WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AN EFFICIENT IMAGE DENOISING METHOD USING EDGE PRESERVING FILTER

Sathyakala D, Mandati Vinay, Kotaru Naga Malleswara Rao, Patlaveeti Jabeer, Yetukuri Aravind Kumar

Department of ECE, Dhanalakshmi Srinivasan Engineering College , Perambalur

ABSTRACT:

Images are often corrupted by impulse noise in the procedures of image acquisition and transmission. In this project, we propose an efficient denoising scheme and its VLSI architecture for the removal of random-valued impulse noise. To achieve the goal of low cost, a low-complexity VLSI architecture is proposed. We employ a decision-tree-based impulse noise detector to detect the noisy pixels, and an edge-preserving filter to reconstruct the intensity values of noisy pixels. Furthermore, an adaptive technology is used to enhance the effects of removal of impulse noise. Our extensive experimental results demonstrate that the proposed technique can obtain better performances in terms of both quantitative evaluation and visual quality than the previous lower complexity methods.Moreover, the performance can be comparable to the higher complexity methods. The design requires only low computational complexity and two line memory buffers. Its hardware cost is low and suitable to be applied to many real-time applications.This Design is implemented in FPGA XC3S 200 TQ-144 using Verilog HDL and simulated by Modelsim 6.4 c and Matlab Tool. Area, Delay analysis is done by Synthesize Xilinx tool.

Keywords: Impulse noise, VLSI architecture, Edge Preserving Filter, FPGA XC3S 200 TQ-144, Verilog HDL, Modelsim 6.4 c, Xilinx tool.

INTRODUCTION:

Image processing is widely used in many fields, such as medical imaging, scanning techniques, printing skills, license plate recognition, face recognition, and so on. In general, images are often corrupted by impulse noise in the procedures of image acquisition and transmission. The noise may seriously affect the performance of image processing techniques. Hence, an efficient denoising technique becomes a very important issue in image processing. According to the distribution of noisy pixel values, impulse noise can be classified into two categories: fixed-valued impulse noise and random-valued impulse noise. The former is also known as salt-and-pepper noise because the pixel value of a noisy pixel is either minimum or maximum value in grayscale images. Impulsive noise, which can be caused by malfunctioning camera photosensors, optic imperfections, electronic instability of the image signal, aging of the storage material, faulty memory locations in hardware or transmission errors due to natural or man-made processes. Noise removal researches have been performed in the area of image processing. Traditional method of noise removal is median filter.This filter is effective in impulse noise removal only for low density impulse noise.

PURPOSE OF THE PROJECT

Here we are going to see some points regarding to purpose behind choosing this topic. In this project, we propose an efficient denoising scheme. Our goal is to suppress noise, while preserving the image details. Our extensive experimental results demonstrate that the proposed technique can obtain better performances in terms of both quantitative evaluation and visual quality than the previous lower complexity methods. We are trying to get a better reconstructed image as output, so that its suitable to be applied to many real-time applications.

PROPOSED SYSTEM

The noise considered in this paper is random-valued impulse noise with uniform distribution. Here, we adopt a 3X3 mask for image denoising. The quality of denoised images of Edge preserving denoising algorithms, a variety of simulations are carried out on the six well-known 512 x 512 8-bit gray-scale test images. For a single test image, the corrupted versions of it are generated in Matlab environment with random-valued impulse noise at various noise densities from 5 to 20 percent with increments of 5 percent. Then, we employ different approaches to detect impulse noise and restore the corrupted image. Thus, we can easily compare the restored images with the source image for denoising methods. Our method Decision-tree-based detector (DTBDM) to detect the noisy pixel and employs an effective design to locate the edge.

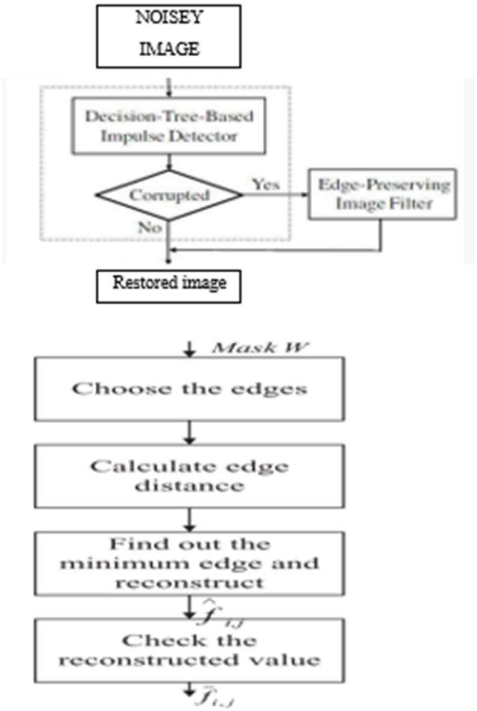


FIG.1.PROPOSED FLOW

IMAGE DENOISING

One of the fundamental challenges in the field of image processing and computer vision is image denoising, where the underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of the image. Image noise may be caused by different intrinsic and extrinsic conditions which are often not possible to avoid in practical situations. Therefore, image denoising plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification, where obtaining the original image content is crucial for strong performance. While many algorithms have been proposed for the purpose of image denoising, the problem of image noise suppression remains an open challenge.

PROPOSED ALGORITHM

- **Step 1:** Select 2-D window of size 3*3. Assume that the pixel being processed.
- **Step 2:** If then is an uncorrupted pixel and its value is left unchanged.
- **Step 3:** If or then is a corrupted pixel then two cases are possible as given in Case i) and ii).
- **Case i):** If the selected window contains all the elements as 0″s and 255″s. Then replace with the mean of the element of window.
- **Case ii):** If the selected window contains not all elements as 0″s and 255″s. Then eliminate 255″s and 0″s and find the median value of the remaining elements. Replace with the median value.
- **Step 4:** Repeat steps 1 to 3 until all the pixels in the entire image are processed.

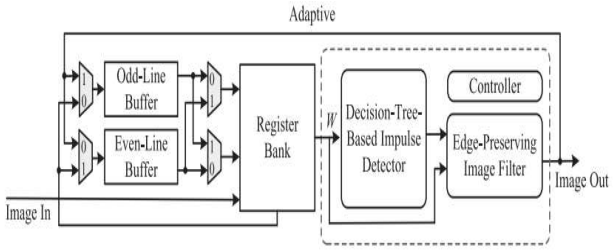


FIG.2.BLOCK DIAGRAM

PROPOSED SYSTEM TECHNIQUE

The architecture of DTBDM shown in fig consists of five main blocks:

- Line Buffer
- Register Bank
- DTBID
- Edge Preserving Image Filter
- Controller

LINE BUFFER

DTBDM adopts a 3×3 mask, so three scanning lines are needed. Therefore mask contains 9 pixels. The centre pixel i.e 5^{th} pixel is processed.

REGISTER BANK

It consists of nine registers to store the 3×3 pixel values of the current mask.

DECISION TREE BASED IMPULSE DETECTOR

In order to determine whether the centre pixel of the current mask is noisy or not, the correlations between the centre pixel and its neighbouring pixels in the mask are considered. This DTBID consist of three modules namely i) Isolation module, ii) Fringe module and iii) Similarity module. Three concatenating decisions of these modules build a decision tree. The decision tree is a binary tree and can determine the status of centre pixel of the mask as noisy or noise-free. If the result is positive then the pixel is noisy. Otherwise if the result is negative the pixel is considered to be noise free. Isolation module is used to decide whether the pixel is in smooth region or not. The Fringe module is used to decide whether the pixel is situated at the fringe or not. Finally if both isolation and fringe module fails to detect the noise the similarity module is used to decide the result.

EDGE PRESERVING IMAGE FILTER

The pixel value of noise-free pixel is unaltered. whereas the pixel value of noisy pixel is reconstructed using the Edge Preserving Image Filter. It consist of two modules i) minED generator and ii) Average generator. The minED generator generates the edge with minimum value out of eight directions. Then the pixels in minimum value direction is used to reconstruct the value of a noisy pixel. The median value is used for reconstruction of noisy pixel if all other pixels in mask are also noisy.

CONTROLLER

It sends control signals to control pipelining and timing statuses of the VLSI DTBDM circuit. It also sends signals to schedule reading and writing statuses of the data that are stored in register bank or in line buffers. The pixels are read row by row in a raster scan order. For a 3×3 sliding window, two First-In-First-Out (FIFO) buffers are used. The FIFO buffers are used to reduce the memory access to one pixel per clock cycle. The depth of the FIFO buffer is chosen as (W-3), where W is the width of the image. In order to access all values of the window for every clock cycle, the two FIFO buffers must be full. The architecture places the lowest demand on external memory bandwidth, but the highest demand on internal memory bandwidth. This approach does not cause problems, as FPGA devices contain large amount of embedded memory.

PROPOSED SYSTEM ADVANTAGES

- Low-power Design
- Low-energy high-throughput hardware
- Digital image proseeing made digital image can be noise free.
- It can be made available in any desired format. (X-rays, photo negatives, improved image, etc)
- Digital imaging is the ability of the operator to post process the image .It means manipulate the pixel shades to correct image density and contrast.
- Images can be stored in the computer memory and easily retrieved on the same computer screen.

SOFTWARE REQUIREMENT

- **MODELSIM 6.4c**
- **XILINX ISE 13.2**
- **MATLAB**

MODELSIM 6.4c

Modelsim SE - High Performance Simulation and Debug

ModelSim SE is our UNIX, Linux, and Windows-based simulation and debug environment, combining high performance with the most powerful and intuitive GUI in the industry.

XILINX ISE 13.2

It is used as synthesis tool.

XILINX ISE Design Tools:

Xilinx ISE is the design tool provided by Xilinx. Xilinx would be virtually identical for our purposes.

There are four fundamental steps in all digital logic design. These consist of:

1. Design – The schematic or code that describes the circuit.
2. Synthesis – The intermediate conversion of human readable circuit description to FPGA code (EDIF) format. It involves syntax checking and combining of all the separate design files into a single file.
3. Place & Route – Where the layout of the circuit is finalized. This is the translation of the EDIF into logic gates on the FPGA.
4. Program – The FPGA is updated to reflect the design through the use of programming (.bit) files.

MATLAB

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development, including Graphical User Interface building

The MATLAB system consists of five main parts:

- The MATLAB language
- The MATLAB working environment
- Handle Graphics
- The MATLAB mathematical function library
- The MATLAB Application Program Interface (API)

VERILOG HDL

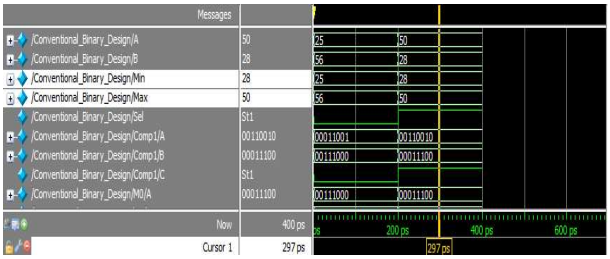
Verilog HDL is a Hardware Description Language (HDL). A Hardware Description Language is a language used to describe a digital system, for example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an Integrated Circuit (IC) chip, i. e., the switch level. Or, it might describe the logical gates and flip flops in a digital system, i. e., the gate level. An even higher level describes the registers and the transfers of vectors of information between registers. This is called the Register Transfer Level (RTL). Verilog supports all of these levels. However, this handout focuses on only the portions of Verilog which support the RTL level.

VERILOG HDL

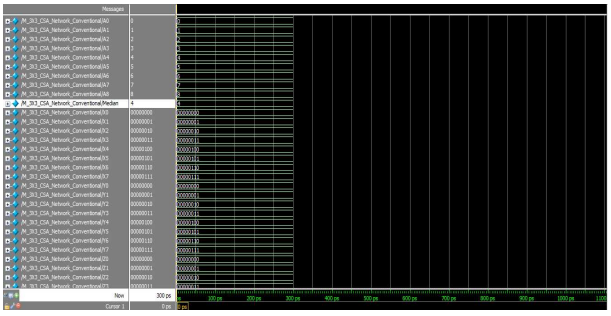
Verilog HDL is a Hardware Description Language (HDL). A Hardware Description Language is a language used to describe a digital system, for example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an Integrated Circuit (IC) chip, i. e., the switch level. Or, it might describe the logical gates and flip flops in a digital system, i. e., the gate level. An even higher level describes the registers and the transfers of vectors of information between registers. This is called the Register Transfer Level (RTL). Verilog supports all of these levels. However, this handout focuses on only the portions of Verilog which support the RTL level.

RESULTS:

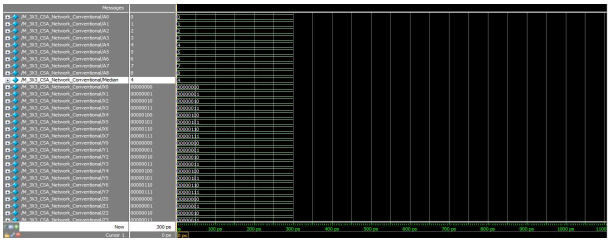
CONVENTIONAL BINARY DESIGN



CONVENTIONAL CAS DESIGN

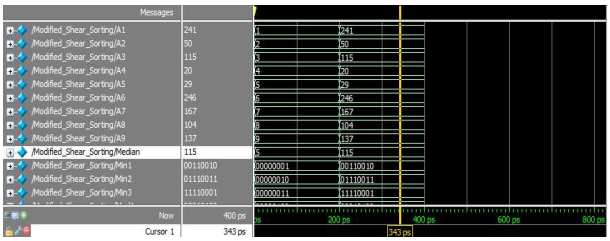


CSA 3X3 Network Conventional

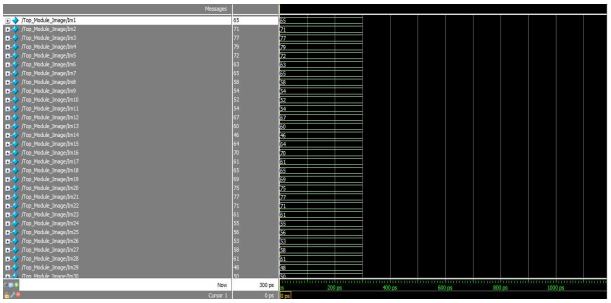


MAIN COMPARTOR

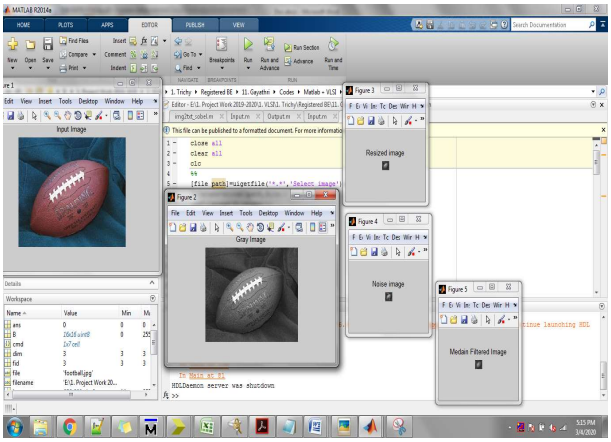
PROPOSED SHEAR SORTING BASED MEDIAN FILTER



VERILOG MEDIAN OUTPUT



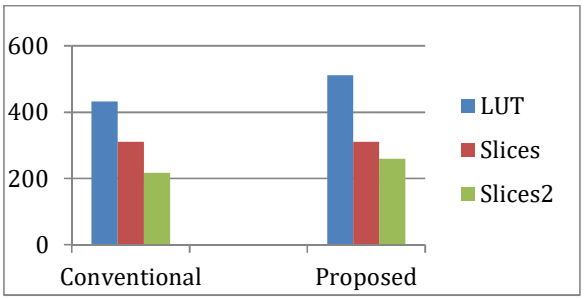
OVERALL OUTPUT



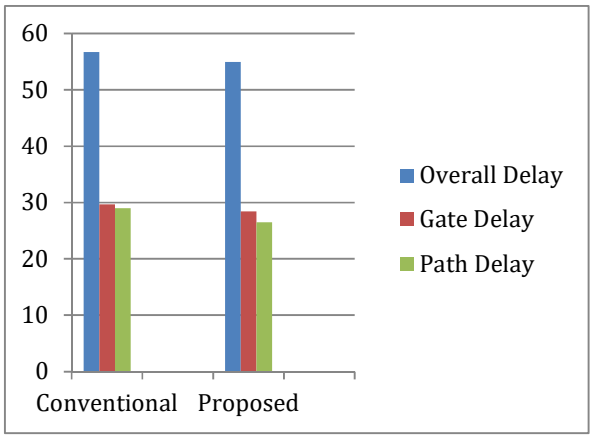
SYNTHESIS REPORT OF PROPOSED SYSTEM

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	511	66,560	1%	
Logic Distribution				
Number of occupied Slices	259	33,280	1%	
Number of Slices containing only related logic	259	259	100%	
Number of Slices containing unrelated logic	0	259	0%	
Total Number of 4 input LUTs	512	66,560	1%	
Number used as logic	511			
Number used as a route-thru	1			
Number of bonded IOBs	80	633	12%	
Total equivalent gate count for design	3,849			
Additional JTAG gate count for IOBs	3,840			

AREA COMPARISON



DELAY COMPARISON



AREA AND DELAY COMMPARISON TABLE

METHOD NAME	AREA IN NUMBER OF LUT			DELAY		
NAME	LUT	GATE COUNT	SLICES	DELAY	LOGIC DELAY	ROUTE DELAY
CONVENTIONAL	432	311	217	58.688 ns	29.693 ns	28.995 ns
PROPPOSED	511	3849	259	54.946 ns	28.442 ns	26.504 ns

CONCLUSION:

A low-cost VLSI architecture for efficient removal of random-valued impulse noise is proposed in this paper. The approach uses the decision-tree-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. The VLSI architecture of our design yields a processing Spartan 3 XC 3S200 TQ144 technology. It requires only low computational complexity and two line memory buffers. Therefore, it is very suitable to be applied to many real-time applications.

REFERENCES:

1. 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.
2. N.I. Petrovic and V. Crnojevic, “Universal Impulse Noise Filter Based on Genetic Programming,” IEEE Trans. Image Processing, vol. 17, no. 7, pp. 1109-1120, July 2008.
3. 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.
4. A.S. Awad and H. Man, “High Performance Detection Filter for Impulse Noise Removal in Images,” IEEE Electronic Letters, vol. 44, no. 3, pp. 192-194, Jan. 2008.
5. X. Zhang and Y. Xiong, “Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter,” IEEE Signal Processing Letters, vol. 16, no. 4, pp. 295- 298, Apr. 2009.

6. Synopsys Inc., “Design Ware Building Block IP,” http://www. synopsys.com, 2012.
7. T. Matsubara, V.G. Moshnyaga, and K. Hashimoto, “A FPGA Implementation of Low-Complexity Noise Removal,” Proc. 17th IEEE Int’l Conf. Electronics, Circuits, and Systems (ICECS ’10), pp. 255-258, Dec. 2010.
8. 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.