A Hardware/Software Codesign for Image Processing in a Processor Based Embedded System for Vehicle Detection

Hosun Moon*, Sunghwan Moon*, Youngbin Seo*, and Yongdeak Kim*

Abstract: Vehicle detector system based on image processing technology is a significant domain of ITS (Intelligent Transportation System) applications due to its advantages such as low installation cost and it does not obstruct traffic during the installation of vehicle detection systems on the road[1]. In this paper, we propose architecture for vehicle detection by using image processing. The architecture consists of two main parts such as an image processing part, using high speed FPGA, decision and calculation part using CPU. The CPU part takes care of total system control and synthetic decision of vehicle detection. The FPGA part assumes charge of input and output image using video encoder and decoder, image classification and image memory control.

Keywords: Embedded System, ITS, Image Processing, Vehicle Detect

1. Introduction

In order to promote efficiency on limited road capacities and traffic utilities, some policies that make traffic condition uniform are necessarily required. The establishment of dedicated traffic information system eliminates factors that can obstruct the traffic flow or damage the traffic utilities. To accomplish these objects, it is important to convert traffic condition into useful information. In this regard, it is essential to detect vehicles by using image processing technologies. Nowadays most of installed vehicle detector systems use sensing technology and the sensors are laid under the road or roadside. Sensing technology detects the vehicles which use induction type detector sensor, supersonic wave sensor, microwave sensor and optic detector sensor. The sensing technology is normally used in the traffic fields but the sensing methods have some handicaps. It needs lots of human resources and cost to maintain the detecting sensors. It also needs complicated equipment that includes many sensors. However, it is possible to install the video-based vehicle detector system without damage on the road.

Moreover, it also has many advantages:

1) Simple installation methods
2) Moderate installation cost
3) Easy to maintain
4) Low-price detecting equipment

Despite its many advantages, vision-based vehicle detector system is sensitive to weather, shadow and background image. For this reason, recognition rate can be affected by those environmental factors and hence it is necessary to develop compensating algorithms.

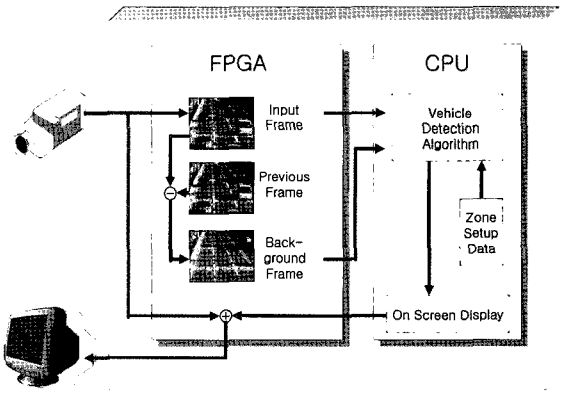


Fig. 1. Functional Block Diagram

In this paper, we propose a cost effective real-time video-based vehicle detector system consisting ARM7 micro-processor [3][4] and Field Programmable Gate Array (FPGA).

The paper is organized as follows. Section 2 describes system architecture and functions of each block. Vehicle detection algorithm based on background abstraction is presented in Section 3. In section 4, experimental results of the proposed system are shown. Finally, in section 5, we describe our conclusions and the directions for further study.

2. System Architecture

The proposed characteristics of the system are described below:

Manuscript received October 4, 2005; accepted November 8, 2005.
This work was supported in part by participation within the limits of the project "Key industrializetion technologies for ubiquitous computing systems on advanced vehicles" sponsored by IITA, Korea.

* Department of Electrical and Computer, Ajou University, Suwon Korea (who@ajou.ac.kr, rome097@hotmail.com, {kamanai, yongdkim}@ajou.ac.kr)

1) Real-time vehicle detection
2) Easy to install without construction
3) Enable to measure various traffic parameters

The main design objective [5] of the system is not only to achieve goal performance but also to obtain stable operation due to the fact that the system runs in the real fields.

The functional block diagram of system is shown in Fig. 1. It has two component parts such as CPU and FPGA. Current input frame is saved to the memory by FPGA. Then, the background frame is obtained by subtracting previous frame and current frame.

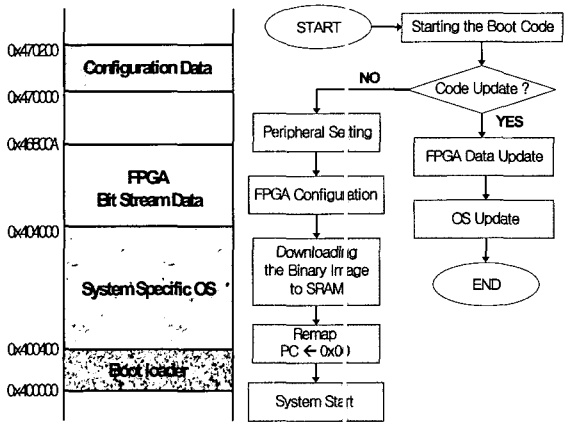


Fig. 2. Memory Map and Booting Sequence

CPU determines the detection of vehicle by comparing current frame image and background frame image and these images depend on the selected zone area which is controlled by system operator. CPU also saves display data to memory for an on screen display.

After reset, the bootloader program [2] is executed and the memory of the system is mapped [9].The Booting Sequence is shown in Fig. 2.

We implement FPGA in Xillinx's Spartan-II series, and it has five important features [6] such as, Input and output of images, save images to memory, Memory buses control, On Screen Display (OSD) and Update background images.

Fig. 3 shows FPGA interface scheme [8] among CPU, memory and NTSC CODEC. Input image from NTSC decoder enter FPGA synchronizes with decoder's clock signal. In order to display, FPGA passes through inputted signal to NTSC encoder. At the same time, input image data have to be saved into memory for image processing by CPU. A new image frame is started by VRESET signal of NTSC decoder.

Since transmitted input byte image must be saved per each clock, FPGA [6] creates memory address for each byte data in order to save it into memory. For this reason, FPGA initializes the memory address as soon as activated VRESET signal is detected, change memory bus into another space for image frame saving, and then it saves input data per every clock cycle by increasing memory address. The FPGA part of memory control has to handle 320*240 bytes per each image frame. The amount of data that must be processed mounts up to 4500Kbytes (320* 240*60) since image is inputted by 60 frames per second. Because it has a risk for data loss during real-time processing, we implement a dual image memory scheme for saving in order to stabilize CPU's processing time. For example, during FPGA save image data [6] into the first memory, CPU accesses the second memory. The same way, during FPGA saves image data into the second memory, CPU access the first memory. This dual memory scheme leads it to be more reliable for system performance. FPGA use these consequent two images in order to update background image by subtracting. This means that FPGA takes partial charge of simple heavy calculation. As a result CPU can concentrate on its calculation power to execute vehicle detection algorithm stably for selective zone.

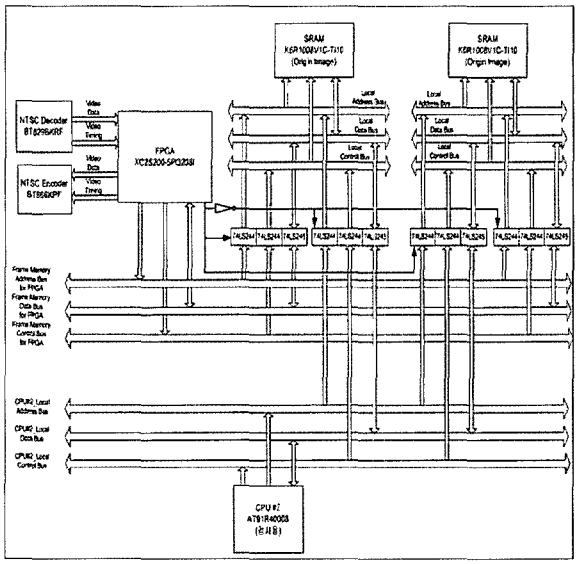


Fig. 3. FPGA Interface Scheme

Also, FPGA is in charge of On Screen Display. The system has two external interfaces; with PC by RS-232 and with an operator by PS2 mouse. These two interface methods enable to conFig. system settings. The first communication method is used for development. The second method is used for communication with an operator through a monitor which is connected to the system. This method will be used in the actual field. The system offers graphical communication functions to an operator by FPGA. In order to display communication message on the monitor, it marks with a dot on predefined position. The proposed system has separate On Screen Display memory that can be accessed by both CPU and FPGA. CPU updates display message into the OSD memory address corresponding with actual monitor pixel position (320*240) And FPGA sums up input NTSC data to OSD data for displaying various kinds of information on the monitor.

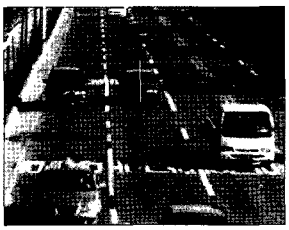
3. Vehicle Detecting Algorithms

When a vehicle enters the virtual rectangle zone which is

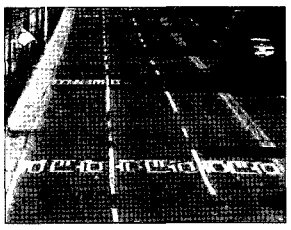
configured by operator for distinction, system can be recognized. When every frame of the system is monitored, the variation of pixel in each selective zone detects whether the vehicle is exist or not. If no vehicles exist in the selective zone, the variation of pixel is as same as the background. In an opposite case, the variation of pixels can be changed up or down in comparison with road background image. Therefore it is necessary to decide the lowest and highest threshold limits in order to prepare both dark and bright vehicles. These threshold ranges of values are obtained by using distribution range [10] while no vehicles are passed by. While pixel data abstracted in the selective zone is out of threshold range, it determines whether the vehicle enters into zone or not. It has two methods that compare the samples on the selective zone and threshold values. Firstly, it compares threshold values and pixel data by sequence. Secondly, it compares pixel data and the value of threshold limit [1]. We apply neural network theory for smooth updating of background image that depends on environmental parameters such as shadow and intensity of illumination. As the result of smooth updating, background image [1] is more applicable on various situations. The comparison of original background image with the obtained background image is achieved by subtracting consequent two images as shown in Fig. 4.

The detecting algorithm is described as below:

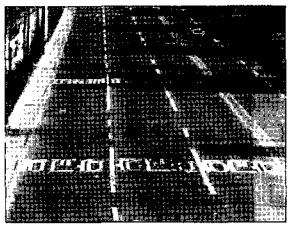
1) Normalize input image that represents each zone area into 32 pixel data.
2) Count pixel number that it is out of threshold range
3) If the count rate exceeds 25%, it is estimated as the line being occupied (vehicle flow)



(a) Input Image



(b) Original Background Image



(c) Obtained Background Image

Fig. 4. Background image comparison that calculated existing background image

Threshold value and occupied rate are not fixed value and they are variables in order to be adapted to the change of background image. When a new image is inputted, normalization is executed for 8 images. Background image is updated every 20 minutes. [7]

4. Experiment Results

In order to verify the proposed system, we tested the system. The input image use video tape for recoding actual road for testing. Fig. 5 shows the system environment.

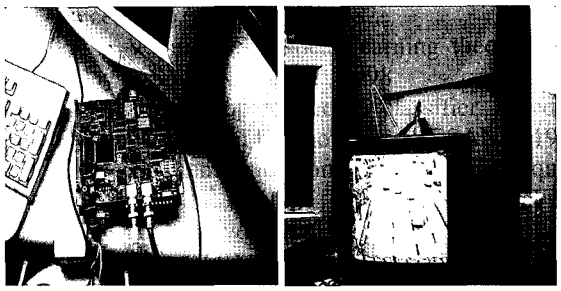


Fig. 5. Experimental Environment

Test conditions are as follows:

1) Input image: 320*240 pixels, 60 frame/sec
2) 8 selected zone per one frame
3) Period: 10 days, day time and night time
4) Number of tested vehicles: 100

The results are shown in Fig.s 6, 7and 8. As a result, detecting error rate is low under fine and cloudy weather conditions.

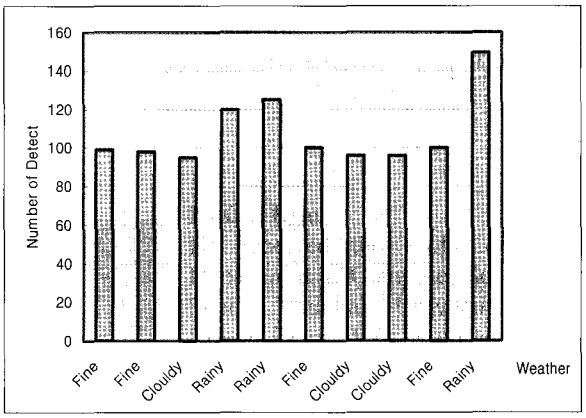


Fig. 6. Number of Detected Vehicles in Day time

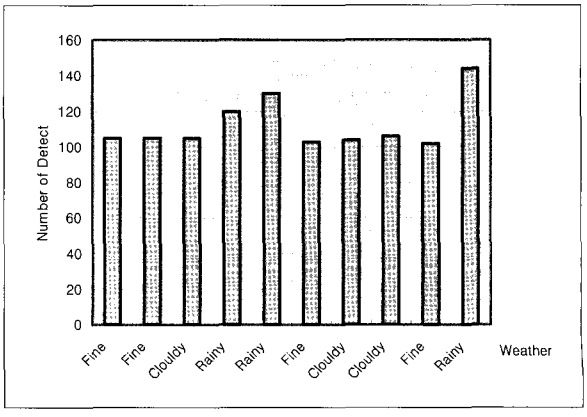


Fig. 7. Number of Detected Vehicles in Night time

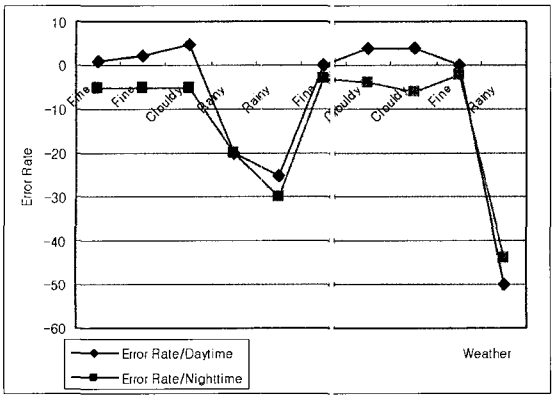


Fig. 8. Error Rate

It is observed that raining makes the error rate high. This result indicates that it is necessary to improve vehicle detecting algorithm under raining conditions.

5. Conclusions

In this paper, we propose a design for stable high-performance image processing system on Real- time. For high-performance, FPGA takes charge of pre-processing which abstracts the background CPU only calculates selective part of each frame while the FPGA calculates the whole part of each frame. According to our results, the system can calculate over 30 frame of 320*240 per second. We find that it could be possible for data loss if the system receives the image data from NTSC decoder directly. This is due to the fact that one pixel enters the system from camera module by 0.035 μsec, the system saves image form NTSC to image data memory by FPGA. CPU deals with not only direct image data from NTSC decoder but also image data of memory. In addition, FPGA saves input image data to two image data memory. Only one image data memory saves odd field image frame while the other save even field image frame from NTSC by FPGA. If FPGA saves odd field image in one memory, CPU treats image data in another memory. This can guarantee the stability of the image processing on a system. Also, if image processing algorithm requires more complex calculating process, FPGA can execute additional pre-processing for the algori-thm. This could improve the performance of the proposed system without changing any hardware part except the FPGA.

References

[1] Surendra Gupte, et al., "Detection and Classification of Vehicles", *IEEE Transactions on Intelligent Transportation Systems*, pp.37~47, iss.3, no.1, 2002

[2] Ed Sutter, *Embedded Systems Firmware Demystified*, CMP Books. 2002

[3] ATMEL, *AT91 ARM Thumb Microcontrollers Data Sheet*, 2002

[4] ARM, *ARM Architecture Reference Manual*, February 2000

[5] M.Schlett, "Trends in Embedded Micropro-cessor Design", *Computer, vol. 31, no. 8*,pp.44-49, Aug, 1998

[6] S. C. Chan, et al., "A programmable image processing system using FPGA"

[7] Lipton, A.J. et al., "Moving target classification and tracking from real-time video", *Applications of Computer Vision,* pp.8-14,1998. WACV '98. Proceedings., Fourth IEEE Workshop on, 1998.

[8] Douglas J. Smith, *HDL Chip Design*, Doone Publications, 1996.

[9] ARM, *ARM7TDMI Data Sheet*, Aug 1995

[10] Webber H C(ed), *Image processing and transputers*, IOS Press, 1992.

Hosun Moon
He received B.S. in industrial engineering from Ajou University in 1997, M.S. in electronics engineering from Ajou University in 2004. He is currently pursuing a Ph.D. at Ajou University, where a focus his has been in embedded software and Real-time OS and Intelligent Transport System.

Sunghwan Moon
He received B.S. in electronics engineering from Ajou University in 2004, He is currently pursuing a M.S. at Ajou University, where a focus his has been in image processing and embedded system.

Youngbin Seo
He received B.S. in electronics engineering from Ajou University in 2004, He is currently pursuing a M.S. at Ajou University, where a focus his has been in image processing and Real-time OS and embedded system.

Yongdeak Kim

He received B.S., M.S. and Ph.D. from De-partment of Electronics Engineering at Yonsei University, Korea, in 1971,1973 and 1976 respectly. From 1973 to 1974, he was a research member at Department of Electronics Ecole Superieur D'electricite in France. Frome 1970 to 1980, he was a visiting Professor of Department of Electronics Engineering at Standford University. Since 1974, he his been professor of Department of Electronics Engineering at Ajou University, Korea. His current research interests include Intelligent Transport System and Computer Network.