Research Article ISSN: 2319-507X
Sunitha D, IJPRET, 2013; Volume 2 (4): 89-99 IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

AN ADAPTIVE WEIGHT ALGORITHM FOR REMOVAL OF IMPULSE NOISE

D. SUNITHA, Mr. B. KAMALAKAR

1. M. Tech-Student (VLSI System Design), CVSR College of Engineering
2. Asst. Professor, CVSR College of Engineering

Accepted Date: 27/10/2013 ; Published Date: 01/12/2013

Abstract: This paper proposes the design of VLSI architecture for removal of random valued impulse noise using an Adaptive weight algorithm (AWA)and it is implemented in Spartan 3EDK kit. The adaptive weight algorithm has the advantage of lower computational complexities and higher efficiencies. We employ decision based impulse noise detector to detect noisy pixels and an adaptive weight median filter (AWMF) The new impulse detector, which uses multiple thresholds with multiple neighborhood information of the signal in the filter window to classify the noise free and noisy pixels. And secondly, denoising the detected impulse noise by employing adaptive weight median filter (AWMF) technique. So that noise free pixels are getting preserved and only noisy pixels are replaced with the estimated central noise-free ordered median value in the current filter window. This filter is good at detecting noise even at a high noise level. This technique not only has lower computation complexity but also require less memory. Here we designed core processor Micro blaze and implemented using XILINX platform studio Design suite. The algorithm is written in system C Language and tested in SPARTAN-3 FPGA kit by interfacing a test circuit with the PC using the RS232 cable. The test results are seen to be satisfactory.

Keywords: Algorithm, Impulse, Adaptive Weight

Corresponding Author: Ms. D. SUNITHA

Access Online On:

www.ijpret.com

How to Cite This Article:

Sunitha D, IJPRET, 2013; Volume 2 (4): 89-99

PAPER-QR CODE

INTRODUCTION

Images are often corrupted by impulse noise when they are recorded by noisy sensors or sent over noisy transmission channels. Many impulse noise removal techniques have been developed to suppress impulse noise while preserving image details .The median filter, the most popular kind of nonlinear filter, has been extensively used for the removal of impulse noise due to its simplicity. However, the median filter tends to blur fine details and lines in many cases. To avoid damage to good pixels, decision-based median filters realized by thresholding operations have been introduced in some recently published works .In general, the decision-based filtering procedure consists of the following two steps: an impulse detector that classifies the input pixels as either noise-corrupted or noise-free, and a noise reduction filter that modifies only those pixels that are classified as noise-corrupted. In general, the main issue concerning the design of the decision-based median filter focuses on how to extract features from the local information and establish the decision rule, in such a way to distinguish noise-free pixels from contaminated ones as precisely as possible. In addition, to achieve high noise reduction with fine detail preservation, it is also crucial to apply the optimal threshold value to the local signal statistics. Usually a trade-off exists between noise reduction and detail preservation. In this proposed technique a novel decision-based filter, named the adaptive weighted median filter, to overcome the drawbacks of the above methods. Basically, the proposed filter takes a new impulse detection strategy to build the decision rule and practice the threshold function. The new impulse detection approach based on multiple thresholds considers multiple neighborhood information of the filter window to judge whether impulse noise exists. The new impulse detector is very precise without, while avoiding an increase in computational complexity. The impulse detection algorithm is used before the filtering process starts, and therefore only the noise-corrupted pixels are replaced with the estimated central noise-free ordered median value in the current filter window. Extensive experimental results demonstrate that the new filter is capable of preserving more details while effectively suppressing impulse noise in corrupted images.

The median filter is a simple nonlinear smoothing operation that takes a median value of the data inside a moving window of finite length. This filter has been recognized as a useful image enhancement technique due to its edge preserving smoothing characteristics and its simplicity in implementation ***[5]***. Median filtering preserves edges in images and is particularly effective in suppressing impulsive noise. Application of median filtering to an image, however, requires some caution because median filtering tends to remove image details such as thin lines and corners while reducing noise. Moreover, Manuscript received August 9, 1988; revised

November 12, 1990. This work was supported by the National Science Foundation under Grant DCI-8611859 and the Korea Science and Technology Foundation. This paper was recommended by Associate Editor H. Gharavi. S.-J. KO is with the Department of Electrical and Computer Engineering, University of Michigan, Dearborn, MI 48128. Y. H. Lee is with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Seoul, Korea. IEEE Log Number 9101486. the performance of median filtering is unsatisfactory in suppressing signal-dependent noise. Recently, in response to these difficulties, several variations of median filters have been introduced. Specifically, the max/median **[4],**FIR-median hybrid and multistage median filters have been developed for detail preserving smoothing. These filters preserve more image details at the expense of noise suppression. In [2], an adaptive median filter that adjusts the window size depending on the input is proposed to trade between detail preservation and noise suppression. For signal-dependent noise reduction some adaptive median-type filters, such as adaptive double window modified trimmed mean and signal adaptive median 141 filters, have been proposed. The outputs of the adaptive median-type filters are obtained based on the results of median filtering, and as a consequence these adaptive filters tend to remove image details just like median filtering. The ***weighted median*** (WM) filter **is** an extension of the median filter, which gives more weight to some values within the window [5]-[1]. This WM filter allows a degree of control of the smoothing behavior through the weights that can be set, and therefore, it is a promising image enhancement technique. In this paper, we focus our attention on a special case of WM filters called the ***center*** weighted median (CWM) filter. This filter gives more weight only to the central value of a window, and thus it is easier to design and implement than general WM filters. We shall analyze the properties of CWM filters and observe that CWM filters preserve more details at the expense of less noise suppression like the other non adaptive detail preserving filters.

II ADAPTIVE WEIGHT ALGORITHM

The noise considered in this paper is random-valued impulse noise with uniform distribution as practiced in [3] Here, we adopt a 3×3 mask for image denoising. Assume the pixel to be denoised is located at coordinate (*i*, *j*) and denoted as *pi,j*, and its luminance value is named as *fi,j*,.DTBDM consists of two components: decision-tree-based impulse detector and edge-preserving image filter. The detector determines whether *pi,j* is a noisy pixel by using the decision tree and the correlation between pixel *pi,j* and its neighboring pixels. If the result is positive, edge-preserving image filter based on direction-oriented filter generates the reconstructed value. Otherwise, the value will be kept unchanged. The design concept of the DTBDM is displayed in Fig. 1.

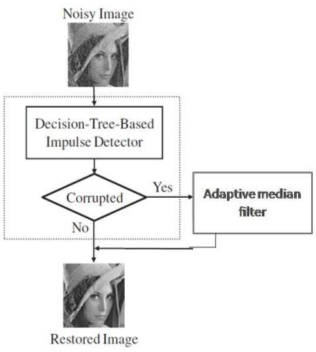

Fig.1 data flow of adaptive weight algorithm

Noise Filtering

If the input pixel is classified as an impulse *B*(*i*, *j*), the pixel value is replaced by the estimated central noise-free ordered median value. Otherwise, its original intensity is the output. However, the estimated median value here is computed from only the noise-free pixels within the filter window *w*(*k*). The noise-free pixels can be sorted in ascending order, and replace the noisy pixel by the median value. This is called Switching based median filter. Here the size of filtering window is adaptive in nature and its size is depend on the number of pixels which are noise free in current filtering window. The maximum window size shouldn't be more than 7×7 to reduce blurring effect. Steps are given below for Adaptive weight median Filtering:

1. Start with (3×3) filtering window form x (*i*, *j*) and corresponding (3×3) window from binary flag image f (*i,j)* .

2. Find out how many pixels are detected as noise free in current filtering window from corresponding binary flag window.

3. Iteratively extends window size outward by one pixel in all the four sides of the window, if the no. of uncorrupted pixels is less than half of the total number of pixels (denoted by Sin=1/2[3×3]) within the filtering window .These all above three steps should be repeat again if condition are not satisfy.

4. So since the current pixel has been marked noisy, it will not participate in filtering process. Only the pixels that are classified as noise free in filtering window will participate in median filtering process. This will, in turn, yield a better filtering result with less distortion.

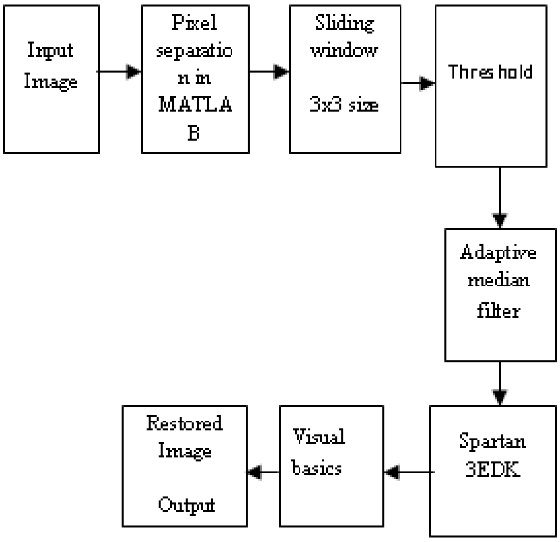

Fig. 2 blockdiagram

III.PROPOSED SYSTEM

In this proposed system a VLSI architecture is designed and implemented, which is to perform the impulse noise removal in images with high speed and reduced complexity. For that, the coprocessor Micro blaze is converted into adaptive weight algorithm using Xilinx platform studio in system C language and then tested in Spartan 3EDK FPGA kit. RS232 cable is used for interfacing the test circuit with PC. Instead of median filtering technique here adaptive weight median filtering (AWMF) method is used for the designing. This hardware implementation can overcome the shortages of previous works not only that it can achieve high speed in computation and low power consumption also.

A. Micro blaze processor design

The Micro Blaze embedded soft core is a reduced instruction set computer (RISC) optimized for implementation in Xilinx field programmable gate arrays (FPGAs). See Fig.9. for a block diagram depicting the Micro Blaze core. Field-programmable gate arrays (FPGA'S) are flexible and reusable high-density circuits that can be easily re-configured by the designer, enabling the VLSI design / validation /simulation cycle to be performed more quickly and less expensive. Increasing device densities have prompted FPGA manufacturers, such as Xilinx and Altera, to incorporate larger embedded components, including multipliers, DSP blocks and even embedded processors. One of the recent architectural enhancements in the Xilinx Spartan, Virtex family architectures is the introduction of the Micro Blaze (Soft IP) and PowerPC405 hard-core embedded processor.

The Micro blaze processor is a 32-bit Harvard Reduced Instruction Set Computer (RISC) architecture optimized for implementation in Xilinx FPGAs with separate 32-bit instruction and data buses running at full speed to execute programs and access data from both on-chip and external memory at the same time.

An interrupt controller is available for use with the Xilinx Embedded Development Kit (EDK) software tools. The processor will only react to interrupts if the Interrupt Enable (IE) bit in the Machine Status Register (MSR) is set to 1. On an interrupt the instruction in the execution stage will complete, while the instruction in the decode stage is replaced by a branch to the interrupt vector (address Ox 10). The interrupt return address (the PC associated with the instruction in the decode stage at the time of the interrupt) is automatically loaded into general-purpose register. In addition, the processor also disables future interrupts by clearing the IE bit in the MSR. The IE bit is automatically set again when executing the RTID instruction.

Due to the advancement in the fabrication technology and the increase in the density of logic blocks on FPGA, the use of FPGA is not limited to anymore to debugging and prototyping digital circuits. Due to enormous parallelism achievable on FPGA and the increasing density of logic blocks, it is being used now as a replacement to ASIC solutions in a few applications. Soft cores are technology independent and require only simulation and timing verification after synthesized to a target technology.

B. Xilinx platform studio

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system. B. Embedded Development Kit Xilinx Embedded Development Kit (EDK) is an integrated software tool suite for developing embedded systems with Xilinx Micro Blaze and PowerPC CPUs. EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional. A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation, software application creation, and software verification. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors]. The Platform Generation tool creates the hardware platform using the MHS file as input.

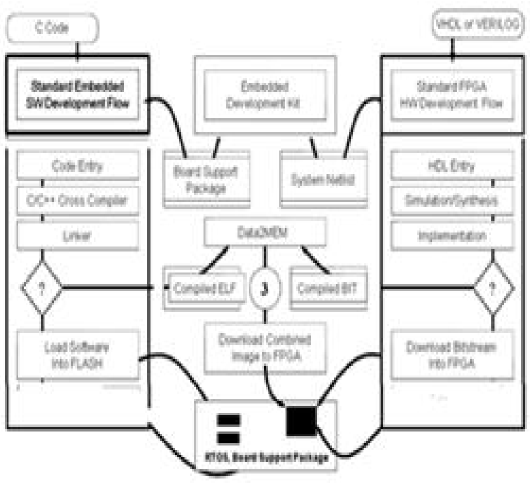

Fig.3. Embedded Development Kit Design Flow

The creation of the verification platform is optional and is based on the hardware platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific simulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK projects. Bit stream Initialize tool initializes the instruction memory of processors on the FPGA. GNU Compiler tools are used for compiling and linking application executables for each processor in the system . There are two options available for debugging the application created using EDK namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor. C. Software Development Kit Xilinx Platform Studio Software Development Kit (SDK) is an integrated development environment, complimentary to XPS, that is used for C/C++ embedded software application creation and verification. The software application can be written in a "C or C++" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

IV. SIMULATION RESULTS

Experiments are performed on gray level images to verify the proposed method. These images are represented by 8 bits/pixel and size is 128 x 128. Image used for experiments are shown in below figure. The architectures were implemented in system c and placed and routed on Xilinx spartan3 XC3S200 FPGA, using Xilinx platform studio v.10.The input image which is the original image is as shown in the Fig.4.The measurements used for proposed method are as follows: An often used global objective quality measure is the mean square error (MSE) defined as

MSE=((i/p-o/p)^2)/n*m

Where, nxm is the number of total pixels. f (i,j) and f(i,j)'are the pixel values in the original and reconstructed image. The peak to peak signal to noise ratio (PSNR in dB) is calculated as

PSNR=Progressive SNR=10log(255^2/MSE))

Usable gray level values range from 0 to 255.And the input noisy image after adding the impulse noise using MATLAB as shown in Fig.5.Then the Adaptive weight algorithm is applied to the noisy image. It first detects the possible noisy pixels by employing the impulse detector and reconstruct the intensity values of noisy pixels by employing the adaptive median filter. Then we get output noise free image which is as shown in Fig.6.

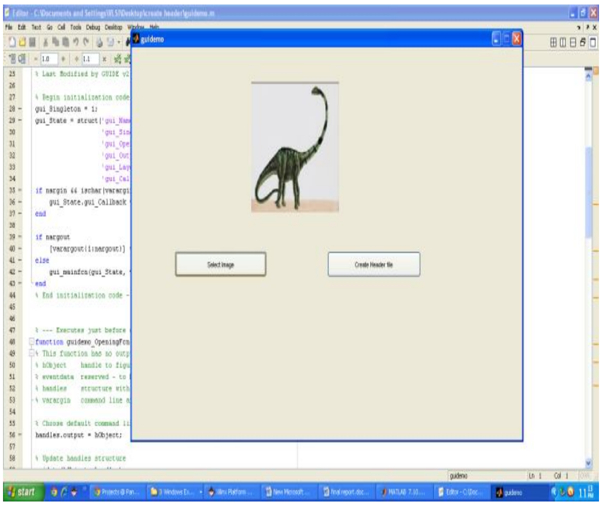

Fig.4 Original image

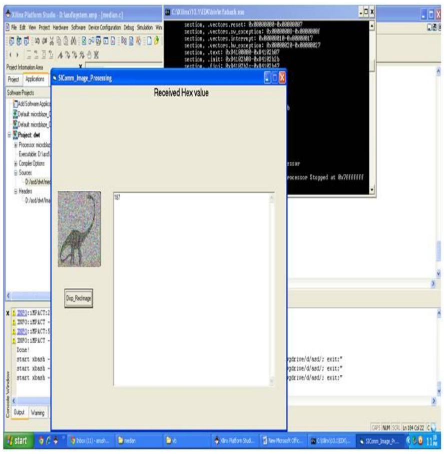

Fig.5 input noisy image(noise density 50%)

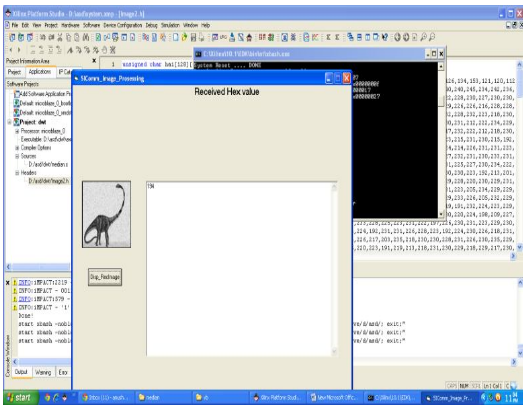

Fig.6 output image

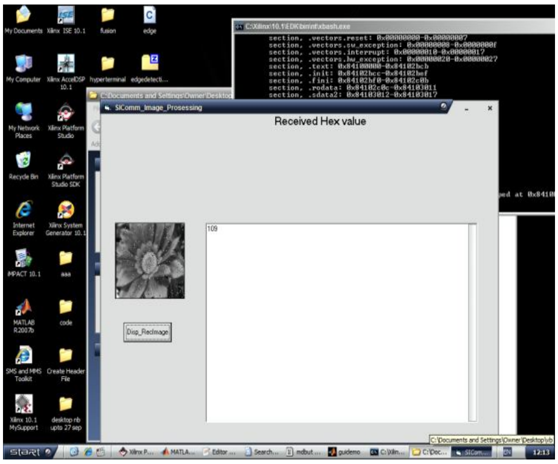

Fig.7 input noisy image(noise density 10%)

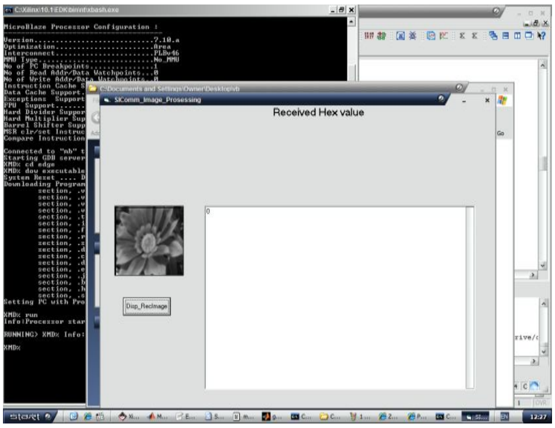

Fig.8 output image

V Conclusion

A low-cost VLSI implementation for efficient removal of random- valued impulse noise is proposed in this paper. The approach uses the decision based detector to detect the noisy pixel and employs an effective design to locate the edge.With adaptive skill,the quality of the reconstructed images is notable improved. Our extensive experimental results demonstrate that the performance of our proposed technique is better than the previous lower-complexity methods and is comparable to the higher-complexity methods in terms of both quantitative evaluation and visual quality. we have presented a new efficient decision-based filter called the adaptive weight median filter, for impulse noise removal. Because the new impulse detection mechanism can accurately tell where noise is, only the noise-corrupted pixels are replaced with the estimated central noise-free ordered mean value. As a result, the restored images can preserve perceptual details and edges in the image while effectively suppressing impulse noise.

VI REFERENCES

1. K. Arakawa, "Median filters based on fuzzy rules and its application to image restoration,"*Fuzzy Sets and Systems*, Vol. 77, 1996, pp. 3-13.

2. PENG Lei (ID: 03090345), "Adaptive Median Filtering", pp.2,pp.8, machine vision, 140.429, digital image processing.

3. Chih-Yuan Lien, Chien-Chuan Huang, Pei-Yin Chen, Yi-Fan Lin, "An Efficient Denoising Architecture for Removal of Impulse Noise in Images," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 631-643, April 2013, doi:10.1109/TC.2011.256

4. PENG Lei (ID: 03090345), "Adaptive Median Filtering",pp.2,pp.8, machine vision, 140.429, digital image processing.

5. E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," IEEE Transactions on Image Processing, Vol. 5, 1996, pp. 1012-10