ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

Efficient Hardware for Impulse Noise Removal Using Median Filter

Anu Joy[1],Smitha Cyriac[2], Ansal K A[3]

Asst. Professor, Medical Electronics Department, MVJ College of Engineering, Bangalore, India[1].
Asst. Professor, Electronics& Communication, Viswajyothi College of Engineering & Technology, Vazhakulam, India[2].
Assistant Professor, Department of ECE, Ilahia College of Engineering and Technology, Muvattupuzha,India[3]

Abstract: Salt and pepper noise is one of the most common types of noise in image processing. In order to eliminate salt and pepper noise, median filtering concept is used. But in most of the median algorithms implemented on FPGA, the hardware resource occupancy rate is high and the computational speed is less. So it is hard to realize the real time noise reducing. Sort optimization algorithm is used to design an efficient median filter algorithm on hardware FPGA with less consuming area and high speed parallel architectures. The processing time can be reduced considerably on FPGA. This algorithm greatly decreases the amount of sorting, reduces the hardware resources consumption. The algorithm is illustrated through examples in MATLAB. It is also designed and realized using Xilinx ISE VLSI software and implemented in Spartan 3E FPGA kit. This algorithm is compared with bubble sort algorithm which is the traditional sorting algorithm of median filtering and is proved that hardware occupancy is less for sort optimization algorithm. An advanced method also simulated using MATLAB in which the median value is computed using only noise free pixels. Only the corrupted pixels are replaced by median calculated using only k-nearest noise free pixels of the selected window. This method exhibits better performance when noise removal is considered.

Keywords: Image Preprocessing, Median Filtering, Salt & Pepper noise, Bubble sorting, FPGA, Decision Based Median Filter, Impulse noise

I. INTRODUCTION

In the image processing system, due to imaging systems, signal transmission, working environment and other external conditions, there would inevitably produce many kinds of noise, resulting in lower image quality. Thereby, noise can affect the effects and accuracy of subsequent image processing. To reduce this influence, it is necessary to reduce the noise before further processing. There are many kinds of image noise, such as Gaussian noise, multiplicative noise, salt and pepper noise [2], [5], etc. Salt and pepper noise is one of the most common types of noise in the image processing.

In order to eliminate salt and pepper noise [3], median filtering concept is used. But in most of the median algorithms implemented on FPGA, the hardware resource occupancy rate is high and the computational speed is less. So it is hard to realize the real time noise reducing. The median filter is anonlinear image smoothing technology, its main principle isthat consider each pixel in the image as the centre, build an odd number of samples (3 × 3, 5 × 5, 7 × 7, etc.) observation window around its neigh boring area, sort the gray value of each point, and then use the median value instead of the original value [1]. The key of the median filtering algorithm is to sort the value of each pixel in the observation window to obtain the median value

II. BUBBLE SORT ALGORITHM

It is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted [6]. The algorithm gets its name from the way smaller elements "bubble" to the top of the list. Because it only uses comparisons to operate on elements, it is also known as a comparison sort. Although the algorithm is simple, some other algorithms are more efficient for sorting large lists.

The traditional median filtering algorithm is bubble algorithm, whose main purpose is comparing all data with each other to make an order [1]. The whole processing needs 36 times comparisons totally when a 3×3 window is considered. Bubble sort is an elementary sorting algorithm. It works by repeatedly exchanging adjacent elements, if necessary. When no exchanges are required, the file is sorted.Using this method, the hardware resource occupancy rate is high, and the processing cost is huge, it can hardly to be used in a practical engineering system.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

III. SORT OPTIMIZATION ALGORITHM

The sort optimization algorithm introduces a real-time median filtering using pipelining processing technology according to the FPGA's work characteristics. This algorithm greatly decreases the amount of sorting, reduces the hardware resources consumption. Therefore, it has a good real-time performance even when dealing with high-resolution images.

The advantage of our proposed algorithm is that, in this algorithm the traditional sorting algorithm of median filteringis optimized according to the hardware structure features of FPGA. FPGA is used to acquire the data parallel for comparing the data of the same column in the median filtering window. Comparing results shared by adjacent filter window are saved temporarily to match the new round of median filtering by using FPGA internal resources. This method can reduce the number of comparisons to 13, and thereby improve the algorithm efficiency. The disadvantage is that each pixel in the image will be replaced by the median value of its neigh boring pixels.

The main application is to design a dedicated VLSI chip for the removal of noise in satellite cameras as soon as the image is captured. Currently research work is in progress for filtering aerial images and medical images.

A. Algorithm

The formula of median filtering can be expressed as:

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

where f (x,y) and g(x,y) represent for the original image and the filtered image respectively. W is a two-dimensional template, usually is a 3 × 3, 5 × 5 region.

The sort optimization algorithm works by considering a 3×3 window. According to the work characteristics of FPGA and the transmission way of the image data, we can get the following image processing sequence.

In figure shown below, 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 [1].

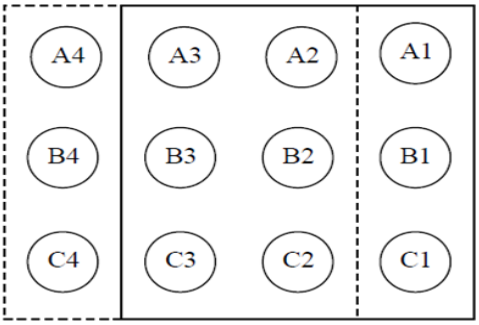

Fig. 1 Sequence of Image Data Processing

First, sort the data by column. It is necessary to do 9 comparisons to get the following result. The result is

A1> B1> C1
A2> B2> C2
A3> B3> C3

- Secondly, we need to find the maxim data in C 1, C2 and C3 by doing 2 comparisons, and find the minimum data in A1, A2 and A3 by doing 2 comparisons, and also find the median data in B1, B2 and B3 by doing 3 comparisons. The formula is as following :

  Amin = min (A1,A2,A3)

  Bmed = med(B1,B2,B3)

  Cmax = max (C1,C2,C3)

- Lastly, we need to find the median data in Amin, Bmed and Cmax by doing 3 comparisons. The formula is as following.

  Fmed = med (Amin, Bmed,Cmax).

The resulted Fmed is the final result. In next circle, resultsof (A2 , B2 , C2) & (A3 , B3 , C3) have been obtained, we only need to seek the sorting of the updating data A4, B4, C4, and then use to calculate the required median. Obviously, in each next processing cycle, only one column data need to be sorted [1]. According to that, the comparison times can be reduced from to 13, and the system's operating efficiency can be greatly improved.

IV. HARDWARE IMPLEMENTATION

The proposed algorithm is executed as an image pre-processing module on FPGA. In this implementation the filter is designed as 3x3 filter kernel. According to that, we need to save temporary data in 3 lines, which are line n-1, line n and line n+1. We can define 3 first input first output (FIFO) memory to store these data. When the data of line n

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

have been processed, drop the data of line n-1 and acquire the data of line n+2 to begin a new round processing [1].

It is possible to get the maxim value or the minimum value of 3 data by doing 2 comparisons, and get the median value or the ordering of 3 data by doing 3 comparisons, we design 4 different comparators, such as ordering comparator, minimum comparator, median comparator and maxim comparator.

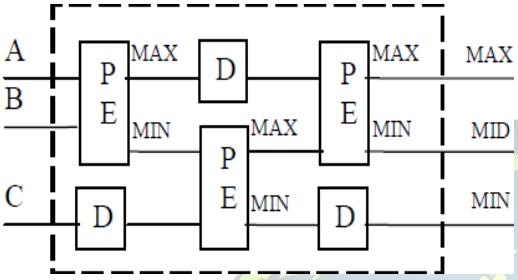

Fig. 2(a) Ordering Comparator

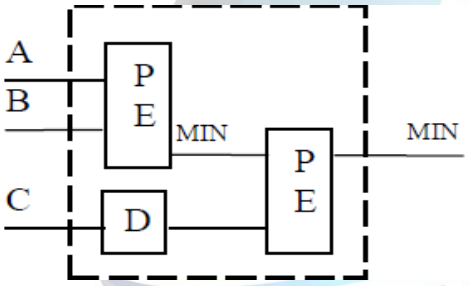

Fig. 2(b)Minimum Comparator

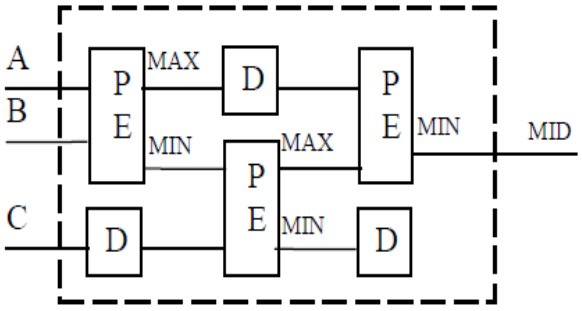

Fig. 2(c) Median Comparator

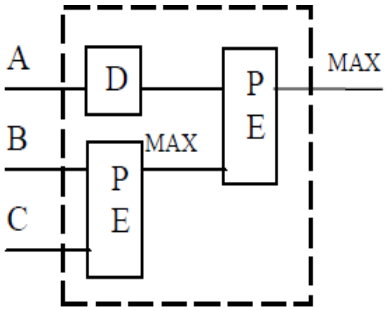

Fig. 2(d) Maximum Comparator

In these figures, the processing element (PE) is basic processing unit, which is designed for comparing 2 input data. D stands for D flip-flop, whose function is to make a single-circledelay. It is used for synchronizing the calculation here.

In the hardware structure of the median filter module we are using two D flip-flops because the algorithm needs to storage two groups temporary data in current processing, we design two D flip-flops to realize this function.

The workflow of the median filtering module is that firstly, three lines of data are wrote into the internal FIFO of FPGA. The data in the 3 FIFO will be sent to the ordering comparator ordinally for data ordering, and the results will be send to next different comparators. Before the second comparing, the data need 2 circle delay to distinguish the circle order of input data by two D flip-flops. The second comparing results will be sent to the final median comparator to get the final result. That means the minimum, median and maximum comparators outputs are given to the final median comparator to get the final median output.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

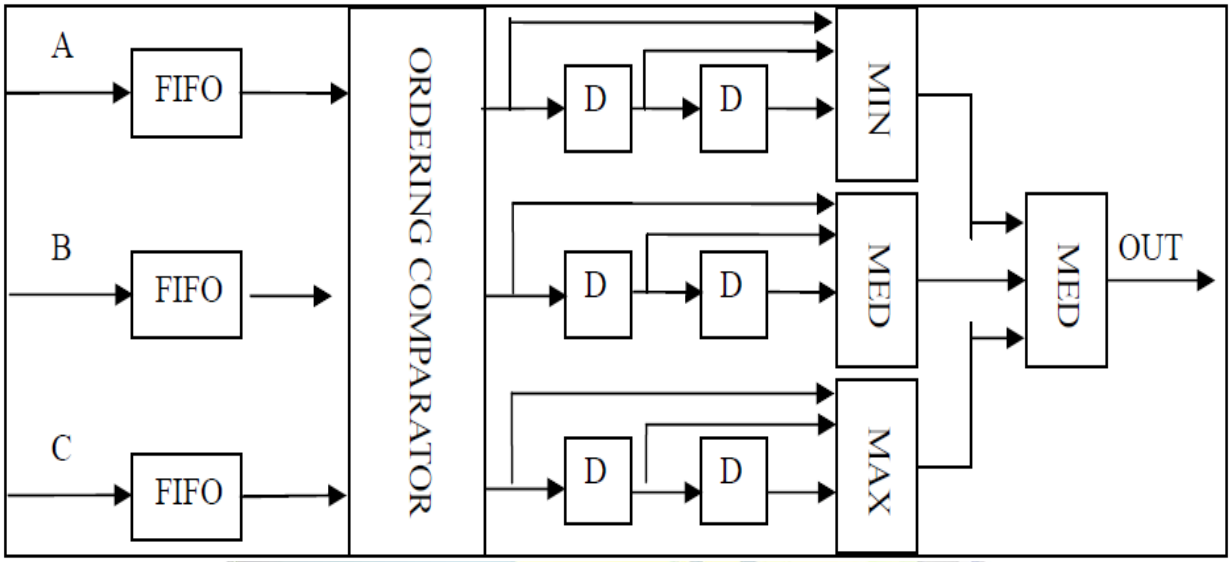

Fig. 3 Hardware Structure of Median Filter Module

The hardware structure of the median filter module is shown in figure above.This method can reduce the number of comparisons to 13, and thereby improve the algorithm efficiency.The disadvantage is that each pixel in the image will be replaced by the median value of its neigh boring pixels.

V. ADVANCED METHOD

Traditional median filter replaces each pixel in an image with the median value of their k-nearest pixels. The problem associated with this approach is that the restored pixel is noise if median value of their k-nearest pixels is a corrupted pixel. In this advanced method the median value is computed using only noise free pixels. Only the corrupted pixels are replaced by median calculated using only k-nearest noise free pixels of the selected window [4].

In this algorithm xi,jand yi,jdenote the values of the pixel at the position (i, j) in thecorrupted and restored images respectively. Si,jrepresents the set of the *k*-nearest pixelslocating in the window of the pixel (i, j) with the window size w. Therefore, if k = (2w + 1) × (2w + 1)nearest pixels are taken into account when restoring the pixel (i, j), then the set Si,jcontains the following pixels:

$$S_{i,j}= \{x_{i-w,j-w}, ..., x_{i,j} , ..., x_{i+w,j+w}\} \quad (2)$$

where w is the size of window and k is the number of the nearest pixels that are taken into account.

A. Noise-free Pixels Detection

To determine whether a pixel (i, j) is noise-free or noise, the value of the pixel (i, j) is compared with the maximum & minimum values of (2w + 1)×(2w+ 1) nearest pixels. This is because the corrupted pixels can take only the maximum or minimum values in the dynamic range (0, 255). The maximum value in its (2w+1) × (2w+1) nearest pixels can be calculated as follows:

$$Maxx_{i,j}= \max\{x_{i-w,j-w}, ..., x_{i,j} , ..., x_{i+w,j+w}\} \quad (3)$$

The minimal value is equal to:

$$Minx_{i,j}= \min\{x_{i-w,j-w}, ..., x_{i,j} , ..., x_{i+w,j+w}\} \quad (4)$$

If a binary matrix *B* is employed to represent whether a pixel is noise or not, then the binary matrix *B* can be computed as follows:

$B_{i,j}=1$ if $Minx_{i,j}<x_{i,j}<Maxx_{i,j}$;

=0 otherwise

Where the value 1 of $B_{i,j}$denotes the pixel (i, j) is noise-free and the value 0 means that the pixel (i, j) is noise.

B. Decision-based Median Filter Using k-nearest Noise-free Pixels

After the binary matrix *B* is obtained, the smallest window

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

Sij= {xi−w,j−w, ..., xi,j , ..., xi+w,j+w} (5)

of the pixel (i,j) containing k noise-free pixels can be obtained by an iterative process based on the following two inequalities:

{Bi−w,j−w, ...,Bi,j , ...,Bi+w,j+w} ≥ k (6)

and

{Bi−w+1,j−w+1, ...,Bi,j , ...,Bi+w−1,j+w−1} <k (7)

Then all noise-free pixels in the window Si,jcan be detected and stored into a new set S*i,j. The pixels in S*i,jare ranked and median value is returned as the final pixel of the corrupted pixel (i, j). Therefore, decision-based median filter using k-nearest noise-free pixels replace the pixels in the original corrupted image with the following rule employed:

yi,j= MEDIAN(S*i,j) if Bij= 0;

= xi,jif Bij= 1;

Where MEDIAN(S*i,j) returns the median value of the pixels in the set S*i,j. The restored pixel value is same as that of original one when Bi,j is equal to one otherwise the restored value is equal to MEDIAN(S*i,j).This advanced algorithm exhibits better performance compared to sort optimization and bubble sort algorithms.

VI. RESULTS

The performance of the median filter is tested in MATLAB using Bubble Sort Algorithm, Sort Optimization Algorithm and Advanced method using the version MATLAB 7.5.0 for both grey scale and colour images. The algorithms are tested on different images of size 256x256, 8bits/pixel.A comparison of MATLAB simulation and VHDL simulation is also done. Bubble Sort and Sort Optimization algorithms are designed using VHDL and executed logic simulation with use of XILINX's Model sim.The hardware implementation of both Sort Optimization and Bubble Sort Algorithms are performed in Spartan 3E XC3S250E development board. The implementation results proved that the hardware occupancy is less for Sort Optimization Algorithm (SOA) and is shown in table1.

XC3S250E-5TQ144		
Logic Utilization	**Bubble Sort**	**SOA**
Number of Slices	331	295
Number of 4 input LUT's	656	281
Number of Bonded IOB's	83	34

TABLE I
HARDWARE COMPLEXITIES OF BUBBLE SORT AND SORT OPTIMIZATION ALGORITHM

Fig. 4Simulation Result of Lena.jpg colour image Using Bubble Sorta) Original image b) Corrupted image(10% noise) c) Restored image by Bubble Sort Algorithm

Fig. 5 Simulation Result of Lena.jpg colour image Using Sort Optimization Algorithm a) Original image b) Corrupted image (10% noise) c) Restored image by Sort Optimization Algorithm

Fig. 6 Simulation Result of Lena.jpg colour image Using Advanced Method a) Original imageb) Corrupted image (10% noise) c) Restored image by advanced method

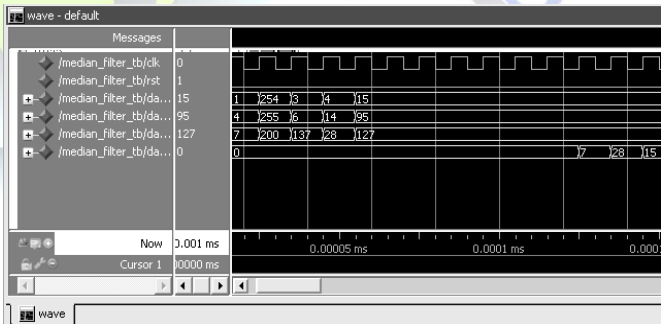

Fig. 7 Simulation result of finding the median by using sort optimization algorithm

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

VII. CONCLUSIONS

Sort optimization algorithm is compared with bubble sort algorithm which is the traditional sorting algorithm of median filtering and realized that the hardware resource occupancy is more for bubble sort algorithm. Sort optimization algorithm greatly decreased the amount of sorting and thereby reduced the hardware resources consumption.

The Sort Optimization Algorithm introduces a real-time image median filter based on FPGA structure. It can be connected with the DSP to process more complex and larger Images. Because the system occupation is small and leave more resources to help DSP for image pre-processing, the future image processing system can run more efficiently.

The simulation results of advanced method in MATLAB is compared with simulation results of sort optimization and bubble sort algorithms to illustrate that the noise removal is better for advanced method. It is possible to obtain better noise removal up to 80% noise densities.

ACKNOWLEDGMENT

The authors would like to thank Mr.Asit Kumar Das, Specialist in DIP, VEDLABS, Bangalore, Mr.SatisBhairannawar, and Specialist in Matlab for their valuable guidance, help and insightful comments.

REFERENCES

[1]. Yan Lu, Ming Dai, Lei Jiang & Shi Li, *Sort Optimization Algorithm of Median Filtering Based on FPGA*, IEEE International Conference on Machine Vision and Human-machine Interface pp.250-253, 2010

[2]. Chan.R.H., Chung-WaHo., Nikolova, *Salt-and-pepper noise removal by median-type noise detectors and detail preserving regularization*,IEEE Transactions on Image Processing, Vol.no:14, 2005.

[3]. Jiang Bo, Huang Wei, *Adaptive Threshold Median Filter for Multiple-impulse Noise*, Journal of Electronics Science and Technology of China. Vol.5, 2007

[4]. YiHong, Sam Kwong, Hanli Wang, *Decision-based median filter using k-nearest noise-free pixels* ,IEEE Trans, ICASSP, 2009

[5]. http:// appnote.avrportal.com/ Image Processing/ *An Introduction to Digital Image Processing with Matlab Notes for SCM2511 Image Processing.pdf*

[6]. http://webspace.ship.edu/cawell/Sorting/bubintro.html

BIOGRAPHY

Anu Joywas born in India. She took her B-Tech in Electronics and Communication Engineering & M-Tech in VLSI and Embedded System from Viswajyothi College of Engineering and Technology, Muvattupuzha, Kerala. She is currently working as Assistant Professor in MVJ College of Engineering, Bangalore. Her special area of interest is Microwaves and Digital Image Processing.

Smitha Cyriacwas born in India. She took her M-Tech from Model Engineering College, Thrikkakara. She is currently working as Assistant Professor in Viswajyothi College of Engineering and Technology, Muvattupuzha.

Ansal K Awas born in India. He received his B-Tech degree in Electronics and Communication Engineering from Mahatma Gandhi University, Kottyam, Kerala and M.Tech degree (Gold Medal) in Communication System from Ann University Coimbatore. He currently pursuing Ph.D. degree in the field of Microwaves in the Department of Electronics and Communication Engineering, Pondicherry Central University, Pondicherry. He has 6 years of experience in teaching as an Assistant Professor in Ilahia College of Engineering and Technology. Muvattupuzha, Kerala. His special area of interest is Antennas for microwave and Millimeter waves, Antennas with Defected Ground Structures, Photonic Band Gap Structures, Metamaterial Inspired Antennas, Monolithic Microwave Integrated Circuits and Digital Image Processing etc. He is member in IEEE, IEICE, IAENG and ATMS