International Journal of Engineering Science Invention Research & Development; Vol. III, Issue IX, March 2017
www.ijesird.com, e-ISSN: 2349-6185

Novel Processor UtilizationMeasurement for Image Denoising

Jingwen Du, New York University (USA), Rongke Xu, John Carroll School (USA)

Zihan Lin, Dalian American International School (China), Shengxue Xiao, Anshan No. 1 High School (China)

Chenshun Hong, Dalian Jiahui M.S. (China)

I. ABSTRACT

Measuring processor utilization is a key factor for making decisions and validating the proposed algorithm. Parameter values for measuring processor utilization can act as base work and can explore different areas where improvement is needed for future work. This requirement increases considerably when the measurement has to be done on a real-time basis. Present work represents some performance metrics such as the percentage of idle time, processor time, maximum frequency, and processor utility that gives many accurate readings for the measurement of processor utilization when an algorithm median filter is executed for image denoising. Utilization of same parameters can be extended to an algorithm, and precise results can be retrieved.Different data sets of images are used to explore values of different performance metrics.

Keywords:Processor utilization, idle time,images, amedian filter.

II.INTRODUCTION

A. Noise

Distraction which comes into existence from number of sources like sensor temperature, switching, image acquisition, atmospheric disturbance while transmission etc. can result in degradation of the quality of image [7][8] is known as noise for a digital image

Suppose animage is given by j(x,y), noise introduced is m(x,y) and distorted image at the location (x,y) is *f(x,y).*

Then the image corrupted by additive noise is given by

$$f(x,y) = j(x,y) + \eta(x,y)$$

Similarly, if noise is multiplicative then the corrupted image will be written as

$$f(x,y) = j(x,y) * \eta(x,y)$$

Thus, a noise manipulates the actual pixel values of an image. To get an original image certain denoising techniques have to be used so that quality of an image can be increased.

B. Median Filter

A nonlinear process known as Median filtering helps in reducing salt and pepper or other impulsive noise from the image. While reducing the random noise this technique preserves the edges of a given image. In the median filter formula in theequation is applied to a given window size of theimage and the median of thevarious intensity value of the pixels becomes the resultant intensity for the pixels under processing.

$$g(a,b) = median\left(\sum_{i=-1}^{1}\sum_{j=-1}^{1} f(a-i, b-j)\right)$$

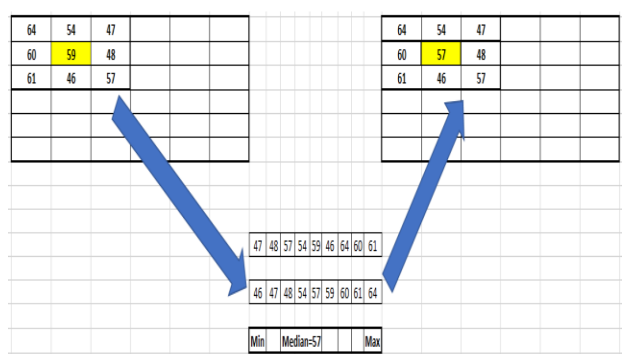

Fig.1. Showing pixel conversion after median calculation

Fig.2. Showing Original, Noisy and Smothed image after denoising

III. LITERATURE SURVEY

Bagga et al. [1] in their research paper defines various parameters like efficiency, speedup, overall execution time and parallel excessive overhead to measure the proposed approach for cache oblivious matrix multiplication. These are the parameters which help in themeasurement of the effectiveness of the cluster deployed. Any distributed computing can make use of these parameters for improvement.

Kaur et al. [6] in their research paper proposed an approach in which image processing can be performed on a cluster in parallel. Images of large size can be further broken down into small images and image filtering algorithms can be applied to all these sub images simultaneously. The effectiveness of the proposed approach is measured using efficiency, speedup, overall execution time and parallel excessive overhead. Authors have used average filtering techniques for denoising the images.

Bagga et al. [3] in their research paper creates virtualization approach for implementing RMI based cluster in the proposed work Winograd's variant of Strassen's method is used for multiplication of the two matrices on a virtual cluster. A cluster from a single system is being created using the virtualization technique. Parallel execution of the number of tasks can be done in the proposed approach. Efficiency, speedup, overall execution time and parallel excessive overhead are the parameters used for measurement. Virtualization helps in sharing of same hardware(NIC, CPU, Memory) among various guest operating systems, as a result, efficiency increases.

Arora et al. in their research paper focus on demand adjustment schemes by considering synthetically generated workload and processor availability mapped with discrete clock frequency. The main goal of the proposed work is to make consistent processor availability management i.e. processor offered space to analyze various scheduling algorithms with different performance parameters. Load adjustment(LAD), Processor required space(PRS), Processor offered space(POS), throughput, currently, active list(CAL), Excessive cycle length(ECL), Processor utilization per process(PUPP) are the various parameters used for measuring the effectiveness.

Cheema et al. [2] in their research paper explains interlaced graphics mechanism with run-length encoding to achieve high compression benefits. Row-Column Interlace Speedup Variations Speed up, Row Interlace Parallel Overhead, Row-Column Interlace Efficiency per cluster machine, Row-Column Interlace Parallel overhead, overall execution time are some of the performance metrics parameters being discussed to validate the proposed work.

Present work explores the performance evaluation parameters provided by the Microsoft windows for real time analysis of processor, memory, network utilization of the image processing technique used. Definition of such parameters is explained in section VI.The effectiveness of any algorithm, working on cache miss improvement, network traffic management can be checked using these parameters.

IV. APPLICATIONS OF DIGITAL IMAGE PROCESSING

In a digital world, where everything is transformed into the digital form so that processing can be done using digital computers. Digital images play a crucial role in collecting the data in the digital form. Digital data is more reliable and can be reproduced and reused as many times as required. There is awide range of applications of in which the image processing plays an important role, some of them are:

- Medical Applications
- Robot vision
- Digital cinema
- Remote Sensing
- Registration Techniques
- Pattern Recognition
- Restorations and Enhancements

V. System configuration and dataset used for Observations

System Configuration along with the detail of the images used is given in Table I.

Table I. Detail of System and dataset

S.No.	Size of image (MB)	Dimensions of image (Pixels)	System Configuration
1	1.79	4233 x 4233	**System Type:** x64-based PC
2	3.47	5106 x 5106	**Processor:** Intel(R) Core(TM) i5-2450MCPU @ 2.50GHz, 2501 Mhz, 2 Core(s), 4 Logical Processor(s)
3	7.18	6003 x 6003	
4	8.95	7005 x 7005	
5	10.40	8004 x 8004	
6	13.30	9003 x 9003	
7	29.00	10023 x 10023	**Physical Memory (RAM)**: 4.00 GB

VI. PERFORMANCE METRICS

% Idle Time**:**It is the percentage of time the processor is idle during the sample interval

% Processor Time**:** It is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then subtracting that value from 100%. (Each processor has an idle thread to which time is accumulated when no other threads are ready to run). This counter is the primary indicator of processor activity and displays the average percentage of busy time observed during the sample interval. It should be noted that the accounting calculation of whether the processor is idle is performed at an internal sampling interval of the system clock tick

% of Processor Utility: It is the amount of work a processor is completing, as a percentage of the amount of work the processor could complete if it were running at its nominal performance and never idle. On some processors, Processor Utility may exceed 100%.

% of Maximum Frequency: It is the percentage of the current processor's maximum frequency. Some processors can regulate their frequency outside of the control of Windows. % of Maximum Frequency, will not accurately reflect actual processor frequency on these systems.

VII. WORKFLOW IN THE GIVEN PROPOSED WORK

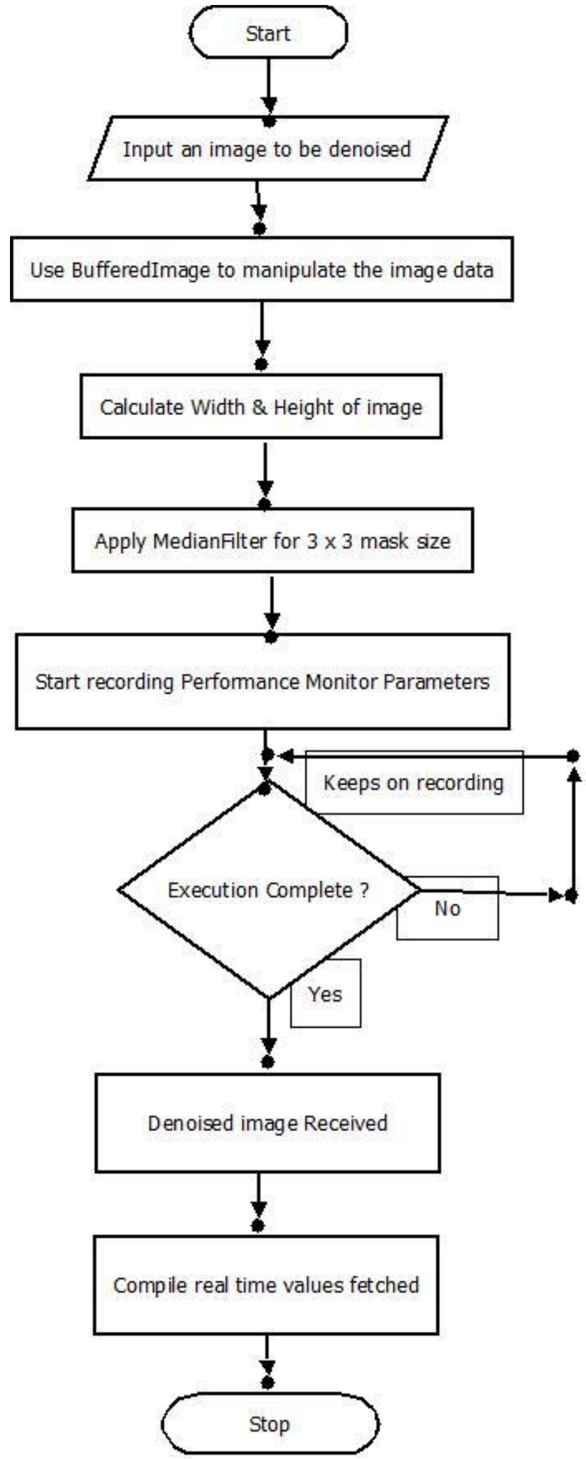

Fig.3. Showing workflow of proposed approach

1. Image as an input is taken from the user,to find out its length and width.

2. As soon as the Java program for applying MedianFilter is started, parallely recording of the various processor's performance evaluation parameter's is started.
3. A BuffferedImage object is created corresponding to the given image for manipulation of an image.
4. Selection of a mask of 3x3 for a given image is calculated.
5. Including the target pixel, there will be atotal of 9 pixels.
6. The colours corresponding to the 8 pixels around the target pixel are captured using getRGB() function.
7. After isolating the R,G,B values of each pixel an array is created for each color.
8. An array is then sorted using the Arrays.sort() function available in java .
9. The Middle value of all arrays is calculated i.e R[4],G[4],B[4].
10. This value is used for replacing all the 9 pixels.
11. The value of thetarget pixel is set and thesame process is repeated for the whole image using the mask of 3x3.
12. After completion of the given algorithm, the recording of various parameter's is also stopped.
13. Values corresponding to various parameters is compiled.
14. The final denoised image is being calculated write() function.

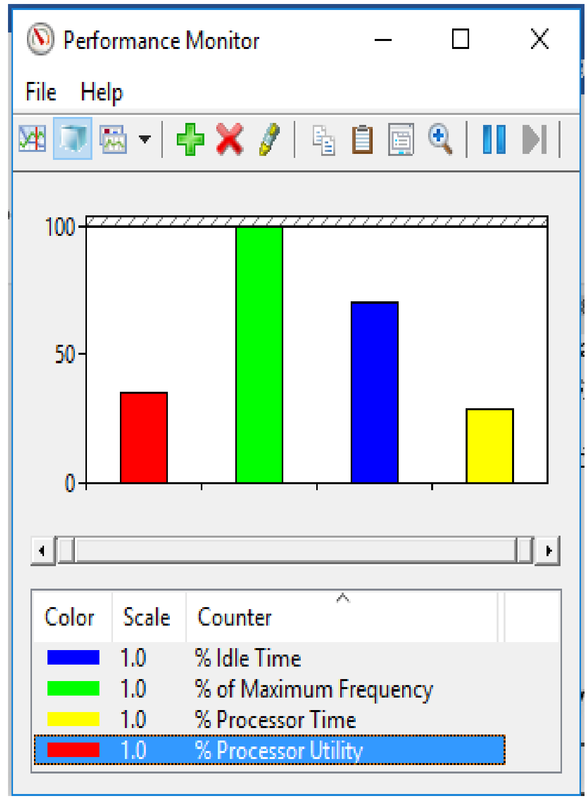

Fig.4. Showing Performance metrics for image dimensions 5106 x 5106

VIII. RESULTS AND DISCUSSION

Values of performance metrics for various images is shown in TABLE II,III,IV,V,VI,VII.

TABLE II. Showing values of performance metrics for image dimensions 5106 x 5106

Parameter 5106	Value
% Idle Time	70.649
% Processor Time	28.495
% Maximum Frequency	100
% Processor Utility	35.366

TABLE III. Showing values of performance metrics for image dimensions 6003 x 6003

Parameter 6003	Value
% Idle Time	72.151
% Processor Time	26.906
% Maximum Frequency	100.00
% Processor Utility	33.737

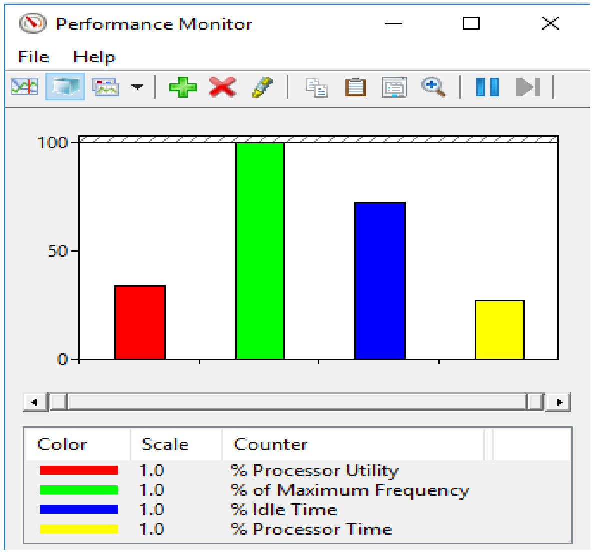

Fig.5. Showing Performance metrics for image dimensions 6003 x 6003

TABLE IV. Showing values of performance metricsfor image dimensions 7005 x 7005

Parameter 7005	**Value**
% Idle Time	65.746
% Processor Time	33.299
% Maximum Frequency	100.0
% Processor Utility	40.907

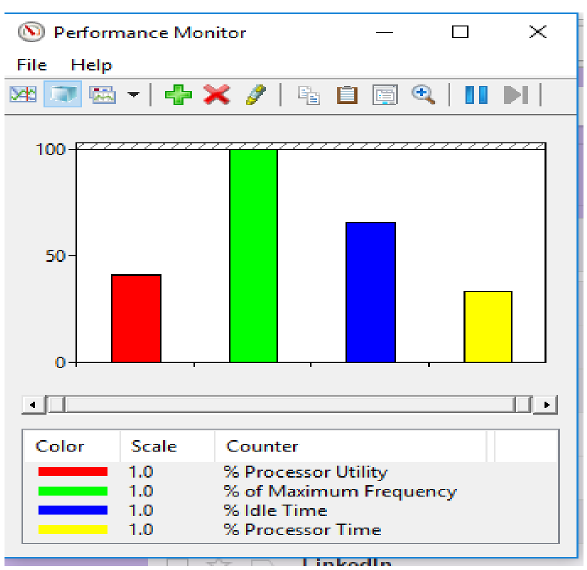

Fig.6. Showing Performance metrics for image dimensions 7005 x 7005

TABLE V. Showing values of performance metrics for image dimensions 8004 x 8004

Parameter 8004	**Value**
% Idle Time	72.33
% Processor Time	27.166
% Maximum Frequency	100.0
% Processor Utility	33.605

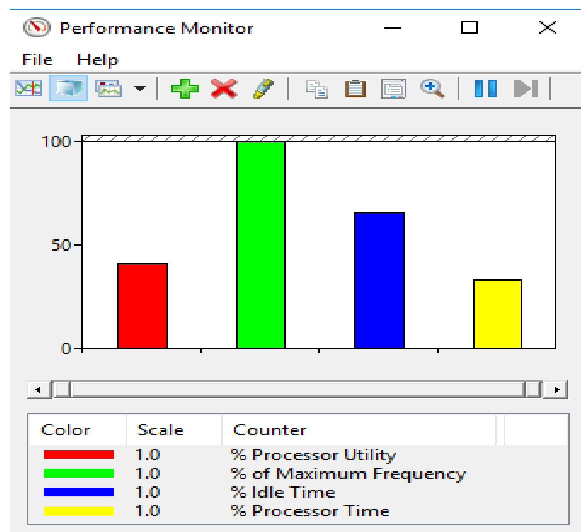

Fig.7. Showing Performance metrics for image dimensions 7005 x 7005

TABLE V1. Showing values of performance metricsfor image dimensions 9003 x 9003

Parameter 9003	**Value**
% Idle Time	70.584
% Processor Time	28.392
% Maximum Frequency	99.145
% Processor Utility	35.229

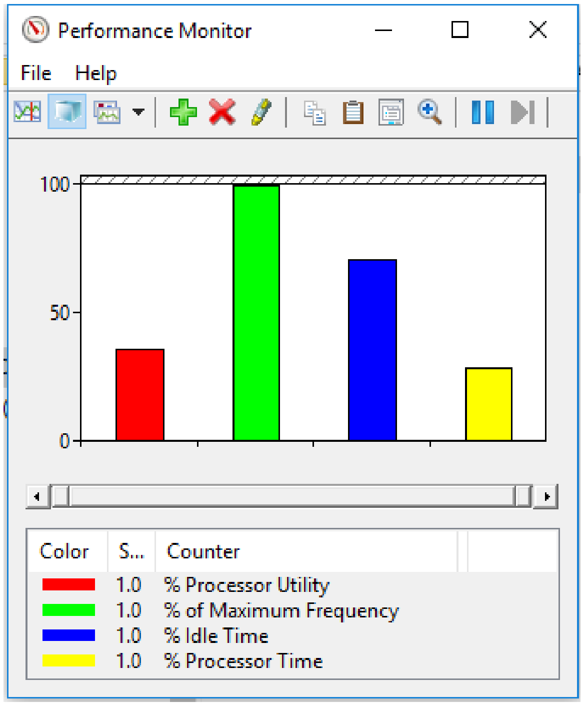

Fig.8. Showing Performance metrics for image dimensions 9003 x 9003

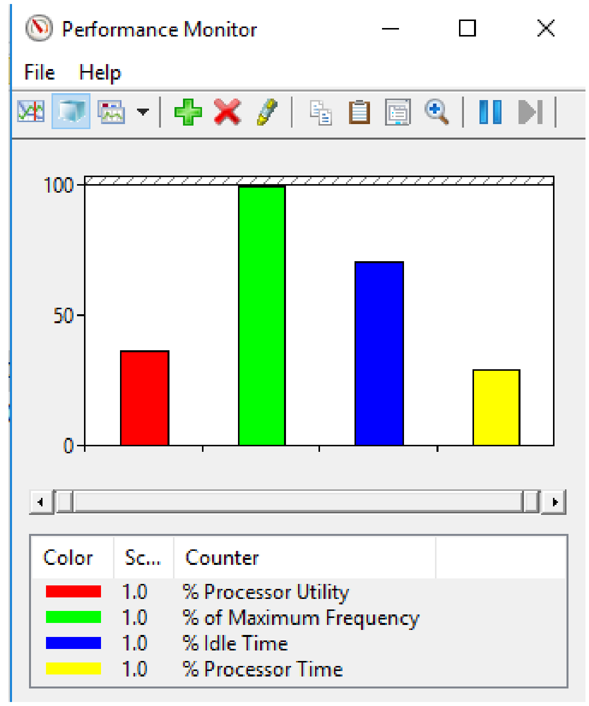

Fig.9. Showing Performance metrics for image dimensions 10023 x 10023

TABLE VII. Showing values of performance metricsfor image dimensions 10023 x 1003

Parameter 10023	**Value**
% Idle Time	70.126
% Processor Time	28.710
% Maximum Frequency	99.289
% Processor Utility	36.105

IX. CONCLUSION AND FUTURE SCOPE

From the above work, it can be concluded that although during the sample interval Maximum Frequency reached is about 100 % still for themaximum amount of time system remains idle and as per the values are shown by the parameter processor utility, it can be infered that processor is highly under utilized by the given algorithm. As per the parameter Processor Time percentage of elapsed time that the processor spends to execute a non-Idle thread is very low and it must be increased. Thus, we can conclude that for an algorithm to be effective processor utilizer for the given value of Maximum Frequency utilization under given span of time, an algorithm it must have alow value of Idle time. The high value of Processor Time and maximum processor Utility.

References

[1]	S. Bagga, A. Girdhar, M. Trivedi and Y. Yang, "RMI Approach to Cluster-Based Cache Oblivious Peano Curves," in *International Conference on Computational Intelligence & Communication Technology*, Ghaziabad, 2016.
[2]	Ankit Arora, Sachin Bagga and Rajbir Singh Cheema. Article: Distributed Cluster Processing to Evaluate Interlaced Run Length Compression Schemes. International Journal of Computer Applications 46(6):26-33, May 2012
[3]	S. Bagga, A. Girdhar, R. Yan and Z. Lin, "Virtualization Approach to Cluster-Based Winograd's Variant of Strassen's Method Using RMI," *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*, Ghaziabad, 2016, pp. 82-88.
[4]	H. Kaur, S. Bagga and A. Arora, "RMI approach to cluster based Winograd's variant of Strassen's method," *MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on*, Amritsar, 2015, pp. 156-162.
[5]	Bagga, Sachin, Deepak Garg, and Ankit Arora. "Moldable load scheduling using demand adjustable policies." *Advances in Computing, Communications and Informatics (ICACCI, 2014 International Conference on*. IEEE, 2014.
[6]	Kaur, Manmeet, Akshay Girdhar, and Sachin Bagga. "Cluster Based Approach to Cache Oblivious Average Filter Using RMI." *The International Symposium on Intelligent Systems Technologies and Applications*. Springer International Publishing, 2016.
[7]	V. K. D. a. H. B. Prajapati, "Dissection of the internal workings of the RMI,possible enhancements and implementing authentication in standard Java RMI," *Int. J. Networking and Virtual Organisations*, pp. 514-534, 2010.
[8]	C. Z. Michael Bader, "Cache Oblivious matrix multplication using an element ordering based on a Peono curve," *Elsevier*, pp. 1-13, 2006.