RESEARCHERID
THOMSON REUTERS

[Bist* *et al.*, 6(7): July, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

HARDWARE IMPLEMENTATION OF REAL TIME WINDOW BASED SWITCHING MEDIAN FILTER

Rashmi Bisht *, Ritu Vijay
Department of Electronics, Banasthali University, India

DOI: 10.5281/zenodo.834473

ABSTRACT

Processing of a digital image by the use of computer algorithm is usually referred as digital image processing. However, some unwanted signals capture the image which is termed as noise. Thus the digital image filtering technique is frequently used to eliminate noise. Generally image filters are following up software approach in systems. But hardware implementation prefers in comparison with software implementation for better processing speed. With a boost in the VLSI technology hardware implementation has given a better choice. In the present paper, a hardware implementation method of a Switching non linear Image Filter is proposed with 3X3 window size. Virtex-7 VC707 was used for design, simulation and synthesis process. The performance analysis of the switching median filter was evaluated for different image set (Lena, Cameraman, and Barbara). The max operating frequency attained was 397 MHz. The minimum delay achieved was approximately 2.41ns. The Verilog language was utilized to aim two-dimension switching image filter using ISE (Xilinx) tool. The proposed algorithm for the Switching Image Filter is working on sorting and finding the median values. The comparison has been done with peak signal to noise ration, processing time and the visual inspection observed on each filter.

KEYWORDS: Image filter, impulse noise, FPGA, Virtex-7.

I. INTRODUCTION

A digital image can specify as a numerical representation of 2-D picture, I (x, y). The magnitude of the function I (x, y) has defined the intensiveness of the image at that particular point. Therefore, digital image has finite values for all x, y and amplitude of function f. A digital image is considered as a collection of elements, called pixels. The pixel is a very common term used to refer the smallest addressable elements of a digital image [1, 2]. Digital images are usually corrupted by a separate number of noises, including fixed valued impulse noise. Image Processing is the field, where digital data is modify with the help of computer for improvement in image qualities. The aim of image processing is maximize clarity, sharpness and points of features of concern towards information extraction and advance analysis. Impulse or Salt & pepper noise is collection of random pixels which has very high contrast compared to the surroundings. So, it has high impact on images even if present in small percentage. Noise can capture image by a camera, scanner, and when the image is transmitted over a noisy channel. The need to remove impulse noise is most important so that it cannot effect adversely on the information or data contained by the original image [3].

Non Linear digital filter used in digital images to remove noise and preserve its quality. Median filter comes in the category of non linear digital filter. It removes impulse noise by altering the detected noisy pixel with the median value of neighborhood pixels or window. All median filters are computationally expensive because of sorting procedures.

High speed image processing task is required in different field like computer vision, medical (MRI, radiography, ultrasound etc.), traffic management industry, space exploration, military, automated industry and more domains. These applications demand different action like image enhancement and object detection etc [4].

Such applications can implement on software easily. But this is not time efficient. As it have extra constraints on memory and other peripheral devices. Concurrency offered by hardware implementation result, much faster

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

compare to software implementation, which is dependent on operating systems and micro controller characteristics [4].

Full custom and semi custom design are two technologies used for hardware design. ASIC (Application specific hardware design) is known as full custom hardware design. Programmable device like DSPs and FPGA are known as semi custom hardware device [4].
A FPGA board, working as a coprocessor in company with the host, can result faster computational intensive and data intensive algorithms. The design procedure demonstrated in this paper provides the design cycle to be reduced and an idea of the speedup to be obtained before circuit is actually implemented.

II. RELATED WORKS

Generally, Non linear median filter used to remove the impulse noise. Median filter is the popular nonlinear filters, applied to reject impulse noise due to its better computational efficiency. Median filtering is widely used in image processing applications due to its edge preservation quality while removing noise. Median filter substitute corrupted pixel by the median value of pixels present in a window. If the window has odd numbers of data, then the median is the center value after all the data in the window are sorted. For an even number, there are two possible median values. Median filters are widely used for smoothing the image and to preserve its quality [5].Standard Median Image filter [6] gives a blurring and inadequate performance for images containing the high intensity impulse (salt & pepper) noise. A median filter, which utilize 3×3 pixel window is performed well only with approx 10-40% noise intensity. It result edge loss, blur and many unfiltered shots when noise intensity increased. Adaptive Median Filter [7] result good at low noise presence. But the increment in window size with high noise densities leads to smudging the image again. In Paper [6], standard median filter is implemented and improvement in the computational speed of different image Enhancement Techniques on FPGA was reported. The hardware circuits of median filter were realized and modeled using Altera System. The synthesis and simulation results were obtained and designs are successfully validated using hardware simulation feature of using FPGA Virtex-7 Evaluation platform (VC707). In switching Image filter [3, 4] the noise detection is based on a pre-defined value. In paper [8], a content based median filter with its hardware implementation is presented. It is suitable for real time impulse noise suppression. The adaptive filter was designed and implemented in FPGA.The median filter operation is applies to only detected noisy pixels. The clock frequency is 50 MHz.

This paper presents less complex hardware implementation of universal 3 × 3 size spatial filter. This implementation contains of primary components: hardware memory management unit and arithmetic block.

III. ALGORITHM

Principal Idea

The suggested algorithm aims at taking out fixed-valued impulse noise. Likewise to the other filtering methods, it contains noise detector and filtering component. The noise sensor component finds whether the pixels are spoilt by the fixed-valued impulse noise or not. Only the detected spoilt pixel is reconstructed by non linear median filter. The non-noisy pixel left unchanged and filtering operation is skipped for them. The algorithm discuss in briefly below.

Noise Sensor

If the sensor finds that the processing pixel (p) value is 255 or 0, it fixes a binary flag S to 1. Other than that, the S is 0. i.e.

$$S=\begin{cases}1, & if\ p=255\ or\ p=0\\ 0, & otherwise\end{cases} \qquad (1)$$

Filtering action is activating whenever binary flag S set to 1.

Switching Median Filter

Median filters are the popular nonlinear filters, are applied to eliminate impulse noise due to its excellent computational efficiency. Median filter replaces noisy pixel by the median value of pixels present in a window. If the window has not even number of entries, then the median is the middle after all the entries in the window

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

are sorted. For an even count of window, there is more than one possible median value. Median filters are widely used for smoothing the image and to preserve its quality [3, 7, and 9].

Switching Median filter is a type of nonlinear image filter. It is used to remove salt and pepper noise. It replaces the corrupted pixel in an image by its median or central value. Salt and pepper noise is an impulse noise having random scattered black and white dots called pixels. This noise is produced because of sudden disturbance in image signal [1, 3]. The steps of the SMF algorithms are:

- Choose a two-dimensional window S_{xy} of size 3X3, with processing pixel P (x, y).
- Calculate the median value P_{med} of window $S_{xy.}$
- The value of processing pixel is substituted by P_{med}.
- Repeat the above step till the above steps is completed for the complete image. The flow chart is shown in Figure 1.

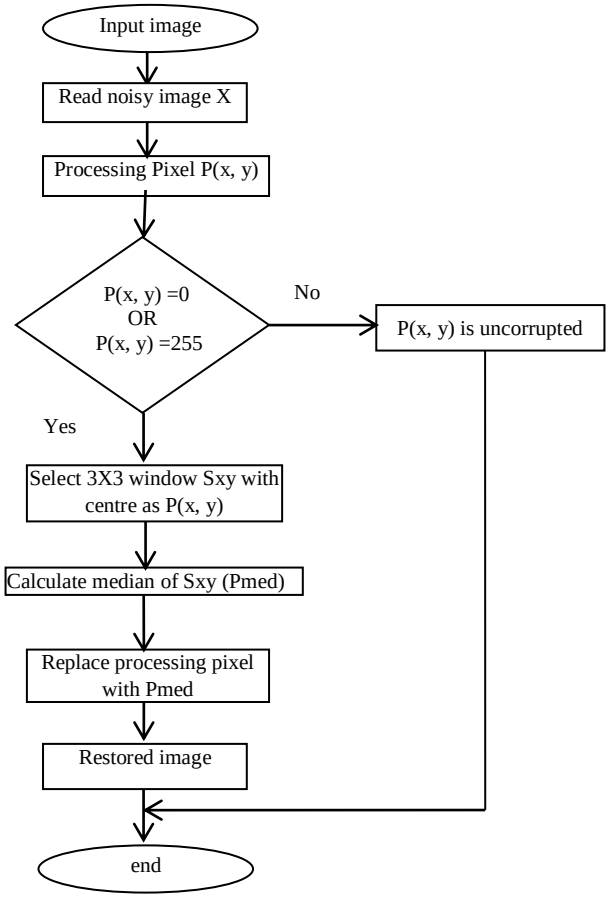


Figure 1 Switching Median Filter

IV. FPGA BASED IMAGE FILTER

FPGAs are integrated circuits, belongs to reconfigurable devices that contain LUT (Look-Up Tables), Registers, BRAM (block RAM) and more complex embedded functions like multipliers etc. The programmer fundamentally plan's the connections and parameters for these blocks. FPGAs are good at high parallel processing tasks. They typically work at low frequencies compare to CPUs. For any project they can easily

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

exceed the performance of a CPU by orders of magnitude. The proposed filter does not consider frequency domain approach. It adopts a spatial domain approach due to less complexity. It utilizes the overlapped window to take out the spoilt pixel in the image.

The construction of the switching median filter contain four basic working blocks

- The forward moving window block
- The median calculation block
- The arithmetic operations block
- The output selection block

The input value of the filter system is the grayscale values of the pixels of the noisy image. For the calculation of the filter output (median) a 3×3 pixel window neighborhood of a noisy pixel can be selected. Image input data are taken in serial manner into the first stage. The moving window block is the internal memory of the system. It used for putting the input values of the pixels and for realizing the moving window process. The 3X3 moving window consists nine registers. When the window is moved forward into the adjacent image pixel just three values stored in the memory are altered.
Comparison has been done between filtered and the original image and the filtering fitness are evaluated. The correct value of noisy pixels of the filtered image will be computed using that pixel and its surrounding neighbors.

Sliding Window Architecture

The input image pixel values of 8-bits wide are carried out in serial manner into the sliding window unit and stored in register. The 3×3 sliding window demands nine registers to keep pixels before it is located in median calculation unit as shown in Figure 2. One clock cycle is needed to read one image pixel from memory. The pixels are record row by row in a raster scan order.
FPGA certain features are utilized for the hardware design of the sliding window. The BRAM feature of the Xilinx Virtex FPGA has been used to perform read, write operation of RAM in Single clock cycle. Even so the similar consequence can also accomplish using LUTs based RAM implementation on FPGA. But the role of BRAM is more effective compared to possible benefits of look up table-FPGA implementation.

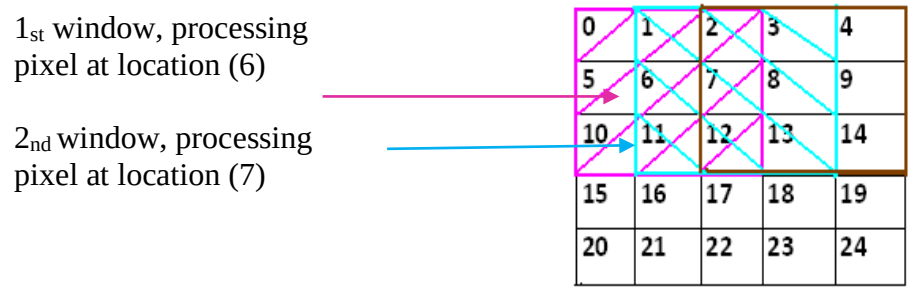


Figure 2 Example of sliding Window operation

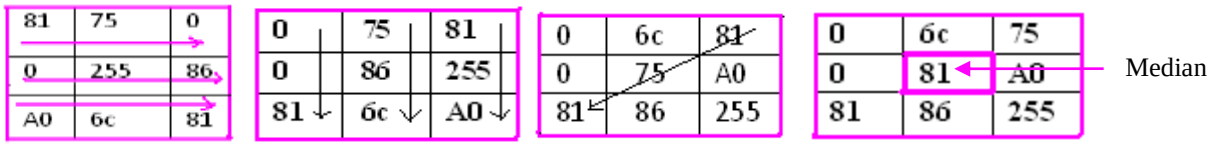
Sorter architecture

The sorting network unit is a 9-inputs/1-output combinational module with a data word length of 8 bits. This process can divide into three stages:

- Firstly, each row data are sorted horizontally, *i.e.* in the ascending order (Figure 3 (a)).
- After completion of vertical sorting, each column is sorted vertically, *i.e.*in the ascending order, Figure 3(b).
- Now diagonally sort elements and collect the second element as the median element of the window Figure3(c, d).

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

The first and last element in the window is minimum and maximum respectively of the nine elements. The working of the sorter is shown in Figure 3.



(a) (b) (c) (d)

Figure 3 Example of Sorter to find out median

V. IMPLEMENTATION

The entire implementation of image processing with FPGA is shown in diagram of Figure 4. In order to simplify the data transactions, RAM is implemented on FPGA using IP core.

The median filter is implemented using 3X3 window. The proposed architecture was examined on the Lena, Barbara and the cameraman image of 512 X 512 pixels for median filter. The image was transmitted to the target Xilinx Virtex-7 FPGA VC707 Evaluation Kit. After filtering operation, the filtered image was taken back to the PC for comparison purposes. MATLAB 7.9 is used to convert images to COE file. That COE file is used with Verilog code for filtering operation. The filtered image pixels are collected in MATLAB for display.

The process of building an image processing system that performs median filtering operations (is divided into two phases:

1. FPGA implementation of BRAM as image memory
2. FPGA implementation of image processing.

In the first phase the original image of size 512X512 is stored in BRAM which involves creating a .coe file in MATLAB, using Xilinx core generator. [6]

The second phase involves taking an image and performing median filtering and image processing on that image using 3x3 window, and displaying input and processed image through MatLab[4].

The median filter can be implemented using different windows size 3x3, 5x5 and 7x7. Larger size masks have a characteristic to the destruction of small edges [1]. The proposed architecture for median filter was examined on the Lena, cameraman, and Barbara images of 512x512 pixels. The output filtered images were sent back to the PC for analysis intention. The clock frequency was 50 MHz. The processing time of filtering with the proposed architecture was recorded for each of the image. Also the percent utilization of the target device was approximated.

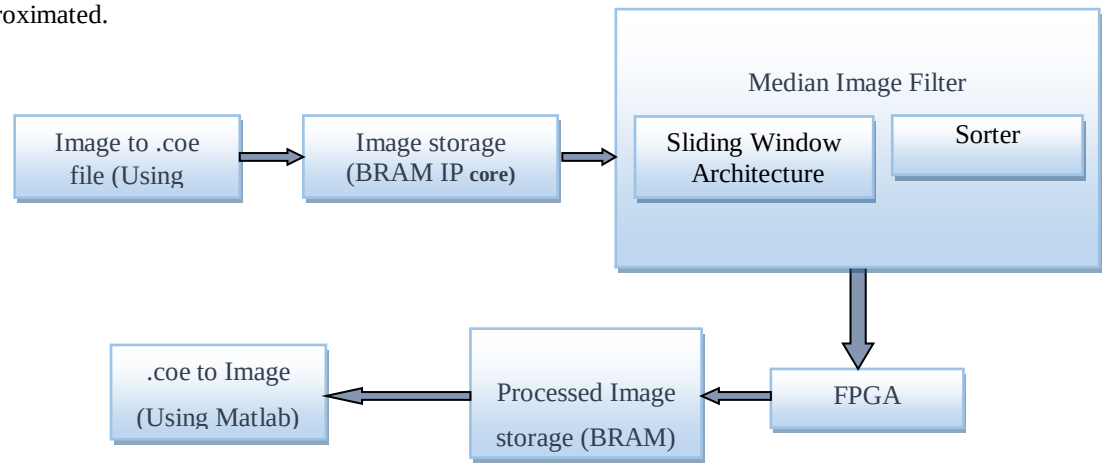
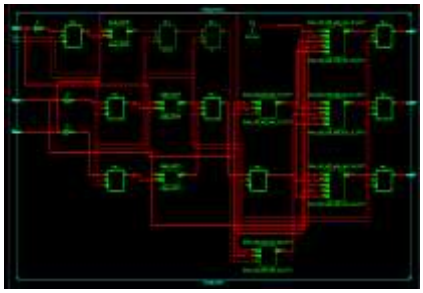


Figure 4 Architecture of the System

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

VI. RESULT AND DISCUSSION

The module of the architecture is described with Verilog and synthesized on virtex-7 series of FPGA device. All the synthesis, simulation, placement and routing are done with an ISE foundation from Xilinx. Figure 5 and 6 shows the RTL of the sorter and median calculation Unit. Figure 7 represents the resource utilization of proposed architecture that indicates the number of LUTs, flip flop and so on.



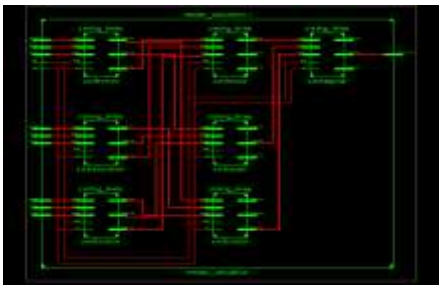


Figure 5 RTL of Sorter Unit ***Figure 6 RTL of Median Calculation Unit***

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	589	607200	0%
Number of Slice LUTs	1025	303600	0%
Number of fully used LUT-FF pairs	501	1113	45%
Number of bonded IOBs	36	700	5%
Number of Block RAM/FIFO	64	1030	6%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 7 Resource Utilization for Image of size 512X512

The maximum frequency obtained after the synthesis is 397.1 MHz, with minimum delay of 2.41 ns. Figure 8 shows the result when the image filter applied to 10-50 % corrupted Lena image by impulse noise, which is easily a high level of noise. The results demonstrate that visually switching median filter produced the accurate output.

Noise Density (%)	**Noisy Image**	**Filtered Image**
10		
20		

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

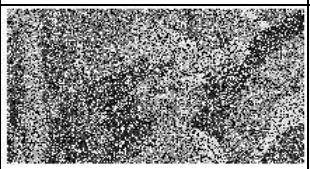
30		
40		
50		

Figure 8 shows visual results taken from FPGA

The pixels Q' (i, j) for 1 ≤i ≤M and 1 ≤j ≤N, of the original image is corrupted by adding impulse noise, with noise density starting 10 to 90%. The Peak signal to noise ratio value (PSNR) is used to equate the relative filtering operation of various filters. The PSNR of the filtered output image Q (i, j) and the original image Q' (i, j) of dimensions M X N pixels is defined as

$$PSNR = 10log10(255 * 2/\sqrt{mse}) \text{ ..... (1)}$$

Where MSE stands for mean square error and given as

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[Q'(i,j)-Q(i,j)]^2}{M\times N} \text{ ..... (2)}$$

It can be seen that Peak signal to noise ratio (PSNR) is closely related to mean square error (MSE).

TABLE II: PSNR for different images

Image	Noise density								
	10%	20%	30%	40%	50%	60%	70%	80%	90%
Lena	41.87	37.45	33.96	30.12	26.47	22.40	18.25	14.65	10.54
Cameraman	41.20	36.97	32.76	29.30	25.52	21.70	17.91	14.14	10.10
Barbara	33.13	29.76	27.46	25.32	23.04	20.34	17.24	14.40	10.17

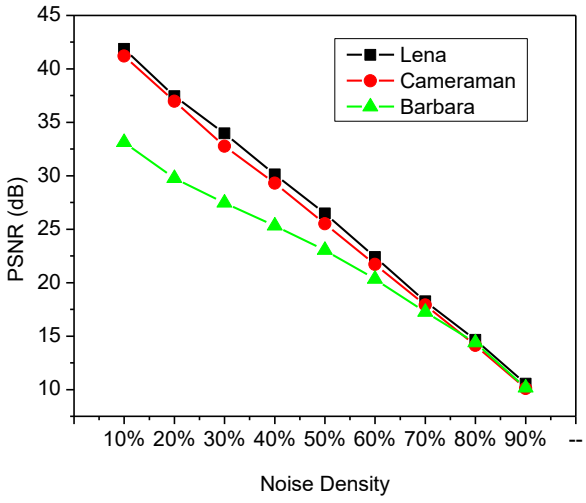


Figure 9 Graph between noise density and PSNR

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

TABLE III: EXPERIMENTAL RESULTS (LENA IMAGE) OF FPGA, MATLAB FOR THE SWITCING MEDIAN FILTER (3X3WINDOW SIZE)

Noise density (%)	Processing time (FPGA) (in sec)	Processing time (MATLAB) (in sec)
10	0.027	0.514684
20	0.043	0.840715
30	0.057	1.256706
40	0.071	1.582003
50	0.083	1.94383
60	0.093	2.322069
70	0.10	2.663722
80	0.11	3.030224
90	0.119	3.41398

Table III shows the results for the above implementation for a median filter window size of 3x3. It is observed that the processing time and percentage utilization of FPGA resources are linearly and directly proportional to image size.
The processing time of a mentioned algorithm with FPGA is compared with Matlab processing time. All the images use 512×512 size for their implementation.

VII. CONCLUSION

The successful implementation of image processing operations (median filtering) illustrates that a series of image processing algorithms useful for various applications can be efficiently implemented on FPGA hardware. A hardware implementation of algorithm is offered which gives improved result in terms of peak signal to noise ration. This work can be applied in pre-processing task in many applications, to perform operations like noise filtering, shape analysis, edge detection and many more.
The performance of the algorithm has been checked at different noise densities for gray-scale images on Virtex-7 FPGA VC707 kit with clock frequency 50MHz. At higher noise percentage amount, the algorithm reveals good outcome in equivalence with other existing algorithms. Visible and quantitative results are discussed. With regards to window size, for very small windows, like a 9-sample 3×3 window, and for median calculation, proposed method is both quick and compact.
The design can be extended to make it possible to automatically change mask sizes, for example 5x5, 7x7 or 9x9 whichever is needed. This can be useful when operating large size images where bigger mask size is useful or for adaptive image filter. Also this work can be modified to fit to real time applications where the user can directly load the input image by interfacing FPGA with the camera.

VIII. REFERENCE

[1] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing prentice hall." *Upper Saddle River, NJ* (2002).
[2] Petrou, Maria, and Costas Petrou. *Image processing: the fundamentals*. John Wiley & Sons, 2010.
[3] Vasicek, Zdenek, and Lukas Sekanina. "An area-efficient alternative to adaptive median filtering in fpgas." *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*. IEEE, 2007.

[4] Jinghong, Duan, Deng Yaling, and Liang Kun. "Development of image processing system based on DSP and FPGA." *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*. IEEE, 2007.
[5] Huang, T., G. J. T. G. Y. Yang, and G. Tang. "A fast two-dimensional median filtering algorithm." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.1 (1979): 13-18.
[6] Bittibssi, Tarek M., et al. "Image enhancement algorithms using FPGA." *International Journal of Computer Science & Communication Networks* 2.4 (2012): 536-542.
[7] Hwang, Humor, and Richard A. Haddad. "Adaptive median filters: new algorithms and results." *IEEE Transactions on image processing* 4.4 (1995): 499-502.
[8] Louverdis, Gerasimos, Ioannis Andreadis, and Antonios Gasteratos. "A new content based median filter." *Signal Processing Conference, 2004 12th European*. IEEE, 2004.

CITE AN ARTICLE

Bisht, Rashmi , and Ritu Vijay. "HARDWARE IMPLEMENTATION OF REAL TIME WINDOW BASED SWITCHING MEDIAN FILTER." *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY* 6.7 (2017): 732-40. Web. 25 July 2017.