The Design of the IP with high efficiency for IoT using Data Separation Prewitt Edge Detection

Chang-hee Park[1] , Jin-sung Jung [1]
and Sang-Bock Cho [2]
[1] Department of Electronic Engineering, Ulsan University
93 Daehak-ro, Nam-gu, Ulsan 680-749,Republic of Korea
[2]Department of Electronic Engineering, Ulsan University
93 Daehak-ro, Nam-gu, Ulsan 680-749,Republic of Korea
E-mail: [1] pc0pc@naver.com, [2] sbcho@ulsan.ac.kr

Abstract: In this paper, in order to implement the IP of edge detection algorithm for surveillance in image using the camera in CCTV or vehicle black box, we designed pre-processing step that is the edge extraction algorithm. First, after input image converts into the input signals of R, G, and B, three inputs are combined, and converted to gray scale. Also, these data is divided in half and apply to each Prewitt edge[1] based on X, Y axis through 3 line buffer, the output image is determined edge image through a combination of two result. A proposed algorithms was implemented using matlab program. This was converted and comfirmed to verilog through xilinx ISE

1. Introduction

Recently, human life are becoming more and more convenient, tanks to the development ICT(Information Communication Technology), new technology that combines the ICT in a variety of industries are being developed. Especially among technologies such as ITS (Intelligent Transportation System) and IOT (Internet of Things) is actively conducted research around the image processing system, Using image processing and image information are investing a lot of manpower and cost to the IP core development. This image processing system is a pre-treatment step of the process is important to provide an accurate and clear image information.

In this paper, we designed an edge detection algorithm in order to implement a preprocessing step for obtaining the image for security in IP. A proposed algorithms was implemented using matlab program. This was converted and comfirmed to Verilog through xilinx ISE

2. Main Subject

Figure 1 is block diagram for Edge detection
According to Block diagram, this processing is divided into input Data separation buffer line Priwitt filter output five-step process .
The most important process during this process is data separation process(3rd process). This process was in the reference machine data transfer method.
Machine data transfer method , there are a variety of ways such as FIFO, FILO, parallel transfer, serial transfer, and etc, refer to the parallel transfer method.

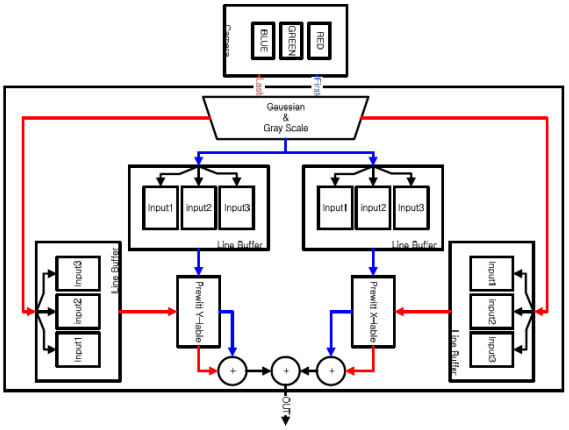

Figure 1.The block diagram of algorithm for edge extra

2. 1 Input data

The original image is converted to grayscale and Deteriorated for noise cancellation to RGB input of gausian filter.

2.1-1 Gaussian filter

Using Gaussian filter for noise cancelation is described as a canny edge detection
The Canny algorithm uses an optimal edge detector based on a set of criteria which include finding the most edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization, and marking edges only once when a single edge exists for minimal response[2].
According to Canny, the optimal filter that meets all three criteria above can be efficiently approximated using the first derivative of a Gaussian function.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

$$\frac{\partial G(x,y)}{\partial x} \alpha\, x e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \frac{\partial G(x,y)}{\partial y} \alpha\, y e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

The first stage involves smoothing the image by convolving with a Gaussian filter. This is followed by finding the gradient of the image by feeding the smoothed image through a convolution operation with the derivative of the Gaussian in both the vertical and horizontal directions. The 2-D convolution operation is described in the following fomula 3.

$$I'(x,y) = g(k,l) \otimes I(x,y)$$
$$= \sum_{k=-N}^{N} \sum_{l=-N}^{N} g(k,l) I(x-k, y-l) \quad (3)$$

where: g(k,l) = convolutional kernel
I(x,y) = original image
I'(x,y) = filtered image
2N + 1 = size of convolutional kernel

Both the Gaussian mask and its derivative are separable, allowing the 2-D convolution operation to be simplified.

This optimization is not only limited to software implementation, but also applies to hardware implementation as well.[4]

2.1-2 gray scale

Grayscale conversion is the simplest method of method of reducing the capacity of the video image.

$$gray = (Red + Green + Blue) / 3 \quad (4)$$

2. 2 Data separation

Image area separation of the above and below pressure along the Formula 5

$$\sum_{i=0}^{n} K(i) \quad benchmark\ \boldsymbol{m}$$
$$= \sum_{i=0}^{m} K(i) + \sum_{i=m+1}^{n} K(i) \quad (5)$$

Figure 2 is a schematic view showing the Formula 1. Schematized has the x-direction and y - direction. Among them, only the direction of x-direction shows. Existing data transfer method fills the 3 pixel by one pixel to be moving in one direction. The convolution these pixels with the filter mask for filtering

Separated system has from two directions above and below the pixel extraction. So it merges meet at the reference point. According to the theory, the system has the advantage that the transmission time is shorter than the existed system process near half time.

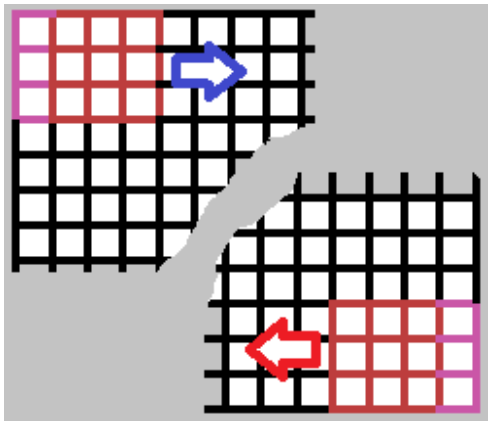

Figure 2. Schematized the separation system

2. 3 buffer line

Using 3 line buffer receives the converted value suitable easier filtering use to 3x3 matrix as shown in the figure 2.

This figure is a description which the receiving image data received 3pixel / clock instead of receiving the image data received by 1pixel / clock. Therefore, it is possible to efficiently receive giving the next image data.

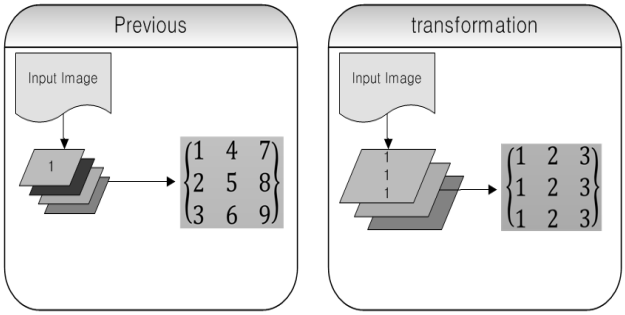

Figure 3. 3buffer line flow

2. 4 Prewitt filter

Edge detection is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation.

This process detects outlines of an object and boundaries between objects and the background in the image. An edge-detection filter can also be used to improve the appearance of blurred or anti-aliased video streams.

The basic edge-detection operator is a matrix area gradient operation that determines the level of variance between different pixels. The edge-detection operator is calculated by forming a matrix centered on a pixel chosen as the center of the matrix area. If the value of this matrix area is above a given threshold, then the middle pixel is classified as an edge.

Examples of gradient-based edge detectors are Roberts, Prewitt, and Sobel operators.

All the gradient-based algorithms have kernel operators that calculate the strength of the slope in directions which are orthogonal to each other, commonly vertical and horizontal. Later, the contributions of the different components of the slopes are combined to give the total value of the edge strength.

$$K_X = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * A \quad (6)$$

$$K_Y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * A \quad (7)$$

$$Result = |K_X| + |K_Y| \quad (8)$$

The Prewitt operator measures two components. The vertical edge component is calculated with Kx (X-axis direction)(7) and the horizontal edge component is

calculated with Ky(Y-axis-direction (8). It, the sum of the absolute values between the two kernel, gives an indication of the intensity of the gradient in the current pixel.

According to Prewitt operator, Accept the pixels, out from each of the buffer line, is calculated by X-direction and Y-direction, It is respectively calculated through Formula 6, 7 and 8 to derive the results.[1][3]

2. 5 Output

The incoming pixels are stored in memory. This memory size is required as the total number of pixels.

All results are added together at the adder tree to produce the filter middle point output result. [4]
Typically, scaling is applied at the final output.

3. Experimental Result

Figure 4 shows the results of a RTL-level verification in ISE 14.3. Figure 5 can check the source image and results of the edge detection. Table 1 is compared process area size and The time for processing between previous methods and the present method

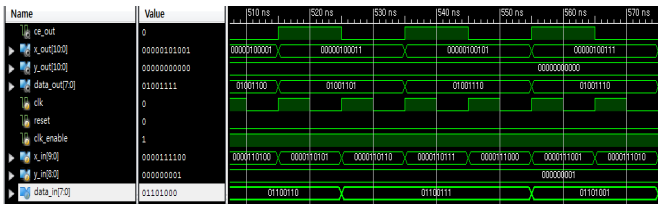

Figure 4. The simulation result of RTL-level

Figure 5. The original image and result image

Table 1. Previous methods and Present method

	Previous	Present
Area	13668.178	14128.14
Process Time	13.79	7.9

4. Conclusion

In this paper, it designed to implement to IP Edge detection algorithm to process at a faster rate of the pre-treatment step to obtain an image for security. Input image converted into the input signals of R, G, and B, three inputs are combined, and converted to gray scale. Also, these data is divided in half starting from top and bottom and apply to each Prewitt edge based on X, Y axis through 3 line buffer, the output image is determined edge image through a combination of two result.

The algorithm was validated by RTL-level through the ISE 14.3 and proved to be the latest method faster than exited methods.

It also plans to draw them into FPGA soft IP

5. Acknowledgement

This work was supported by the Industrial Core Technology Development Program (10049009, Development of Main IPs for IoT and Image-Based Security Low-Power SoC) funded by the Ministry of Trade, Industry & Energy

References

[1] E. Argyle, “Techniques for edge detection,” Proc. IEEE, vol. 59, pp. 285-286, 1971.

[2] J. F. Canny,“A computational approach to edge detection”, IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 6, pp. 679-697, 1986.

[3] J. Canny, “Finding edges and lines in image”. Master’s thesis, MIT, 1983.

[4] Hong Shan, “ Adaptive Edge Detection for Real-Time Video Processing using FPGAs” , Neoh Asher Hazanchuk ,2010