

IMPLEMENTATION OF FAST DOUBLE PARALLEL IMAGE PROCESSING USING SORT OPTIMIZATION ALGORITHM

¹Sawant Sapna Sham & ²Roopa M

Dept. of Electronics & Communication, Dayananda Sagar College of Engineering Bangalore, India
E-mail : ¹sapnasawant12@gmail.com, ²surajroopa@gmail.com

Abstract – Image processing plays a vital role in many applications like automation, visual inspection. Thus the Image processing system should meet the real time demand. Many factors like salt and pepper noise will affect the image processing system. This paper introduces the double parallel architecture using sort optimization algorithm to accelerate the median filter and edge detection. The double parallel architecture is divided in to two parts image level parallel and operation level parallel. The image level parallel is a high level parallel, which divides one image in to two parts and process them simultaneously.

Keywords – double-parallel, image filter, edge detection, median filter.

I. INTRODUCTION

Image filtering and edge detection are important steps in the field of image processing. Many algorithms are proposed to speed-up and parallelize the median filter and edge detection method. But many edge detection method lack the image level parallelization. This paper introduces implementation of efficient algorithm for image filter and edge detection. To overcome this problem double parallel architecture is introduced which will speed-up the image processing system. Double parallel architecture is divided in to two parts image level parallel and operation level parallel. The experimental result shows the double parallel architecture is very suitable for accelerating high image processing task and meets the real time demand in many application such as intelligent vehicles, robotics vision system[10], [11]. The section two describes the median filter and edge detection methods. Section three represents the whole system. And in section four experimental results are shown.

II. OVERVIEW OF THE SYSTEM

Here the architecture contains two levels: image-level and operation-level parallel. The image-level parallel is a high- level parallel, which divides one image into

different parts and processes them concurrently. The operation-level parallel is inside each thread of image-level parallel, which fully explores every parallel part inside the concrete algorithms. The following subsections sequentially introduce the implementation of the whole system and main modules in the system. Imaging processing modules in figure 1 respectively process different parts of input data in parallel, which are the image-level parallel.

The image-level, which divides an input image into two parts. The two parts are respectively transferred to their own image processing modules. Then, the two same image processing modules process different image parts, respectively. This parallel scheme is very effective in accelerating image processing.

The operation-level shown in figure 2, which exists in both median filter and edge detector. During the operation of median filter 9 neighbouring pixels including the centre pixel are assigned to the three row extractors for reducing the searching time of the median value. Firstly each row extractor extracts the median value of three pixels in its row. The three row extractors work in parallel. Then, the final median extractor calculates the median value of the output values of three row extractors. This improvement will greatly increase the process of median searching. Although the final result is not the exact median of nine pixel values, it is among the middle three values and brings very little errors for an edge detection step. After a median filter step filtered pixels are sent to an edge detector.

III. DESCRIPTION OF MEDIAN FILTER

In this paper median filter and prewitt edge detection methods are used for image filtering and edge detection. The median filter is a non-linear filter. Here the median filter is designed using the sort optimization algorithm [2]. In image processing system while the transmission of image noises are added like salt and

pepper noise. This can affect the image processing system. So Yan lu [2] developed the sort optimization algorithm for median filter. The formula of median filter can be expressed as.

$$g(x,y) = \text{med } f[(x-i),(y-j)], i, k \in W \quad (1)$$

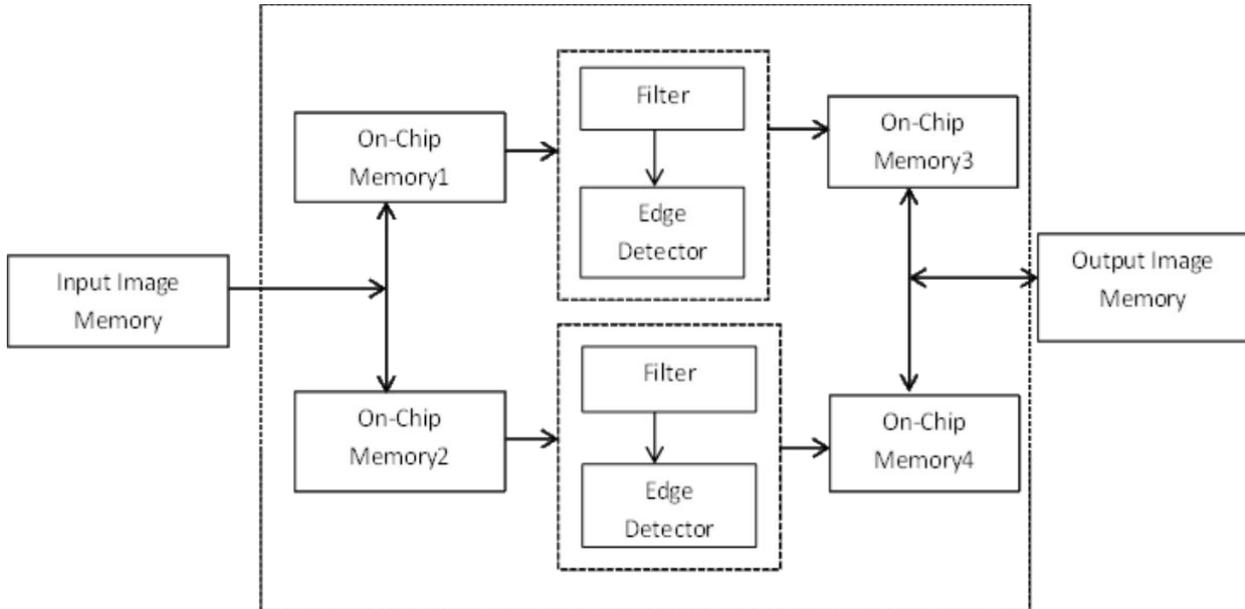


Fig. 1 : Double parallel image processing system

According to the work characteristics of FPGA and the transmission way of the image data, we can get the following image processing sequence:

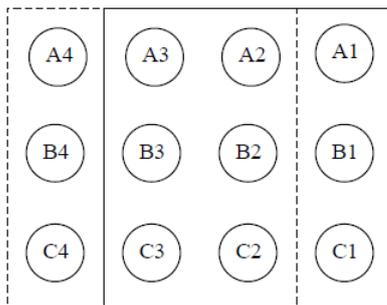


Fig. 2: The sequence of the image data

In above figure 2 A, B, C represent for the data of 3 different adjacent lines of the image respectively. The data in the real-line square is currently processing data and the data in the dashed square is the data will be processed in the next circle. It is easy to find out that there is 6 common data between the two processing circle. Hence, some parts of the current result can be utilized to the next processing for computation reduction. From the above image sequence of data we need to proceed data by column. 9 comparisons are needed to get following results. We assume

$$A1 > B1 > C1$$

$$A2 > B2 > C2$$

$$A3 > B3 > C3$$

We need to find the maximum data in C1, C2 and C3 by doing 2 comparisons, and find the minimum data in A1, A2 and A3 by doing 2 comparison and find the median data in B1, B2 and B3 by doing 2 comparisons. So the formula is

$$A_{\min} = \min(A1, A2, A3)$$

$$B_{\text{med}} = \text{med}(B1, B2, B3)$$

$$C_{\max} = \max(C1, C2, C3) \quad (2)$$

Finally we need to find median data in A_{\min} , B_{med} and C_{\max} by doing 3 comparisons.

$$F_{\text{med}} = \text{med}(A_{\min}, B_{\text{med}}, C_{\max}) \quad (3)$$

The resulted F_{med} is the final result. In the next circle, the comparison results of (A2, B2, C2) and (A3, B3, C3) have been obtained, we only need to find the sorting of the updating data A4, B4, C4 and then use (2) and (3) to calculate the required median. Apparently, in each next processing cycle, only one column data need to be sorted. According to that, the comparison times can be

reduced from the original 21 down to 13 and the system's operating efficiency can be greatly improved

IV. HARDWARE IMPLEMENTATION OF MEDIAN FILTER

The basic building blocks of median filter are, ordering comparator, minimum comparator, maximum comparator and median comparator. Using Xilinx 12.2 the below building blocks are implemented by the use of VHDL language.

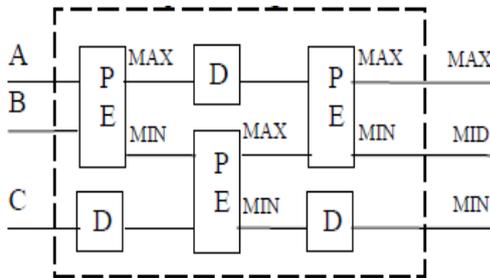


Fig. 3(a) : Ordering comparator

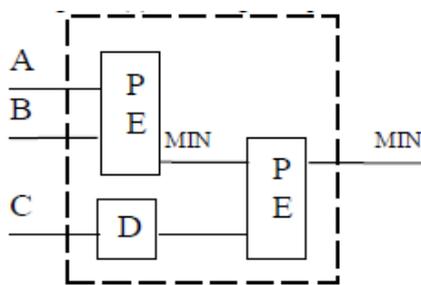


Fig 3(b) Minimum comparator

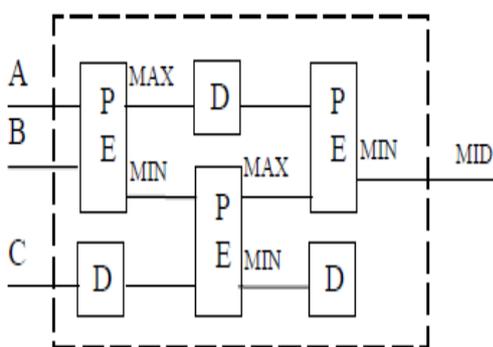


Fig 3(c) Median comparator

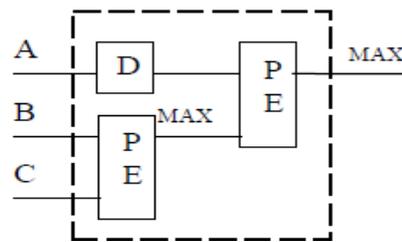


Fig 3(d) Maximum comparator

V. PREWITT EDGE DETECTION

The prewitt edge detection is used for calculating the maximum response of the edge template, it finds a direction of each pixels in which the intensity changes rapidly. Here we are using input image convolved with eight 3, 3 convolution kernels each of which sensitive to different orientation.

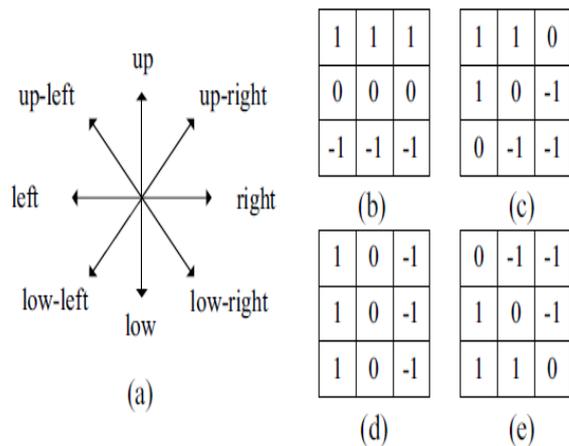


Fig.: Convolution kernels of the Prewitt edge detector. (a)Eight edge orientation. (b) Left orientation kernel. (c) Low-left orientation kernel. (d)Up orientation kernel. (e) Up-left orientation kernel.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. Boundry processing of the divided image

During image-level parallel operation, the input image is divided into two sections for parallel median filter and edge detection. In this paper, both median filter and edge detector utilize 3x3 masks, which means that processing each pixel, as Well as the pixels in the boundary of each section, needs eight pixels around it. In order to deal with the boundary problem, each section contains four more rows of pixels than half input image. The four extra rows are responsible for processing a boundary row

B. Matlab result



Fig 5(a) Original Image



Fig 5(b) Divided Image (1)



Fig 5(c) Divided Image (2)

C. Matlab result for median filter

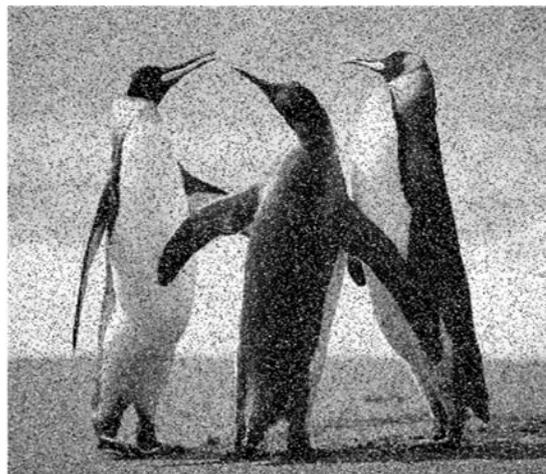


Fig 6(a) Noisy image



Fig 6(b) De-noised image

D. Matlab result for edge detection



Fig 7 Edge detection of the image

E. VHDL Simulation Result

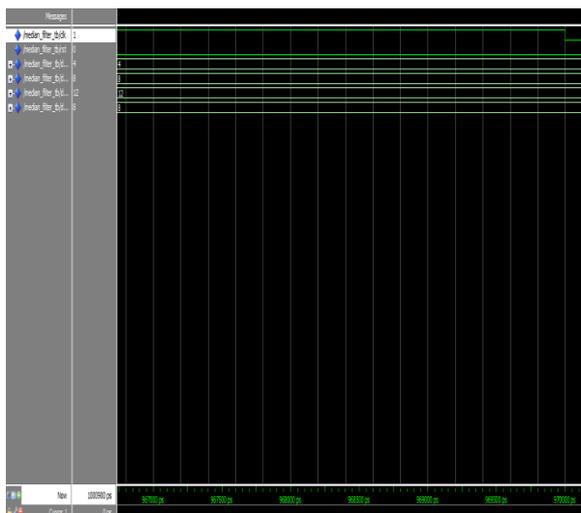


Fig. 8 : Median filter output

VII. CONCLUSION

A double parallel architecture for median filter an edge detection using sort optimization is proposed in this paper. The experimental result shows high performance of the proposed architecture. In next step we can do a trial of dividing the input image in to more than two parts to accelerate the processing speed.

VIII. REFERENCES

- [1] Ye Li, Qingming Yao, Bin Tian and Wencong Xu, "Fast Double-Parallel image processing based on FPGA" IEEE conference 2011.
- [2] Y. Lu, M. Dai, L. Jiang, and S. Li, "Sort optimization algorithm of median filtering based on fpga," in Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on, 2010, pp. 250–253.
- [3] P. Wei, L. Zhang, C. Ma, and T. S. Yeo, "Fast median filtering algorithm based on fpga," in Signal Processing (ICSP), 2010 IEEE 10th International Conference on, 2010, pp. 426–429.
- [4] C. J. Juan, "Modified 2d median filter for impulse noise suppression in a real-time system," Consumer Electronics, IEEE Transactions on, vol. 41, no. 1, pp. 73–80, Feb. 1995.
- [5] R. Maheshwari, S. Rao, and P. Poonacha, "Fpga implementation of median filter," in VLSI Design, 1997. Proceedings., Tenth International Conference on, Jan. 1997, pp. 523–524
- [6] Y. Hu and H. Ji, "Research on image median filtering algorithm and its fpga implementation," in Intelligent Systems, 2009. GCIS '09. WRI Global Congress on, vol. 3, May 2009, pp. 226–230.
- [7] Z. Shanshan and W. Xiaohong, "Vehicle image edge detection algorithm hardware implementation on fpga," in Computer Application and System Modeling (ICCA SM), 2010 International Conference on, vol. 14, 2010, pp. 184–188.
- [8] K. El Houari, B. Cherrad, and I. Zohir, "A software-hardware mixed design for the fpga implementation of the real-time edge detection," in Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on, 2010, pp. 4091–4095.
- [9] Z. Guo, W. Xu, and Z. Chai, "Image edge detection based on fpga," in Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on, 2010, pp. 169–171.
- [10] L. Li, J. Song, F.-Y. Wang, W. Niehsen, and N.-N. Zheng, "Ivs 05: new developments and research trends for intelligent vehicles," Intelligent Systems, IEEE, vol. 20, no. 4, pp. 10–14, july-aug. 2005.
- [11] F.-Y. Wang, P. J. Lever, and B. Pu, "A robotic vision system for object identification and manipulation using synergetic pattern recognition," Robotics and Computer-Integrated Manufacturing, vol. 10, no. 6, pp. 445–459, 1993.
- [12] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol. 43, no. 3, pp. 157–192, Mar. 1996.
- [13] T. Nodes and J. Gallagher, N., "Median filters: Some modifications and their properties," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 30, no. 5, pp. 739–746, Oct. 1982.
- [14] T. Peli and D. Malah, "A study of edge detection algorithms," Computer Graphics and Image Processing, vol. 20, no. 1, pp. 1–21, 1982. [Online]. Available: <http://www.sciencedirect.com/science/article/B7GXF-4D7JP09-27/2/f4a8a3ff110b8635fea9eeca0259b305>
- [15] Z. Guo, W. Najjar, F. Vahid, and K. Vissers, "A quantitative analysis of the speedup factors of fpgas over processors," in Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays, ser. FPGA '04. New York, NY, USA: ACM, 2004, pp. 162–170. [Online]. Available: <http://doi.acm.org/10.1145/968280.968304>.

