Implement of Vehicle Blind Spot Detection System Using SDSoC

Chayanin Youngyai* and Wannarat Suntiamorntut

Department of computer engineering, Faculty of Engineering, Prince of Songkla University, Hatyai Songkhla 90112 Thailand

* Corresponding author. E-mail address: c.youngyai@gmail.com

Received: 10 October 2019; Revised: 22 April 2020; Accepted: 29 April 2020; Available online: 5 June 2020

Abstract

Embedded system and image processing is the technology used to apply for designing systems to facilitate drivers in the future by replacing wing mirrors with the camera to detect objects at the blind spot at the back. When the object is detected within the blind spot zone, the system is alarm drivers. These are a concept of The blind spot detection vehicle system (BSDS). This research presents BSDS to help drivers, which is designed on Field Programmable Gate Array (FPGA). There are two main modules for creating. The first module is pre-processing image using grayscale to resize the dataset in terms of the bit stream in every pixel and region-of-interest (ROI) for specifying the area of pixels wanted to reduce the noise of the data set or data quantity that needs to use in calculation. The second module is vehicle detection and alarm by Sobel operator for object and shadow detection to detect the shadow of the vehicle and confirm the existence of it. This research shows details of each algorithm, flow chart, and effectiveness of vehicle detection. This idea has been designed and developed by a software called Software-Defined System-On-Chip (SDSoC) on the Zybo board. In conclusion, the system can detect vehicles at resolution 1920*1080 within 12 ms/frame and accuracy of vehicle detection at 100% recall.

Keywords: SDSoC, FPGA, BSDS, image-processing, pre-processing image, region-of-interest, Sobel operator, shadow detection

Introduction

Advanced Driver Assistance Systems (ADAS) are the automation system that installs inside the vehicle and assists the driver in the driving process. ADAS technology helps in reducing accidents. The ADAS has received much attention, and many driving assist systems are developed, including traffic sign monitoring system, automatic parking system, lane detection system, BSDS, and so forth.

In Asian countries, especially ASEAN, there are many vehicles on the road leading to high chances of accidents, respectively. In fact, if the driver is not aware of the vehicle in his blind spot, it can easily lead to accidents when changing lane or U-turn. So, having BSDS can alarm driver to be more careful to reduce the loss on the road. There are many research papers in the present that use the sensor to design BSDS, including Song, Chen, and Chiu Huang (2004), that uses many ultrasonic sensors to detect vehicles or Ajay and Ezhil (2016) use ultrasonic sensors and Arduino MEGA to design BSDS. But since sensors have a relatively high cost, many research turns to use image processing instead of sensors, such as Chang and Hsu (2010) using SURF algorithm to detect moving objects and alarm, Nong, Osman, Yusof, and Sidek (2015) designs motorcycle detector on FPGA using image processing, Baek, Lee, Park and Seo (2015) uses HoG cascade classifier and Kalman filter to detect vehicles, Fernández et al. (2013) use SVM classification to detect vehicles during daytime and headlight detection for vehicle detection during the night, Sotelo and Barriga (2008) uses optical flow to design BSDS, Jung and Yi (2018) have also developed BSDS by using histogram of oriented gradients (HOG) and the support vector machine (SVM), Poonam and Gajanan (2017) provides vehicle detection and tracking the movement of objects by different location of the object between the current frame and previous frame.

Therefore, this research uses image processing in BSDS design by working on a small computer that can install in the vehicle. Field Programmable Gate Array (FPGA) can develop the BSDS because FPGA technology builds parallel operating modules, executing its work quickly and respond in real-time. Our research uses SDSoC tools to create the FPGA platform for development BSDS. The system has cameras that install on wing mirrors of the vehicle to receive image data to process as designed. The system is alert when the vehicle is approaching in the blind spot area, which is called the detection area, as in figure 1.

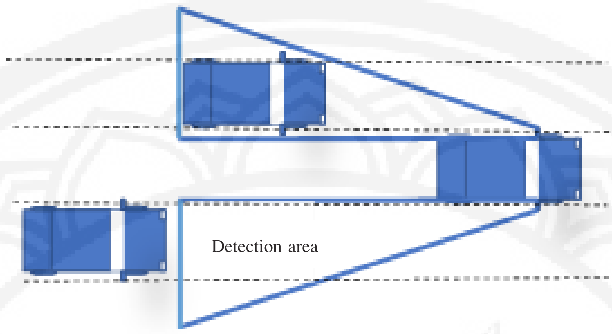


Figure 1 Side vision for detection area

Methods and Materials

1. Hardware and Software Materials

1.1 Zybo Board Architecture

Zybo Board is a System on a chip (SoC). SoC is an integrated circuit, which integrates all parts of the computer and electronics circuit within a single chip. Zybo Board consists of two parts including, 1) Innovative ARM®, which operates software well, and can install the operating system. 2) FPGA is the technology, which designs the hardware circuit. FPGA can design the system using gate inside to create the hardware platform, which does not require the processing unit of calculation for the ability to real-time respond effectively with structures, as in figure 2.

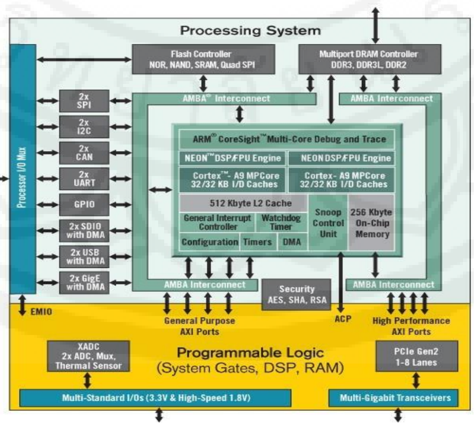


Figure 2 Zybo board architecture

1.2 SDSoC

SDSoC is a software, which helps to design the hardware platform on the SoC board supports C/C++ language for the effectiveness of the system in the aspect of power-saving and speed in work that it reduces the time for designing. It is a good alternative that helps design systems that require high speed instead of former hardware platforms that uses Vhdl or Verilog language, which takes a long time and highly professional. SDSoC can adjust modules from hardware to software or software to hardware for flexibility of the system's effectiveness, as in figure 3. However, we can use C/C++ language to design the system. Still, the developer needs to consider parallelism of each condition in certain functions if the developer does not consider it, developed functions won't generate to HW of that function, or even it can generate, work of the module will not be as designed.

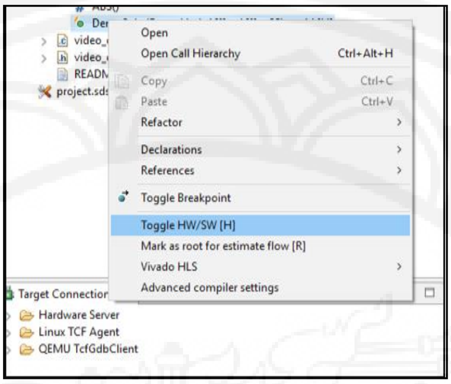


Figure 3 Toggle function to hardware on SDSoC

2. System Architecture and Procedures

BSDS divides into two modules. The first one is the pre-processing image module for data input, noise reduction, and resizing the data. The second module is called the image-processing module, and it is part of vehicle detection with BSDS with the flowchart, as shown in figure 4.

I. Receiving data from cameras on wing mirrors of vehicles.

II. Process obtained data at pre-processing image module using Gray Scale and ROI.

III. Detect vehicle by image-processing using Sobel operator and shadow detection

IV. Alarm driver when a vehicle enters detection area.

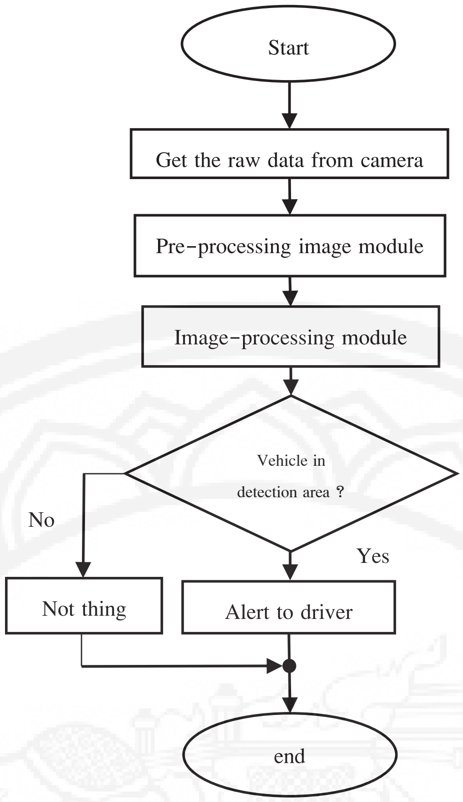


Figure 4 The flowchart of BSDS

2.1 Pre-processing image

Pre- processing image is the method for preparing the dataset, as reducing noise, Resizing dataset, filtering, and format transforms, and so forth. This module includes two sub-modules, which are grayscale and ROI. Grayscale uses to resize the dataset from 24 bit/pixel to 8 bit/pixel in every pixel. ROI uses to reduce the quantity of data.

2.1.1 Grayscale

Grayscale is the algorithm, which is converting the format of data image in each pixel of the original figure in RGB color space, as in figure 5 to grayscale space in figure 6. It can be observed that the resizing of data from 24 bit to 8 bit.

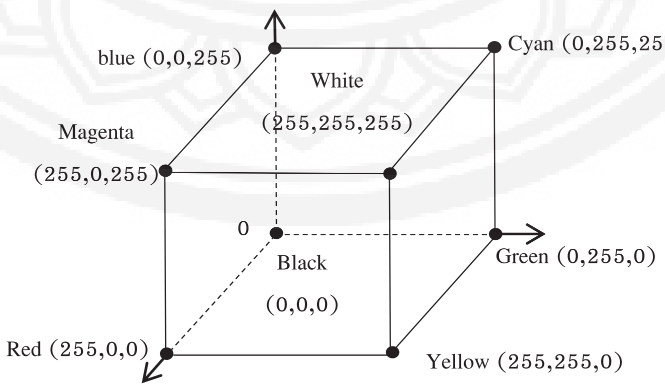


Figure 5 RGB color space

Figure 6 Gray scale space

This research uses Luma algorithm to design the grayscale module, which loads weights of R, G, B values of each pixel as in equation 1. This research developed the system that needed to work in real-time. We adjust the algorithm with the shift bit technique instead of multiplication for reducing processing, as shown in equation 2. The result of the grayscale module, as shown in figure 7.

$$G = 0.2126*R + 0.7152*G + 0.0722*B \quad (1)$$

$$G = (54*R + 183*G + 18*B + 128) >> 8 \quad (2)$$

(a) (b)

Figure 7 (a) original image (b) Gray scale image

2.1.2 ROI

The region of interest (ROI) is another algorithm that's often used as it is an algorithm for filtering data in the area of interest in each frame. The ROI defines the borders of data in the area of interest with concept, as in figure 8. In case original figures have the resolution of 400* 800 pixels, but We focus the data in the lower right area with the width of 200 pixels and height of 400 pixels. We can send only specific data in the field of interest for reducing the processing time in the next module.

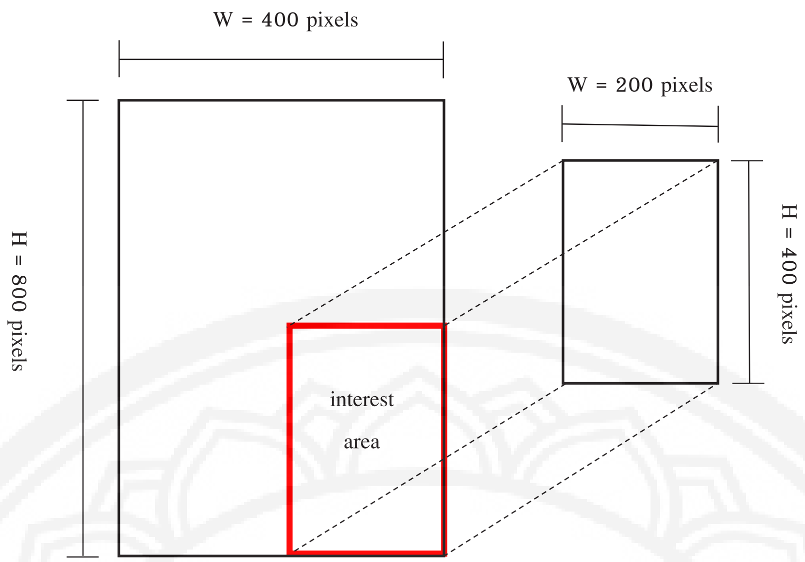


Figure 8 concept of ROI

This research paper uses the concept above by counting to state data-position used to process in the module since original data is in the data streaming form at the resolution of 1920*1080 pixel/frame. We calculate the area of interest image from counting by determining data value in each pixel as follows. In case it is in an uninterested area, it will be black, while in the field of interest, it will be the same as the original data. The result of ROI module, as shown in figure 9.

(a) (b)

Figure 9 (a) Output of gray scale module (b) Output of pre-processing image module

The pre-processing image module receives the data from the camera in real-time. This module can support the data, which is the resolution of 1920*1080 pixel/frame. We use grayscale and ROI for reducing the size of data and the noise of data. Therefore, the result of the pre-processing image module is in terms of the grayscale format.

2.2 Image-processing

Image-processing is a module used to process data for vehicle detection and alert to driver when the vehicles enter the detection area. The image-processing module uses the Sobel operator to detect vehicles, and shadow detection confirms the detected object that is the vehicle and tracks their location. The image-processing module has the flowchart, as shown in figure 10.

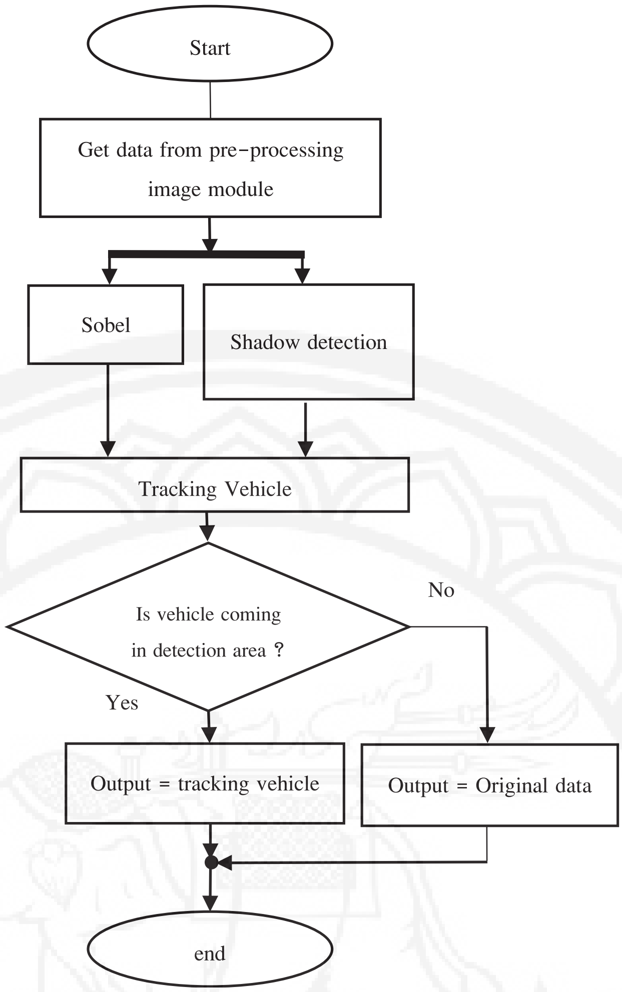


Figure 10 The flowchart of image processing module

2.2.1 Sobel operator

Sobel operator is an algorithm used widely to calculate for object in each frame using two 3*3 templates as in figure 11 by separating into horizontal differences and vertical differences and using both values to calculate magnitude like equation 3.

-1	0	1
-2	0	2
-1	0	1

(a)

1	2	1
0	0	0
-1	-2	-1

(b)

Figure 11 (a) horizontal template (b) vertical template

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \qquad (3)$$

In Anusha, JayaChandra, and Satya (2012) are the implementation of the Sobel algorithm on FPGA with pipeline structure. So our research use pipeline technique to help the system process faster as input data is in streaming form by collecting data of (2*width)+3. After that, next to the clock cycle that imports data will be calculated, and data saved in FIFO(first in first out) form will be transferred with flow chart like figure 12 and usage result like figure 13.

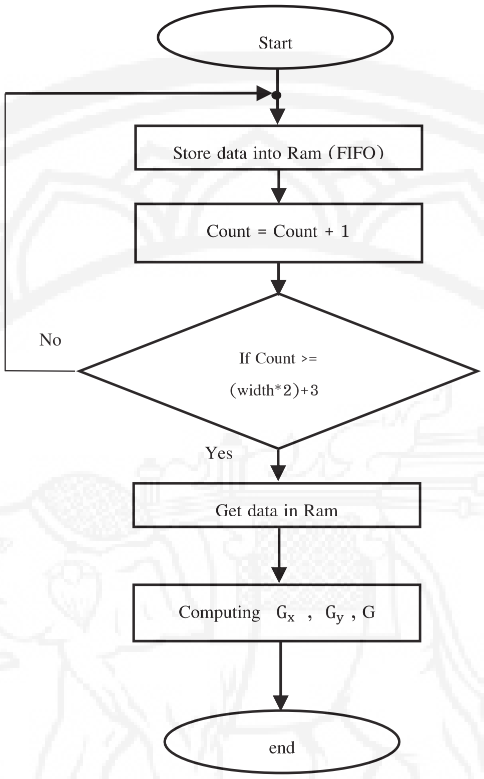
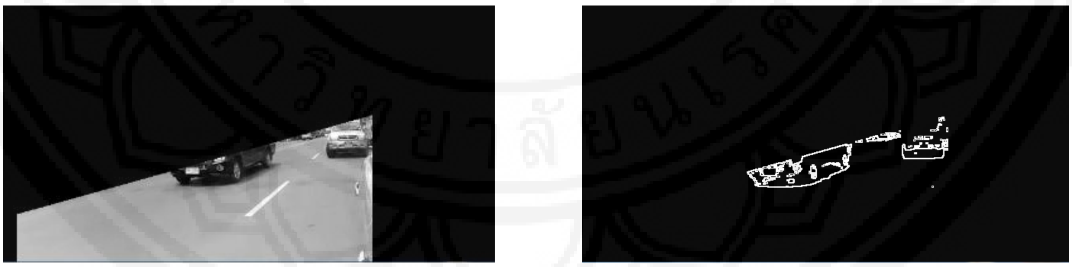


Figure 12 The flowchart of Sobel operator module



(a) (b)

Figure 13 (a) Output of pre-processing image module (b) Output of Sobel operator module

2.2.2 Shadow detection

Shadow detection is a method, which detects the shadow of an object. This research uses the technique, which calls threshold filter to filter determining color value. This research paper uses the interval estimation, as in equation 4. The interval estimation is the estimation of a specific range of population parameters using data finding the lowest and highest value. The interval estimation has less chance of inaccuracy than point estimation, and the width of scale could be more upper or lower based on the level of confidence meaning chance. The population parameter is in range of estimated value, such as P(L< μ <U) = 0.95 means chance that μ will be between L and U is 0.95 or 95%, and opportunity that μ will be less than L or more than U is 5%, as in figure 14.

$$\bar{x} \pm z_{\alpha/2} * \frac{\sigma}{\sqrt{n}} \quad (4)$$

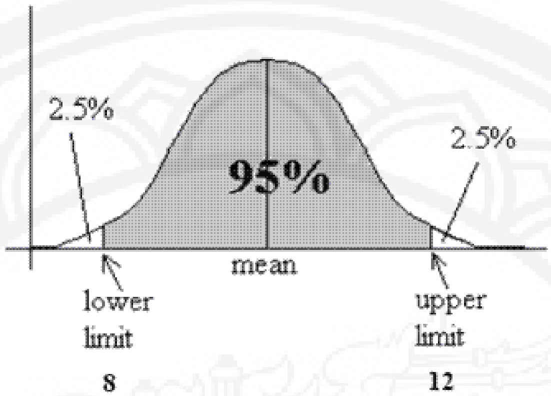


Figure 14 95% confidence level graphs

This research paper uses the technique above to design the flowchart and create shadow detection module, as in figure 15. The result, as shown in figure 16.

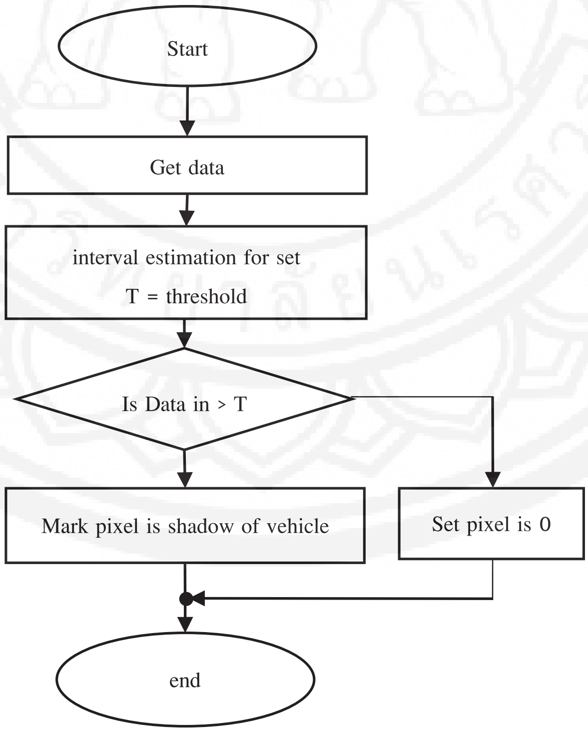
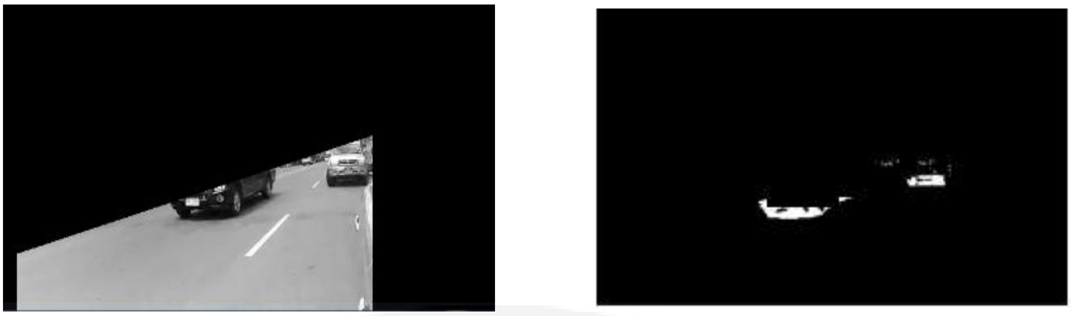


Figure 15 The flowchart of shadow detection module



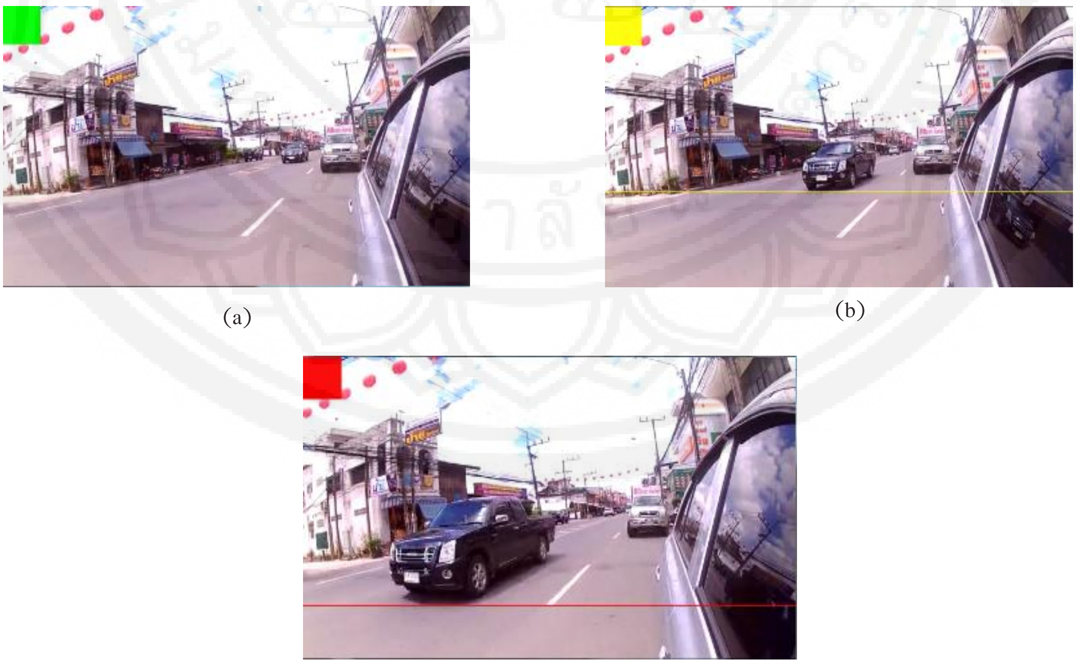
(a) (b)

Figure 16 (a) Output of pre-processing image module. (b) Output of shadow detection module

For detecting the vehicles. We use both results from Sobel operator and shadow detection. The result of the Sobel operator has been examined to determine that it is a vehicle. We calculate the direction locomotion of the object from the position of the object between the previous frame and the current frame. However, using shadow detection contributes to the increased accuracy of the system. The object that is detected will confirm that it is the vehicle when the object moves into the detection area and has the shadow.

Result

This system is tested on the domestic road. The dataset, which resolution at 1920x1080 @ 60fps receives from the camera in real-time. The cameras install on the side mirror and send data via HDMI port to the Zybo board for processing in BSDS. Notifications of BSDS divide into three levels, including 1) green means there are no vehicles in detection area, 2) yellow means there is a vehicle in the detection area that driver should be careful and 3) red means there is a vehicle near the driver's vehicle. The notification of BSDS in each level shown in figure 17.



(a) (b)

(c)

Figure 17 (a) green status (secure) (b) yellow status (beware) (c) red status (warning)

There are two aspects of calculating effectiveness of system including resources used to design BSDS as in figure 19 and accuracy measurement of BSDS for vehicle detection using statistical process called confusion matrix. Confusion matrix is result evaluation from correct work and program's mistake including three types as follows.

1) Accuracy is value telling accuracy of the program for vehicle detection calculated using equation 5.

$$ACC = \frac{T_p+T_n}{T_p+T_n+F_p+F_n} \quad (5)$$

2) Recall (True Positive Rate) is value telling ratio of correctness of the program for vehicle detection to total calculated using equation 6.

$$TPR = \frac{T_p}{T_p+F_n} \quad (6)$$

3) Precision is value telling correctness of the program using equation 7.

$$PPV = \frac{T_p}{T_p+F_p} \quad (7)$$

True Positive (T_p) is number of vehicles that system can detect and alarm when there are other vehicles in detection area.

True Negative (T_n) is number of vehicles that system cannot detect and did not alarm when there are no other vehicles in detection area.

False Positive (F_p) is number of vehicles that system alarm driver when there are no other vehicles in detection area meaning the system detected vector incorrectly. If there are a lot of False Positive, it means program made a lot of mistakes but did not cause damage.

False Negative (F_n) is number of times that system did not alarm driver when there are other vehicles in detection area. This case is the most dangerous as it can affect life of the driver. This case is the worst result of system's mistake.

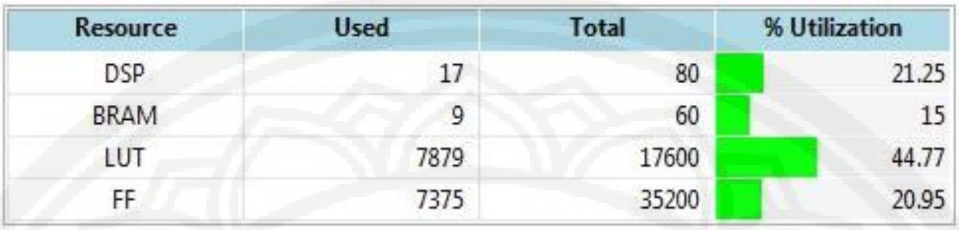
Using accuracy measurement above can measure accuracy of BSDS, which shows result on table 1 and table comparing with other papers that designed BSDS in table 2.

Table 1 The result of the measurement of the accuracy of the vehicle detection vehicle system

type of vehicle	T_p	F_n	F_p
motorcycle	30	0	-
vehicle	90	0	-
sum	120	0	8
TPR	**PPV**		**ACC**
100%	93.75%		93.75%

Table 2 The table compare measurement of the accuracy with other research

Methods	Chen and Chen	Cayir and Acarman	Proposed
Processing time	10 ms	16 ms	12 ms
TPR(%)	100	82.35	100
PPV(%)	79.70	77.78	93.75



Resource	Used	Total	% Utilization
DSP	17	80	21.25
BRAM	9	60	15
LUT	7879	17600	44.77
FF	7375	35200	20.95

Figure 18 the resources are used on the Zybo board

Discussion

This research paper presented BSDS on Zybo board. BSDS is designed by FPGA technology with image processing, which developed the system to choose different algorithms and tools from the research of both Chen and Chen (2009); Cayir and Acarman (2009). This research uses the grayscale, ROI to initially filter data and use Sobel operator, shadow detection for vehicle detection. SDSoC tool to help design BSDS based on FPGA technology. This research paper measures two parts of effectiveness, including working accuracy and processing time. There are two parts of the accuracy aspect, 1) recall which system could do 100% and 2) precision, the system could do 93.75%, which were better than the research of Chen and Chen (2009); Cayir and Acarman (2009) because we use two image-processing methods that work as conditions for vehicle detection. For processing time measured from total time used to process and result of the image in each frame, this research used 12 ms/frame which is better than research of Cayir and Acarman (2009) that used 16 ms/frame but longer than research of Chen and Chen (2009) which took only 10ms/frame. The test result showed that the system could perform well in both time and precision aspects to detect vehicles in blind spot. Although, designing a system using FPGA technology results in the fast processing system being able to operate in parallel. It has to be exchanged for the time to implement because it is very complicated to develop. However, our study had limitations. This system is not able to assess the behavior of the vehicle because we use not using machine learning. Therefore, machine learning is another technique that should add to BSDS for improving the efficiency of this system to learn the behavior of vehicles in the blind spot and predictions about the movement path of the vehicle. Will warn the driver or not.

Conclusion and Suggestions

This research paper presented tools and algorithm used to design BSDS which system divided it into two main modules, including 1) pre-processing image, module used to prepare data for processing and 2) image-processing is main module for processing to detect vehicles. The system was designed using SDSoC tool on

FPGA technology for the system to respond real-time. This research used resources of Zybo board which includes dps 80, Bram 60, LUT (look up table) 17600 and FF 35200 and measured 2 aspects of including accuracy for vehicle detection of 93.75% precision, 100% recall, 93.75% accuracy and for aspect of time, system could operate image resolution of 1920*1080 pixel/frame and process each frame within 12 ms. Suggestions for future work are to add the camera to the rear of vehicles and increase the function of predicting, such as the vehicle will move from the left lane to the right, speed up, and so forth by machine learning technology.

Acknowledgements

This work would not have been possible without the financial support of Department of computer engineering, Faculty of Engineering, Prince of Songkla University, Hatyai. Also, the author thanks Dr. Wannarat Suntiamorntut for help in in order to solve problems, the system can work effectively.

References

Anusha, G., JayaChandra, P., & Satya, N. (2012). Implementation of Sobel Edge Detection on FPGA. *International Journal of Computer Trends and Technology, 3*(3), 472-475. Retrieved from http://www.ijcttjournal.org/Volume3/issue-3/IJCTT-V3I3P127.pdf

Ajay, T. S., & Ezhil, R. (2016). Detecting Blind Spot by Using Ultrasonic Sensor. *International Journal of Scientific & Technology Research, 5*(5), 195-196. Retrieved from https://www.ijstr.org/final-print/may2016/Detecting-Blind-Spot-By-Using-Ultrasonic-Sensor.pdf

Baek, J. W., Lee, E., Park, M. R., & Seo, D. W. (2015). Mono-camera based side vehicle detection for blind spot detection systems. In *2015 Seventh International Conference on Ubiquitous and Future Networks, 7-10 July 2015* (pp. 147-149). Sapporo, Japan: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICUFN.2015.7182522

Cayir, B., & Acarman, T. (2009). Low cost driver monitoring and warning system development. In *2009 IEEE Intelligent Vehicles Symposium, 3-5 June 2009* (pp. 94-98). Xi'an, China: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IVS.2009.5164259

Chang, W. C., & Hsu, K. J. (2010). Vision-based side vehicle detection from a moving vehicle. In *2010 International Conference on System Science and Engineering, 1-3 July 2010* (pp. 553-558). Taipei, Taiwan: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/icsse.2010.5551779

Chen, C. T., & Chen, Y. S. (2009). Real-time approaching vehicle detection in blind-spot area. In *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC '09), 4-7 October 2009* (pp. 24-29). St. Louis, MO, USA: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ITSC.2009.5309876

Fernández, C., Llorca, D. F., Sotelo, M. A., Daza, I. G., Hellín, A. M., & Álvarez, S. (2013). Real-time vision-based blind spot warning system: Experiments with motorcycles in daytime/nighttime conditions. *International Journal of Automotive Technology, 14*(1), 113-122. https://doi.org/10.1007/s12239-013-0013-3

Jung, K. H., & Yi, K. (2018). Vision-Based Blind Spot Monitoring Using Rear-View Camera and Its Real-Time Implementation in an Embedded System. *Journal of Computing Science and Engineering, 12*(3), 127-138. https://doi.org/10.5626/JCSE.2018.12.3.127

Nong, M. A. M., Osman, R., Yusof, J. M., & Sidek, R. (2015). Real time motorcycle image detections on field programmable gate array. In *2015 IEEE Regional Symposium on Micro and Nanoelectronics (RSM), 19-21 August 2015* (pp. 1-4). Kuala Terengganu, Malaysia: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/RSM.2015.7354961

Poonam, A. K., & Gajanan, P. D. (2017). Image Processing Based Vehicle Detection and Tracking System. Retrieved from https://iarjset.com/upload/2017/november-17/IARJSET%2026.pdf

Song, K. T., Chen, C. H., & Chiu Huang, C. H. (2004). Design and Experimental Study of an Ultrasonic Sensor System for Lateral Collision Avoidance at Low Speeds. In *IEEE Intelligent Vehicles Symposium, 14-17 June 2004* (pp. 647-652). Parma, Italy: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IVS.2004.1336460

Sotelo, M. A., & Barriga, J. (2008). Blind spot detection using vision for automotive applications. *Journal of Zhejiang University SCIENCE A, 9*(10), 1369-1372. https://doi.org/10.1631/jzus.A0820111