# Analysis of Segmentation Parameters Effect towards Parallel Processing Time on Fuzzy C Means Algorithm

Cepi Ramdani[1], Indah Soesanti[2], Sunu Wibirama[3]

***Abstract* - Fuzzy C Means algorithm or FCM is one of many clustering algorithms that has better accuracy to solve problems related to segmentation. Its application is almost in every aspects of life and many disciplines of science. However, this algorithm has some shortcomings, one of them is the large amount of processing time consumption. This research conducted mainly to do an analysis about the effect of segmentation parameters towards processing time in sequential and parallel. The other goal is to reduce the processing time of segmentation process using parallel approach. Parallel processing applied on Nvidia GeForce GT540M GPU using CUDA v8.0 framework. The experiment conducted on natural *RGB* color image sized 256x256 and 512x512. The settings of segmentation parameter values were done as follows, weight in range (2-3), number of iteration (50-150), number of cluster (2-8), and error tolerance or epsilon (0.1 – 1e-06). The results obtained by this research as follows, parallel processing time is faster 4.5 times than sequential time with similarity level of image segmentations generated both of processing types is 100%. The influence of segmentation parameter values towards processing times in sequential and parallel can be concluded as follows, the greater value of weight parameter then the sequential processing time becomes short, however it has no effects on parallel processing time. For iteration and cluster parameters, the greater their values will make processing time consuming in sequential and parallel become large. Meanwhile the epsilon parameter has no effect or has an unpredictable tendency on both of processing time.**

***Keywords*--FCM, Processing Time. Segmentation Parameters, Parallel Processing, Fuzzy C Means.**

## I. Introduction

Image segmentation have many variety of methods, one of them is clustering. Clustering is a method of grouping data into several groups or clusters which is the data in one cluster has the degree of maximum similarity but the data between clusters has a minimum degree of similarity and there is no intersection data between clusters [1]. One of the most effective and popular clustering algorithm is Fuzzy C Means, FCM [2]--[4].

FCM have some shortcomings, one of them is the large amount of time consumption [5]--[7]. Many studies have done to reduce the processing time of FCM algorithm. The direction of the studies divided into two categories [3]. The first direction, some researchers doing improvement to the algorithm itself, making changes on algorithm by doing editing, addition, or subtraction. The second direction, moving the heavy computation tasks to the suitable machine to handle the tasks such as GPUs.

Some previous studies on doing reduction time processing by moving the heavy computation tasks on GPU or in other word, using parallel approach were not optimal [3],[6],[8],[9]. There are many optimization parallelize aspects that can be applied to make parallelization process become more effective and optimal.

The other things that can be taken into consideration on doing time reduction on FCM algorithm is knowing the appropriate values of segmentation parameters. Segmentation parameter values were not only important for segmentation accuracy but also for processing time. Knowing the relation between segmentation parameter values towards processing time, particularly in parallel processing time, will provide plenty of positive contributions on doing time reduction.

This research was conducted with two objective point, first to reduce the processing time of FCM algorithm segmentation process using parallel approach applied optimization parallelize aspects optimally also adopted double precision floating-point format number [10]. The second objective is to find the relation between segmentation parameters values towards processing time in sequential and parallel by evaluating the pair of graph of each parameters towards processing time.

## II. Related Work

In this section presented previous studies conducted by other researchers on the same field. The differences are shown on Table I.

## III. Literature Review

### A. Image Segmentation

The definition of image segmentation, as explained by some researchers can be derived as a process of partitioning an image or pixel data of image into several groups based on interest characteristics for example image intensity, color, or texture and there is no overlapping data between groups [2], [11], [12].

Image segmentation usually used to separate the interest of regions from other regions on an image. So, the analysis to the image becomes easy.

### B. Sequential Fuzzy C Means

Fuzzy C-Means (FCM) is a data clustering techniques that allowed a data become a member of more than one clusters at

[1]*Student, Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Jl. Grafika 2 Yogyakarta 55281 INDONESIA (tlp: 0274-555 225; e-mail: cepiramdani.ti14@mail.ugm.ac.id)*

[2,3]*Lecturer, Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Jl. Grafika 2 Yogyakarta 55281 INDONESIA (tlp: 0274-555 225; e-mail: indahsoesanti@ugm.ac.id, sunu@ugm.ac.id)*

once, but actually it is belongs to only one cluster. The belonging of data to the cluster is determined by membership degree. There are three main calculation processes in FCM, objective function calculation, membership function calculation, and cluster center calculation. Flowchart of sequential FCM is shown on Fig. 1.

TABLE I
RELATED WORK

| Researcher | Algorithms | Object | Parallelize Optimization Aspects |
|---|---|---|---|
| M. Al-Ayyoub et al [3] | FCM, brFCM | Lung CT images | Memory management, tasks division CPU-GPU based on profiling apps |
| N. Ali et al [13] | Bias Correction Fuzzy C-Means | Brain medical MRI size 256x256, 512x512, 1,024x1,024, 2,816x2,816 | Memory management, minimize data exchange between CPU-GPU memories, thread count, summation process using Cublas function. |
| H.Li et al [6] | FCM | Natural color image | Memory management, thread count, parallel reduction |
| N.Aitali et al [14] | Spatial FCM | Natural color image, Lena. | Minimize data exchange between CPU-GPU memories |
| M.Shehab et al [7] | FCM, T2FCM | Brain MRI | Summation process conducted in CPU. |
| M. Almazrooie et al [15] | FCM | Brain MRI images | Data parallelism, management memory, minimize bank conflict, sequential access memory (coalesced), minimize data exchange between CPU-GPU memories, and parallel reduction |
| This research | FCM | Natural color image, Lena, size 256 x 256 and 512 x 512 | Data parallelism, memory management, thread count, sequential access memory (coalesced), minimize bank conflict, minimize data exchange between CPU-GPU, minimize thread divergence, avoid race condition, parallel reduction, atomic operation, and double precision floating point number format. |

*1) Objective Function Calculation [15], [16]:*

$$J_{fo} = \sum_{k=1}^{c}\sum_{y=1}^{n} u_{ky}^{m} \left| d_y - v_k \right|^2 \qquad (1)$$

where:
$J_{fo}$ = Objective function
$u_{ky}$ = Membership values of data y at kluster k
$d_y$ = Data value at y from data dimension
$v_k$ = Cluster center at k from data dimension
n = Amount of data
c = Amount of kluster
m = Power or Weight

*2) Membership Function Calculation [15], [16]:*

$$u_{ky}^{m} = \frac{1}{\sum_{r=1}^{c} \left( \frac{\left| d_y - v_k \right|}{\left| d_y - v_r \right|} \right)^{\frac{2}{m-1}}} \qquad (2)$$

where:
$u_{ky}$ = Membership values of data y at kluster k
$d_y$ = Data value at y from data dimension
$v_k$ = Cluster center at k from data dimension
$v_r$ = Cluster center at r from data dimension
c = Amount of kluster
m = Power or Weight

*3) Cluster Center Calculation [15], [16]:*

$$V_k = \frac{\sum_{y=1}^{n} U_{ky}^{m} \bullet d_y}{\sum_{y=1}^{n} U_{ky}^{m}} \qquad (3)$$

where:
$v_k$ = Cluster center at k from data dimension
$u_{ky}$ = Membership values of data y at kluster k
$d_y$ = Data value at y from data dimension
m = Power or Weight
n = Amount of data

## IV. PARALLEL IMPLEMENTATION

Parallel processing interpreted as processing many data simultaneously in one time by many processors, followed by communication between processor and comprehensive control or coordination. There are some parallelize optimization aspects applied on this research [17]--[22], which are data parallelism, thread count, branch divergence or thread divergence, memory management, minimize synchronization, coalesced accessing memory, reduce bank conflict, reduce the data exchange between CPU and GPU memory, calculating the amount of threads and thread blocks, and avoid race condition.

Parallel implementation used in this research is hybrid. There is a division of tasks between CPU and GPU. The GPU handles heavy computing processes with large amount of data. All processes handled by GPU i.e. cluster center computation, objective function computation and distribute the membership values to all members.

At objective function computation there is one process that handled by CPU, it is final total computation. CPU handles it because the amount of data is too small. Reduction parallel technique and atomic operation are applied on cluster center and objective function computation process. There are summation operations with a lot of amount of data.

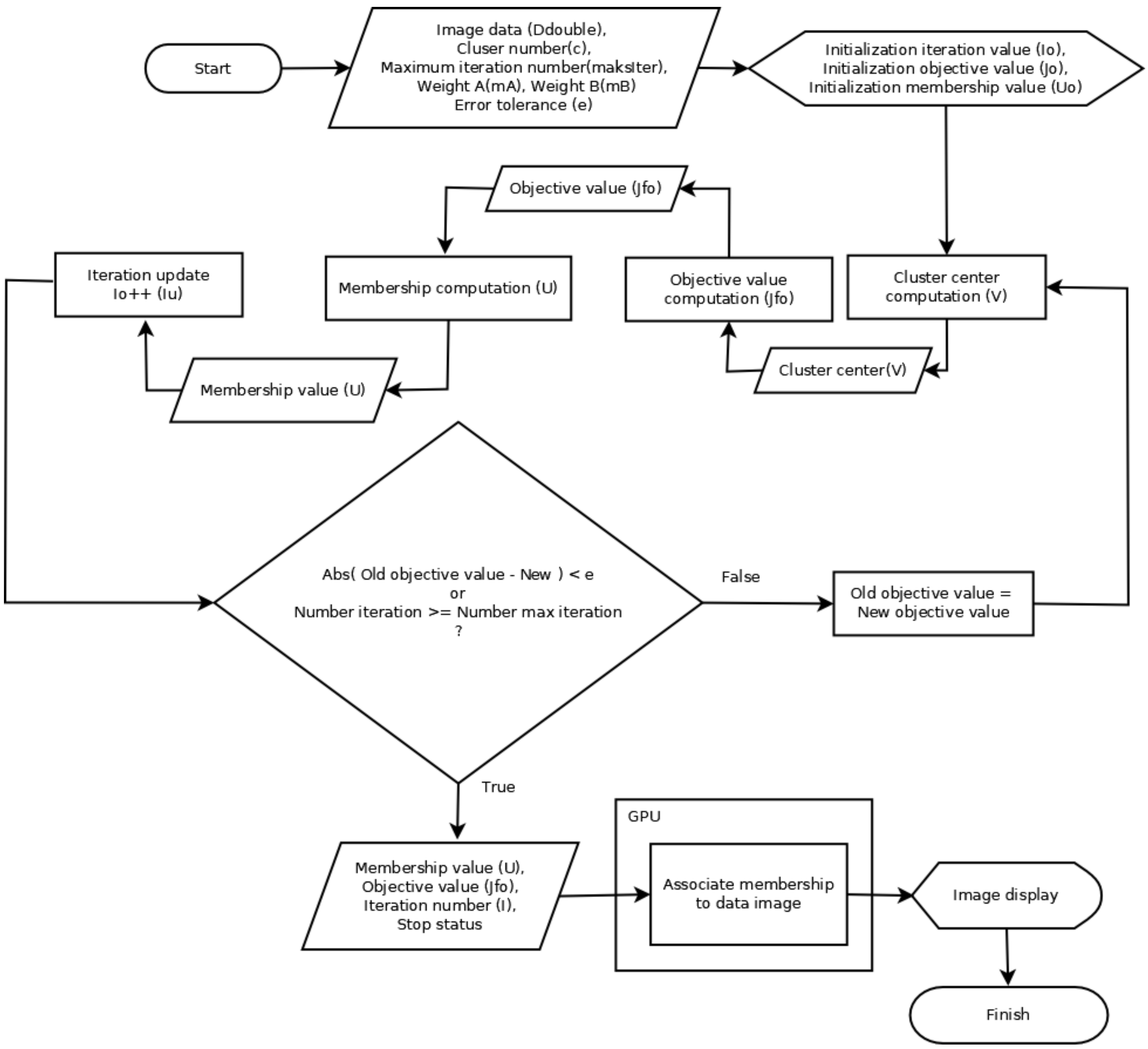



Fig. 1 Flowchart of sequential fuzzy C-Means.

The computation stages of cluster center and objective function have no difference in parallel and sequential. Nevertheless, there is a difference in membership function computation. In parallel version, there is an addition computation stage at the end, to compute square new membership value with weight. The flowchart of parallel fuzzy C-Means is shown on Fig. 2.

## V. Experiment And Analysis

### A. Experiment

The experiment stage is done to get the test data that will be analyzed to answer the hypothesis. The objects of testing are natural RGB color images, Lena, sized 256 x 256 and 512 x 512.

The conditions applied on both algorithms were equal. Segmentation parameter values are set up under certain condition, the set up shown on Table II.

The total amount of experiment data result is 1,512. Segmentation parameter values combination can be seen at Fig. 3.

TABLE II
SEGMENTATION PARAMETER VALUES SETTING

| Parameters | Start Values | Last Values | Increase Values | Total |
|---|---|---|---|---|
| Cluster | 2 | 8 | +1 | 7 |
| Weight | 2 | 3 | +0,5 | 3 |
| Iteration | 50 | 150 | +50 | 3 |
| Error tolerance | 0.1 | 0.1 x $10^{-5}$ | x0.1 | 6 |
| Amount of combination | | | | 378 |

### B. Processing Time Analysis

The analysis conducted to find out the difference between sequential and parallel processing time. It was done by compare sequential and paralel time consuming from experiment data, using (4). The result is shown on Table III. Parallel processing time is faster 4.5 times than sequential process.

$$H_b = \frac{\sum_{i=1}^{n} (t_{serial\ n} \div t_{paralel\ n})}{n} \quad (4)$$

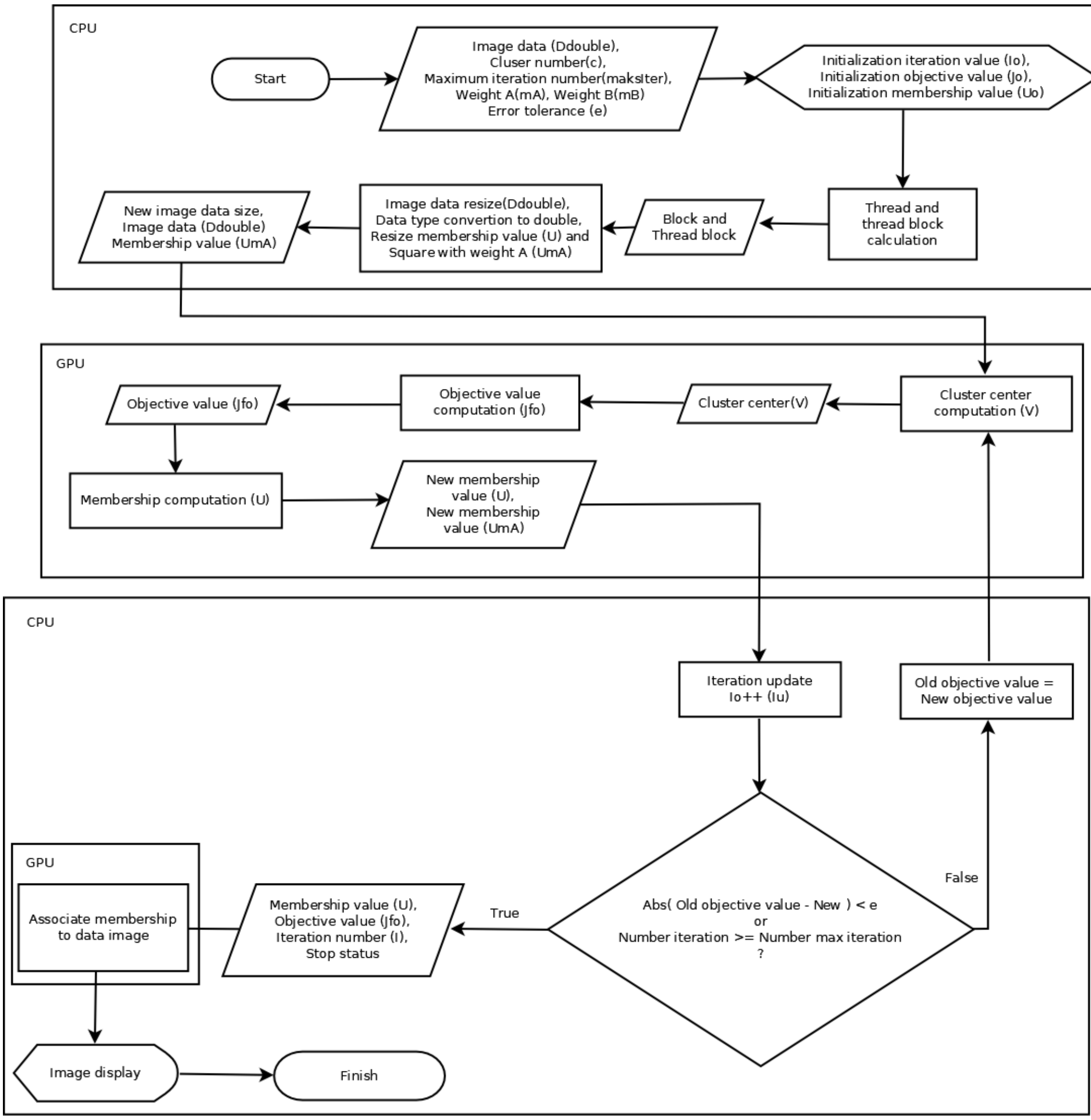



Fig. 2 Flowchart of Parallel Fuzzy C Means.

where:

$H_b$ = Comparing result
$t_{serial}$ = Sequential time
$t_{paralel}$ = Parallel time
n = Amount of data

TABLE III
SEQUENTIAL AND PARALLEL PROCESSING TIME

| Image | Means of Sequential | Means of Parallel | Comparison | Means |
|---|---|---|---|---|
| 256 | 25,971.4735 | 5,735.77772 | 4.52797768 | 4.5234256 |
| 512 | 104,931.704 | 23,220.7651 | 4.51887368 | |

The graph of comparison segmentation processing time displayed on Fig. 4 and Fig. 5. The blue line is sequential algorithm and red line is parallel algorithm, meanwhile Y-axis meaning time of process and X-axis is number of data, the amount is 378 each.

The position of the red line or parallel processing time graph is always under the blue line from the first data to the next data until the end of data on both pictures. We can take a conclusion, parallel processing faster than sequential processing.

## C. Image Segmentation Similarity Level Analysis

Similarity analysis conducted in two ways, first using cosine similarity algorithm to compute the exact value of similarity level [23]--[25] and second comparing the image histograms. Cosine similarity, Equation (5) or (6) computes two values that are similarity value and angle value. The similarity value generated by this computation has value in range between 0 to 1 and angle values from 0 to 90 degree. If the similarity value is 0 which means the angle value is 90, then both images are very different, the similarity is 0%.

TABLE IV
COMPARISON IMAGE HISTOGRAM 256 X 256 PIXEL

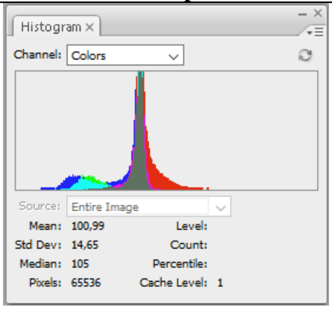
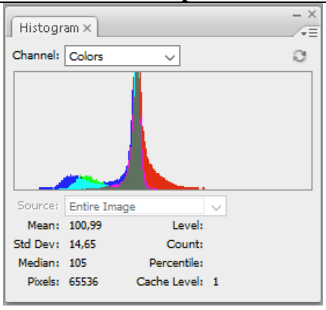
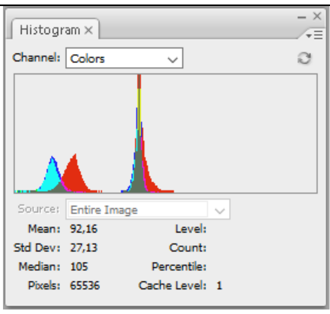
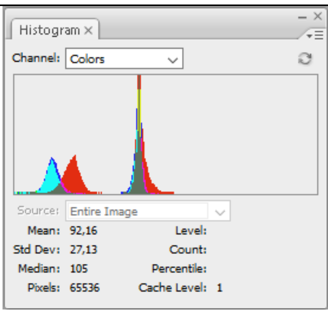
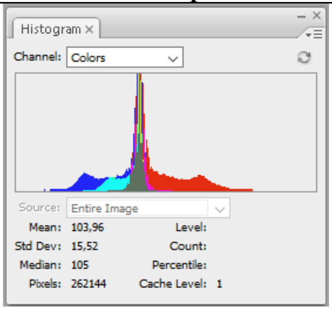
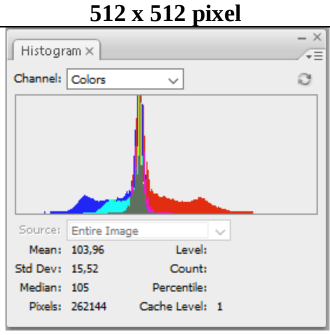
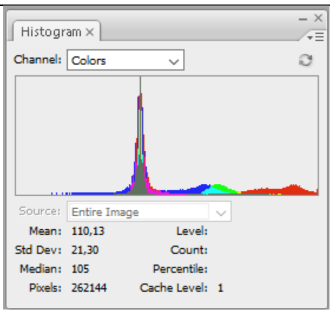
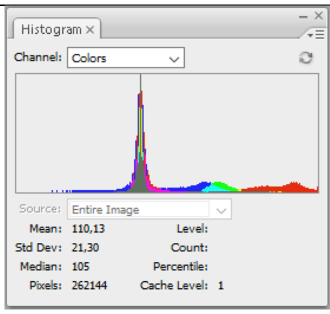
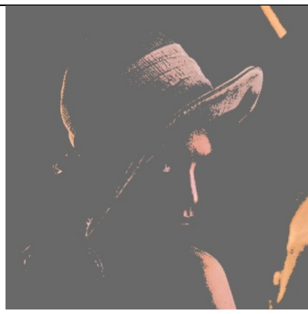
| **Sequential Segmented Images 256 x 256 pixel** | **Sequential Image Histogram 256x256 pixel** | **Parallel Image Histogram 256x256 pixel** | **Parallel Segmented Images 256 x 256 pixel** |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |

TABLE V
COMPARISON IMAGE HISTOGRAM 512 X 512 PIXEL

| **Sequential Segmented Images 512 x 512 pixel** | **Sequential Image Histogram 512 x 512 pixel** | **Parallel Image Histogram 512 x 512 pixel** | **Parallel Segmented Images 512 x 512 pixel** |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |

Generated by the computation of cosine similarity equation, the similarity level of each pair of segmented image is 1 with angle value is 0 degree. Meanings the similarity is 100%. All segmented images are equal with each pair.

For image histogram comparison, can be seen on the Table IV and Table V. Each table contains the segmented image and their histogram. The table has four columns, the first column contains segmented image generated by sequential algorithm, the second column contains image histogram of its segmented image, the third contains image histogram of segmented image generated by parallel algorithm, and the last column contains its segmented image.

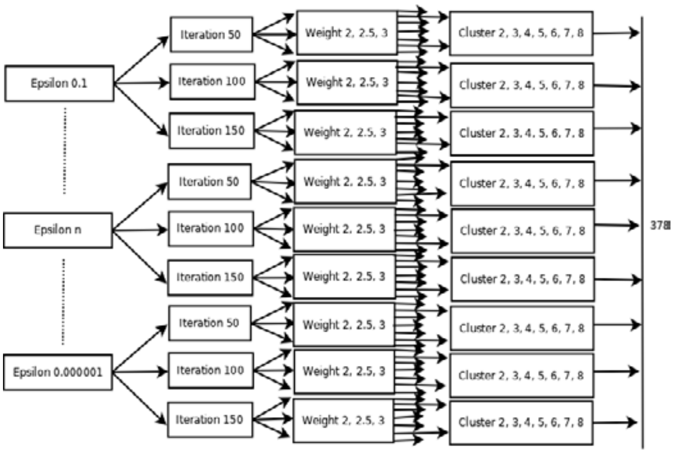



Fig. 3 Parameter values combination.

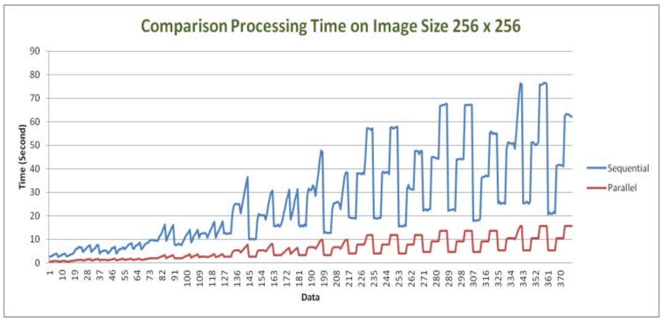



Fig. 4 Comparison processing time on image 256 x 256.

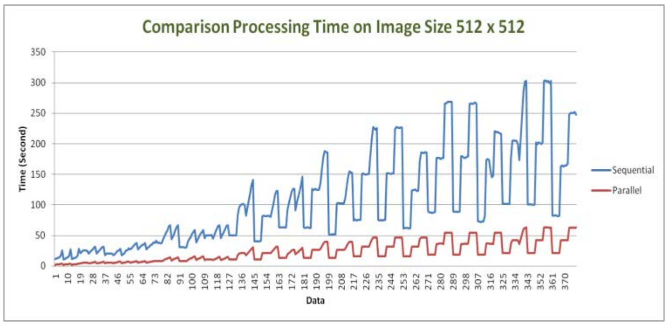



Fig. 5 Comparison processing time on Image 512 x 512.

From Table IV and Table V, it can be seen each pair segmented image has the same histogram, in value such as mean, standard deviation, median, pixel, and even in the shape. The conclusion is the segmented images of both processes are equal.

$$CosSM = \frac{A \bullet B}{\|A\| \times \|B\|} \quad (5)$$

Implementation,

$$CosSM = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (6)$$

where:

A = Matrix A
B = Matrix B
A • B = *Dot product* matrix A and matrix B
||A|| = *norm* A
||B|| = *norm* B
||A|| x||B|| = *Cross product* of ||A|| and ||B||
n = amount of data
i = index number

### *D. Correlation of Weight Parameter and Processing Time*

The weight parameter has three values that are 2, 2.5, and 3. The analysis is done by evaluating the change of processing time graph toward the change of weight parameter value. Weight parameter act as an independent variable, its value is set to vary whereas other parameter which is act as dependent variables, their values are constant.

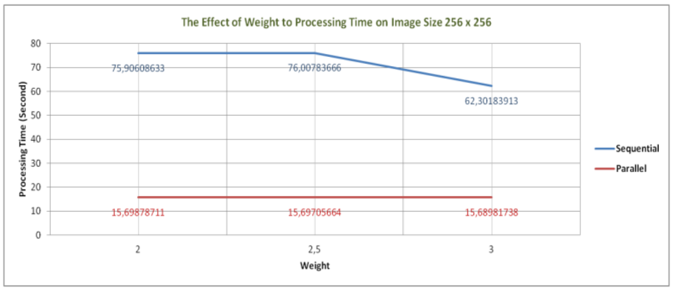



Fig. 6 The effect of weight to processing time on image size 256x256.

Fig. 6 and Fig. 7 show the effect of weight to the processing time at image sized 256x256 and 512x512. The blue lines show sequential and the red lines show parallel processing time. X-axis show the weight value and Y-axis show the time in second. The settings of dependent parameter values on graphs as follows, cluster parameter in 8, iteration in 150, and error in 0.1 x $10^{-5}$.


Fig. 7 The effect of weight to processing time on image size 512x512.

From both pictures can be seen the effect of weight. On sequential processing time, the greater weight values the time become short. However, it has no effect on parallel processing time. The red graph looks straight and the values almost constant.

### *E. Correlation of Error Tolerance Parameter and Processing Time*

The error tolerance or epsilon parameter has six values that are 0.1, 0.1x$10^{-1}$, 0.1x$10^{-2}$, 0.1x$10^{-3}$, 0.1x$10^{-4}$, and 0.1x$10^{-5}$. Here, epsilon parameter act as an independent variable, where its value is set to vary whereas other parameter, which is act as dependent variables, their values are set in constant.

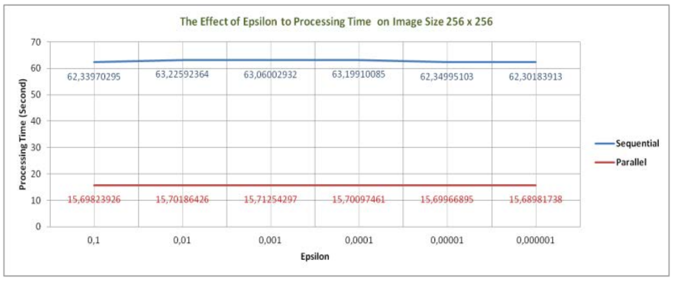



Fig. 8 The effect of epsilon to processing time on image size 256x256.

Fig. 8 and Fig. 9 show the effect of epsilon to the processing time at image sized 256x256 and 512x512. The blue lines show sequential and the red lines show parallel processing time. X-axis show the epsilon value and Y-axis show the time in second. The settings of dependent parameter values on graphs as follows, weight parameter in 3, cluster in 8, and iteration in 150.

The shape of all lines, blue and red, from both pictures visually looks straight but there are up and down in the middle. It can be concluded that epsilon has no significant effect on processing time in sequential or parallel. The tendency is unpredictable.

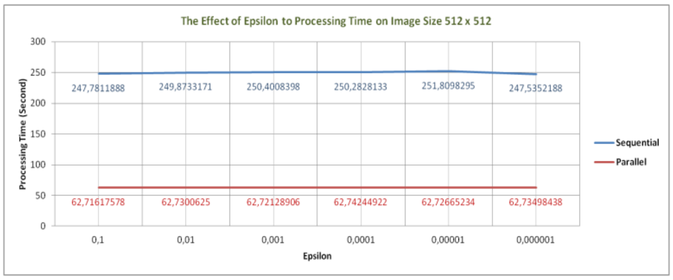



Fig. 9 The effect of epsilon to processing time on image size 512x512.

## F. Correlation of Iteration Parameter and Processing Time

The iteration parameter act as an independent variable, its value is set to vary and the other parameters, which act as dependent variables, their values are set in constant. The values of iteration parameter are 50, 100, and 150.

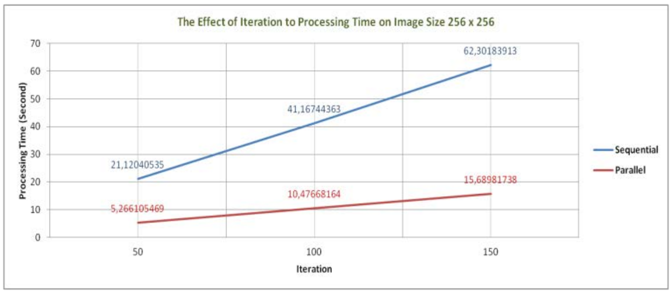



Fig. 10 The effect of iteration to processing time on image size 256x256.

Fig. 10 and Fig. 11 show the effect of iteration parameter values to the processing time at image sized 256x256 and 512x512. The blue lines show sequential and the red lines show parallel processing time. X-axis show the iteration value and Y-axis show the time in second. The settings of dependent parameter values on graphs as follows, weight parameter in 3, cluster in 8, and error tolerance in $0.1x10^{-5}$.

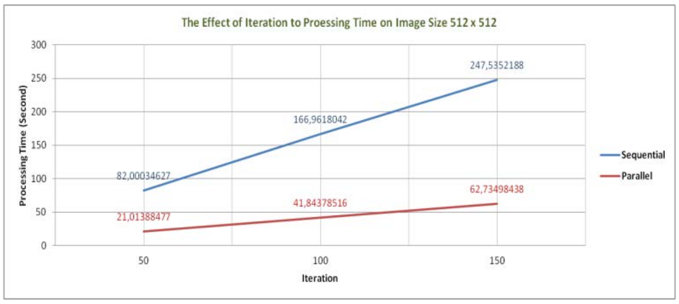



Fig. 11 The effect of iteration to processing time on image size 512x512.

The conclusion concluded by evaluating both of images and lines, blue and red lines, the greater value of iteration parameter will make the processing time consumption become large. It is occurring to all processing time, in sequential and parallel, but with different influence value. The change on sequential process is more significant than parallel process.

## G. Correlation Cluster Parameter and Processing Time

Fig. 12 and Fig. 13 show the effect of cluster parameter values to the processing time at image sized 256x256 and 512x512. The blue lines show sequential and the red lines show parallel processing time. X-axis show the cluster value and Y-axis show the time in second. The values of cluster parameter are 2, 4, 5, 6, 7, and 8. Meanwhile the settings of dependent parameter values as follows, weight parameter in 2.5, iteration in 50, and error tolerance in $0.1x10^{-5}$.

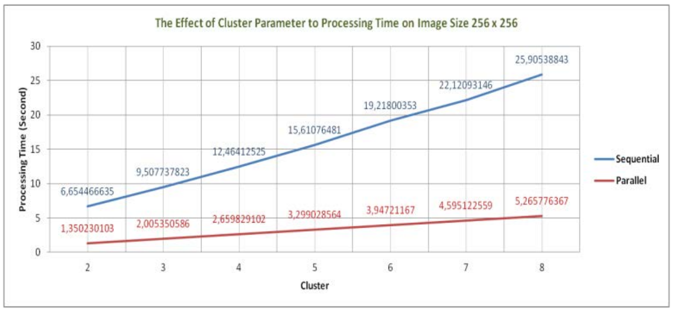



Fig. 12 The effect of cluster to processing time on image size 256x256.

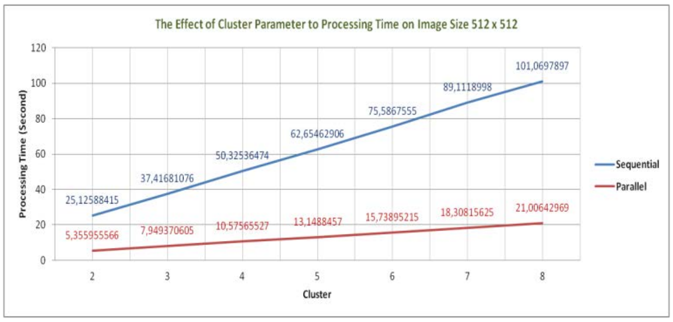



Fig. 13 The effect of cluster to processing time on image size 512x512.

From both pictures can be seen the effect of cluster parameter towards processing time. The greater value of cluster will make the processing time consumption become large. It is occurring to all processing time, in sequential and parallel, but with different influence value. The change on sequential process is more significant than parallel process.

## VI. Conclusion

The series of research stages that have been conducted give some conclusion as follows. Processing time reduction on parallel approach in GPU by following parallelize optimization aspect gives the expected results, which is parallel processing time shorter 4.5 time than sequential process. The similarity level value of segmented image computed by cosine similarity algorithm is 100%, which means all pair of segmented image generated by all processes is equal.

The effect of segmentation parameter values towards processing time in sequential and parallel concluded as follows,

- Weight parameter. The greater its value will make the sequential processing time become short. However, it has no effect on parallel processing time.
- Error tolerance parameter. It value has no effect on both processing time. The tendency is unpredictable.
- Iteration parameter. The greater their values then the processing time get longer.
- Cluster parameter. Equal with the effect iteration parameter. The greater its value then the processing time get longer.

The amount of influence value that occurs on both processes is different. The changes on sequential processing more significant than parallel processing.

## Acknowledgment

The authors are thankful to the management of Department of Electrical Engineering and Information Technology Universitas Gadjah Mada for providing facilities and many supports.

## References

[1] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Boston: Pearson Addison-Wesley, 2006.

[2] H. Shaaban, A. Habib, F. Obaid, "Performance Evaluation of K-Mean and Fuzzy C-Mean Image Segmentation Based Clustering Classifier", *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 12, pp.176-183, 2015.

[3] M. Al-Ayyoub, A. M. Abu-Dalo, Y. Jararweh, M. Jarrah, and M. Al Sa'D, "A GPU-based Implementations of the Fuzzy C-Means Algorithms for Medical Image Segmentation," *Journal of Supercomputing*, Vol. 71, No. 8, pp. 3149–3162, 2015.

[4] W. Cai, S. Chen, and D. Zhang, "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation," *Pattern Recognition*, Vol. 40, Issue 3, pp. 825–838, 2007.

[5] M. A. Balafar, "Fuzzy C-mean Based Brain MRI Segmentation Algorithms," *Artificial Intelligence Review*, Vol. 41, No. 3, pp. 441–449, 2014.

[6] H. Li, Z. Yang, and H. He, "An Improved Image Segmentation Algorithm Based on GPU Parallel Computing," *Journal of Software*, Vol. 9, No. 8, pp. 1985–1990, 2014.

[7] M. A. Shehab, M. Al-Ayyoub, and Y. Jararweh, "Improving FCM and T2FCM Algorithms Performance Using GPUs for Medical Images Segmentation," *6th International Conference on Information and Communication Systems, ICICS 2015*, 2015, pp. 130–135.

[8] M. A. Alsmirat, Y. Jararweh, M. Al-Ayyoub, M. A. Shehab, and B. B. Gupta, "Accelerating Compute Intensive Medical Imaging Segmentation Algorithms Using Hybrid CPU-GPU Implementations," *Multimedia Tools and Applications*, Vol. 76, No. 3, pp. 3537–3555, 2017.

[9] M. Al-Ayyoub, M. A. Shehab, Q. Yaseen, and Y. Jararweh, "Accelerating Clustering Algorithms Using GPUs," *Conference: 2016 IEEE High Performance Extreme Computing Conference (HPEC-2016)*, 2016, pp. 1–2.

[10] N. Whitehead and A. F. Florea. (2011) Floating Point and IEEE-754 Compliance for NVIDIA GPUs, [Online], http://docs.nvidia.com/cuda/floating-point/index.html, access date: 14 November 2017

[11] L. Lalaoui, T. Mohamadi, and A. Djaalab, "New Method for Image Segmentation," *Procedia - Social and Behavioral Sciences*, Vol. 195, pp. 1971–1980, 2015.

[12] M. S. H. Al-Tamimi and G. Sulong, "Tumor Brain Detection Through MR Images: A Review of Literature," *Journal of Theoretical and Applied Information Technology*., Vol. 62, No. 2, pp. 387–403, 2014.

[13] N. A. Ali, B. Cherradi, A. El Abbassi, O. Bouattane, and M. Youssfi, "New Parallel Hybrid Implementation of Bias Correction Fuzzy C-Means Algorithm," *Proceedings - 3rd International Conference on Advanced Technologies for Signal and Image Processing, ATSIP 2017*, Vol. 1, pp. 1–6, 2017.

[14] N. Aitali, B. Cherradi, and A. El Abbassi, "GPU Based Implementation of Spatial Fuzzy C-Means Algorithm for Image Segmentation," *Information Science and Technology, 2016 4th IEEE International Colloquium*, 2016, pp. 460–464.

[15] M. Almazrooie, M. Vadiveloo, and R. Abdullah, "GPU-Based Fuzzy C-Means Clustering Algorithm for Image Segmentation," *Arxiv preprint arXiv:1201.2050*, 2016, pp. 1–13.

[16] K.Chuang, H.Tzeng, S.Chen, J.Wu, T.Chen, "Fuzzy C-Means Clustering with Spatial Information for Image Segmentation," *Computerized Medical Imaging and Graphics*, Vol. 30, pp. 9-15, 2006.

[17] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth, "Medical Image Segmentation on GPUs - A Comprehensive Review," *Medical Image Analysis*, Vol. 20, No. 1, pp. 1–18, 2015.

[18] H. L. Khor, S. C. Liew, and J. M. Zain, "A Review on Parallel Medical Image Processing on GPU," *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, Vol. 17, No. 8, pp. 45–48, 2015.

[19] P. Afshani and N. Sitchinava, "Sorting and Permuting without Bank Conflicts on GPUs," *23rd Annual European Symposium*, 2015, Vol. 9294, pp. 13–24.

[20] X. Mei and X. Chu, "Dissecting GPU Memory Hierarchy Through Micro benchmarking", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 1, pp. 72–86, 2017.

[21] Will Landau, (2013) CUDA C: race conditions, atomics, locks, mutex, and warps, [Online], https://wlandau.github.io/gpu/lectures/cudac-atomics/cudac-atomics.pdf, Retrieved 28 June 2016.

[22] James Balfour, (2011) CUDA Threads and Atomics, [Online], https://mc.stanford.edu/cgi-bin/images/3/34/Darve_cme343_cuda_3.pdf, Retrieved 14 November 2017.

[23] R. T. Wahyuni, D. Prastiyanto, and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," *Jurnal Teknik Elektro Universitas Negeri Semarang*, Vol. 9, No. 1, pp. 18–23, 2017.

[24] R. V. Imbar, Adelia, M. Ayub, and A. Rehatta, "Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks," *Jurnal Informatika*, Vol. 10, No. 1, pp. 31–42, 2014.

[25] Sugiyamto, B. Surarso, and A. Sugiharto, "Analisa Performa Metode Cosine dan Jacard pada Pengujian Kesamaan Dokumen," *Jurnal Masyarakat Informatika*, Vol. 5, No. 10, pp. 1–8, 2014.