INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

Parallel image filtering for faster image communication and comparative analysis

Akshat Singhal
akshat.singhal2016@vitstudent.ac.in
Vellore Institute of Technology, Vellore, Tamil Nadu

Nimit Agrawal
nimit.agrawal2016@vitstudent.ac.in
Vellore Institute of Technology, Vellore, Tamil Nadu

Jivjot Singh
jivjot.singh2016@vitstudent.ac.in
Vellore Institute of Technology, Vellore, Tamil Nadu

Saksham Sharma
saksham.sharma2016@vitstudent.ac.in
Vellore Institute of Technology, Vellore, Tamil Nadu

ABSTRACT

We all know that image processing is the field of massive calculations. So, in the major requirements list have parallel processing on the top. In this paper, we have demonstrated the parallel execution of image denoising. Also, the comparative analysis is there, to ensure that the algorithm takes sufficient time and show proficient results when parallel computing is done. Also, instead of taking a single image and applying all filters to it, we used multiple image array and loop through all the filters and followed by PSNR comparison of the same. We used parallel MATLAB for the parallel programming and then listed out all the parameters and their execution cost to be specifically determined.

Keywords— *Hybrid Filter, Median Filter, Mean Filter, Parallel, Social Media, Noise Removal*

1. INTRODUCTION

As in the literature review, we have addressed some problems and one of the main problems is distinctive properties available in different methodologies. This causes a problem that there is no optimal solution which can deal with all types of images and at the same time, it can address all types of densities in any situation. If the proposed methodology is properly executed, then using a single filter, we can process every type of image. Mainly and most widely, it is used in the application of chatting and sending images through any messenger application. When the user tries to send the image, the image is taken for image pre-processing, necessary filters and image layers are applied and then it is sent to the receiver. During transmission, there is some sort of disturbances and other factors which lead to distortion of images. Then, at the receiver's side, the image is corrected using image filters. Though the study of correct filters to be used to de-noise the image is very large, there are some filters that are used very frequently. Other applications of image filtering are medical treatments, etc. But another problem which arises is that the amount of data communicated daily is very large, and we need to use a technique which can help us to perform image filtering on many images simultaneously and in this paper, we have demonstrated the same.

2. EASE OF USE

2.1 Selection of noise

As, we have focused mainly on the application of different filters on the different noises, so for the ease of use and user understanding ability, we have chosen two noises which are Salt and pepper noise and Gaussian noise. And also, talking about the application, these two noises are mostly available in the distorted image. So, for better implementation and ease of use, we have chosen these noises.

2.2 Selection of filters

This is related to the selection of noises part, because we have to show the usage of all the filters and their application on different noises, so we have chosen averaging/mean filter, median filter, hybrid median filter and Gaussian filter which have different effects on different noises and thus, comparative study is presented later.

3. LITERATURE REVIEW

3.1 Image filtering

In early researches and surveys, about around 2012, the main considerations were the applications of image filtering along with the ways to achieve the same. But the problem they faced and dealt with is noise removal and parallel execution of the process [1] [3] [16]. Also, one of the early paper, they addressed and stated that a critical issue in image restoration is the removal of Gaussian noise. They proposed that to filter image is not a problem but to maintain its information integrity is a problem [16]. But later on, when the applications began to widen, some authors proposed an algorithm called HMF (Hybrid Median Filtering). Their fundamental point was to expel the impulsive noise from the image which can be used in transmission or after transmission to correct the polluted image [1]. The main thing they focused on was edge detection and correction. But there were some problems as only grey scales images are taken for

© 2019, www.IJARIIT.com All Rights Reserved

consideration [1] [16] [23]. But in a later study, this problem was considered and the plan was to develop a filter which can work on both grey scale images and coloured images [2] [3] [17]. Then, another hybrid filter, which consists of fuzzy and Kalman filter was considered and some good results were obtained [3]. But, the problem was completely solved here. A problem which is yet to be addressed was parallel execution of all the process and along with that, the improvisation of image filtering technique. Then, taking the fuzzy filter to de-noise the image, parallel execution of image noise calculation was done in order to make the algorithm of image filtering more efficient [2]. But there is also a problem that the parallelization was to detect only the noise, which, in some cases do not help, but only increases the complexity of the algorithm. Later on, an additional algorithm was discovered to address the issue of mixed noise which pollutes the image [17]. Then, in 2018, another algorithm was proposed, which address the problem of SAP noise and also, but the problem with the algorithm is that its complexity is very high [23]. In consequent research, the algorithm was proposed, generally which was not hybrid but applying all the filters sequentially till the noise is not removed which was called DAMF filter, the Different Applied Median Filter [24].

3.2 Applications

The detailed study of 25 research papers gave us a new insight into the astonishing amount of applications that the parallel image processing field has to offer. We got to know about the application of parallel image processing using ARM Processors wherein a new parallel architecture of parallel image processing was discussed with the assistance of ARM processors incorporated with ordinary parallel engineering and the ARM Processors were utilized as standard innovation [7]. Next, we got to know about another application in the medical imaging domain, in brain anatomy. Parallel imaging was used to enhance the Echo Planner Imaging-Diffusion Weighted Images (EPI-DW) of the brain. The technique proved to be very successful in enhancing the images of the occipital lobe of the brain [8]. Next, we saw the use of GPGPU in Parallel Imaging. We saw a method to decrease the harm done to the brain cells amid a cerebrum stroke which decreased the examination time for a stroke with a specific end goal to build the potential for recuperation. The method used a GPGPU to perform fast calculations in parallel to accelerate the cerebrum examining process [9]. Then, we saw a methodology to use parallel block processing which in the present day scenario is dominated by the K-means approach. The proposed technique has been used on Ortho Imagery Satellite Images [10]. Then we saw a parallel approach to Medical denoising using NLM (Non-Local Means). In the method, the established NLM calculation is enhanced to denoise medical images by including a novel noise weighting function and parallelizing and this method was found to be better than conventional methods using NLM [14]. Next, we saw Parallel Processing for High-Performance Image Processing wherein the authors have discussed the way in which optimum performance can be achieved in this field. A mix of custom equipment and advanced programming instrument would give elite for the computationally concentrated ongoing frameworks and hence, we get to know how high-performance computing works in image processing [18]. Then we saw the application of medical processing on GPU. In this method, five key factors were identified: Data parallelism, thread count, branch divergence, memory usage and synchronization. The papers reviewed, investigated GPU computing in several medical image processing areas such as image registration, segmentation, denoising, filtering, interpolation and reconstruction. The conclusion was that a hybrid CPU-GPU system works best [21]. Last but not least, we saw the application of parallel imaging in Satellite Image Processing. In this method, we got to know about support vector machines (SVM) and about genetic algorithms (GA). This method reviewed various SVM implementations and GPU based algorithm approaches. A simple GA is also implemented on CPU and GPU and their results were compared and studied [25]. Hence, the above applications were found for the domain of parallel image processing.

3.3 Parallel matlab

Matlab is an omnipresent numerical processing device utilized by researchers, designers and others in a wide scope of utilization from computational material science to sociologies. The expanding unpredictability and size of these reproductions require extra computational power - not accessible on a work area - through the utilization of parallel figuring. Matlab/Octave is basically utilized in consecutive mode. Be that as it may, Matlab gives worked in parallelization bolster through Parallel Computing Toolbox (PCT) [4] The Parallel Computing Toolbox gives parallel builds in the MATLAB dialect, for example, parallel for loops, distributed clusters, and message passing, empowering quick prototyping of parallel code through an intelligent parallel MATLAB session. [13] Utilizing PCT, the MATLAB can permit unravelling picture handling issues utilizing multi-centre processors, GPUs, and PC groups. To help this capacity, the abnormal state builds, e.g. parallel for-circles, extraordinary exhibit composes, are bolstered, which enable us to parallelize MATLAB applications without CUDA or MPI programming.[19] HPC Matlab demonstrates preferable execution over the parallel processing tool kit and the disseminated figuring server from Math works, for both MPI correspondence schedules and also P threads module.[4] The PID exhibited a critical speedup in preparing medicinal pictures contrasted and successive capacities.[19]

3.4 Other parallel methods

Images are stored and processed as matrices of pixels. As the image resolution and noise increase, the computation for enhancement, classification or even filtering becomes highly complex. To reduce time complexity, parallel computation is becoming important. Most methods in Image Processing and filtering discussed in the papers under this category use hardware which is better suited to large computations, like GPUs (Graphics Processing Unit) [11][21][25][20][22] and MICs (Many Integrated Cores) [15]. With such hardware which facilitate huge computations through multiple threads, there arise certain problems like - low RAM capacity to store intermediate values and high transfer time for data due to this memory latency. In most cases, a majority of the computation time is dedicated to memory latencies, with up to 92% in paper [25]. Although there are memory latencies and considerable communication costs for data transfers between the CPU and the GPU, the speed up in almost all papers is significantly good (30-35%). Other methods include the programmer to write algorithms with separate serial and parallel ports [5] [12] for multicore systems (OpenMP). This method may not have the support of dedicated hardware like GPUs but gives the programmer freedom of choice with synchronization (blocking or lock-free), chunk sizes and decomposition techniques. This helps the programmer to tailor his algorithm according to the problem for higher performance and speedup. However, this generates problem and hardware specific algorithms, not universal algorithms which have the same performances across all data sets and images.

4. NOISES

4.1 Gaussian Noise

Gaussian commotion is one kind of factual noise. It is equitably appropriated over the flag. The Gaussian moves are additionally

© 2019, www.IJARIIT.com All Rights Reserved

called the ordinary noise and it is a noteworthy piece of reading commotion of a picture sensor that is of the steady commotion level, in the dull region of the picture. The likelihood thickness work (PDF) of Gaussian commotion is equivalent to that of the ordinary conveyance and furthermore known as Gaussian dissemination. It is normally utilized as added substance repetitive sound give Added Substance White Gaussian Commotion (AWGN).

$$P(z) = \frac{1}{\sqrt{2\pi\sigma}}\, e^{-\frac{Z-\mu^2}{2\sigma^2}}$$

Where z speaks to grey level, μ is the mean of normal estimation of z and σ is its standard deviation.

4.2 Salt pepper

At the point when a simple picture flag is transmitted in a direct dispersive channel, the picture edges (step-like or heartbeat like a flag) get obscured and the picture flag gets debase with added substance white Gaussian clamor since no viable channel is without commotion. On the off chance that the channel is poor to the point that the clamor difference is sufficiently high to make the flag excurse to high negative or positive esteem when the thresholding task at the front end of the recipient will contribute soaked min and max esteem. This kind of clamor is called salt and pepper commotion. In the meantime, the picture contain the dim is called pepper and the picture contain the splendid pixel is known as salt. There for the simple picture flag is transmitted and the flag gets undermined with Additive White Gaussian Noise and Salt and Pepper too. At that point, there is an impact of blended commotion.

The PDF of salt and Pepper noise function

$$P(z) = \begin{cases} P_x & \text{for } z = x \\ P_y & \text{for } z = y \\ 0 & \text{otherwise} \end{cases}$$

4.3 Basis of comparison

- **Mean Square Error (MSE):** MSE measures the contrast between the estimations of the unique picture and the resultant picture. MSE is a capacity relating to the normal estimations of the squared blunder. MSE measures the normal of the squares of the blunders. The blunder is the distinction between the esteem suggested by the estimator and the amount to be evaluated.
- **Peak Signal to Noise Ratio (PSNR):** PSNR is utilized to quantify the nature of recreated picture from the first picture. PSNR is communicated as far as the logarithmic decibel scale. Higher PSNR, by and large, demonstrates that the yield picture is of higher quality.

5. FILTERS

5.1 Averaging filter

Averaging filtering takes a window containing average values of the input image's pixels and passes it over the image. This is a low pass filter and is quite a simple one.

The 3X3 averaging filter window:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

5.2 Median filter

Median filtering takes a window containing the median value of the pixel values of the input image and passes it over the input image as a mask. It is most effective for Speckle noise and salt and pepper noise. It is used for smoothing techniques.

•	•	•
•	X	•
•	•	•

5.3 Hybrid Median Filter

Hybrid median filtering employs the usage of a hybrid median filter. It calculates 3 medians.

MR- Median of horizontal and vertical row column values

MD- Median of Diagonal values

The final result is the median of MR, MD and central pixel.

This process is known as hybrid median filtering.

5X5 Hybrid Median Filtering window:

•				•
	•		•	
•	•	X	•	•
	•		•	
•				•

6. METHODOLOGY

We have followed a simple methodology for the implementation of our idea in which we have divided our work into modules and then, we have implemented modules separately and after that integrated them to work as a final module. So the steps we followed are as follows:

Step 1: Dividing the work in a module on the basis of filter usage, noises and based on error comparison

Step 2: For each applied noise and image polluted, we further divided the modules into parts on the basis of filters used in recovering those images

Step 3: After that, for each image and each filter applied on it, we divided the filter application into two parts, which is serial execution and parallel execution

Step 4: Then, the next module, which is error comparison part, it was divided into sub-modules which are, for each noisy image and every filter applied, Mean-squared error and PSNR ratio was calculated.

Step 5: Then, the next module was the error comparison part, for which tabular column was pre-formed and the values were filled when we integrated al the modules and obtained the work distribution output.

Step 6: Then, both the serial and parallel implementation was processed and results were obtained

Step 7: Then, the results were analysed on part of bytes transferred to each worker and also the timely execution of each filter by the parallel workers. And then, the time taken for the serial execution for the same was recorded.

7. ALGORITHM

FOR i=1 to Number_of_images

    Image = image_database(i)
    INITIALIZE GPU_ARRAY
    INITIALIZE PARALLEL POOL
    IB = POLLUTE(Image , Salt and pepper)
    IC = POLLUTE(Image , Gaussian)

```
    PLOT(Image,IB,IC)
    SET N_WINDOW = 3X3
    IF = APPLY_MEAN(IB, N_WINDOW)
    IF2 =APPLY_MEAN(IC, N_WINDOW)
    PLOT(IF1)
    PLOT(IF2)
    PARALLELFOR(every_layer_of_image)
       IMG_M= APPLY_MEDIAN(IB,N_WINDOW)
      UPDATE IMG_M
    END PARALLELFOR
    PARALLELFOR(every_layer_of_image)
       IMG_P= APPLY_MEDIAN(IC,N_WINDOW)
       UPDATE IMG_P
    END PARALLELFOR
    PLOT(IMG_M)
    PLOT(IMG_P)
    IH = APPLY_HYBRID_MED(IB, N_WINDOW)
    IH2 =APPLY_HYBRID_MED(IC, N_WINDOW)
    PLOT(IH1)
    PLOT(IH2)
    IG = APPLY_GAUSSIAN(IB)
    IG2 =APPLY_GAUSSIAN(IC)
    PLOT(IG)
    PLOT(IG2)
    ERR1= COMPUTE(MEAN_ERROR(Image,IB))
    ERR2 = COMPUTE(PSNR(Image,IB))
    ERR3= COMPUTE(MEAN_ERROR(Image,IC))
    ERR4 = COMPUTE(PSNR(Image,IC))
    ERR5= COMPUTE(MEAN_ERROR(Image,IF))
    ERR7 = COMPUTE(PSNR(Image,IF2))
    ERR8= COMPUTE(MEAN_ERROR(Image,M))
    ERR9 = COMPUTE(PSNR(Image,M))
    ERR10= COMPUTE(MEAN_ERROR(Image,P))
    ERR11= COMPUTE(PSNR(Image,P))
    ERR12=COMPUTE(MEAN_ERROR(Image,IH)
    ERR13= COMPUTE(PSNR(Image,IH))
    ERR14=COMPUTE(MEAN_ERROR(Image,IH2
    ERR15= COMPUTE(PSNR(Image,IB))
    ERR16=COMPUTE(MEAN_ERROR(Image,IG)
    ERR17= COMPUTE(PSNR(Image,IG))
    ERR18=COMPUTE(MEAN_ERROR(Image,IG2
    ERR19= COMPUTE(PSNR(Image,IG2))
END FOR
```

8. EXPERIMENTAL RESULTS

8.1 Original image

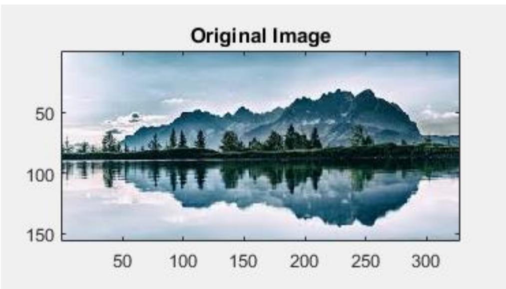

Fig. 1: Original image

8.2 Salt and Pepper Noise

(a) Noisy image

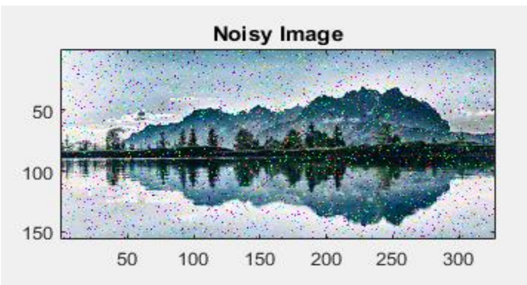

Fig. 2: Noisy image

(c) After Averaging filter

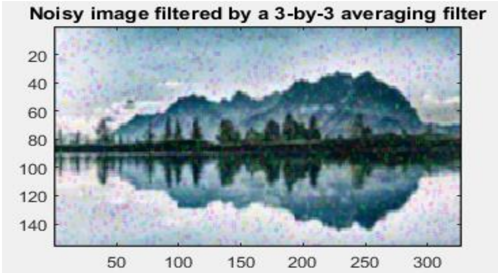

Fig. 3: After Averaging filter

(d) After Median filter

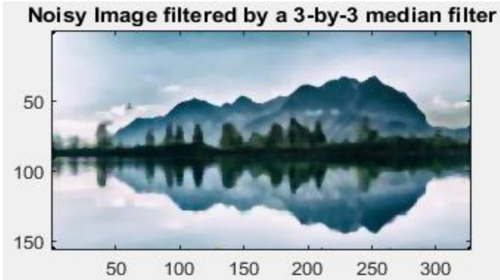

Fig. 3: After Median filter

(e) After Hybrid median filter

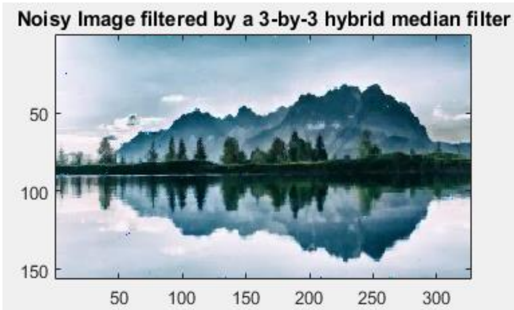

Fig. 4: After Hybrid median filter

(f) After Gaussian Filter

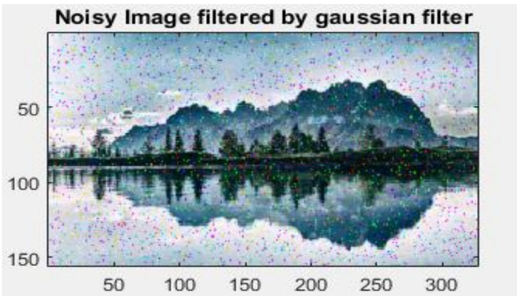

Fig. 5: After Gaussian filter

8.3 Gaussian Noise

(a) Noisy image

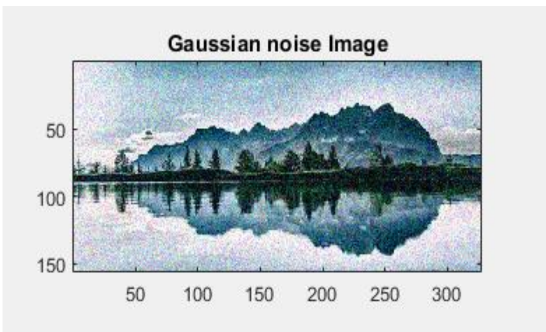

Fig. 6: Noisy image

© 2019, www.IJARIIT.com All Rights Reserved

(b) After Averaging filter

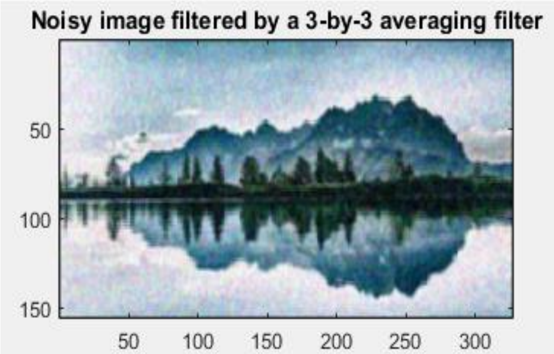

Fig. 7: After Averaging filter

(c) After Median filter

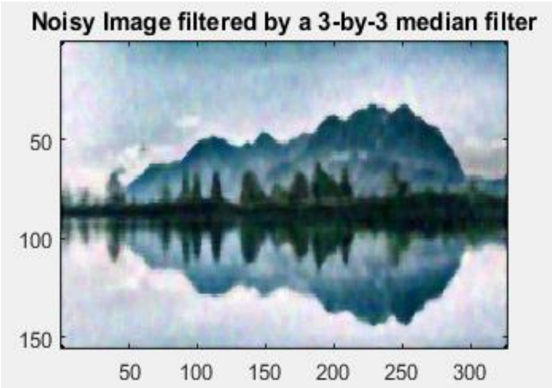

Fig. 8: After Median filter

(d) After Hybrid Median Filter

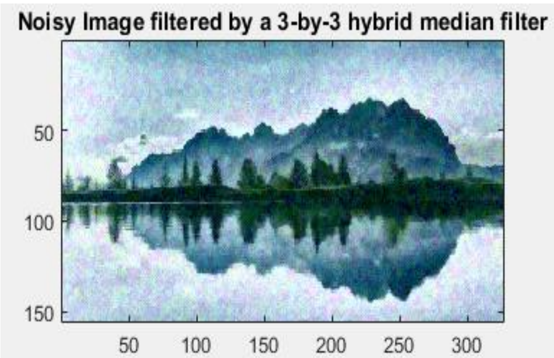

Fig. 9: After Hybrid Median filter

(e) After Gaussian filter

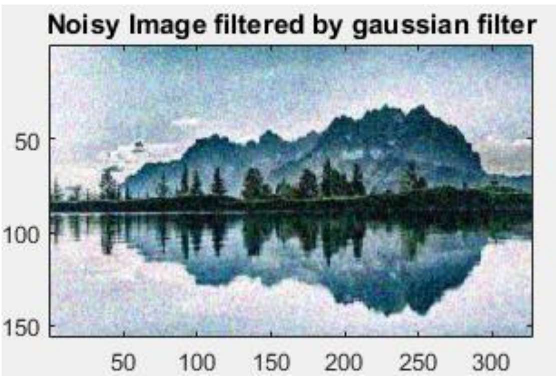

Fig. 10: After Gaussian filter

The noise-added images and their filtered versions are given below:

Table 1: Error and the PSNR of the filters on Salt and Pepper noise

	Noisy	**Mean**	**Median**	**Hybrid Median**	**Gaussian**
Mean square error	1136.4755	403.0284	277.7739	103.3620	511.7213
PSNR	17.5752	22.0774	23.6939	27.9872	21.0405

Table 2. Error and the PSNR of the filters on Gaussian noise

	Noisy	**Mean**	**Median**	**Hybrid Median**	**Gaussian**
Mean square error	567.4853	321.5199	330.4323	302.2432	266.3625
PSNR	20.5913	23.0587	22.9400	23.3272	23.8761

Table 3. Comparison of Serial and parallel execution

Filter	**Time taken in serial construct**	**LOC**	**Parallelisation method**	**Time taken in parallel construct**	**LOC**
Mean	0.915254	14	GPU	0.861611	14
Median	2.7313	20	Parfor loop	2.6789	28
Hybrid-median	5.7198	13	Not applicable	5.7198	13
Gaussian	0.519087	13	GPU	0.159007	13

9. CONCLUSION

From the above results, we can conclude that serial execution takes a lot more time than parallel execution. Also, Gaussian filter was found to be the best for Gaussian noise affected images and hybrid median filter was found to be the best for salt and pepper noise affected images. Also, the averaging filter had the least efficiency while being used on Gaussian noise and salt and pepper noise. We also tried GPU programming and we found out that we could easily parallelise mean and Gaussian filtering using GPU. However, hybrid median filtering and median filtering processes did not run on GPU. Also, upon executing filters in GPU, the time fluctuated a lot and was not very reliable as a source of parallel-serial comparison. Therefore, parallel Matlab was found to be a much reliable way of comparing parallel and serial executions. Hence, we can conclude that parallel programming is a much better way of processing images as compared to serial programming.

10. REFERENCES

[1] Rakesh, M. R., Ajeya, B., and Mohan, A. R. (2013). Hybrid Median Filter for Impulse Noise Removal of an Image in Image Restoration, 5117–5124.
[2] Uddin, M. S. (2015). An Efficient Approach for Removal of Impulse Noise from Highly Corrupted An Efficient Approach for Removal of Impulse Noise from Highly Corrupted Images, (December). https://doi.org/10.15764/STSP.2015.01004
[3] Ali, E. H., Karam, E. H., and Mohammed, M. S. (2013). Mixed-Noise Reduction by Using Hybrid (Fuzzy and Kalman) Filter For Gray and Color Images, 4(11).
[4] Guo, X., Dave, M., and Sayeed, M. (2016). HPCmatlab : A Framework for Fast Prototyping of Parallel Applications in Matlab. Procedia - Procedia Computer Science, 80, 1461–1472. https://doi.org/10.1016/j.procs.2016.05.467
[5] Du, J., Li, W., and Xiao, B. (2018). The fusion of anatomical and functional images using parallel saliency features. Information Sciences, 430–431, 567–576. https://doi.org/10.1016/j.ins.2017.12.008

© 2019, www.IJARIIT.com All Rights Reserved

[6] Saxena, S., and Sharma, N. (2016). Parallel Image Processing Techniques, Benefits and Limitations. Research Journal of Applied Sciences, Engineering and Technology, 12(2), 223-238.
[7] Ahmad, Z.A., Elshaikh, M., nor, C.M., Mustafa, M.S., Romli, M.A., and Jamlos, M.F. (2012). Low-Cost Parallel Processing System for Image Processing Applications. Procedia Engineering, 41, 771-776.
[8] Zaki, M.M., Hegazy, E.A., Abdelhamid, M.I., El-Mogy S.A. (2018). The role of parallel imaging technique in decreasing geometric distortion artifact in diffusion-weighted imaging of the brain. The Egyptian Journal of Radiology and Nuclear Medicine, https://doi.org/10.1016/j.ejrnm.2018.05.009
[9] Zhu, F., Gonzalez, D.R., Carpenter, T., Atkinson, M., Wardlaw, J. (2012). Parallel Perfusion Image Processing using GPGPU. Computer Methods and Programs in Biomedicine, http://dx.doi.org/10.1016/j.cmpb.2012.06.004
[10] Rashmi, C., Chaluvaiah, S., Kumar, G.H. (2016). An Efficient Parallel Block Processing Approach for K-Means Algorithm for High-Resolution Ortho imagery Satellite Images. Procedia Computer Science, 89, 623-631.
[11] Aldinucci, M., Spampinato, C., Drocco, M., Torquati, M., and Palazzo, S. (2012). A parallel edge preserving algorithm for salt and pepper image denoising. 2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA). doi:10.1109/ipta.2012.6469567
[12] Nguyen, T., Nakib, A., and Nguyen, H. (2016). Medical image denoising via the optimal implementation of non-local means on hybrid parallel architecture. Computer Methods and Programs in Biomedicine, 129, 29-39. doi:10.1016/j.cmpb.2016.02.002
[13] Mustafa, Z. A., Abrahim, B. A., Tom, K. M., and Garma, F. B. (2015). A New Technique for Parallel Image Denoising. Journal of Clinical Engineering, 40(3), 144-149. doi:10.1097/jce.0000000000000101
[14] Mingliang, X., Pei, L., Mingyuan, L., Hao, F., Hongling, Z., Bing, Z., Liwei, Z. (2016). Medical image denoising by parallel non-local means. Neurocomputing, 195, 117-122. doi:10.1016/j.neucom.2015.08.117
[15] Huang, F., Lan, B., Tao, J., Chen, Y., Tan, X., Feng, J., and Ma, Y. (2017). A Parallel Nonlocal Means Algorithm for Remote Sensing Image Denoising on an Intel Xeon Phi Platform. IEEE Access, 5, 8559-8567. doi:10.1109/access.2017.2696362
[16] Prajapati, H. B., and Vij, S. K. (2011). Analytical study of parallel and distributed image processing. 2011 International Conference on Image Information Processing, (Iciip), 1–6. https://doi.org/10.1109/ICIIP.2011.6108870
[17] Mustafa, Z. A., Abrahim, B. A., and Kadah, Y. M. (2012). K11. Modified Hybrid Median filter for image denoising. National Radio Science Conference, NRSC, Proceedings, (Nrsc), 705–712. https://doi.org/10.1109/NRSC.2012.6208586
[18] Hemnani, M. (2016). Parallel processing techniques for high-performance image processing applications. 2016 IEEE Students Conference on Electrical, Electronics and Computer Science (SCEECS). doi:10.1109/sceecs.2016.7509316
[19] Deng, X., and Liu, Z. (2016). An improved image denoising method applied in resisting mixed noise based on MCA and median filter. Proceedings - 2015 11th International Conference on Computational Intelligence and Security, CIS 2015, 162–166. https://doi.org/10.1109/CIS.2015.47
[20] Goes, M. F., Sinhoreti, M. A., Consani, S., and Silva, M. A. (1998). Morphological effect of the type, concentration and etching time of acid solutions on enamel and dentin surfaces. Brazilian Dental Journal, 9(1), 3–10. https://doi.org/10.1088/1742-6596/755/1/011001
[21] Khor, H. L., Liew, S., and Zain, J. M. (2015). A review on parallel medical image processing on GPU. 2015 4th International Conference on Software Engineering and Computer Systems (ICSECS). doi:10.1109/icsecs.2015.7333121
[22] Sarjanoja, S., Boutellier, J., and Hannuksela, J. (2015). BM3D image denoising using heterogeneous computing platforms. 2015 Conference on Design and Architectures for Signal and Image Processing (DASIP), 1–8. https://doi.org/10.1109/DASIP.2015.7367257
[23] Kumar, S. V., and Nagaraju, C. (2018). Support vector neural network-based fuzzy hybrid filter for impulse noise identification and removal from the gray-scale image. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci 2018.05.011
[24] Gökrem, L., and Engino, S. (2018). Different applied median filter in salt and pepper noise, 0, 1–10. https://doi.org/10.1016/j.compeleceng 2018.01.019
[25] Buche, S. B., Dhondse, S. A., and Khobragade, A. N. (2016). Satellite image processing on parallel computing: A technical review. 2016 Online International Conference on Green Engineering and Technologies (IC-GET). doi:10.1109/get.2016.7916718.

© 2019, www.IJARIIT.com All Rights Reserved