International Journal of Automotive Technology, Vol. 6, No. 2, pp. 171–181 (2005)

1229-9138/2005/020-11

EXTRACTION OF LANE-RELATED INFORMATION AND A REAL-TIME IMAGE PROCESSING ONBOARD SYSTEM

U. K. YI[1)] and J. W. LEE[2)*]

[1)]Department of Digital Animation, Busan Kyungsang College, Busan 611-701, Korea
[2)]Department of Industrial Engineering, Automotive Research Center, Chonnam National University, Gwangju 500-757, Korea

(Received 5 January 2004; Revised 18 May 2004)

ABSTRACT – The purpose of this paper is two-fold: 1) A novel algorithm in order to extract lane-related information from road images is presented; 2) Design specifications of an image processing onboard unit capable of extracting lane-related information in real-time is also presented. Obtaining precise information from road images requires many features due to the effects of noise that eventually leads to long processing time. By exploiting a FPGA and DSP, we solve the problem of real-time processing. Due to the fact that image processing of road images relies largely on edge features, the FPGA is adopted in the hardware design. The schematic configuration of the FPGA is optimized in order to perform 3 × 3 Sobel edge extraction. The DSP carries out high-level image processing of recognition, decision, estimation, etc. The proposed algorithm uses edge features to define an Edge Distribution Function (EDF), which is a histogram of edge magnitude with respect to the edge orientation angle. The EDF enables the edge-related information and lane-related to be connected. The performance of the proposed system is verified through the extraction of lane-related information. The experimental results show the robustness of the proposed algorithm and a processing speed of more than 25 frames per second, which is considered quite successful.

KEY WORDS : Onboard image processing unit, Image pre-processing, FPGA, DSP, EDF

1. INTRODUCTION

Recently, a large amount of work on the image processing of road traffic scenes has been done for intelligent safety vehicle (ISV), which aims to improve vehicle safety and may even provide for autonomous navigation. Although it is difficult to extract precise road information due to various noise sources, image processing provides familiar visual information to people (Ozawa, 1999). In order to adopt the image sensor to the ISV, the image processing system should process lots of data in real-time with a robustness and be constructed small enough in order to be loaded onto a vehicle.

In this paper, an EDF-based algorithm (Lee, 2002a; Lee *et al.*, 2003) capable of extracting lane-related information from road images and the design specifications of image processing hardware system implementing the algorithm are presented.

The structures for high-speed image processing hardware system for road images are largely classified into two types. One type is the parallel processing type with multiple processors. The other type is composed of two main processors, a low-level and a high-level processor, according to the processing steps of an algorithm applied to the hardware. The low-level processor's main role is to extract image features such as the edge. The high-level processor carries out recognition or decision using the image features from the low-level processor. Because the extraction of image features requires repeated operations of a single function, ASIC or an FPGA, which are good at repeated operation, are adopted in the design of the low-level processor. In this paper, our system has a similar structure to this latter type.

Examples of the parallel processing type system are ASSERT-2 system (Smith, 1995) and GOLD system (Bertozzi, 1996). In the ASSERT-2 system, for motion segmentation and object tracking in real-time, a VME board and two PowerPC601 are used. In this system, the VME board carries out image acquisition and image storing while the PowerPC601 takes on the role of feature tracking, segmentation and shape tracking. In the GOLD system, PAPRICA (PArellel PRocessor for Image Checking and Analysis) performs image processing at early stages and the SPARC station processes intermediate stages. Even though these systems realize real-time processing through parallel processing, miniaturization for

*Corresponding author. e-mail: joonlee@chonnam.ac.kr

practical use seems to be quite difficult.

The system composed of two main processors has been designed by several researchers (Hanawa *et al.*, 2001; Shin *et al.*, 2000; Kang *et al.*, 1999). Hanawa *et al.*, proposed a stereo image recognition system for ADA (active driving assist). They designed a system which consists of a stereo camera and an image recognition device. In their system, a 16-bit microcontroller was used for shutter speed control of the stereo camera. The image recognition device carried out fault diagnosis and communication. By using a gate array and two RISC chips, the image recognition device obtained a processing speed of 10 frames per second. Shin *et al.* implemented a real-time image processing board that made multi-resolution processing possible for an autonomous land vehicle. Their system was composed of an FPGA and a RISC. In their systems, image acquisition and pre-processing were implemented by using FPGA that could be reconfigurable. High-level image processing was also realized in real-time by a general-purpose RISC or DSP. As the system composed of two main processors is greatly affected by an algorithm in use, the system is generally utilized for special-purposes. Even though it is not easy to expand this system's applications, this type of structure has been the norm because it satisfies the requirements of miniaturization and optimal design.

The proposed hardware system in this study is composed of many components such as a line buffer, a memory for an LUT (look-up table), an FPGA, and a DSP. It has two main processors, one low-level and one high-level processor. The line buffer, the memory for LUT, and the FPGA constitute the low-level processor. The FPGA mainly calculates the gradient magnitude and orientation of the 3×3 Sobel edge (Faugeras, 1993) at high speed. Meanwhile, the DSP of the high-level processor performs reprocessing of low-level features for an advanced task. The proposed system is also designed to make up for the shortcomings of the PC-based image processing systems that have been used in the past. PC-based image processing systems are difficult to load onto vehicles and they are not efficient for the convolution operation, like edge detection. In Sobel edge detection for a 640×480 image, about 300 million multiplication and addition operations and 30 thousand tangent operations are required. These operations deteriorate the performance of the conventional PC-based image system. However, if these operations are implemented by an FPGA that can be reconfigured, and real-time processing can be realized through the parallel processing of a single function. We optimized the proposed system as a small-sized onboard unit to make it easy for loading onto a vehicle.

The proposed system uses Sobel edges in the extraction of lane-related information in real-time. While the Sobel edge operator has good detection capabilities, it has some trouble with localization. Nevertheless, the Sobel edge operator was selected in this research because the localization problem could be solved by using the non-local maxima suppression (Lee and Kweon, 1997; Faugeras, 1993). Furthermore, good detection is considered more important than good localization in the proposed EDF-based algorithm.

For the realization of the EDF-based algorithm to extract lane-related information, the following assumptions concerning lane marks on roads have been made: 1) Lane marks are painted in a brighter color than other parts on road surface; 2) Changes in the orientation of lane marks are small and smooth along the lane; 3) Lane marks are parallel to the left and right from the center of the lane. These assumptions can be thought of the properties of lane marks. Based on the assumptions, the EDF as a one-dimensional function is defined. Formulated by accumulating the edge magnitude of pixels of the same edge orientation in the region of interest, EDF plays four important roles: 1) It reduces the noise-related effects of a dynamic road scene; 2) It makes possible the measuring of a lane orientation without camera-related parameters; 3) It leads the lane-departure alert or lane-keeping problem to a mathematical problem by using its two important shape features local maxima and symmetry axis; 4) It enables the edge-related information and the lane-related information to be connected (Lee, 2002a).

The first two assumptions of lane marks indicate that the EDF has distinctive peak values at the locations corresponding to the orientation of lane boundaries. Accordingly, the local maxima of the EDF are estimates of the orientation of lane boundaries.

Performance verification of the proposed system was carried out through experiments to extract lane-related information in a road traffic image. It is well-known that image processing of road images is influenced by various noise sources such as an abrupt change of a light source, shadow, rainy weather, pseudo marks, and road features which makes road boundary information difficult to extract. Overcoming these noise effects requires many image features, which eventually leads to long processing time. Under these conditions, the proposed hardware system could extract the image features of an intensity and edge-related information such as the gradient magnitude, gradient orientation and sign of the gradients in real time.

2. SYSTEM CONFIGURATION

The goal of the image processing system pursued in this paper is the implementation of an onboard hardware system that can perform image processing algorithms for road images at high speed. In addition, the system should be loaded onto a vehicle. The architecture of the propos-

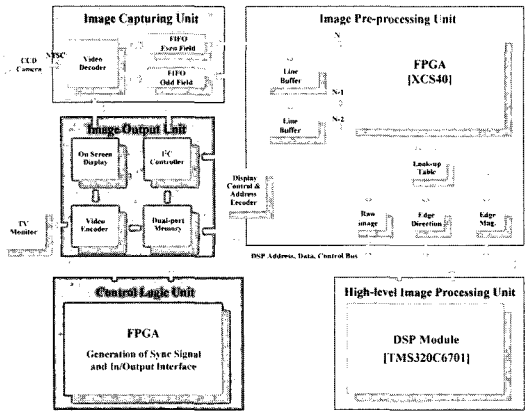


Figure 1. System architecture for a real-time image processing.

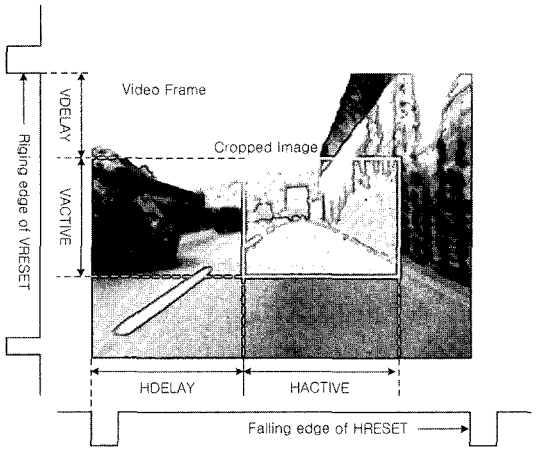


Figure 3. Setting up the region of interest.

ed system is configured as shown in Figure 1.

The system is composed of several units. An image capturing unit converts the analog NTSC signal from a CCD camera into a digital signal through a video decoder. A low-level image processing unit carries out the Sobel edge extraction, and a high-level image processing unit performs an image processing algorithm. An image output unit sends the processing results out to external display devices, and a control signal generation unit controls the flow of control signals.

2.1. Image Capturing Unit

A black and white CCD camera is used to take pictures. The output from the CCD camera, the NTSC analog signal, is converted into a digital signal in the image capturing unit which is made up of a video decoder, field memory, multiplexer, and control circuitry as shown in Figure 2.

The NTSC composite signal from the CCD camera is in interlace mode and the signal is transferred to the board by one coaxial cable. However, since the image signal is composed of vertical and horizontal synchronization signals in addition to the intensity signal, separation of these signals at high speed. In order to manage an analog composite signal with a signal level of

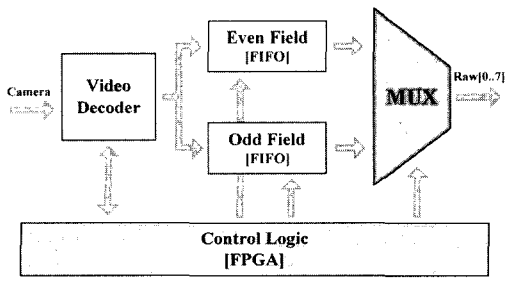


Figure 2. Structure of image capturing unit.

1.0Vpp (75Ω termination), a video decoder is required to digitalize the analog signal. When the video decoder is designed by using a general purpose A/D converter, the detection circuit of vertical and horizontal synchronization signals has to be included and an additional complicated circuit is also included because an A/D converter with a fast conversion speed is generally used. Moreover, if the circuits are not designed to exact specification, the captured image is not of a good quality. This can affect the whole image processing system. Therefore, the design of a video decoder using a general purpose A/D converter is carried out carefully (Johnson and Graham, 1993).

In this paper, a special-purpose video decoder, a Conexant BT829 decoder, was used for the composite signal processing. The chip produces a good quality captured image as well as being a simple design circuit. The control mode of the chip makes it possible to set up the various image sizes by hardware control. Accordingly, it is easy to fix or alter the region of interest. As shown in Figure 3, the chip can fix the region of interest by using VDELAY, VACTIVE, HDELAY and HACTIVE signals which are controlled by I^2C, that is, a bus-interface for data communication between DSP and video-decoder.

In order to form one frame of a standard image of 2:1 interlace, input-capture-unit has to alternately read line by line from even and odd fields. Two memories for the even and odd fields are used to store image data. This structure makes it possible to process field or frame-based processing at need. Even though a general purpose SRAM with the possibility of random access can be used as the filed memory to store digital images from the video decoder, a dual port FIFO memory was selected in this paper. This was done because the FIFO memory reads and writes image data independently, which leads to real-time processing. The image of the even field generated from the video decoder is stored in a high-rank FIFO

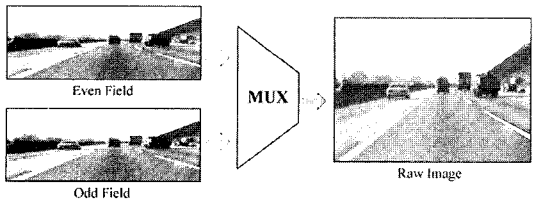


Figure 4. Function of a multiplexer.

memory and the image of the odd field is stored in a low-rank FIFO memory. Then, the data stored in the two field memories is read by a multiplexer as shown in Figure 4.

2.2. Image Pre-processing Unit

Generally, the issue with respect to the selection of an edge operator is whether the edge operator detects true edge and whether the detected edges lie at the exact location. Since it is hard to satisfy these two conditions at the same time, there is a trade-off between the detection of the true edge and exact location (Lee and Kweon, 1997). In this study, the 3×3 Sobel edge operator is selected based on the experiences of road image processing. This selection gives rise to the question about the localization issue because the Sobel operator provides edges distributed like a ridge at the boundaries of an object. An ideal edge operator detects edges only on the boundaries. However, most operators produce wide thick edges, so it is difficult to determine the exact location of the boundaries of an object from detected edges. In this situation, the edge pixel with the largest local gradient value is generally considered to be the true edge boundary. Using this basic idea, the localization problem can be solved. The non-local maximum suppression (Lee and Kweon, 1997; Faugeras, 1993) is a method which utilizes this idea. In addition, good detection is considered more important than good localization in our image processing algorithm for road images. Therefore, we use the Sobel operator as the edge operator.

As the unit, which has exclusive control of an edge operation, the image pre-processing unit is optimized for 3×3 Sobel edge operation. As shown in Figure 5, the pre-processing unit is made up of a line buffer, FPGA, and the memory for an LUT.

In this paper, pre-processing of the hardware for Sobel operation is implemented as follows: Image data stored in

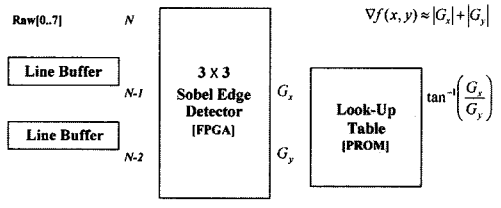


Figure 5. Structure of pre-processing unit.

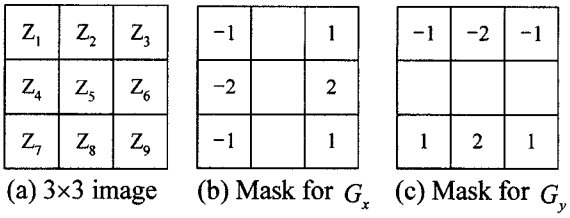


Figure 6. 3×3 Sobel edge operator.

even and odd field memories is read line by line in turn according to the output control signal for EVEN/ODD fields. As shown in Figure 5, the previous two lines, N-1 and N-2 on the figure, are read from field memories and transferred to line buffers. If the third line, N, is read from field memories, three bytes at a time from each line are fed into the FPGA according to the control signal from the FPGA and the 3×3 Sobel edge operation is performed. Figure 6 shows the Sobel operator. Using eight neighbors of Z_5 of Figure 6(a) G_x and G_y, which are vertical and horizontal gradients, respectively, are calculated as follows:

$$G_x = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7), \quad (1)$$
$$G_y = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3).$$

Two important physical quantities, gradient magnitude, $\nabla f(x, y)$ and orientation, $\alpha(x, y)$ are obtained by

$$\nabla f(x, y) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \quad (2)$$

$$\alpha(x, y) = \tan^{-1}(G_x/G_y). \quad (3)$$

Figure 7 shows the 3×3 Sobel edge detector of FPGA in detail. The edge detector consists of a staging register, a gradient calculator and a magnitude calculator. The edge detection is synchronized with a clock. If a certain nth clock is inputted, three pixels of 8bit each are inputted to the staging register, and at the same time, the data in each latch in the staging register is shifted. After n + 1th to n + 3th clocks are inputted, the gradients G_x and G_y are calculated using adders of 8 to 10 bits. Moreover, if n + 4th clock is inputted, the gradient magnitude is calculated and the gradients are transmitted to the LUT at the same time to obtain the gradient orientation.

As shown in Figure 8, the staging register made up of

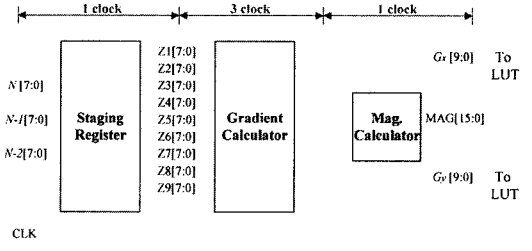


Figure 7. Structure of the Sobel edge detector.

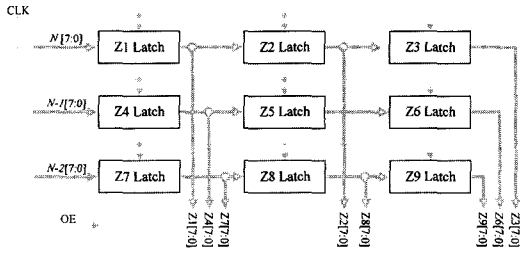


Figure 8. Structure of staging register.

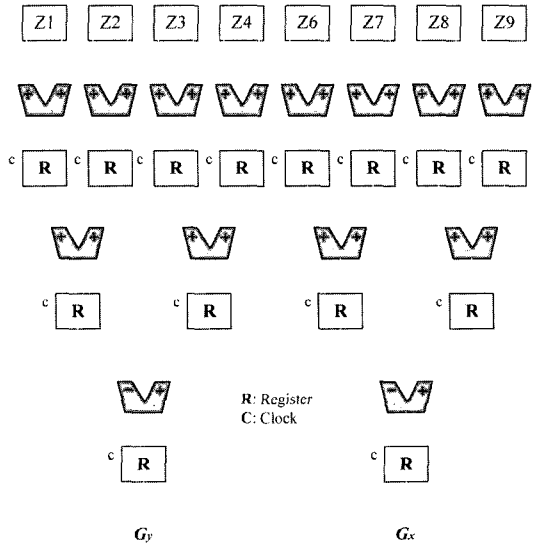


Figure 9. Structure of gradient calculator.

nine latches outputs eight data to the gradient calculator and shifts the eight data at the same time by synchronization of the clock. Since edge detection is impossible with the first two columns during the period of the first two clocks, the control signal OE (output enable) is added to output data from when the third clock is generated.

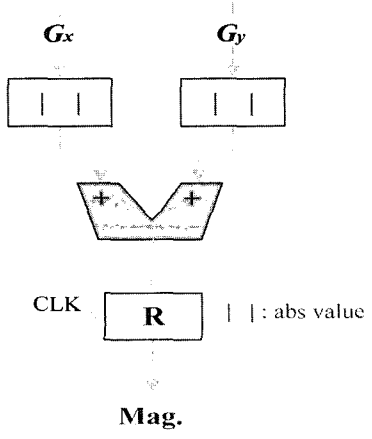


Figure 10. Structure of magnitude calculator.

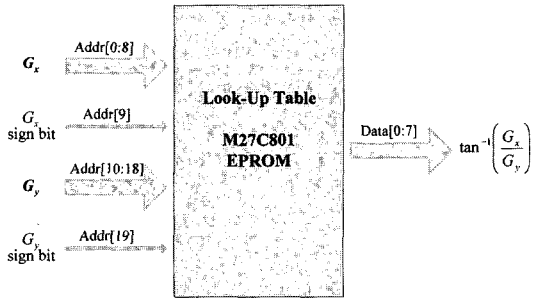


Figure 11. LUT for the gradient orientation.

The structure of the gradient calculator used to implement Equation (1) is shown in Figure 9. Registers are used to fix the timing in the output of intermediate results to the next stage of clock synchronization. Figure 10 shows the magnitude calculator.

Since the gradient orientation requires an arc tangent calculation that is impossible in one clock in the FPGA, we use the LUT instead of direct calculation. Given the LUT in advance, the gradient orientation is easily determined by taking an angle stored at the position corresponding to G_x and G_y in the LUT. The only problem is to determine the memory size of the LUT, which is related to the range of G_x and G_y, and the range of arc tangent. The maximum range of G_x and G_y is -1020 to 1020 and the range of arc tangent is dependent on the function used in the program which comprises the LUT. We use the function of atan of C language, which has a range of $-\pi/2$ to $\pi/2$. The LUT is constructed by setting the value of atan as an integer and the range of G_x and G_y as -512 to 512 to reduce the memory size by normalizing the original range. Then, the LUT has a memory size of 1 Mbytes. Figure 11 shows the flow of an LUT application.

2.3. High-Level Image Processing Unit

The high-level image processing unit controls the general actions of the system and takes charge of various calculations to implement an algorithm. The DSP used in this study, which is the key component of the unit, is the

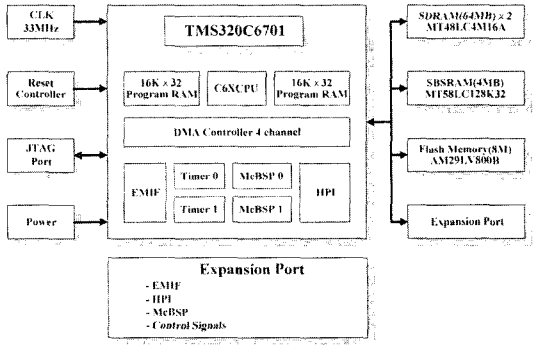


Figure 12. Structure of high-level image processing unit.

TMS320C6701, the floating point processor of Texas Instruments (Texas, 1998). This processor has a performing speed of 1,600 MIPS, and operates at a speed of four times as fast as 33 MHz (or 40 MHz) standard clock with 1.9 V low voltage. SBSRAM (synchronous burst static random access memory) and SDRAM (synchronous dynamic random access memory) were used for the memory. The SBSRAM stores the data of final results or intermediate results from a computation and the SDRAM stores the data needed in a large capacity format like an image, even though it is slower than the SBSRAM. Figure 12 shows the structure of the unit.

The DSP controls the flow of overall actions in the system. It initializes hardware components such as the video decoder, encoder, and OSD (on screen display) and also does the initialization of the actions of the system such as sizing of the input image, the setting of the processing region, and other variables. After initializing the actions, the high-level image processing unit implements an algorithm based on the intensity from the image capturing unit and the edge-related features such as the sign, magnitude, and orientation obtained from the pre-processing unit. Instead of engaging the intensity data transmission and edge-related features by the CPU, the DMA (direct memory access) increases the data transmission speed by letting peripheral equipment manage the memory-bus directly. By using four DMA channels, the DSP makes data exchange possible between internal memories or peripheral equipment and external memories. Various information from the pre-processing unit is transmitted through the DMA to the DSP's internal memory, which contributes to the real-time processing by reducing the number of cycles requiring access to the external memory.

2.4. Image Output Unit

The image output unit is used to monitor the captured image and the results from image processing. Displaying an image requires changing the image into a standard NTSC analog signal. As shown in Figure 13, reading and writing are carried out simultaneously by using the two memories, A and B, and the switching circuitry of the memory bank. While memory A is being read, memory B is being written. The opposite is also possible. BT864 of Conexant was selected as the video encoder and OSD was used for character output.

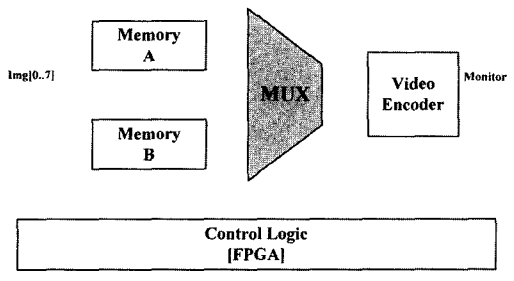


Figure 13. Structure of image output unit.

3. EDF

3.1. Region of Interest (ROI)

If a CCD camera is mounted on a test vehicle such that the optical axis coincides with the centerline of the car body, and roll and tilt angles are 0°, the vanishing point of the road images appears in the center of the vertical direction. Then, it is necessary to limit the processing area below the vanishing point because lanes visible in a road image generally lie in that area. In addition, we assume that there is no horizon or vertical lane in the images. In fact, there are no visible lanes in the horizon except for a steep curved road. Depending on the constraint of camera mounting, the ROI for image processing is confined within the shaded regions as shown in Figure 14. There are two purposes for the establishment of the ROI: 1) to reduce processing time 2) to highlight the ROI with respect to a traveling lane.

3.2. Edge Thresholding

The reasons why the edge is selected as the image-primitive in extracting lane-related information from road images are: 1) the most distinctive edge pixels appear along lane boundaries, 2) the edge itself has directional information that correlates with lane orientation.

While the pixels from lane boundaries in a road image have a large magnitude, their numbers are small compared with other pixels. Therefore, it is necessary to eliminate small magnitude pixels in order to raise the effect of pixels from lane boundaries. However, in an edge operation, choosing a threshold value for edge magnitude is difficult (Faugeras 1993; Lee 2002b). An adaptive method from (Lee *et al.*, 2001) was used to determine the threshold. Even though the method has no theoretical background, it makes a significant contribution because it eliminates the need for human inter-

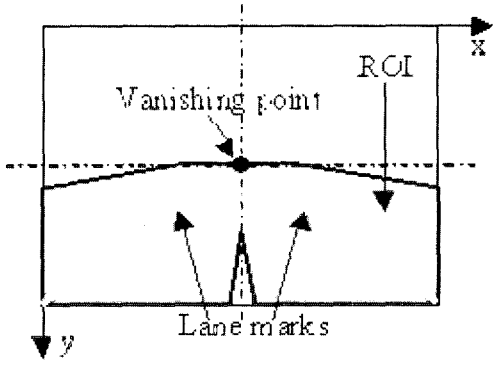


Figure 14. ROI for image processing.

vention.

3.3. EDF

Based on edge information and the three assumptions of lane marks on roads explained in Section 1, an EDF is defined with pixels remained after edge thresholding. In our former algorithm (Lee, 2002a), the EDF was defined as a one-dimensional function as follows:

$$F(d)=\sum_{n(d)} \nabla f(x,y) \tag{4}$$

where $n(d)$ is the number of pixels with orientation, $d = \alpha(x, y)$ of Equation (3), and $\nabla f(x,y)$ is the edge magnitude of Equation (2). To obtain $n(d)$, we set the range of $\alpha(x, y)$ as 0° to 180° and use a quantization of 1°. The EDF is the histogram of the edge magnitude of pixels with respect to the orientation. In this paper, the EDF of Equation (4) is slightly modified into

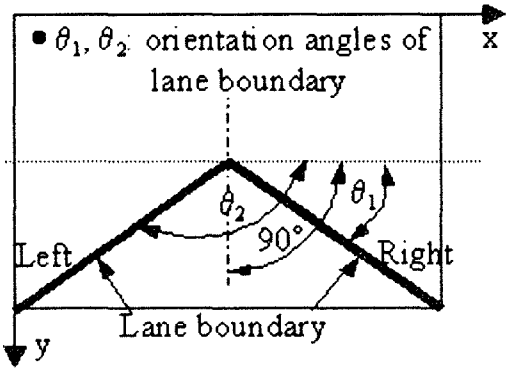
$$F(d)=\sum_{n(d)} \omega\nabla f(x,y) \tag{5}$$

where ω is a weighting factor defined by

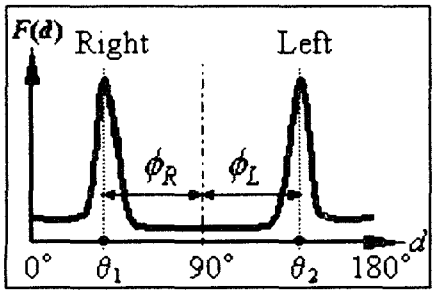
$$\omega=\begin{cases}\omega_1 & \text{if } (x, y) \in S \\ \omega_2 & \text{elsewhere}\end{cases}, \quad \omega_1 >> \omega_2,$$

ω_1 and ω_2 are determined experimentally. The new EDF of Equation (5) raises the effects of pixels belonging to the set, S, which is constructed by an LBPE (lane boundary pixel extractor). In this paper, we exploited the LBPE introduced in (Lee *et al.*, 2003) in order to extract pixels expected to be on lane boundaries.

In a road image, the pixels from lane boundaries have larger edge magnitudes than pixels from other road areas, and almost a similar orientation, but other parts do not have such characteristic. Therefore, the EDF has a rather large value in the vicinity of lane directions. This causes the EDF to have local maxima near the directions θ_1 and θ_2 of the right and left lane boundaries in the road images. In the extraction of lane-related information, the EDF does not need any information related to camera parameters. The graphical form of an EDF is ideally shown in Figure 15(b) for the orientation angles of lane boundaries shown in Figure 15(a). A symmetry axis of the EDF is generally located near .



(a) Orientation angles of lane boundaries



(b) EDF

Figure 15. Lane boundaries and EDF.

4. HARDWARE IMPLEMENTATION AND APPLICATION

4.1. Hardware Implementation

Figure 16 shows the implemented hardware for the high-speed image processing device. The high-level image processing unit was modularized and the rest of the parts were implemented as a structure contained within one board. This system can be applied to the input of three cameras at the same time in order to extend to stereo image processing. Moreover, because the synchronization of a point in time when images are captured is important in stereo image processing, the system was made to control the sync signal of cameras directly. Basic debugging was performed through a JTAG port and a monitoring program was made for the experiment.

4.2. Application of the System to Extracting Lane-Related Information

In order to evaluate the performance of the implemented

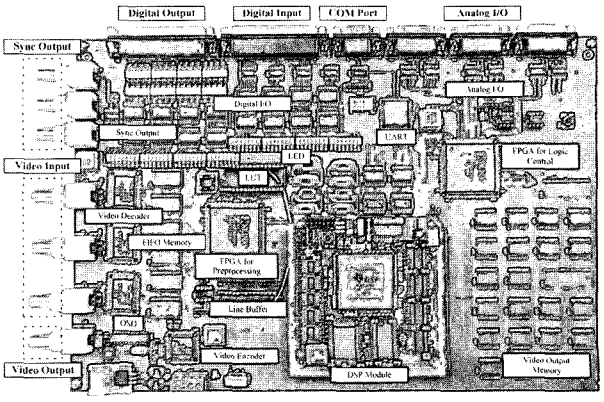
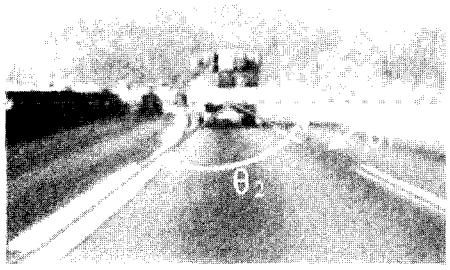


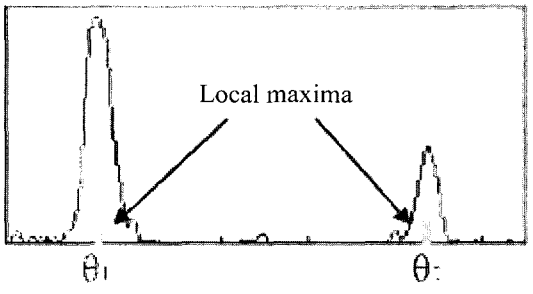
Figure 16. An onboard system implemented for real-time image processing.



(a) Raw image and extracted information

(b) Gradient magnitude of edges

(c) Gradient orientation of edges



(d) EDF

Figure 17. Extraction of lane-related information.

hardware, we carried out experiments with an algorithm to extract lane-related information. Figure 17 shows the results of the algorithm by the hardware shown in Figure 16. Figure 17(a) presents a captured raw image and extracted information superimposed on the image, while figures (b) and (c) show the magnitude and direction of edges produced by the pre-processing unit, respectively. Figure 17(d) shows an EDF and its local maxima, θ_1 and θ_2, which are corresponding to the direction of both lane-boundaries of a lane. The EDF was constructed by using the edge features of Figure 17(b) and (c) in the high-level

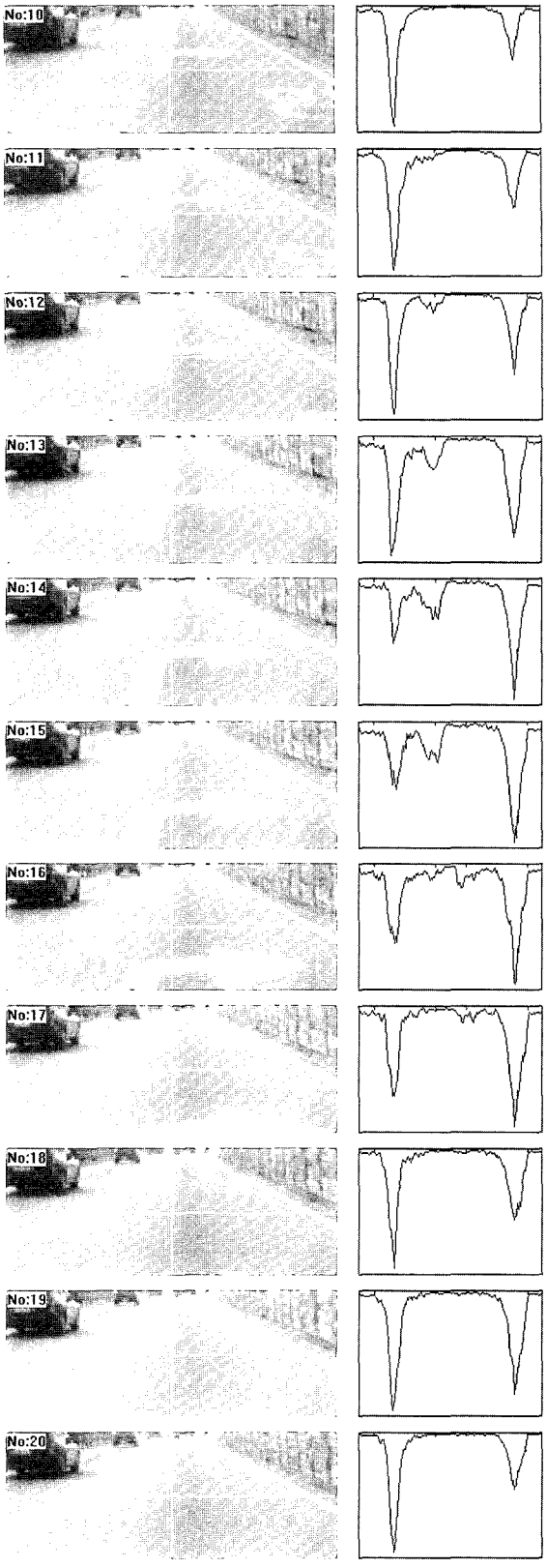


Figure 18. Raw images and their EDFs for daytime sequential images.

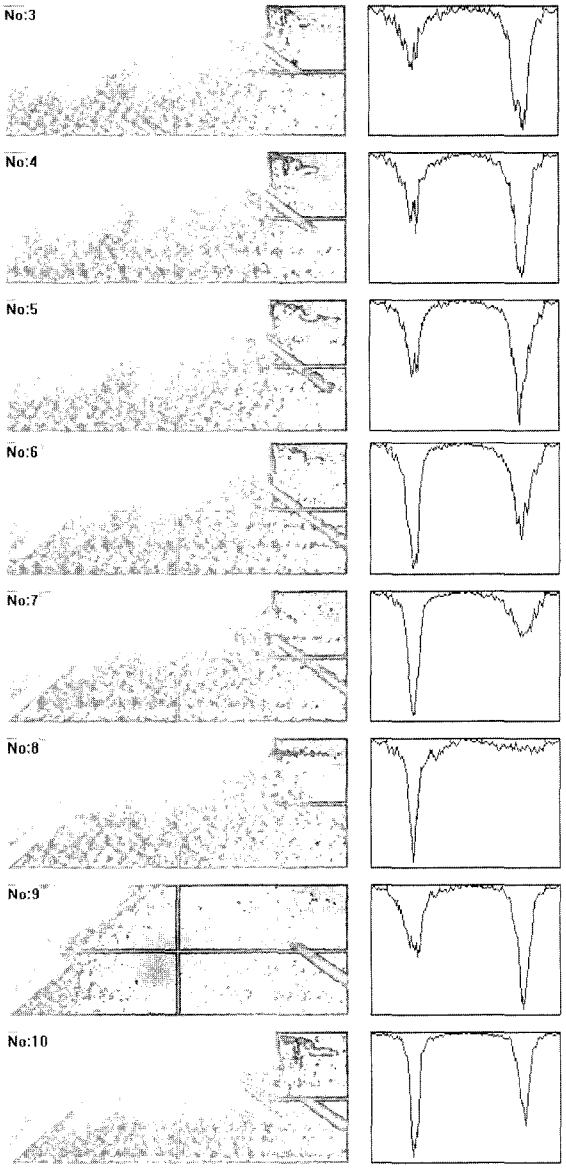


Figure 19. Raw images and image processing results in a rainy day.

image processing unit shown in Figure 12. These experimental results prove that the implemented hardware can provide fine images, reliable edge features, and lane-related information.

The next two experimental results shown in figures 18 and 19 show that the proposed system could be applied in various weather environments such as on a fine day and a rainy day. In the figures, the left column represents the captured raw images and the right column illustrates the EDFs calculated by Equation (5). The most remarkable point in the two figures is that the EDF-based algorithm proposed in this paper performed well with robustness in a noisy environment. Furthermore, the performance is almost the same in both weather environments.

4.3. Analysis and Consideration

As the system is initialized, the video decoder outputs digital images and stores them in the FIFO memories according to the DSP's commands. As shown in Figure 20, it takes about 33.3 ms from capturing an image to storing the image in the FIFO memory. Essentially, the data stored in the FIFO memory is the region of interest for the EVEN/ODD field. After the image is stored in the FIFO, the FPGA waits for the DSP's command to input the image into the FPGA. The waiting time is 4.1 ms. After the FPGA of the pre-processing unit receives the order from the high-level image processing unit, it takes the essential role of generating the control signals for the video decoder and the FIFO memories.

The high-level image processing unit keeps the processing pace with the FPGA of pre-processing unit. The DSP starts to read the pre-processing results such as intensity and edge-related information when the 60th line of the ODD field is written to the FIFO memory (marked A at Figure 20) and brings them into the memories for high-level image processing. The number 60 was chosen based on experience. Because the FIFO memory has a dual port memory structure, it can read and write image data independently. However, reading and writing cannot occur at the same address at the same time, so it is asked to have an address space of reading and writing of more than approximately 600 pixels apart according to the data sheet. Actually, if this space (marked B in Figure 20) is reduced, processing time would be reduced.

The algorithm used to evaluate the proposed system is sequentially processed as follows: construction of the EDF, search for the local maximum point (LMP), collection of edge pixels with the gradient orientation corresponding to LMP to form a scatter diagram, and obtaining lane-related information through a line fitting to the scatter diagram (Lee *et al.*, 2001). Experimental results are transferred to the memories for the image output unit and outputted to a monitor. As shown in Figure 20, conversion time of edge operation from the image pre-processing unit is about 17.7 ms. The processing time of an algorithm from reading the results of pre-processing to displaying the extracted lane-related information is about 14.6 ms. This analysis shows that it takes 37.4 ms to process the image data of one frame of 320×240, therefore, more than 25 frames per second can be processed.

Generally, a reconfigurable computing system can greatly reduce processing time by hard-wiring the part requiring a long processing time by using its reconfiguration power. In this paper, we used Spartan XCS40 of Xilinx FPGA with 784 CLBs in order to solve the problem of real-time processing of the Sobel edge operation.

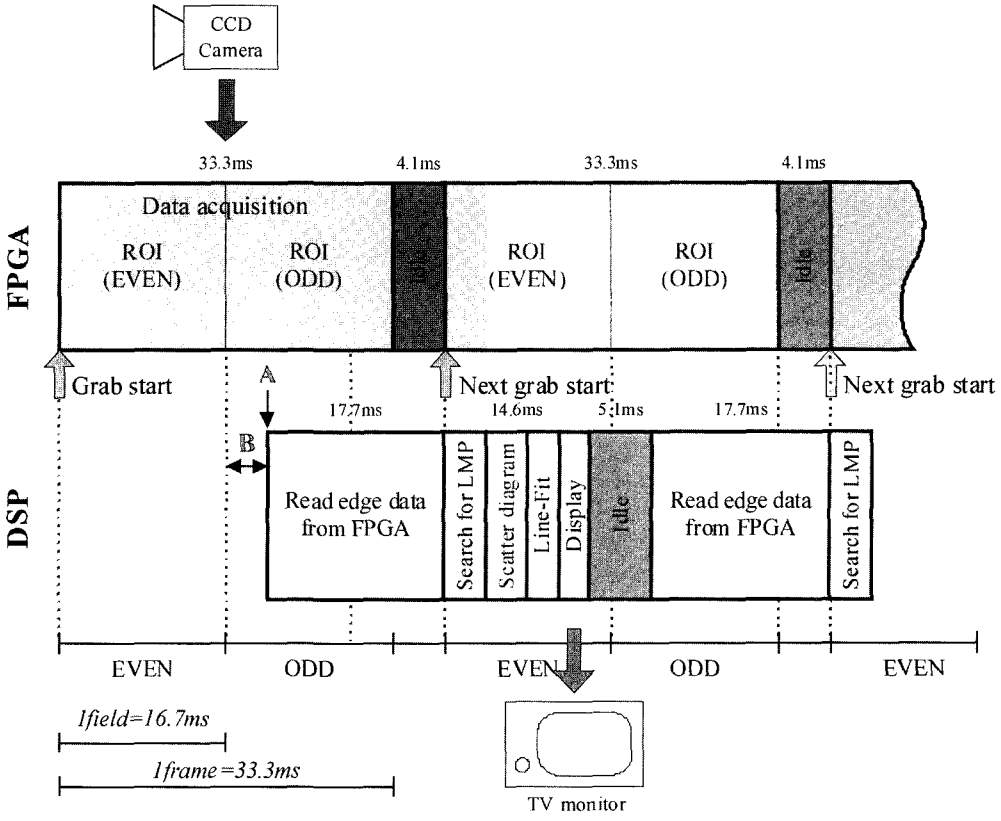


Figure 20. Timing analysis of the onboard image processing system.

The Sobel edge operation was coded and synthesized by VHDL (Xilinx, 2000; Smith, 1996). Consequently, the total number of CLBs used for the Sobel operation, video decoder, and FIFO control are 169, which is 21% of the total.

5. CONCLUSIONS

The contributions of this paper are two-fold: 1) to provide a miniature image processing onboard hardware system capable of performing low-level image processing to high-level image processing; 2) to supply a concise algorithm capable of extracting lane-related information.

In terms of hardware, this paper showed the design specification and structure of an onboard system to implement an image processing algorithm at high speed. Considering that edge-related features from the Sobel edge operation provide efficient cues for image analysis, we designed the pre-processing unit to be able to extract the edge features in real-time by using a FPGA and LUT. Meanwhile, the high-level image processing unit was designed to include DSP and peripheral memories. The size of the onboard system was optimized to make it easy for loading onto a vehicle. The proposed hardware system is not limited only to road traffic image processing. It is possible to apply it directly to various fields in need of real-time processing based on various image information such as intensity and edge-related features. Additionally, the system was designed for general purposes to make it possible to input three CCD cameras simultaneously and is applicable to stereo and multi-view images.

In order to evaluate the performance of the proposed image processing system, an EDF-based lane-related information extraction algorithm was applied. The experimental results showed that the proposed hardware system could conduct image processing at a rate of more than 25 frames per second, and the proposed algorithm performed well even in noisy environments without human intervention and camera-related parameters.

ACKNOWLEDGEMENT–This work has been supported by grant no. 505-2004-000-16067-0 from the Korea Science & Engineering Foundation.

REFERENCES

Bertozzi, M. and Broggi, A. (1996). Real time lane and obstacle detection on the GOLD system. *Proc. IEEE Intelligent Vehicles '96*. 213–218.

Faugeras, O. (1993). *Three-Dimensional Computer Vision*

A Geometric Viewpoint. The MIT Press. England.

Hanawa, K. and Sogawa, Y. (2001). Development of stereo image recognition system for ADA. *IEEE IVS2001.* 177–182.

Johnson, H. and Graham, M. (1993). *High-Speed Digital Design.* Prentice Hall. New Jersy.

Kang, HI. Ju, Y. W. and Baek, K. R. (1999). An implementation of the high speed image processing board for contact image sensor. *J. Control, Automation and Systems Engineering* **5**, **6**, 691–697.

Lee, J. W. (2002a). A machine vision system for lane-departure detection. *CVIU* **86**, **1**, 52–78.

Lee, J. W. (2002b). A fuzzy neural-network algorithm for noisiness recognition of road images, *Trans. Korean Society of Automotive Engineers* **10**, **5**, 147–159.

Lee, J. W. and Kweon, I. S. (1997). Extraction of line features in a noisy image. *Pattern Recognition* **30**, **10**, 1651–1660.

Lee, J. W. Kee, C. D. and Yi, U. K. (2003). A new approach for lane departure identification. *Proc. IEEE IV03.* 100–106.

Lee, J. W. Kim, K. S. Jeong, S. S. and Jeon, Y. W. (2000). Lane departure warning system: Its logic and on-board equipment (20005331). *Proc. JSAE. Japan.* 9–11.

Lee, J. W. Yi, U. K. and Baek, K. R. (2001). A cumulative distribution function of edge direction for road-lane detection. *IEICE Trans. Information and Systems.* E84-D. 9. 1206–1216.

Ozawa, S. (1999). Image processing for Intelligent Transport Systems. *IEICE Trans. Information and Systems.* E82-D. 3. 629–636.

Shin, C. W. and Kim, K. I. (2000). Design of real-time image processing board with multi-resolution processing method for ALV. *ITS2000.* 1–8.

Smith, D. J. (1996). *HDL Chip Design.* Doone Publications. Madison. AL. USA.

Smith, S. M. and Brady, J. M. (1995). ASSET-2: Real time motion segmentation and shape tracking. *IEEE Trans. on PAMI* **17**, **8**, 814–820.

Texas Instruments (1998) TMS320C6701 Floating-Point Digital Signal Processor. Texas Instruments Ltd.

Xilinx. (2000). The programmable logic data book 2000. Xilinx. Inc. 2000. E. Beuville, K. Borer, E. Chesi, E.H.M.