Salt and Pepper Noise Removal in Digital Images Using Modified Decision Based Filter

George Levi. D
VLSI and Embedded Systems
B. S. Abdur Rahman University
Chennai, India
Grg.levi@gmail.com

Selvakumar. M
Electronics and Communication
B. S. Abdur Rahman University
Chennai, India
mselvakumar@bsauniv.ac.in

***Abstract*—Digital images are corrupted frequently by impulse noise during the procedures of acquisitions and their transmissions. An effective VLSI architecture is designed in this paper to remove the salt and pepper impulse noise from the images. Modified Decision tree is used for detecting the noisy pixels and the noise density. Mean and Median filtering technique is applied to restore the noisy pixels. Moreover with this architecture we can determine the high density and low density impulse noise. It is implemented using the Xilinx ISE design suite and tested in Spartan-3E FPGA kit by interfacing it with the PC using the RS232 cable. The implemented design shows better results in visual quality in the noise removed image also it has less complexity and improved performance.**

Index Terms— Impulse noise; Modified decision tree; Median filter.

I. INTRODUCTION

In digital image processing for better visual quality of the image, noise has to be removed from it. An effective impulse noise removal from the corrupted images is very important for the image processing operations [1]. The intensity of the impulse noise can be either high or low. Based upon the pixel values and its characteristics impulse noise can be two common types the salt and pepper noise and the random valued impulse noise. In the images with salt and pepper noise the noise pixel values can be maximum and minimum. Noise pixels in the images with random valued impulse noise can be any random value between 0 and 255 in grey scale images. Many methods have been proposed to eliminate the salt and pepper noise [2], [3] and some of them are using the decision tree [4], [5]. The median filters were the popular filters used for removing impulse noise, because of its computational efficiency and de-noising power. However, when the noise level reaches over 50 percent, edge details of the original image are smeared by the filter. Different modifications of the median filter [6], [7] have been proposed, in these approaches they first identify the possible noisy pixels and then replace them by using the median filter or its variants but the details and edges are not recovered satisfactorily when the level of noise is high also they affect the noise free pixels.

A differential rank impulse detector (DRID) [8] compares the rank and brightness of the pixel of interest in various series to the rank of the median to the closest ones. An alpha-trimmed mean based method (ATMBM) [9] uses the alpha trimmed mean in impulse detection, a linear combination of its original value to replace the noisy pixel and the median of its local window. A differential rank impulse detector (DRID) was presented in [10] consists of the impulse detector which compares the signal samples within a narrow size rank window with absolute value and rank. Rank-ordered relative differences (RORD) [11] used to identify pixels that are likely to be corrupted by impulse noise. Decision based algorithm (DBA) is proposed in [12] removes the corrupted pixel by the median from its decisions. In [13] a decision tree is designed which deals with the salt and pepper noise. In [14] Lien and Huang proposed an decision tree based de-noising method (DTBDM) to detect the noisy pixel using the decision modules and replace the noisy pixels by the edge preserving filter. The main drawback of this architecture is that we cannot recognize the difference between the high density noise and the low density noise. Also this architecture is designed only to remove the random valued impulse noise leaving out the salt and pepper noise.

Based on the basic concepts used above we propose a modified decision tree based de-noising method and their VLSI architecture design for the removal of impulse noise in images using the median filter techniques. The main advantage of our architecture is that our design can identify the high density noise and low density noise individually and then we use the edge preserving median filter to replace the noisy pixels.

II. PROPOSED MDBDM

The noise which is considered in this paper is the impulse noise. MDBDM is modified decision based de-noising method. Corrupted images are detected if it is noisy or not with pixel matrix Fig. 1. If it is noise free pixel then the value is directly taken to the output image Fig. 2. If the processing pixel is a noisy pixel then the noise analysis is made and the respective values are restored using the median filter.

25	255	20
34	60	87
0	112	43

Fig. 1. 3X3 sliding window

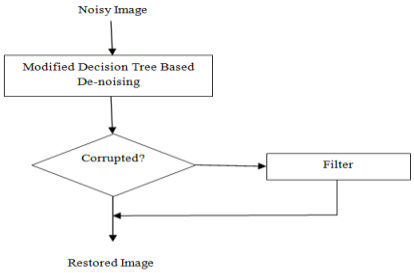

Fig. 2. Dataflow of MDBDM

Images to be de-noised are initially taken as 3x3 sliding window as in Fig. 1. This sliding window is taken for all the individual pixels that are available in the image. MDBDM consists of three main parts: decision making, noise analysis, noise restoration Fig. 3. The dataflow concepts of the MDBDM are shown in Fig. 2.

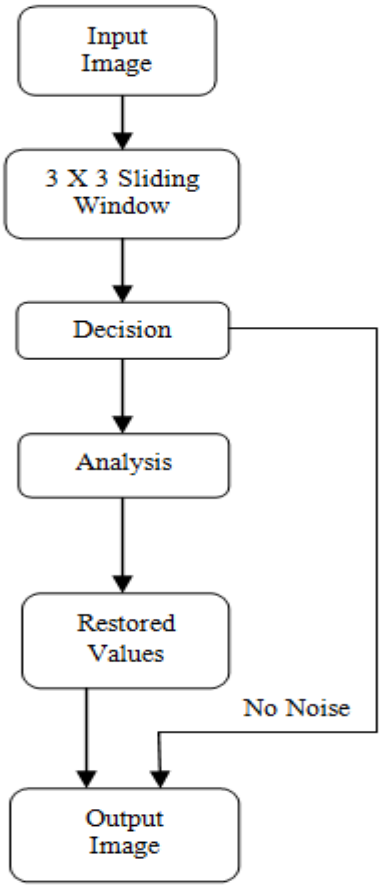

Fig. 3. MDBDM block diagram.

The MDBDM design block diagram is shown in Fig. 3. After the image received been converted into sliding window, pixels in an image that have been affected by impulse noise can be determined using the threshold values. Each pixel in the sliding window is compared with the neighboring, the neighboring pixel value size are adjustable as well as the threshold values are also adjustable. When the pixel values are greater or lesser than the threshold values then it is considered as the noise. If the pixel value does not cross the threshold value then it is considered as the noise free pixel. Noise free pixel values are directly taken to restore the output image and the noisy pixels are passed to the noise analysis. From the noise analysis the filter replaces the mean and median values based on the density of the noise in the particular sliding window.

III. NOISE ANALYSIS

The noise analysis is the most important part of this architecture after the decision making. Once the noisy pixel is identified from the decision making block it is analyzed, in the sliding 3X3 matrix the number of noisy pixels are determined Fig. 4. If the total number of the noisy pixels in the matrix are greater than the noise free pixels then it is categorized as the high density noise, else when the noise free pixels are higher than the noisy pixels they are categorized under the low density noise.

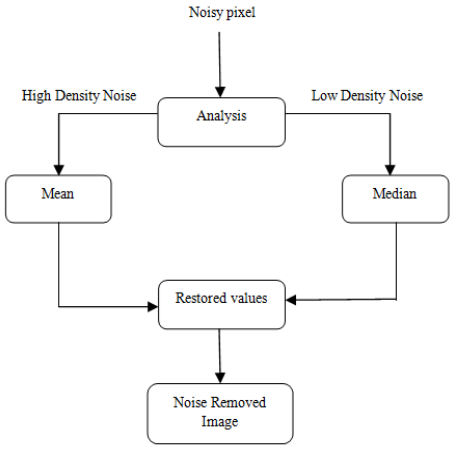

Fig. 4. Noise analysis structure

Fig. 4 shows the structure of the noise analysis in the MDBDM. When there is high density noise occurring in the images their noisy values is replaced with the mean of the noise free pixels in the matrix. If there is low density noise in the matrix then those noise values are replaced with the median of the pixels in the matrix. These analysis steps are followed for all the noisy pixel values. All the restored values together form the noise removed image.

IV. VLSI IMPLEMENTATION OF MDBDM

MDBDM has the low computational complexity it does not need any additional iterations or serious of operations to be performed. Here we design Micro blaze core processor and

then implement it using the Xilinx platform studio design suite. The algorithm is written using system C Language and they are tested in Spartan-3E FPGA kit by interfacing them in the PC with the help of RS232 cable. The median filter technique is used for this designing

A. Micro Blaze processor design

The Micro Blaze is an embedded soft core processor with reduced instruction set computer (RISC) optimized for the implementation in Xilinx FPGAs. Since field programmable gate arrays (FPGA'S) are flexible and can be easily re-configured by the designer, it enables the VLSI design to be performed quickly and less expensive. Due to increasing device densities Xilinx has incorporated larger embedded components and processors. One of such enhancement in the Xilinx Spartan, virtex family architecture is the introduction of the new Micro Blaze (Soft IP) hard-core embedded processor. Micro blaze processor is a 32-bit RISC architecture for implementation in Xilinx FPGAs with 32-bit instruction and data buses, to execute programs and to access data from on-chip and external memory. Interrupt controller is available to use with the Xilinx Embedded Development Kit (EDK) software tools.

The advancement in fabrication technology and increase in the density of logic blocks on FPGA and its use in debugging and prototyping digital circuits. Due to the enormous parallelism achievable on FPGA and the increasing density of logic blocks, it is currently used now as a replacement to ASIC solutions in a few applications. Soft cores are technology independent and it requires only simulation and timing verification after been synthesized to the target technology.

B. Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is a development environment which we use to design the hardware portion of the system. Embedded Development Kit (EDK) is the integrated software tool we use for developing the embedded systems using Xilinx Micro Blaze. EDK is preferred due to its variety of tool content and applications for developing the embedded system from creating the hardware to final implementation on FPGA. Our design of the system consists of hardware and software component creation on the embedded processor system. Our design involves: creation of hardware platform, verification of hardware platform (simulation), creation of software platform, its application and verification. MHS (Microprocessor Hardware Specification) file created from base system wizard defines the architecture of the system and the peripherals. Using the MHS file as input the Platform Generation tool creates the hardware platform. The MSS (Microprocessor Software Specification) file defines the software platform that includes driver, library customization parameters for processor, peripherals, interrupt handler routines, and other software routines. The MHS file is taken as input for creating simulation files as a specific simulator. Bit stream tool is used to initialize the instruction and memory of the processors on the FPGA. Compiling and linking application executables for each processor in the system are done by GNU Compiler tools. Xilinx Microprocessor Debug (XMD) is used for debugging the application software Fig. 5.

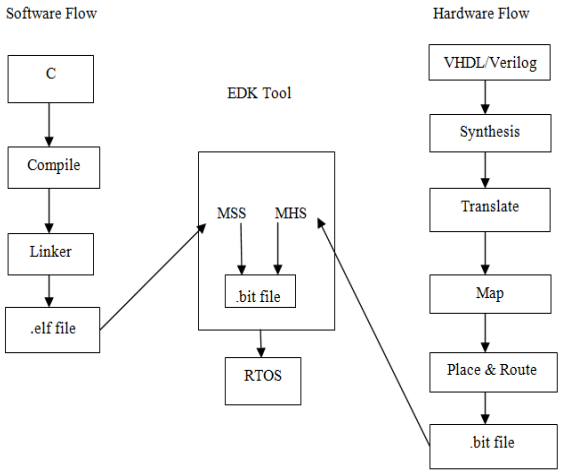

Fig. 5. Implementation Design Flow

Software Development Kit (SDK) consists of tools that enable us to design the necessary software application for selected Soft IP Cores in Xilinx Embedded Development Kit (EDK). The hardware and software implementation design with the EDK tool is shown in Fig. 5. The software application is written in system C, by debugging & downloading the bit file into FPGA the complete embedded processor system for user application is implemented. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

V. RESULTS

The Algorithm is implemented using Micro blaze Processor and is performed on gray scale images to verify this proposed method. Images are represented with 8 bits/pixel and the size considered is based on the availability of the SRAM in the kit. In general we take 128 x 128, image used for this process are shown in Fig. 6. This modified architecture is implemented in system c, placed and routed on Xilinx spartan3E XC3S500 FPGA, using the Xilinx platform studio.

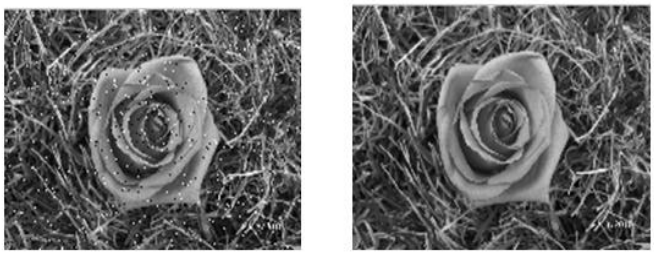

Fig. 6. Noisy and Noise free Image result

VI. CONCLUSION

A low-cost, low complexity VLSI architecture implementation for effective removal of impulse noise is proposed in this paper. This approach uses the modified decision based noise detection to detect the noisy pixel and employs an efficient design to preserve the noise. With adaptive skills, the quality of the reconstructed images is improved. Our experimental results demonstrate that our performance of the proposed technique is much better than the previous lower-complexity techniques and is comparable to the other higher-complexity methods. We have presented a new modified decision-based filter and noise analysis for the impulse noise removal. This new impulse detection and analysis can determine where the noise is and what is the density of the noise either high density or low density noise, also only the noisy pixels are replaced with the estimated central noise-free mean and median values. As a result, the restored image preserves entire details and edges in the images while effectively suppressing impulse noise.

REFERENCES

[1] R.C. Gonzalez and R.E. Woods, Digital Image Processing. Pearson Education, 2007.

[2] R.H. Chan, C.W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," *IEEE Trans. Image Processing*, vol. 14, no. 10, pp. 1479-1485, Oct. 2005.

[3] P.-Y. Chen and C.-Y. Lien, "An Efficient Edge-Preserving Algorithm for Removal of Salt-and-Pepper Noise," *IEEE Signal Processing Letters*, vol. 15, pp. 833-836, Dec. 2008.

[4] H.-H. Tsai, X.-P. Lin and B.-M. Chang, "An Image Filter with a Hybrid Impulse Detector Based on Decision Tree and Particle Swarm Optimization," Proc. IEEE Int'l Conf. Machine Learning and Cybernetics, July 2009.

[5] P.-Y. Chen, C.-Y. Lien and H.-M. Chuang, "A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise," *IEEE Trans. Very Large Scale Integration Systems*, vol. 18, no. 3, pp. 473-481, Mar. 2010.

[6] P.E. Ng and K.K. Ma, "A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images," *IEEE Trans. Image Processing*, vol. 15, no. 6, pp. 1506-1516, June 2006.

[7] R.H. Chan, C.W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," *IEEE Trans. Image Processing*, vol. 14, no. 10, pp. 1479-1485, Oct. 2005.

[8] I. Aizenberg and C. Butakoff, "Effective Impulse Detector Based on Rank-Order Criteria," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 363-366, Mar. 2004.

[9] W. Luo, "An Efficient Detail-Preserving Approach for Removing Impulse Noise in Images," *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 413-416, July 2006.

[10] I. Aizenberg and C. Butakoff, "Effective Impulse Detector Based on Rank-Order Criteria," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 363-366, Mar. 2004.

[11] H. Yu, L. Zhao, and H. Wang, "An Efficient Procedure for Removing Random-Valued Impulse Noise in Images," *IEEE Signal Processing Letters*, vol. 15, pp. 922-925, 2008.

[12] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision based algorithm for removal of high-density impulse noises," *IEEE Signal Process. Lett.*, vol. 14, no. 3, pp. 189–192, Mar. 2007.

[13] P.-Y. Chen, C.-Y. Lien, and H.-M. Chuang, "A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise," *IEEE Trans. Very Large Scale Integration Systems*, vol. 18, no. 3, pp. 473-481, Mar. 2010.

[14] Chih-Yuan Lien, Chien-Chuan Huang, Pei-Yin Chen and Yi-Fan Lin, "An efficient denoising architecture for removal of impulse noise in images" *IEEE Transactions on computers*, vol.62, No. 4, April 2013.

[15] Xilinx Inc. Xilinx ISE and Xilinx EDK tools.

[16] Spartan-3E Starter Kit Board User Guide, Reference Manual, Xilinx, Inc.