Nidhi Verma, 2024, 12:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Implementation of Wavelet Filter for Image Denoising and Analysis of Performance of Parameters

Scholar Nidhi Verma, Professor Dr Bharti Chourasia
EC Department
RKDF institute of Science & Technology Constituent unit of SRKU

Abstract- During acquisition or transmission, motive noise taints image and video signals. An effective and user-friendly motivating commotion ejection process is therefore necessary. Due to the disturbance, the main exhibition is closed or occasionally there is disappointment. An effective VLSI filter execution with execution upgrade for picture denoising is presented in this paper. The test images vary in size and purpose. The Daubechies 6-tap wavelet filter wavelets are found to beat the approximated denoising execution, which is unbiasedly top sign to commotion proportion and abstractly visual nature of picture. MATLAB and VLSI-Xilinx 14.7 software are used for simulation. The filter architecture will be adjusted with Xilinx version-14.7, and the imaging process will be visualized with MATLAB software.

Keywords- that Daubechies, wavelet filter, VLSI, DWT

I. INTRODUCTION

In Digital Image Processing computer algorithm are used to improve the quality of image and extract fruitful information for further processing. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image-based tasks. The principle advantage of Digital Image Processing methods is its versatility, repeatability and the preservation of original data precision. A two-dimensional function, F(x,y), is defined as follows: x and y are spatial coordinates; the amplitude of F at any pair of coordinates (x,y) represents the image's intensity at that location. A digital image is one where the values of F's amplitude, x, and y are all finite. The effect of digitization is shown in Figure 1

Despite their higher cost compared to analog filters of the same kind, digital filters enable numerous designs that would be unfeasible or unworkable with simpler analog filters. Finite impulse response filters, which enable linear phase response, are among the many high order digital filters that can be created. Sometimes, due to the associated analog-to-digital and digital-to-analog conversions, anti-aliasing filters, or other delays in their implementation, digital filters used in real-time analog systems have problematic latency.

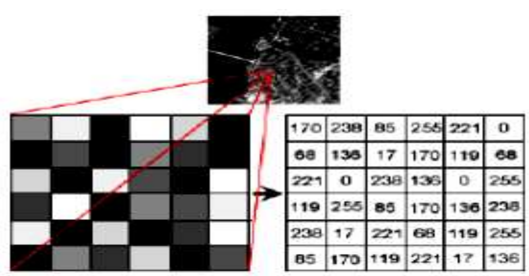

Figure 1: Digital Image

1. Discrete Wavelet Transform

Using the Discrete Wavelet Transform approach, image pixels can be converted into wavelets, which can then be utilized for wavelet-based coding and compression. Discrete wavelet transform (DWT) is the name given to any wavelet transform for which the wavelets are sampled sequentially. Similar to other wavelet transforms, temporal resolution is a major benefit over Fourier transforms since it records both frequency and location data (location in time). Wavelets are functions that enable data analysis of images or signals based on resolutions or scales.

2. Haar wavelets

Lossless and lossy picture compression can be effectively achieved with Haar wavelet compression. It creates a sparse or almost sparse matrix by averaging and differencing data in an image matrix. Haar, a Hungarian mathematician, created the first DWT.

3. Daubechies wavelets

The discrete wavelet transformations that are most frequently employed were developed in 1988 by Ingrid Daubechies, a mathematician from Belgium. This formula is depends up on producing increasingly more discrete samplings of an implicit base wavelet function, twice as fine as the preceding scale, using recurrence relations.

II. PROPOSED METHODOLOGY

In this section we discuss the proposed methodology. The flow chart of methodology is given below figure no 2. The proposed work is divided into the following sub modules-Image initialization, Extract Feature points, Implement WAVELET, Implement, WAVELET, Apply inverse Feature transform, Compute MSE and PSNR.

1. Image Initialization

In this module, we do preprocessing steps for image such as Image noise reduction, Histogram equalization and Image size estimation.

In this we extract the pixels in the given image and arrange it in a matrix. Here we detect whether the input image was single sample image or not. If it was more than one sample image, then this was converted into single sample intensity image (Gray scale image).

Figure 2: Flow Chart

2 Extract Feature Points

Initially we acquire the binary points from the input image and produce the features by applying Feature transform in a range of. This Feature transform denotes the image representation as a collection of projections along different directions.

The Feature transform encompasses a Feature function that computes the projections of an image along specified directions of axis and axis.

$$R_\theta(x') = \int_{-\infty}^{\infty} f\,(x'\cos\theta - y'\sin\theta, x'\sin\theta + y'\cos\theta)\,dy'$$

A projection in Feature transform is computed using the equation 3 at an angle. θ.

$$\text{Here}\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}.$$

The Feature points are obtained by applying Feature transform using Feature function.

3. Implement Wavelet

In the encoding process of WAVELET, the image is broken down into K*K blocks of pixels, where K denotes 2, 4, 6, etc. The WAVELET computation for a sequence f(i) of length K. The WAVELET coefficients are obtained from each block of input data. Then we encode the Feature points by applying this computed WAVELET. WAVELET is having best energy compaction capability for highly correlated images.

4. Implement IWavelet

At the receiver, the projections are retrieved (decoded) by IWAVELET and is used to reconstruct the image.

$$D(u) = \alpha(u)\sum_{i=0}^{K-1} f(i)\cos\left[\frac{\pi(2i+1)u}{2K}\right]$$

Here, u ranges from 0, 1 ...K-1 and the WAVELET coefficients is D (u). The inverse WAVELET (IWAVELET) is expressed as,

$$f(i) = \sum_{u=0}^{K-1} \alpha(u)D(u)\cos\left[\frac{\pi(2i+1)u}{2K}\right]$$

5. Apply Inverse Feature Transform

Explicit and computationally efficient inversion formulas for the Feature transform and its dual are available. The Feature transform in n dimensions can be inverted by the formula and the power of the Laplacian (−Δ)(n−1)/2 is defined as a pseudo differential operator if necessary by the Fourier transform.

Then finally we obtain the reconstructed image.The proposed work is Daubechies 6-tap wavele technique used for the wavelet transform. The working procedure of this proposed scheme is halved into following process namely, selection of input image (Original image), and acquisition of binary points, apply Feature transform to obtain the Feature points. The design approach of the proposed framework to minimize the adaptation delay in the error-calculation block.

III. SIMULATION AND RESULTS

For simulation we used MATLAB software which is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. And for Implementation Xilinx ISE (Coordinated Mix Condition) is used.

Figure 3: Sample input image

Figure 4: Image Initialization

Figure 5: Extract Features points

By Xilinx for blend and examination of HDL structures, empowering the specialist to incorporate their arrangements, perform timing examination, investigate RTL graphs, recreate an arrangement's reaction to different redesigns, and organize the target contraption with the product engineer.

The Xilinx® ISE Test framework (ISim) is a hardware description language (HDL) test framework that empowers you to perform valuable and timing amusements for VHDL, Verilog and mixed vernacular structures.

Figure 3 presents the various input sample images, which is already consist in the system picture library. and Figure 4 is showing the image initialization steps, here browse the picture and it comes in the MATLAB environment. Figure 5shows the extract the image features or pixels. Feature points (read corners) in images are points that invariant under view changes, zoom, lightening conditions etc.

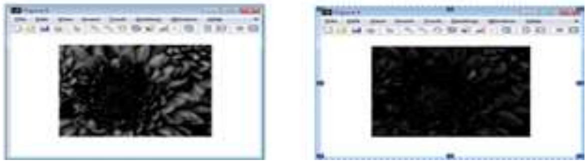

Figure 6: (a) Reshape Image (b) Transformed Image

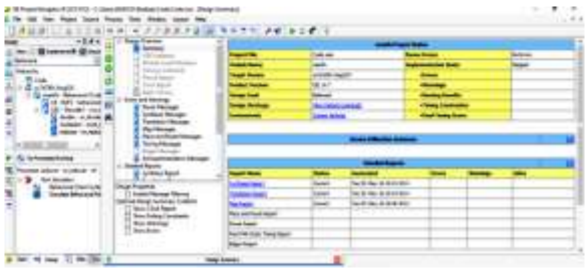

Figure 7: Xilinx software initialization

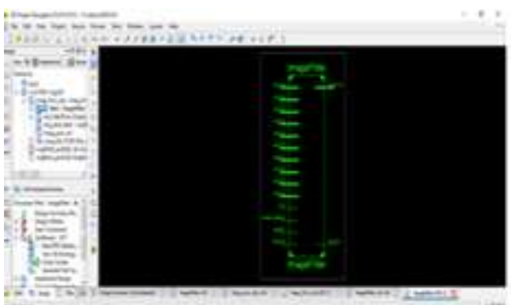

Figure 8: Top module of filters in xilinx environment

Figure 9: Complete RTL View

Figure 6 is provind reshape image and the rransformed image after the preprocessing step of the image.

Figure 7 is showing the xilinx initialization window and opens the filter files. The windows show the all files, implementation and simulation tabs. Figure 8 is showing the top module of the filter design, where see the various input and output combinations. Figure 9 is showing the complete register transfer level view of the proposed filter for image processing

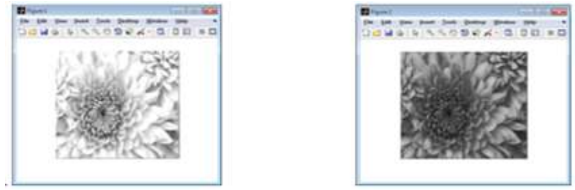

Figure 10: (a) Output resized image (b) Gray Image

Figure 10 shows the resized and the gray image, Grayscaling is the process of converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white. Figure 11 is showing the image transformed and the filtering image view. Image filtering is changing the appearance of an image by altering the colors of the pixels. Increasing the contrast as well as adding a variety of special effects to images are some of the results of applying filters.

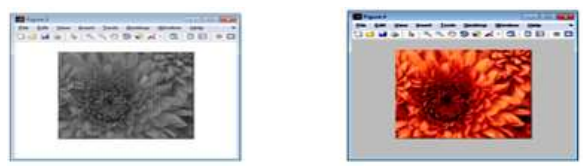

Figure 11: (a) Inverse transformed image (b) Output filtered image

Figure 12: Xilinx Summary report

Figure 12 is presenting the Xilinx summary report of the proposed filter design VLSI architecture.

Figure 13 Xilinx synthesis report


Figure 14 Timing details

Area or number of component

Table1 is showing summary of components using duration proposed approximate multiplier implementation. Total number of slice look up table used 418 while availability is 63400. Look up table and flip flop pairs used 254 while availability is 513. Bonded input output block used 57 while availability is 210. Now total area is calculated from this utilization summary. Therefore, 11.71% area is used for implementation of proposed digital filter.

Table 1: Device utilization summary

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	349	126800	0%
Number of Slice LUTs	418	63400	0%
Number of fully used LUT-FF pairs	254	513	49%
Number of bonded IOBs	57	210	27%
Number of Block RAM/FIFO	1	135	0%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	8	240	3%

Slice Logic Utilization:
Number of Slice Registers: 349 out of 126800 0%
Number of Slice LUTs: 418 out of 63400 0%
Number used as Logic: 418 out of 63400 0%
Slice Logic Distribution:
Number of LUT Flip Flop pairs used: 513
Number with an unused Flip Flop: 164 out of 513 31%
Number with an unused LUT: 95 out of 513 18%
Number of fully used LUT-FF pairs: 254 out of 513 49%
Number of unique control sets: 20
IO Utilization:
Number of IOs: 57
Number of bonded IOBs: 57 out of 210 27%

Specific Feature Utilization:
Number of Block RAM/FIFO: 1 out of 135 0%
Number using Block RAM only: 1
Number of BUFG/BUFGCTRLs: 1 out of 32 3%
Number of DSP48E1s: 8 out of 240 3%
Power


Figure 15 X-power Analyzer

Figure 15 is showing Xilinx power analyzer. There is showing various family of FPGA IC like artix, kintex, spartan, vertex etc available. Input power and consume power is in milli W. Therefore the total average consume power is approximate 0.082W.

Delay or Latency

Total number of paths / destination ports: 9106 / 889

Delay: 4.419ns (Levels of Logic = 0)
Source: t_2/temp_7 (FF)
Destination: kernel0/Mmult_m8[3]_GND_15_o_MuLt_15_OUT (DSP)
Source Clock: CLK rising
Destination Clock: CLK rising
Cell:in->out fanout Delay Delay Logical Name (Net Name)
-- ------------
IBUF:I->O 22 0.001 0.535 m8_3_IBUF (m8_3_IBUF)
DSP48E1:A3 3.722 kernel0/Mmult_m8[3]_GND_15_o_MuLt_15_OUT

Total 4.258ns (3.723ns logic, 0.535ns route)
(87.4% logic, 12.6% route)

```
==================================
================================
Total number of paths / destination ports: 9 / 9
 Data Path: kernel0/RD_EN to RD_EN
                              Gate     Net
Cell:in->out     fanout   Delay   Delay  Logical Name
(Net Name)
 ----------------------------------------  ------------
FDC:C->Q             4   0.478   0.419  kernel0/RD_EN
(kernel0/RD_EN)
OBUF:I->O                   0.000          RD_EN_OBUF
(RD_EN)
 ----------------------------------------
Total                     0.897ns (0.478ns logic, 0.419ns
route)
                                   (53.3% logic, 46.7% route).
```

Memory

Total REAL time to Xst completion: 34.00 secs
Total CPU time to Xst completion: 34.92 secs
Total memory usage is 4649268 kilobytes

Table 2: Simulation Results

Sr No.	Parameter	Value
1	Method	Wavelet filter
2	Area	11.71 mm^2
3	Delay	0.897 ns
4	Power	0.082 W
5	Power Delay product	81.5
6	Frequency	1114 MHz
7	Throughput	89120000 pixels/sec

Table 3: Comparison of simulation results

Sr No.	Parameter	Previous Work [1]	Proposed Work
1	Filter Type	Bilateral Filter	Wavelet Filter
2	Delay	NA	0.897 ns
3	Frequency	236.697 MHz	1114 MHz
4	Slice look up table	5142	418
5	Fully used look up-flip flop pair	1782	254
6	Bounded I/O boxes	69	57
7	Number of DSP48E1s	36	8
8	Throughput	59171103 pixels/sec	89120000 pixels/sec

Table.2 showing comparison of proposed work with previous work, so it can be seen that proposed work gives better result than existing work.

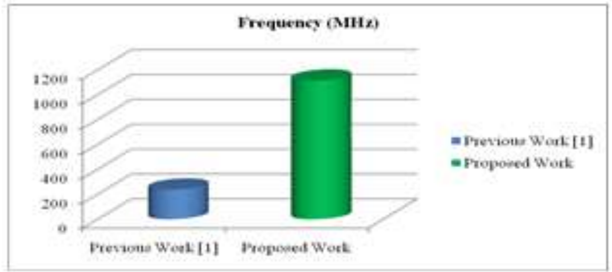

Figure 16: Frequency

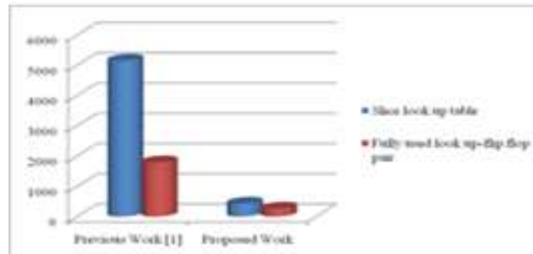

Figure 17: Look up table

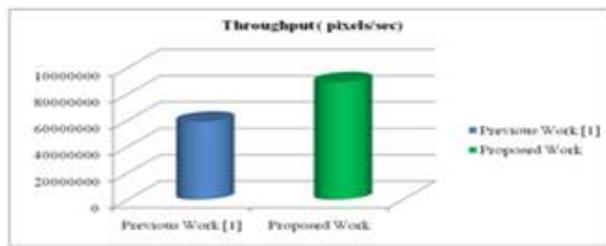

Figure 18: Comparison of throughput

Figure 16 is showing the comparison of the frequency used by the filter operation. So the proposed filter achieves the better frequency than existing. Figure 17 is showing the comparison of the look up table used by the filter operation. So the proposed filter uses less number of components than existing. Figure 18 presents the comparison of the throughput or data speed. The proposed digital filter achieves the 89120000 pixels/sec while previous it is 59171103 pixels/sec.

V. CONCLUSION

The orthogonal filter with DWT is used as a tool to generate feature points and speed up the encoding process in this study, which describes the construction of a VLSI filter for image denoising with performance enhancement. We investigate the use of DWT for encoding feature points and IDWT for decoding feature points. The proposed method performs better than the current DWT with Orthogonal Wavelet filtering picture denoising technique. The performance of our suggested picture denoising technique, which achieves better parameter values than the current image compression technique, is revealed by the experimental findings. We plan to change the internal adder design in the future to carry out wavelet filter architecture propagation. The goal of the filter architecture optimization is to increase the level of filtering performance while lowering power consumption and delay in comparison to the suggested approach. The image filtering method reduces the amount of time needed for the filter architecture. The frequency attained by the suggested filter is 1114 MHz, as opposed to 236.697 MHz with the previous filter. There are 418 lookup tables in the slice, 254 fully utilized lookup-flip flop pairs, 57 bounded I/O boxes, and 8 DSP48E1s. The current work achieves a total throughput of 89120000 pixels/sec, compared to the previous work's 59171103 pixels/sec.

REFERENCES

1. Vulchi, Lakshmi & Aakash, Gonugundla & Kumar, Duvvala & Valiveti, Hima. (2023). Performance Analysis of Real and Complex Wavelet Transform Techniques used for Wavelet-Based Image Denoising. 1-7. 10.1109/ViTECoN58111.2023.10157293.
2. Vilimek D, Kubicek J, Golian M, Jaros R, Kahankova R, Hanzlikova P, Barvik D, Krestanova A, Penhaker M, Cerny M, Prokop O, Buzga M. Comparative analysis of wavelet transform filtering systems for noise reduction in ultrasound images. PLoS One. 2022 Jul 7;17(7):e0270745. doi: 10.1371/journal.pone.0270745. PMID: 35797331; PMCID: PMC9262246.
3. C. Lien, C. Tang, P. Chen, Y. Kuo and Y. Deng, "A Low-Cost VLSI Architecture of the Bilateral Filter for Real-Time Image Denoising," in IEEE Access, vol. 8, pp. 64278-64283, 2020, doi: 10.1109/ACCESS.2020.2984688.
4. K. Rajini and K. Vasanth, "Area Efficient and Low Power Multiplexer based Data Comparator for Median filter in Denoising Application," 2020 International Conference on Communication and Signal Processing (ICCSP), 2020, pp. 0761-0765, doi: 10.1109/ICCSP48568.2020.9182116.
5. M. Monajati and E. Kabir, "A Modified Inexact Arithmetic Median Filter for Removing Salt-and-Pepper Noise From Gray-Level Images," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 4, pp. 750-754, April 2020, doi: 10.1109/TCSII.2019.2919446.
6. P. Sendamarai and M. N. Giriprasad, "FPGA implementation of combined compression & denoising scheme for remote sensing images," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016, pp. 401-404, doi: 10.1109/RTEICT.2016.7807850.
7. A.Mekkalaki and M. Kulkarni, "Design of 16∗16 high speed MLM using KOM for image filter applicaton," 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), 2016, pp. 3504-3509, doi: 10.1109/ICEEOT.2016.7755356. Kamarujjaman, M. Mukherjee and M. Maitra, "An efficient FPGA based de-noising architecture for removal of high density impulse noise in images," 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2015, pp. 262-266, doi: 10.1109/ICRCICN.2015.7434247.
8. R. Pushpavalli and G. Sivarajde, "An intelligent post processing technique for image enhancement," 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013, pp. 1-5, doi: 10.1109/ICEVENT.2013.6496555.
9. P. Deepa and C. Vasanthanayaki, "VLSI Implementation of Enhanced Edge Preserving Impulse Noise Removal Technique," 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems, 2013, pp. 98-102, doi: 10.1109/VLSID.2013.170.
10. C. Lien, C. Huang, P. Chen and Y. Lin, "An Efficient Denoising Architecture for Removal of Impulse Noise in Images," in IEEE Transactions on Computers, vol. 62, no. 4, pp. 631-643, April 2013, doi: 10.1109/TC.2011.256.

11. K. Vasanth, S. N. Raj, S. Karthik and P. P. Mol, "Fpga implementation of optimized sorting network algorithm for median filters," INTERACT-2010, 2010, pp. 224-229, doi: 10.1109/INTERACT.2010.5706144.
12. C. Hsu, C. Lan, Y. Lo and Y. Huang, "Adaptive De-noising Filter Algorithm for CMOS Image Sensor Testing Applications," 2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems, 2010, pp. 136-143, doi: 10.1109/DFT.2010.23.
13. C. Lien, P. Chen, L. Chang, Y. Lin and P. Chang, "An efficient denoising chip for the removal of impulse noise," Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 1169-1172, doi: 10.1109/ISCAS.2010.5537308.
14. P. Chen, C. Lien and H. Chuang, "A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 3, pp. 473-481, March 2010, doi: 10.1109/TVLSI.2008.2012263.
15. T. Q. Vinh, J. H. Park, Y. Kim and S. H. Hong, "FPGA Implementation of Real-Time Edge-Preserving Filter for Video Noise Reduction," 2008 International Conference on Computer and Electrical Engineering, 2008, pp. 611-614, doi: 10.1109/ICCEE.2008.61.
16. Pei-Yin Chen, Chih-Yuan Lien and Yi-Ming Lin, "A real-time image denoising chip," 2008 IEEE International Symposium on Circuits and Systems, 2008, pp. 3390-3393, doi: 10.1109/ISCAS.2008.4542186.
17. C. Hsieh, T. Tsai, S. Chang and T. Shan, "A Cost-Effective Noise-Reduction Filtering Structure Based on Unsymmetrical Working Windows," Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), 2007, pp. 527-530, doi: 10.1109/IIH-MSP.2007.8.
18. A.Hazra, J. Bhattacharyya and S. Banerjee, "Real time noise cleaning of ultrasound images," Proceedings. 17th IEEE Symposium on Computer-Based Medical Systems, 2004, pp. 379-384, doi: 10.1109/CBMS.2004.1311743.
19. K. Rocha, A. Venkatachalam, T. Bose and R. L. Haupt, "Multiplierless algorithms for high-speed real-time onboard image processing," Proceedings, IEEE Aerospace Conference, 2002, pp. 4-4, doi: 10.1109/AERO.2002.1036901.