A Low Energy 2D Adaptive Median Filter Hardware

Ercan Kalali, Ilker Hamzaoglu
Faculty of Engineering and Natural Sciences, Sabanci University
34956 Tuzla, Istanbul, Turkey
{ercankalali, hamzaoglu}@sabanciuniv.edu

Abstract— The two-dimensional (2D) spatial median filter is the most commonly used filter for image denoising. Since it is a non-linear sorting based filter, it has high computational complexity. Therefore, in this paper, we propose a novel low complexity 2D adaptive median filter algorithm. The proposed algorithm reduces the computational complexity of 2D median filter by exploiting the pixel correlations in the input image, and it produces higher quality filtered images than 2D median filter. We also designed and implemented a low energy 2D adaptive median filter hardware implementing the proposed 2D adaptive median filter algorithm. The proposed hardware is verified to work correctly on a Xilinx Zynq 7000 FPGA board. It can process 105 full HD (1920x1080) images per second in the worst case on a Xilinx Virtex 6 FPGA, and it has more than 80% less energy consumption than original 2D median filter hardware on the same FPGA.

Keywords—Median filter, hardware implementation, FPGA, low energy.

I. INTRODUCTION

Digital images are affected by the noise resulting from image sensors or transmission of images. Image denoising is performed to remove the noise from the images. Several linear and non-linear filters are proposed for image denoising [1]. Although non-linear filters are more complex than linear filters, they are more commonly used for image denoising because they reduce the smoothing and preserve the image edges. 2D spatial median filter is the most commonly used non-linear filter for image denoising. It is a non-linear sorting based filter. It sorts the pixels in the given window, determines the median value, and replaces the pixel in the center of the given window with this median value. Therefore, it has high computational complexity.

In this paper, we propose a novel low complexity 2D adaptive median filter algorithm. The proposed algorithm exploits the pixel correlations in the input image to reduce the computational complexity of 2D median filter and to produce higher quality filtered images than 2D median filter. We also designed and implemented a low energy 2D adaptive median filter hardware implementing the proposed 2D adaptive median filter algorithm for 5x5 window size. The proposed hardware is implemented using Verilog HDL. It is verified to work correctly on a Xilinx Zynq 7000 FPGA board. It can work at 263 MHz, and it can process 105 full HD (1920x1080) images per second in the worst case on a Xilinx Virtex 6 FPGA. It has more than 80% less energy consumption than original 2D median filter hardware on the same FPGA.

Several median filter algorithms are proposed in the literature [2]-[5]. These algorithms can be classified into two groups. The median filter algorithms proposed in [2, 3] optimize the sorting process to reduce the computational complexity of median filter algorithm without reducing the quality of the filtered images. The median filter algorithms proposed in [4, 5] increase the quality of the filtered images without increasing the computational complexity of median filter algorithm. These algorithms try to detect the noisy pixels and adaptively filter only these noisy pixels. However, the adaptive median filter algorithm proposed in this paper both reduces the computational complexity of median filter algorithm and increases the quality of filtered images by exploiting the pixel correlations in the input image.

Several median filter hardware are proposed in literature [6]-[10]. In [6], an adaptive median filter hardware that detects the corrupted pixels with some iterations and filters only these pixels is proposed. The proposed median filter hardware uses different sorting algorithms like bitonic and odd-even merge sort. In [7], the sorting process of median filter algorithm is optimized. The proposed median filter hardware only finds the correct positions of input pixels in the sliding window instead of sorting all pixels in the window. In [8], a histogram based median filtering algorithm is proposed. But, it only performs well for large window sizes. Low complexity bit-pipeline (LCBP) algorithm is proposed in [9] in order to decrease hardware complexity and increase operating frequency. The proposed median filter hardware is implemented in different FPGAs for different window sizes. An energy efficient median filter hardware is proposed in [10] by optimizing memory read/write scheduling of median filter algorithm. However, the performance and hardware area of this hardware are not reported. The adaptive median filter hardware proposed in this paper is compared with these median filter hardware in Section III.

The rest of the paper is organized as follows. In Section II, the proposed adaptive median filter algorithm is explained. In Section III, the proposed adaptive median filter hardware is presented, and its implementation results are given. Section IV presents the conclusions.

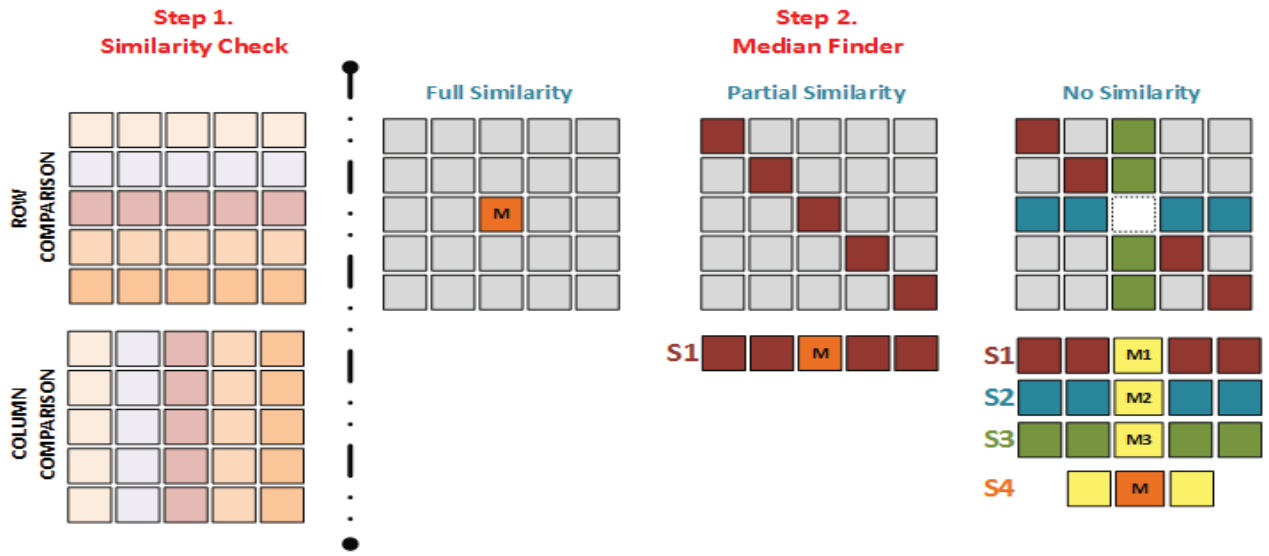

Fig. 1. Proposed 2D adaptive median filter algorithm

II. Proposed Adaptive Median Filter Algorithm

2D spatial median filter algorithm sorts the pixels in the given window, determines the median value, and replaces the pixel in the center of the given window with this median value. The proposed 2D adaptive median filter algorithm consists of two steps as shown in Fig. 1. The pseudo code of the proposed adaptive median filter algorithm for 5x5 window is given in Fig. 2. The proposed adaptive median filter algorithm does not perform any sort in the best case, and it sorts 15 pixels instead of 25 pixels in the worst case for 5x5 window.

```
Adaptive_Median_Filter(Window) {
  RC = compare(MSB 4 bits of pixels in each row)
  CC = compare(MSB 4 bits of pixels in each column)
  PS_R = (RC[0] & RC[1] & RC[2] & RC[3] & RC[4])
  PS_C = (CC[0] & CC[1] & CC[2] & CC[3] & CC[4])
  if (PS_R is 1 and PS_C is 1)
    Median = Window(2, 2)
  else if (PS_R is 1 or PS_C is 1)
    Median = median(Diagonal Pixels (S1))
  else {
    M1 = median(Diagonal Pixels (S1))
    M2 = median(Horizontal Pixels (S2))
    M3 = median(Vertical Pixels (S3))
    Median = median(M1, M2, M3 (S4))
  }
  Window(2, 2) = Median
}
```

Fig. 2. Pseudo code of proposed 2D adaptive median filter algorithm

In the first step, the proposed adaptive median filter algorithm compares the pixels in each row and column of the given window separately. If the pixels in a row are similar, the row comparison signal for that row is set to 1. Similarly, if the pixels in a column are similar, the column comparison signal for that column is set to 1. Then, if pixels in all rows are similar, PS_R signal is set to 1. Similarly, if pixels in all columns are similar, PS_C signal is set to 1. The proposed adaptive median filter algorithm determines the similarity of the pixels in a row or column by comparing only their 4 most significant bits.

In the second step, the median value is determined. If there is full similarity (both PS_R and PS_C are 1), the pixel in the center of the window is determined as the median value of the window. If there is partial similarity (only PS_R or PS_C is 1), the diagonal pixels in the window are sorted, and the median value of the diagonal pixels is determined as the median value of the window. If there is no similarity (neither PS_R nor PS_C is 1), the diagonal (S1), horizontal (S2) and vertical (S3) pixels are sorted, and their median values (M1, M2, M3) are determined separately. Then, the median of M1, M2, and M3 is determined as the median value of the window. Finally, the pixel in the center of the given window is replaced with the median value.

The number of windows with similar pixels in an image varies from image to image. We used the HEVC video compression standard test videos [11] and commonly used image processing benchmark images [12] to determine percentage of similarities for different window sizes. The simulation results for 5x5 and 7x7 window sizes for one image from Traffic (2560x1600), People on Street (2560x1600), Basketball Drive (1920x1080), Tennis (1920x1080), Kimono (1920x1080), Park Scene (1920x1080), Vidyo1 (1280x720), Vidyo4 (1280x720), Kristen and Sara (1280x720), Four People (1280x720) videos [11], and Baboon (512x512), Barbara (512x512), Goldhill (512x512), Lena (512x512), Peppers (512x512) images [12] are shown in Table I.

TABLE I. SIMILARITY PERCENTAGES (%) FOR 5X5 AND 7X7 WINDOW SIZES

		Traffic	People on Street	Basket	Tennis	Kimono	Park Scene	Vidyo 1	Vidyo 4	Kristen and Sara	Four People	Baboon	Barbara	Goldhill	Lena	Peppers
5x5	**F. S.**	13.32	13.30	18.29	25.39	20.23	14.64	19.16	22.16	21.06	20.17	2.21	8.13	7.51	10.31	11.63
	P. S.	2.34	1.68	4.22	4.25	3.67	3.90	4.27	3.71	2.01	4.66	1.00	2.44	2.56	2.46	3.20
	N. S.	84.54	85.02	77.49	70.36	76.10	81.46	76.57	74.13	76.94	75.17	96.79	89.42	89.92	87.23	85.17
7x7	**F. S.**	4.44	4.41	4.78	9.86	6.01	3.31	5.09	6.82	8.32	7.79	2.47	3.39	3.45	3.23	3.77
	P. S.	3.24	1.11	1.54	2.75	1.11	2.15	3.33	2.37	2.26	2.39	2.04	2.10	2.07	2.06	2.04
	N. S.	92.32	94.48	93.68	87.39	92.88	94.55	91.59	90.81	89.42	89.82	95.48	94.51	95.48	94.71	94.19

TABLE II. PSNR VALUES FOR HEVC TEST IMAGES

Image	Window Size	S & P Noise	Matlab Median Filter	Prop. Median Filter	ΔPSNR (dB)
Traffic	5x5	18.189	32.515	34.582	**2.067**
	7x7		29.345	32.864	**3.519**
People on Street	5x5	18.156	32.371	34.947	**2.576**
	7x7		29.157	33.334	**4.177**
Basket	5x5	18.713	31.291	32.054	**0.763**
	7x7		30.046	31.191	**1.145**
Tennis	5x5	17.699	38.145	39.007	**0.862**
	7x7		35.149	37.729	**2.580**
Kimono	5x5	17.929	43.436	45.418	**1.982**
	7x7		39.796	43.904	**4.108**
Park Scene	5x5	18.077	31.648	34.125	**2.477**
	7x7		29.574	32.829	**3.255**
Vidyo1	5x5	18.211	35.080	36.812	**1.732**
	7x7		32.528	35.356	**2.828**
Vidyo4	5x5	18.215	35.200	36.383	**1.183**
	7x7		32.885	35.517	**2.632**
Kristen and Sara	5x5	17.977	31.316	32.677	**1.361**
	7x7		28.457	30.794	**2.337**
Four People	5x5	18.154	30.728	32.265	**1.537**
	7x7		28.601	31.287	**2.686**

TABLE III. PSNR VALUES FOR BENCHMARK IMAGES

Image	Window Size	S & P Noise	Matlab Median Filter	Prop. Median Filter	ΔPSNR (dB)
Baboon	5x5	18.526	21.249	23.098	**1.849**
	7x7		20.563	23.305	**2.742**
Barbara	5x5	18.461	23.142	24.923	**1.781**
	7x7		23.546	25.115	**1.569**
Goldhill	5x5	18.348	28.717	30.701	**1.984**
	7x7		27.226	30.239	**3.013**
Lena	5x5	18.459	30.971	32.927	**1.956**
	7x7		28.894	32.144	**3.250**
Peppers	5x5	18.100	31.801	33.865	**2.064**
	7x7		29.991	33.072	**3.081**

We also quantified the impact of the proposed adaptive median filter algorithm on the PSNR performance for 5x5 and 7x7 window sizes. Salt & pepper noise is added to original images. Then, we filtered the images with MATLAB standard median filter function (medfilt2), and with the proposed adaptive median filter algorithm. The PSNR values are given in Table II and Table III for different images. The results for Basketball Drive image are shown in Fig. 3. These results show that the proposed 2D adaptive median filter algorithm produces higher quality filtered images than standard 2D median filter algorithm. This is because the proposed median filter algorithm tries to avoid replacing the pixel in the center of the given window with the median value if it is not a noisy pixel by exploiting the similarity of the pixels in the window.

III. PROPOSED ADAPTIVE MEDIAN FILTER HARDWARE

The proposed 2D adaptive median filter hardware implementing the proposed 2D adaptive median filter algorithm is shown in Fig. 4. An input pixels buffer is used to store the pixels in a 5x5 window. This on-chip buffer reduces the required off-chip memory bandwidth. After the pixels are loaded into this buffer, 40x4 bit comparators in the comparison unit compare the pixels in each row and column. Based on the comparison results, the similarity control signals PS_R and PS_C shown in Fig. 2 are generated.

If there is full similarity, the pixel in the center of the window is selected in output multiplexer as the median value. If there is partial similarity, only diagonal sort (S1) datapath is enabled, and the other datapaths are disabled to reduce power consumption. If there is no similarity, all datapaths are enabled, and the output of sort (S4) datapath is selected in output multiplexer as the median value.

The proposed median filter hardware sorts 15 pixels in the worst case instead of 25 pixels for 5x5 window. In the proposed hardware, these 15 pixels are sorted in 3 parallel datapaths, and there are 4 pipeline stages to increase the throughput. The proposed median filter hardware has a throughput of 1 median filtering per clock cycle.

The proposed 2D adaptive median filter hardware and the original 2D median filter hardware are designed and implemented using Verilog HDL. The hardware implementations are verified with RTL simulations using Mentor Graphics Questa. The RTL simulations results matched the results of software implementations of median filter algorithms. The Verilog RTL codes are mapped to a Xilinx XC6VLX75T FF784 FPGA with speed grade 3 using Xilinx ISE 13.4. The FPGA implementations are verified with post place and route simulations. The FPGA implementation of the proposed median filter hardware uses 136 slices, 327 LUTs, 150 DFFs, and it can work at 263 MHz. The FPGA implementation of the original median filter hardware uses 208 slices, 634 LUTs, 226 DFFs, and it can work at 250 MHz.

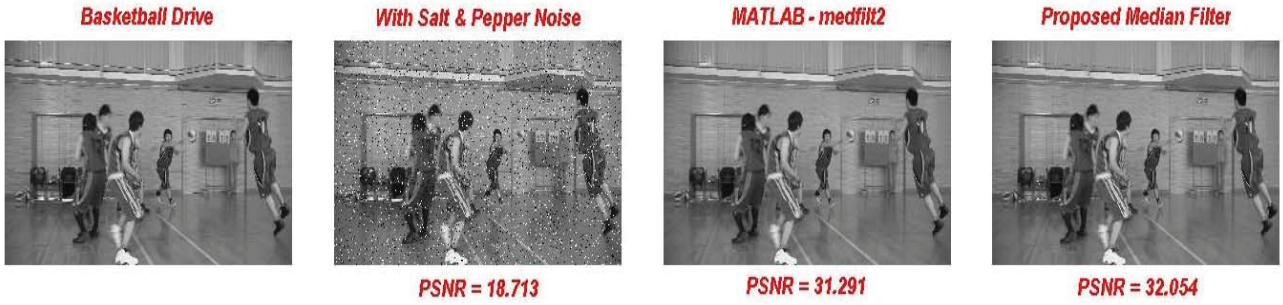

Fig. 3. Example image and its PSNR values

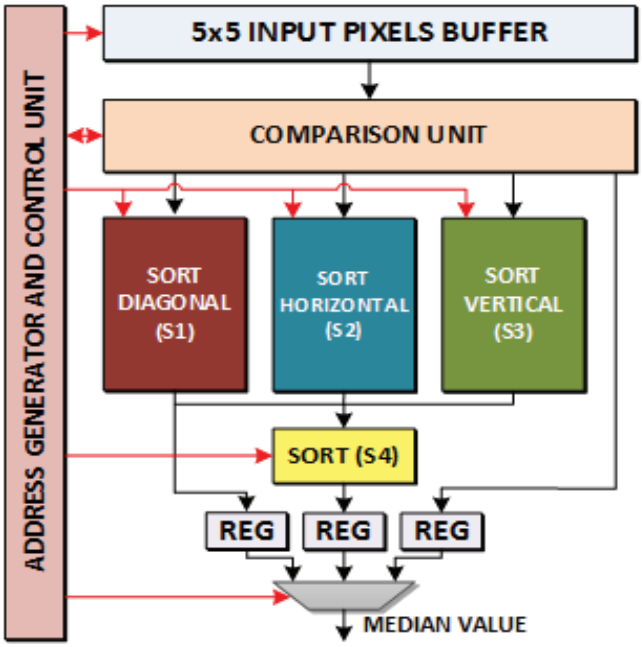

Fig. 4. Proposed 2D adaptive median filter hardware

The proposed adaptive median filter hardware is also implemented on the Xilinx Zynq 7000 FPGA board as shown in Fig. 5. The Zynq board includes 7 series FPGA and ARM Cortex A9 microprocessor on a single chip with high speed AXI bus, 128 MB DDR3, 16 MB Quad Flash Memory, HDMI and Ethernet interface. The VITA 2000 camera captures 60 fps full HD (1920x1080) images. The proposed adaptive median filter hardware filters these images. The filtered images are displayed on HDMI monitor and sent to the computer using Ethernet.

We estimated the power consumptions of both FPGA implementations using Xilinx XPower tool for one image from the Tennis (1920x1080), Kimono (1920x1080), Park Scene (1920x1080) and Basketball Drive (1920x1080) videos [12]. In order to estimate the power consumption of a median filter hardware, its placed and routed netlist is simulated using Mentor Graphics Questa for one image from each video. The signal activities of these simulations are stored in VCD files, and these VCD files are used for estimating the power consumption of that median filter hardware using Xilinx XPower tool. For both FPGA implementations, only internal power consumption is considered, and input and output power consumptions are ignored.

The power and energy consumptions of the proposed 2D adaptive median filter hardware and the original 2D median filter hardware are shown in Fig. 6 and Fig. 7. As shown in these figures, the proposed 2D adaptive median filter hardware has 42% and 85% less power and energy consumption than the original 2D median filter hardware.

The comparison of the proposed adaptive median filter hardware with the median filter hardware proposed in the literature is shown in Table IV. The 2D median filter hardware shown in this table process 5x5 pixel 2D windows whereas 1D median filter hardware shown in this table process 25 pixel 1D windows. Although the adaptive median filter algorithm proposed in [6] increases the PSNR of filtered image, this hardware has a large area. Sorting process is optimized in [7] without reducing image quality. But, its hardware area is 10 times larger than our median filter hardware. Histogram based median filtering proposed in [8] gives better results for large window sizes, but it is very costly for small window sizes. Low complexity bit-pipeline (LCBP) algorithm proposed in [9] has smaller hardware area than the other hardware in the literature. But, our median filter hardware has much smaller area than this hardware. In addition, the median filter hardware proposed in [9] does not increase the quality of filtered images.

TABLE IV. MEDIAN FILTER HARDWARE COMPARISON FOR 5X5 WINDOW

	FPGA	# of Slices	Max. Speed (MHz)	Performance (fps)
[6]	Xilinx VirtexII	1506	305	140 Full HD
[7]	Altera Cyclone II	1309	94	23 Full HD
[8]	Xilinx Virtex II	2300	333	35 Full HD
[9]	Xilinx Virtex II	660	318	Not Given
Proposed	Xilinx Virtex II (Scaled)	366	140	56 Full HD
	Xilinx Virtex 6	136	263	105 Full HD

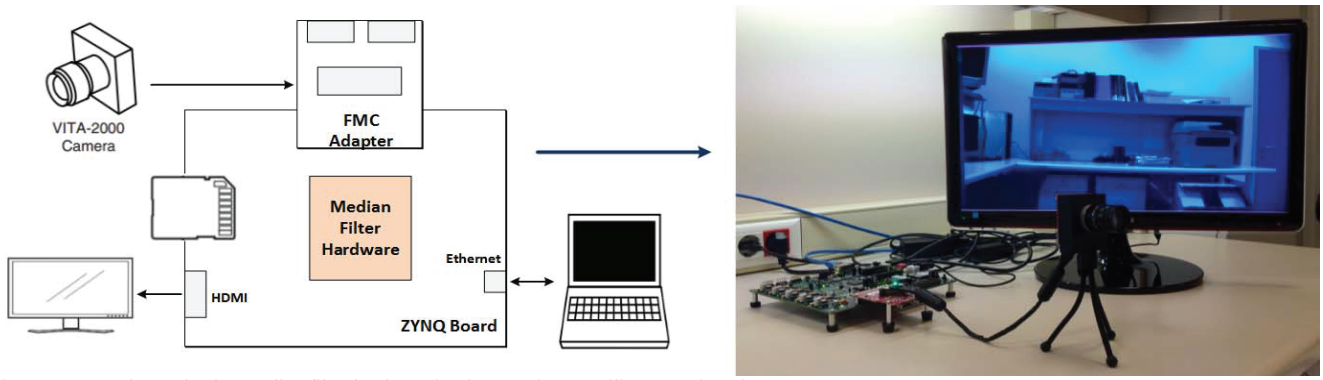

Fig. 5. Proposed 2D adaptive median filter hardware implementation on Xilinx Zynq board

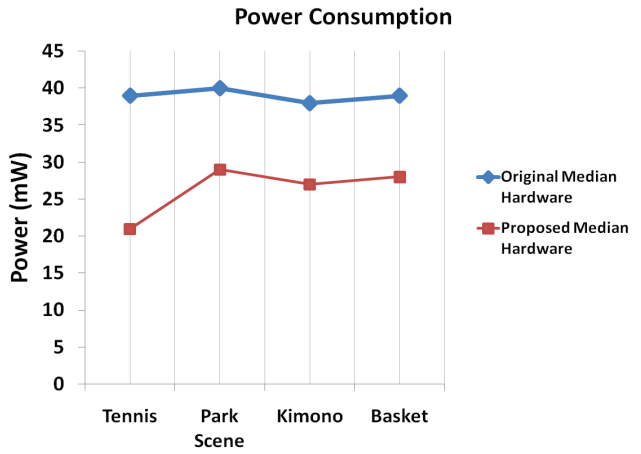

Fig. 6. Power consumptions of median filter hardware for full HD (1920x1080) images

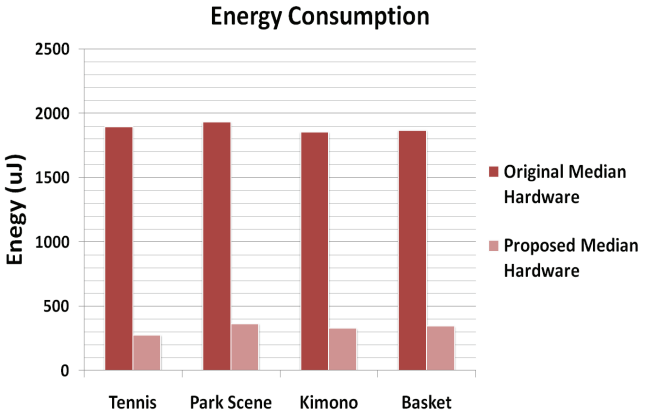

Fig. 7. Energy consumptions of median filter hardware for full HD (1920x1080) images

Optimized memory scheduling based median filter hardware proposed in [10] reduces the energy consumption of median filter hardware up to 53% on Xilinx Virtex 7 FPGA. However, our median filter hardware reduces the energy consumption of median filter hardware more than 80%. In addition, the performance and hardware area of this hardware are not reported.

IV. Conclusions

In this paper, we proposed a novel low complexity 2D adaptive median filter algorithm. The proposed algorithm reduces the computational complexity of 2D median filter, and it produces higher quality filtered images than 2D median filter. We also designed and implemented a low energy 2D adaptive median filter hardware implementing the proposed 2D adaptive median filter algorithm for 5x5 window size. The proposed hardware can process 105 full HD (1920x1080) images per second in the worst case on a Xilinx Virtex 6 FPGA, and it has more than 80% less energy consumption than original 2D median filter hardware on the same FPGA.

References

[1] R. C. Gonzalez, R. E. Woods, Digital Image Processing, Prentice Hall, 2002.

[2] C. Chakrabarti, "Sorting network based architecture for median filters," IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol.40, no. 11, pp. 723-727, Nov. 1993.

[3] J. Scott, M. Pusateri, M. U. Mushtaq, "Comparison of 2D median filter hardware implementations for real time stereo video," 37th IEEE Applied Imagery Pattern Recognition Workshop, Oct. 2008.

[4] S. Esakkirajan, T. Veerakumar, A. N. Subramanyan, C. H. PremChand, "Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter," IEEE Signal Processing Letters, vol. 18, no. 5, pp. 287-290, March 2011.

[5] S. Akkoul, L. Roger, R. Leconge, R. Harba, "A new adaptive switching median filter," IEEE Signal Processing Letters, vol. 17, no. 6, pp. 587-590, June 2010.

[6] Z. Vasicek, L. Sekanina, "Novel hardware implementation of adaptive median filters," 11th IEEE Workshop on Design and Diagnostics of Electronics Circuits and Systems, Apr. 2008.

[7] V. G. Moshnyaga, K. Hashimoto, "An efficient implementation of 1-D median filter," 52nd IEEE Int. Midwest Symp. on Circuits and Systems, Aug. 2009.

[8] S. A. Fahmy, P. Y. K. Cheung, W. Luk, "High-throughput one-dimensional median and weighted median filters on FPGA," IET Computers & Digital Techniques, vol. 3, no. 4, pp. 384-394, June 2009.

[9] D. Prokin, M. Prokin, "Low hardware complexity pipelined rank filter," IEEE Trans. on Circuits and Systems II: Express Brief, vol. 57, no. 6, pp. 446-450, May 2010.

[10] A. Sanny, V. K. Prasanna, "Energy-efficient median filter on FPGA," Int. Conf. on Reconfigurable Computing and FPGAs, Dec. 2013.

[11] F. Bossen, "Common HM test conditions and software reference configurations," JCTVC-L1100, March 2013.

[12] Benchmark Images, http://www.dcs.qmul.ac.uk/~phao/CIP/Images