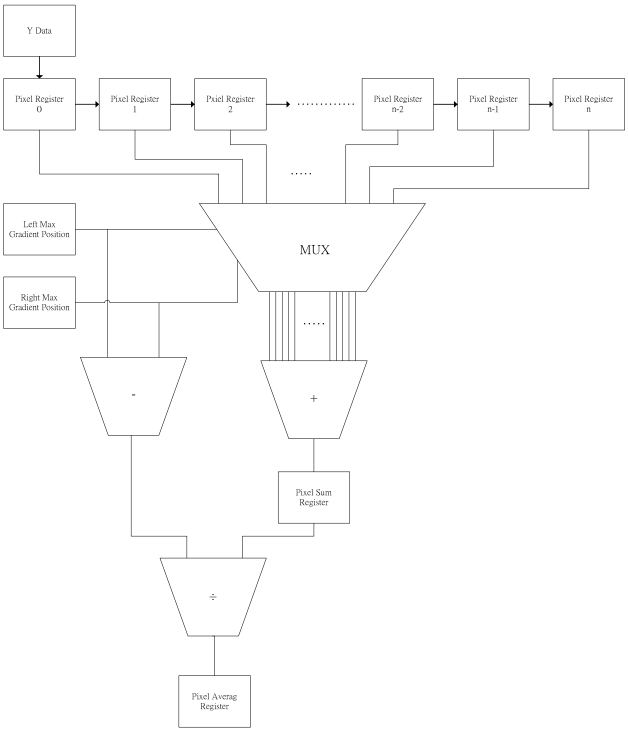


Figure 7 The architecture of pixel sum unit and pixel average unit

If a lane mark boundary is found, the max gradient positions will be converted into image position, and the middle position of a lane mark will also be found.

D. Left and right lane model estimation unit

According to equation (2) and (3), the lane model estimation unit can be implemented by adders, subtractors and multipliers. The architecture of lane model estimation unit is shown as figure (8) and (9).

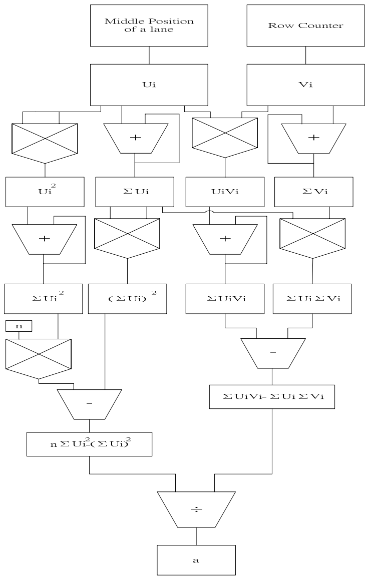


Figure 8 The architecture of lane model estimation unit (“a” estimation)

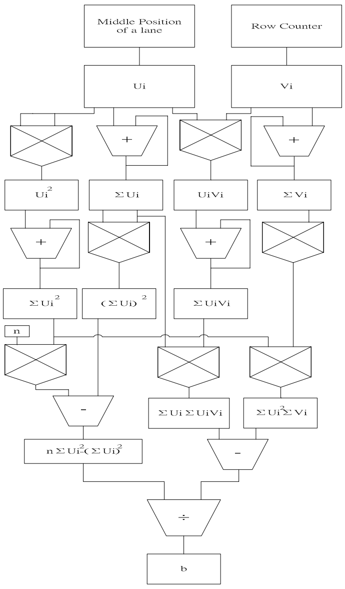


Figure 9 The architecture of lane model estimation unit (“b” estimation)

In this unit, multiplier is implemented by adders and shifters. Two non-restoring dividers are used to calculate slope “a” and constant “b”. A state machine is also used to control the steps of estimation.

E. *Vanishing point estimation unit*

After estimating the equation of both lanes, the vanishing point of both lane marks can be defined. The architecture of vanishing point estimation unit is as shown as figure 10.

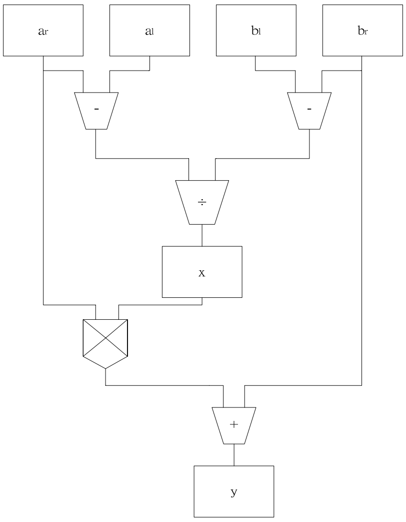


Figure 10 The architecture of vanishing point estimation unit

. In this unit one non-restoring divider is used to estimate ordinate .First, the ordinate of vanishing point x can be estimated by subtractors, adders and multipliers. In order to reduce the chip area, equation (4) or (5) is used to implement the abscissa of vanishing point y calculation unit.

After finding right and left lane marks, the displacement of vehicle can be calculated. Then ΔX can be determined. If values of ΔX larger than the alert limit set by user, the user is alerted by a LED, a buzz or a speaker.

IV. IMPLEMENT RESULTS AND EXPERIMENTAL RESULTS

A. *Implement results*

Our goal is to implement a low cost lane departure warning system in FPGA. The system has been successfully implemented and tested on the SPARTAN6 XC6SLX150 FPGA. The utilization of the device is shown in Table 1.

Table1 Resource Utilization of Device(XC3SD3400A)

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	10,791	184,304	5%
Number of Slice LUTs	27,699	92,152	30%
Number used As logic	26,884	92,152	29%
Number of Occupied Slices	8992	23,038	39%

The lane departure warning system only use 39% logic resources and no block memory of the device.

B. *Experimental results*

In this section, the LDWS was verified by using video database. Our experimental environment is based on the FPGA kit as Fig11. The verifiable video is stored in portable video recorder, and they were played in the form of NTSC and input to the FPGA..

Figure 11 The Xilinx Spartan 6FPGA kit

The video includes various situations (eg. daytime, nighttime, highway and urban road).The picture of test results is shown as figure 12.If both lane marks are detected, two green lines and a square will be shown in the screen. Both green lines are the lane marks and the midpoint of the square is the vanishing point of both lane marks. If the departure of vehicle is detected, the green lines and square will turn to red. The system availability is listed in Table 2. The average availability is 92.5%.

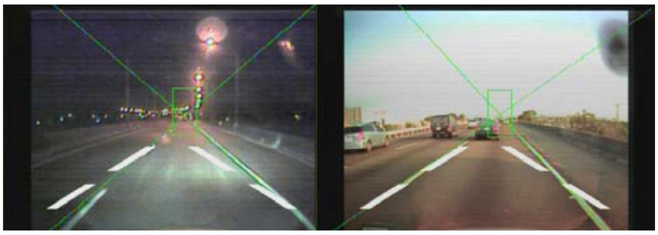


Figure 12 Picture of test results

Table2 System availability

Testing Condition Used	System Availability
Highway in the daytime	97%
Urban road in the daytime	89%
Highway at night	94%
Urban road at night	90%
Average availability	92.5%

According to the experimental results, we can summarize the following point:

(1) The system can be used under various vehicles and environments in highway and urban road.

(2) The system can provide high availability, reliability and accuracy in lane deviation.

(3) The frame of system is more than 30 fps (frame per second), and it meets the requirement of real-time computing.

(4) Only 39% resource of a small FPGA and no memory is used to implemented the system. It meets the requirement of low cost LDWS.

V. CONCULSION

Compared to traditional DSPs and microcontrollers(MCUs), FPGA offers faster processing speed and lower cost. The research proposed FPGA architecture of lane departure warning system and implement the system in FPGA. The system only uses 39% logic resources and needs no memory of the FPGA. And the frame rate is more than 30 frames per second. The average availability of the LDWS is 92.5%. In the future, the architecture would be implemented by ASIC for lower cost.

REFERENCES

[1] B. Yu, W. Zhang, Y. Cai “A Lane Departure Warning System based on Machine Vision” in IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008, pp.197-201.

[2] C.R. Jung, C.R. Kelber “A Lane Departure Warning System Using Lateral Offset with Uncalibrated Camera” in Proc. Int. IEEE Conf. on Intelligent Transportation Systems, Sep. 13–16, 2005, pp. 348-353.

[3] P. Y. Hsioa, C. W. Yeh, S. S. Huang, L. C. Fu “A Portable Vision Based Real-Time Lane Departure Warning System : Day and Night”, IEEE Trans. Intelligent Transportation Systems, vol. 58, no. 4, pp. 2089-2094, May 2009.

[4] W. Zhu, F. Liu, Z. Li, X. Wang, S. Zhang “A Vision Based Lane Detection and Tracking Algorithm in Automatic Drive”, in IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008, pp.799-803..

[5] N. M. Enache, S. Mammar and M. Netto, B. Lusetti “Driver Steering Assistance for Lane-Departure Avoidance Based on Hybrid Automata and Composite Lyapunov Function” in IEEE Trans. On Intelligent Transportation Systems, Mar. , 2010, vol1, pp.28-39.

[6] M. J. Jeng, C. Y. Guo, B. C. Shiau and L. B. Chang, “Lane Detection System Based on Software and Hardware Codesign,” in Proc. 4th Int. IEEE Conf. on Autonomous Robots and Agents, Feb. 10-12, 2009, pp. 319 - 323.

[7] H. Y. Cheng, B. S. Jeng, P. T. Tseng, K. C. Fan, “Lane Detection With Moving Vehicles in the Traffic Scenes,” in IEEE Trans. on Intelligent Transportation Systems, Vol.7 Dec. 2006, pp. 571-582.

[8] A. AM. Assidiq, O. O. Khalifa. Md. R. Islam, S. Khan, “Real Time Lane Detection for Autonomous Vehicles”, in Proc. Int. IEEE Conf. on Computer and Communication Engineering, May. 13-15, 2008, pp. 82 – 88.

[9] J. F. Liu, Y.F. Su, M. K. Ko and P. N. Yu “Development of a Vision-Based Driver Assistance System with Lane Departure Warning and Forward Collision Warning functions” Digital Image Computing: Techniques and Applications, 2008, pp. 480–485.

[10] Sudhir Sharma, Wang Chen, “Using Model-Based Design to Accelerate FPGA Development for Automotive Applications”, in 2009 SAE World Congress, January 1-15,2009.