Design of a 2D Median Filter with a High Throughput FPGA Implementation

Anish Goel, *Student Member, IEEE*, M. Omair Ahmad, *Fellow, IEEE*, and M.N.S. Swamy, *Fellow, IEEE*
Department of Electrical and Computer Engineering
Concordia University, Montreal, QC, Canada H3G 1M8
email: {an_goe, omair, swamy}@ece.concordia.ca

***Abstract* — In this paper, a hybrid technique for median filtering of images affected by impulse noise is proposed. Our technique combines impulse noise detection, histogram-based median calculation and bit-plane processing to obtain approximate median with the aim of optimizing the throughput at minimum cost of image quality. The proposed median filter is implemented on FPGA with pipelining and is significantly faster than existing FPGA based pipelined median filter architectures. Implementation of the proposed median filter hardware provides a throughput of 282 Full High Definition (FHD) frames per second on Zynq-7 FPGA; 48% higher than the throughput of low-latency median filter. Compared to FPGA implementation of a low complexity noise removal, the proposed median filter utilizes only 45% of FPGA slices and provides a speed-up of 2.2 on Zynq-7 FPGA.**

***Keywords*— Noise Detection, Bit-Planes, Histogram-Based, Pipelining, Approximate Median.**

I. INTRODUCTION

Median filtering is one of the most commonly used technique for removal of impulse noise in images [1]. The defect in sensing or capturing device, memory corruptions and shot noise affects an image in such a way that some pixel values are set to minimum while some are set to maximum; a phenomenon that characterizes salt and pepper noise. For recovery of original image from a corrupted image, several techniques have been proposed where the best match for each corrupted pixel is calculated. Finding a pixel from the neighborhood or a close approximation based on the neighborhood pixels to replace the corrupted pixel is the typical approach in image filtering. This approach is characterized by a 2-Dimentional (2D) window, where pixels surrounding the noisy pixel are processed. Several algorithms and techniques have tried to solve the problem of retrieving the corrupted pixels to achieve high quality of filtered images. However, it may be interesting to note that the actual value of all the corrupted pixels may never be known. This is supported by the fact that none of such algorithms have resulted in infinite Peak Signal to Noise Ratio (PSNR) between the original and filtered image, in-spite of low noise density.

For implementation of median filtering algorithms on hardware platforms such as FPGAs, performance parameters like area, frequency and power are equally important as the quality of filtered image. Of these, frequency of the design plays a vital role in many real-time applications. The minimum frequency requirement also known as the 'pixel clock' for a real-time Full High Definition (FHD) vision system at a rate of 60 **f**rames **p**er **s**econd (fps) is 148.5 MHz [2], which must be satisfied by every block of the vision system. For offline processing of massive data, even higher frequencies are desired.

The throughput of algorithms that use sorting-based median filters [3] [4] is limited, as these architectures use cascaded stages of sorting blocks (compare and swap) for median calculation. A Low Energy Adaptive Median Filter (LEAMF) architecture presented in [5] processes only higher 4-bits of pixel values for median calculation. Although LEAMF processes only 4-bits of pixel values, its FPGA implementation results in lower throughput compared to the adaptive median filter presented in [4] which utilizes all 8-bits. On the other hand, FPGA implementation of a histogram-based technique [6] results in large area and low operating frequency due to its design complexity. The quality of images filtered by techniques [3-6] is equivalent to filtered image quality of Conventional Median Filter (CMF) [7]. On the other hand, median filter based on a low complexity noise removal technique [8] provides high quality of filtered images by filtering only noisy pixels (decision-based filtering). Work presented in [9] is a bit-plane processing architecture that processes N number of W-bit integers in smaller blocks of B-bits to achieve area and speed optimization. In the proposed work, we design only one block for processing higher B-bits and ignore the remaining $(W-B)$ lower bits. Although ignoring lower bits compromises the filtered image quality by a small amount, the improvement achieved in hardware performance is significantly higher to balance this trade-off.

The quality of a filtered image can be judged qualitatively by visualizing the image and quantitatively by calculating its PSNR. In general, higher PSNR shows a better approximation to the original image. Qualitatively, there is no significant loss of information even if some of the pixels of a noisy image are not retrieved to near-original values. Following this concept, techniques based on approximate arithmetic [10] have been implemented to achieve optimization in area and power at the cost (< 1dB) of PSNR.

This work was supported in part by the Natural Science and Engineering Research Council (NSERC) of Canada and in part by the Regroupement Stratégique en Microélectronique du Québec (ReSMiQ).

In this paper we implement a 2D decision based median filter that optimizes the delay, using histogram-based technique to calculate median with only higher bit-planes of the image. To achieve higher operating frequencies, pipeline stages are made very shallow. We also examine the effects of the proposed technique on other performance parameters like area, power and output image quality.

The rest of this paper is organized as follows: Section II presents the details of the background techniques with examples of median calculation using only higher *B*-bits and calculation of median using histogram-based technique. Architecture of the proposed median filter is discussed in section III, along-with the architecture of median filter core. Section IV presents the results on filtered image quality and results of implementation of proposed median filter architecture on different generation FPGAs with comparative analysis. Overall conclusion of the work is presented in section V.

II. Background Techniques

A. Decision Based Median Filtering

Fig.1 shows a simple modification around CMF to convert it to decision based median filter where the output pixel is changed to the median of input window only if the *center pixel* is found to be corrupted by impulse noise (0 or 255). The decision based median filter provides better image quality compared to CMF for de-noising images affected by impulse noise.

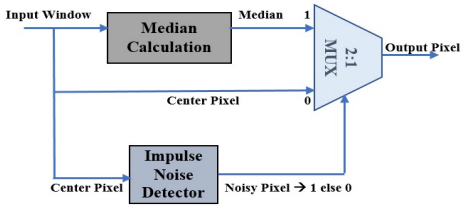

Fig. 1. Decision-based Median Filter using CMF

B. Processing Higher B-Bits for Calculating Approximate Median

The higher bit-planes of images consists of majority of visually significant data and due to the nature of images, most pixels in most windows have values in similar range [11]. Hence approximate median can be calculated using higher *B*-bits of pixel values. To calculate median using higher *B*-bits, consider a 3×3 window of an image which has 9 pixels represented in hexadecimal base in fig.2 (a). The center pixel in this window is to be filtered by processing the window with $B = 4$.

In the first step, the lower hex characters (lower 4-bits) of the input pixels are masked. The resulting 3×3 window is shown in fig.2 (b). The output of previous step is sorted considering only higher 4-bits and reading pixels in order, which results in the window in fig.2 (c). Although the lower nibbles of the pixels are not used for processing or calculation, they are attached to the higher nibble. Sorting using higher 4-bits results in a 4-bit median value of 0x**4.** However, an 8-bit output is required which means one of the three values: 0x**46**, 0x**41** and 0x**42**, is to be selected. Of these values, we select the first value (0x46) in the order of reading input. Although this example refers to *'Sorting'*, our method uses non-sorting based median calculation, discussed in the next section.

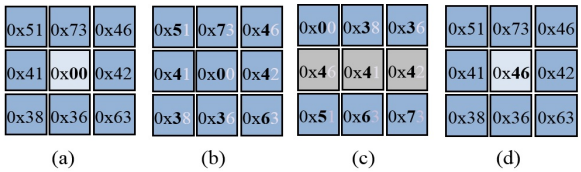

Fig. 2. (a) Input window (b) Masking lower nibble. (c) Sorted window (d) Processed window

It is experimentally observed that there is no significant difference in the performance (Image Quality), if any of the three matching values (0x46, 0x41, 0x42) in above example is chosen as the output. We select the first value in the order of reading input values, as it does not add any overhead to the hardware. In this example, the true/actual median value (0x42) is not chosen as the median and hence we say that approximate median is calculated using higher *B*-bits. However, for most windows in an image, the actual median gets calculated by processing only higher bits. To demonstrate this experimentally, we calculated the median with 5×5 window size and different values of *B* (4, 3 and 2) for several noisy images and video frames from databases [12] and [13]. Fig.3 shows the graph of window number versus actual and approximate median for one such sample image for different values of *B*. It is observed from fig.3 that the difference between the median obtained with different values of *B* and actual median is negligible. In some cases, the median values are exactly equal which shows in general that approximate median is calculated using only higher *B*-bits. Numerically, $B = 4$ provides better results as compared to $B = 2$ or $B = 3$.

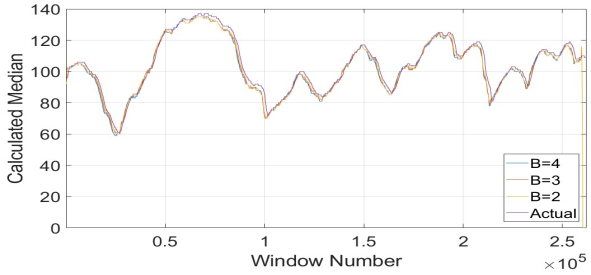

Fig. 3. Actual Median and Median using Higher *B*-bits for a sample image

C. Histogram-Based Median Calculation

The number of cascaded stages of sorting blocks in a classical sorting network for median calculation is equal to the number of input values [14], which results in a low frequency hardware implementation. On the other hand, number of stages in a histogram-based median calculation hardware is independent of the number input values resulting in comparatively higher operating frequencies. The histogram-

based median filter that reads and processes all input pixels of a window simultaneously, is feasible for hardware implementation only if limited bits of input pixel values are considered. For more bits, the hardware grows exponentially.

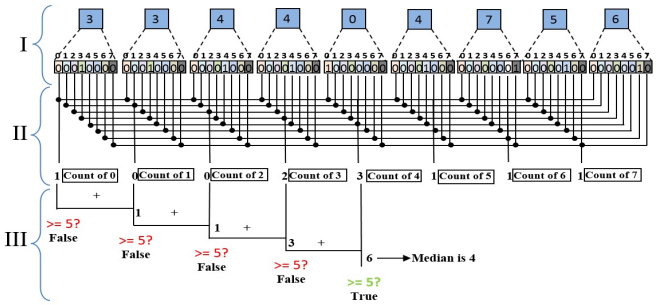

Fig. 4. Histogram-Based Median Calculation

Fig.4 shows median calculation using histogram technique with $B = 3$ as an example. In the first step, all input values are compared with all possible values (0-7) that an input may take; forming Stage-I of the median calculation block. In the next step, the comparison results of Stage-I are added together to calculate the histogram of the input data which forms Stage-II. The Stage-II outputs (count of each value) are added successively and after each addition, the result is compared with value '5' (position of median for 9 inputs) forming Stage-III of the architecture. The value in range 0-7 whose count was added to make the comparison true is the median.

III. Proposed Architecture

We implement the proposed hybrid median filter for 5×5 window and serially process each window for decision based median filtering using histogram calculation with $B = 4$. As the input pixels are available serially, a buffer stores four rows and five pixels of the next row of the image and then provides a 5×5 window to the median filter block in every clock cycle.

A. Proposed Hybrid Median Filter Architecture

Fig. 5 shows the block diagram of the proposed hybrid median filter that processes N W-bit integers. Out of the W-bits, B-bits are processed using histogram-based technique to calculate B-bit median value by the median filter core. Using B-bit median, W-bit median is determined from the input window. Based on the decision made by the impulse noise detector, W-bit median or the *center pixel* is selected as the output. The hardware implementation of the proposed hybrid median filter uses pipeline registers between all stages.

B. Median Filter Core

For simplified representation, block diagram of median filter core for $B = 2$ and $N = 5$ is shown in fig. 6. For $N = 5$, the value of last stage comparators is '3' (position of median for 5 inputs). The hardware design consists of 3 stages with pipeline registers between these stages. The structure of median filter core is same for higher values of B and N with more comparators and adders.

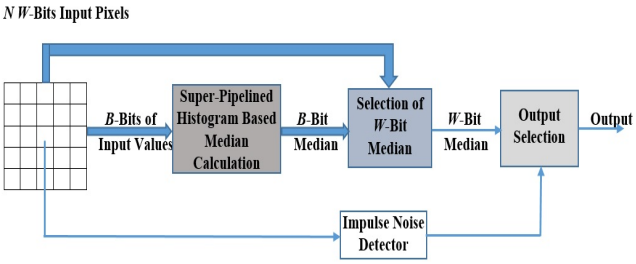

Fig. 5. Proposed Hybrid Median Filter Architecture

The operating frequency of the median filter core is dependent on the critical path delay which is the delay of Stage-II, as it accumulates all comparison bits (25 bits to be added for 5×5 window). To reduce this delay, we exploit the concept of super-pipelining by adding registers inside the adder stage (Stage-II).

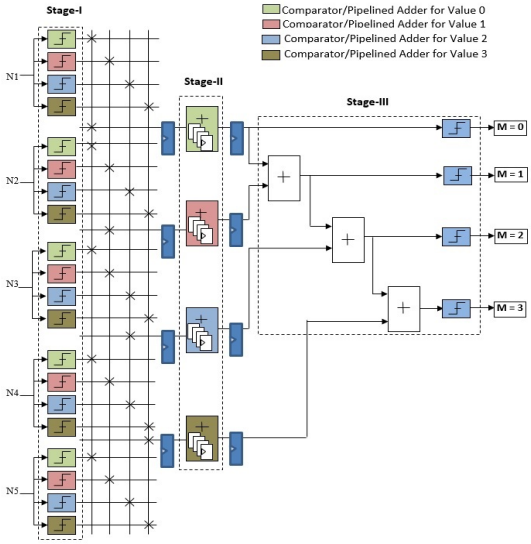

Fig. 6. Median Filter Core with $B = 2$ and $N = 5$

IV. Results

A. Image Quality

To analyze the quality of filtered images by the proposed hybrid median filter, images from database [12] and FHD video frames from database [13] are added with salt and pepper noise of varying noise densities followed by filtering using the proposed technique and calculating the PSNR. The average PSNR of the images filtered by the proposed median filter is at least 8 dB greater than average PSNR of images filtered by median filters [3-6] and is less than the average PSNR of images filtered by Low Complexity Median Filter (LCMF) [8] by approximately 0.7 dB.

B. Hardware Evaluation

To evaluate the proposed hybrid technique on hardware, we designed and implemented it using VHDL for $B = 4$ and $N = 25$ along-with supporting buffers, logic and control on Zybo Z7

evaluation board equipped with Xilinx XC7Z010 SoC. Real-time input to the complete system was provided using APEMAN 1080P FHD camera capturing video at 30 fps with a resolution of 1080×1920 and the frames were fed to the median filter block sequentially with the processed output visualized on Monitor via HDMI interface as seen in fig. 7.

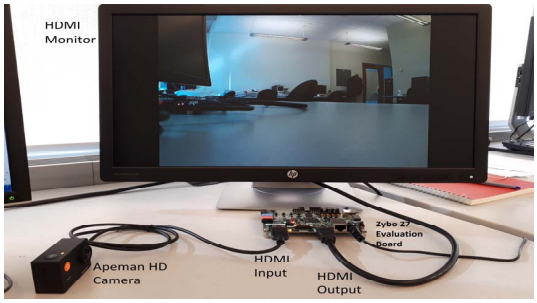

Fig.7. Evaluation of Proposed Hybrid Median Filter on Zybo Z7 Board

C. Experimental Results

The hardware implementations are verified using RTL simulations and the VHDL code for proposed hybrid median filter is mapped to Virtex-2, Virtex-6 and Zynq-7 FPGAs. All experiments were performed on Xilinx Vivado Design Suite. Implementation results of the proposed filter are compared with FPGA implementations of other relevant pipelined median filters for same window size (5×5). These results are presented in Table I. The throughput T_p in **fps** is calculated using (1) from the operating frequency f_{max} (MHz) obtained from timing summary report after inserting the timing constraints. Pipeline latency 'δ' is very small and may be ignored. F_s is 1080×1920; the frame size for FHD video frames. The throughput obtained from (1) is an estimate. The reference implementations also estimate the throughput of their architectures using similar calculations.

$$T_p = [(f_{max} \times 10E6) + \delta] / F_s \quad (1)$$

It is observed from Table I that the proposed hybrid median filter architecture provides the highest throughput compared to reference architectures. Although the proposed median filter hardware utilizes more area compared to some architectures, the throughput achieved in such cases is significantly higher. In comparison with LCMF [8], the proposed median filter requires substantially low area while providing significantly more operating frequency leading to a high throughput on all the three FPGAs. Thus, there is a trade-off between image quality and hardware performance when the proposed median filter is compared with LCMF.

For comparing performance with LEAMF [5], energy required by the proposed architecture is calculated for noisy FHD video frames by importing the simulation activity file in Xilinx XPower analyzer for Virtex-6 implementation. It is observed that the proposed median filter consumes only 30% extra energy compared to LEAMF while providing better image quality and a speed-up of 2.4 as compared to it.

TABLE I. IMPLEMENTATION RESULTS WITH COMPARISON

FPGA	Architecture	# Slices	Frequency (MHz)	Throughput (fps)
Virtex - 2	[4]	1506	305	140
	[5]	366	140	56
	[6]	2300	72	35
	[8]	2718	101	48
	Proposed	**1688**	**384**	**185**
Virtex - 6	[5]	136	263	105
	[8]	1713	169	91
	Proposed	**527**	**531**	**256**
Zynq - 7	[3]	1550	394	190
	[8]	1990	261	125
	Proposed	**900**	**586**	**282**

V. CONCLUSION

This paper presented a hybrid technique for high speed filtering of images affected by impulse noise. The proposed hybrid median filter performs significantly better than many relevant architectures in terms of image quality as well as hardware performance. We validated the proposed hybrid median filter by designing and testing the complete system on Xilinx Zynq-7 SoC. Experimental results show that the proposed median filter hardware can work at a frequency of 586 MHz when implemented on Zynq-7 FPGA with an estimated throughput of 282 FHD frames per second.

REFERENCES

[1] Tai, Y. M., Tu, W.C. and Chien, S. Y., "VLSI Architecture Design of Layer-Based Bilateral and Median Filtering for 4k2k Videos at 30fps", 2017 International Symposium on Circuits and Systems.

[2] Kowalczyk, A., Przewlocka, D. and Kryjak, T., "Real-time Implementation of Contextual Image Processing Operations for 4K Video Stream in Zynq UltraScale+ MPSoC", 2018 Conference on Design and Architectures for Signal and Image Processing (DASIP).

[3] Kumar, V., Asati, A. and Gupta, A., "Low-latency Median Filter Core for Hardware Implementation of 5 × 5 Median Filtering", IET Image Processing, 2017, Vol. 11 Issue 10, pp. 927-934.

[4] Vasicek, Z. and Sekanina, L., "Novel Hardware Implementation of Adaptive Median Filters", 11th IEEE Workshop on Design and Diagnostics of Electronics Circuits and Systems, Apr. 2008.

[5] Kalali, E. and Hamzaoglu, I., "A Low Energy 2D Adaptive Median Filter Hardware", Design, Automation & Test in Europe Conference & Exhibition (DATE) , Grenoble, France, 2015.

[6] Fahmy, S.A., Cheung, P.Y.K and Luk, W., "High-throughput One Dimensional Median and Weighted Median Filters on FPGA", IET Computers & Digital Techniques (Volume: 3, Issue: 4 , July 2009).

[7] T. Nodes, and N. Gallager, "Median filters: Some Modifications and their Properties," IEEE Trans. Acoustics, Speech and Signal Processing, vol.ASSP-30, no.5. (739-746), Oct.l982.

[8] T. Matsubara, V. Moshnyaga, and K. Hashimoto, "A FPGA Implementation of Low-Complexity Noise Removal," in Proc. of 17th International Conference on Electronics, Circuits, and Systems (ICECS), pp. 255 – 258, 2010.

[9] Cadenas, J.O., Megson, G.M. and Sherratt, R.S., "Median Filter Architecture by Accumulative Parallel Counters", IEEE Transactions on Circuits and Systems-II: Express Briefs, VOL. 62, NO. 7, (661-665) July 2015.

[10] Sekanina, L., Vasicek, Z. and Mrazek, V., "Approximate Circuits in Low-Power Image and Video Processing: The Approximate Median Filter", Radio Engineering, VOL. 26, NO. 3, September 2017.

[11] R. C. Gonzalez and R. E. Woods, Eds., Digital Image Processing, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2008.

[12] Image Database: http://www.eecs.qmul.ac.uk/~phao/IP/Images/

[13] Video Database: https://media.xiph.org/video/derf/

[14] Batcher, K.E., "Sorting Networks and their Applications", Spring Joint Computer Conference, 1968.