THE DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI) TO RECONSTRUCT SURFACE OF A HUMAN HEAD USING BETA-SPLINE

[1]MUHAMMAD SYAWAL ABD HALIM, [2]NORMI ABDUL HADI, [3]HANIFAH SULAIMAN, [4]SUHAILA ABD HALIM

[1]Facultyof Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch, Tapah Campus, 35400 Tapah Road, Perak, Malaysia.
[2,3,4]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.
Email: - [1]syawal434@perak.edu.my, [2]normi@fskm.uitm.edu.my, [3]hanifah@fskm.uitm.edu.my, [4]suhaila@tmsk.uitm.edu.my

Abstract - This study focuses on the development of Graphical User Interface (GUI) using MATLAB to reconstruct surface from multi slice CT Scan Images of human head using Beta-spline technique. Since large data were involved in this computation, parallel processing method was introduced in this study. The implementation of computation was applied using "ARS" Cluster, a low-cost parallel laboratory in UiTM Shah Alam. MATLAB "pmode" platform with MATLAB Distributed Computing Server (MDCS) was integrated in the cluster. The developed GUI displaysthe result of the surface of a human head implementation with its computation time. The time taken for the parallel processing method produces minimal computation time as compared to sequential processing.

Keywords - ARS Cluster, Beta-spline, CT Scan, MATLAB, Parallel, Reconstruction.

I. INTRODUCTION

The reconstruction of an image is important since its data structure can be stored and used for future purposes. To view the object better, three-dimensional (3D) reconstruction can be used. This help researcher to study what is inside of the object without breaking it. There are already many reconstruction techniques that have been developed such as image morphing [1] and Beta-spline [2] which can be used by researcher that depends on the suitability of the image or object.

3D scanner, Computerized Tomography (CT) scan or Magnetic Resonance Imaging (MRI) can be used to obtain 3D data points for its reconstruction. The 3D scanner produces unstructured 3D points that are hard to arrange because the data points are scattered. On the other hand, CT-Scan/MRI produce structured 3D points that are easy to handle since the data consists of multi slice images with different height. The first category is images taken from different angles of the object such as axial, sagittal and coronal; the second category is staked images taken using CT and MRI scanners [3]. Since there are many reconstruction techniques, the performance of the computation process has been evaluated and compared in term of processing time and accurateness.

When involving the multi slice of images, the data points from the image must be extracted first. Therefore, large data sets will be obtain from the extraction process. Processing a large datasets using a normal computer would be time consuming and could lead to system crash. This is the reason why parallel computation is introduced in this study in order to handle the operations on large data. Parallel computations can make the data being divided into smaller ones, which are then been solved concurrently. Beta-spline is a family of B-spline curves that offers high degree of continuity [4]. It employs cubic beta spline as the fitted curve based on its capability to maintain the second degree of continuity and it has shape parameters which can be controlled locally. Beta spline gives smooth images and preserves important features of the surface.
Beta-spline technique used large number of curve segments to reconstruct a surface. However, this problem can be solved by using parallel computation to process number of curve segments to different processors. The data are processed using several processors that are distributed among them in parallel computation environment.

In this study, reconstruction of a human head from multi slices CT scan images is processed using "ARS" cluster. The cluster is a low cost parallel laboratory that located at Faculty of Computer and Mathematical Sciences, UiTM Malaysia. The CPU of the "ARS" cluster is Intel Pentium IV processor with 1.8 GHz incorporate with open-source software that is LINUX Fedora 7 with MDCS [5].

This study started with data point's acquisition from the multi slice CT scan images. This collection of data is important before continue to the next step. After that, the data points in term of control points for surface reconstruction by using Beta-spline surface fitting are used in the MATLAB "pmode" platform. Graphical User Interface (GUI) is then developed to display the relevant results. The overall methodology in this study is elaborated in Section 2 to 4 in detail. Section 5 presents the results and discussion. Lastly Section 6 shows the conclusion regarding this study.

II. METHODOLOGY

2.1. Data Acquisition

In data acquisition, 50 layers of a human head from CT scan images acquire from [2]. There are 64 points are extracted as a control points from each layer. The points are usedinbeta spline. The process of extracting boundary data points from CT scan image is illustrated in Figure 1 below.

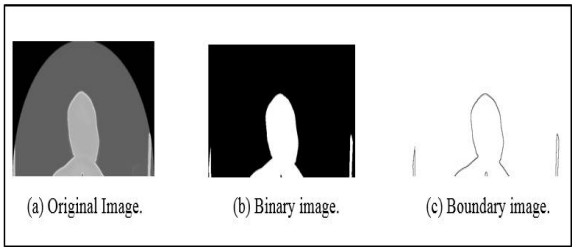


Fig. 1. Process of Extracting Boundary Data Points.

Figure 1 (a) shows the original CT scan image at one layer. The image is then converted tobinary form which contains only two intensity values, either black or white as shown in Figure 2 (b). The conversion consists of a threshold value T set and $L(x, y)$ is the original intensity pixel with (x, y) coordinates. The intensity after conversion is mathematically expressed as,

$$g(x,y) = \begin{cases} 1 & for \;\; L(x,y) \geq T \\ 0 & for \;\; L(x,y) < T \end{cases} \qquad (1)$$

Figure 1 (c) shows the boundary of the image that are extracted by tracing among the white pixels later. From the boundary image, 64 data points were extracted. Then, these steps were repeated for all the 50 otherlayers of the human head.

2.2. Beta-spline Surface Fitting

There are 64 data points provided for each image slice that represent the boundary for each image. The data points are used in this phase where Beta-spline surface fitting technique is taking place.

The Beta-spline surface equation,

$$F(t,u) = [T][M][V][M]^T[U] \qquad (2)$$

where $V_{i,j}$ is a set of 4×4 control points matrix,

$$[V] = \begin{bmatrix} V_{i,j} & V_{i,j+1} & V_{i,j+2} & V_{i,j+3} \\ V_{i+1,j} & V_{i+1,j+1} & V_{i+1,j+2} & V_{i+1,j+3} \\ V_{i+2,j} & V_{i+2,j+1} & V_{i+2,j+2} & V_{i+2,j+3} \\ V_{i+3,j} & V_{i+3,j+1} & V_{i+3,j+2} & V_{i+3,j+3} \end{bmatrix} \qquad (3)$$

$[U]$ is the polynomial matrix,

$$[U] = [u^3 \quad u^2 \quad u \quad 1] \qquad (4)$$

$[T]$ is the polynomial matrix,

$$[T] = [t^3 \quad t^2 \quad t \quad 1] \qquad (5)$$

and $[M]$ is the Beta-spline basis function matrix,

$$[M] = \frac{1}{\delta}\begin{bmatrix} -2\beta_1^3 & 2(\beta_2+\beta_1^3-\beta_1^2+\beta_1) & -2(\beta_2+\beta_1^2+\beta_1+1) & 2 \\ 6\beta_1^3 & -3(\beta_2+2\beta_1^3+2\beta_1^2) & 3(\beta_2+2\beta_1^2) & 0 \\ -6\beta_1^3 & 6(\beta_1^3-\beta_1) & 6\beta_1 & 0 \\ 2\beta_1^3 & \beta_2+4(\beta_1^2+\beta_1) & 2 & 0 \end{bmatrix} \qquad (6)$$

The value of bias and tangent parameters has been set to $\beta_1 = 1, \beta_2 = 0$. This condition is called as unbiased because the surface at the basic position. A patch of surface by using control points is illustrated in Figure 2 below.

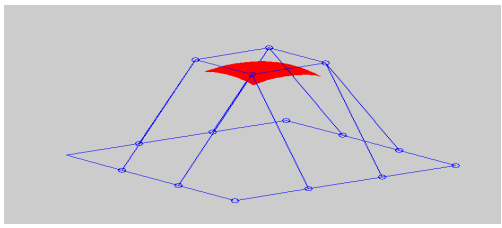


Fig. 2. Surface Generation from 4×4 Control Points.

In order to produce another patch, the last three control points in the previous one are used so that the curve is G^2 continuous. The beta spline curve function is as in equation (7) below

.

$$F_0(t) = V_0 b_0(t) + V_1 b_1(t) + V_2 b_2(t) + V_3 b_3(t)$$
$$F_1(t) = V_1 b_1(t) + V_2 b_2(t) + V_3 b_3(t) + V_4 b_4(t) \qquad (7)$$

The three last control points of $F_0(t)$ are the first three control points of $F_1(t)$ and this continuesuntil.

2.3. Parallel Process

The distribution of data points is easy due to the generation of patches are independent. In order to equalize the work done by the processors, a dyadic segmentation technique is used. The dyadic rational number as stated in [7], D, is a number of the form,

$$D = \frac{a}{2^b} \qquad (8)$$

Where,

a = number of data points

$$b = \log_2 p \qquad (9)$$

where p = number of processor.

2.4. Performance Evaluation

The performance of the algorithm depends on the number of PCS used. The performance is measured and analysed in term of processing time. Then, the

speedup and efficiency of the algorithm are calculated using equations (10) and (11) [8],

$$\text{Speedup, } S(n,p) - \frac{T_\sigma(n)}{T_\pi(n,p)} \quad (10)$$

$$\text{Efficiency, } E(n,p) = \frac{S(n,p)}{p} \quad (11)$$

Where,

p = number of processor(s),

$T_\sigma(n)$ = execution time of the sequential algorithm.

$T_\pi(n,p)$ = execution time of the parallel algorithm with p processors.

III. THE COMPUTATION OF BETA-SPLINE SURFACE FITTING

In computation the beta-spline surface fitting, the first step is to extract the data points from each layer of the CT scan image. Since there are involved 50 slices of human head images, the computation processis done one by one starting from "cut(1)" layer until "cut(50)" layer. The computation process was done using the identification algorithm below.

Identification algorithm

Initialization: Read the CT scan image.
Each image boundary will have 64 data points.
Thus, $Q = 64$.

Step 1: From original image to binary image:
Convert the image pixel by choosing threshold value, $T - 225$ so that the original image will turn into the binary image. Apply the value of threshold into equation (1).

Step 2: Acquiring boundary line:
The boundary between white and black pixel of the binary image will be the boundary line.

Step 3:Finding length and divide them equally:
After the boundary points have been acquired, find the length,

$$L(u) = \int_0^1 \sqrt{x'(u) + y'(u)}\, du \quad (12)$$

Use equation below to make sure that data points have the same length between them.

$$h = \frac{sum\ of\ length}{Q-1} \quad (13)$$

Step 4: Extracting data points:
Find h_i first before substituting into data points, P_i,

$$P_i = [C_i \quad C_{i+1} \quad C_{i+2} \quad C_{i+3}] \quad (14)$$

$$h_i = (i-1) \times h \quad (15)$$

where $i = 1,2,3,\ldots,Q$

For each h_i, and for each curve point j,

If $h_i - CL_j \le 0$, then set j as corresponding point P_j.

After all the 50 layers are completed with the computation process, there involved 3,200 data points. The data are sorted according to their respective height of layers. The first 64 data points in the file are belongs to the first height of the human head and it continues until all the 3,200 data points.

The first patch of a human head uses points $[i \quad i+1 \quad i+2 \quad i+3]$ with the height of $[j \quad j+1 \quad j+2 \quad j+3]$ as the control points. After that, in order to form the second patch, it uses $[i+1 \quad i+2 \quad i+3 \quad i+4]$ and it continues until all patches for each layer are formed. Below is the parallel beta spline algorithm.

Parallel Beta Spline Algorithm

Initialization: Set the value of bias and tension parameters, $\beta_1 = 1$ & $\beta_2 = 0$. Set the value of surface point, N, and the number of processor used for parallel computation.

Step 1: Segmentation of data points:
Use dyadic rational (8) to divide the data points equally depends on the number of processors used.
number of points, Q = 64
n = log(numlabs)/log(2)
dyad = 1/(2^n)

Step 2: Distribution of data to processors:
From master, send the data points that have been divided from step 2, to the available nodes.
processor number 1 do
*process data 1 until Q*dyad*
processor number 2 do
*process data Q*dyad until Q-3*

Step 3: Beta spline computation:
Each worker will perform beta spline computation using equation (2,3,4,5,6) based on the data points they received. Part of human head surface are formed from each node after the computation.

Step 4: Surface of a human head:
Master will collect the results from all node after the computation complete and the result is displayed.

The parallel algorithm started with the initialization of bias and tangent value. After that, the data points that has been extracted from the image is then being segmented by using dyadic technique. This dyadic technique helps in dividing the data and distributed to processors equally. The processors executed the data points and implemented it on beta spline technique. For the sequential and parallel computation, the time elapsed were recorded for analysis purposes. To start recording time in MATLAB, 'tic' is included in the program before the computation start and 'toc' to stop the timer which is been setup after the computation has ended.

IV. THE DEVELOPMENT OF MATLAB GUI

The computation of beta spline was conducted on several available nodes in "ARS" cluster [5]. In this study, 4 nodes were used to implement the computation and each one of them are called "ars1", "ars2", "ars3" and "ars4". The specifications of all nodes are Intel® Pentium® and 2.80 GHz processors using standard Ethernet network. The operating system used is open source LINUX Fedora 7. It also uses MATLAB Distributed Computing Server

(MDCS) as the communication platform for parallel processing. Client pc is used as an interface between user and the cluster workstation. Firstly, all nodes must be turned on. Then, open the terminal window and network window in master client pc. All nodes must be ensured connected to the client pc for the parallel computation. After that, admin centrewas open to add workers from available nodes.

Open MATLAB in client pc must be opened to start the computation. The parallel mode was set in client pc and the validation process is required. The validation process isshowsin Figure 3 below.

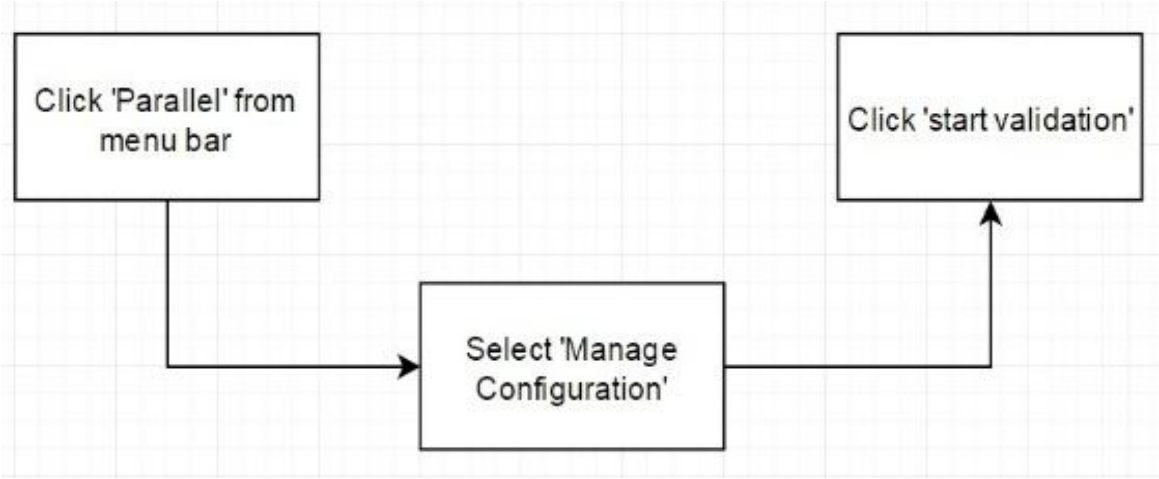


Fig. 3. Flow of Parallel Validation Process.

After validation is successful, MATLAB"pmode" is ready to be used. In the command window, 'pmode start' is used to run the program of beta spline computation.

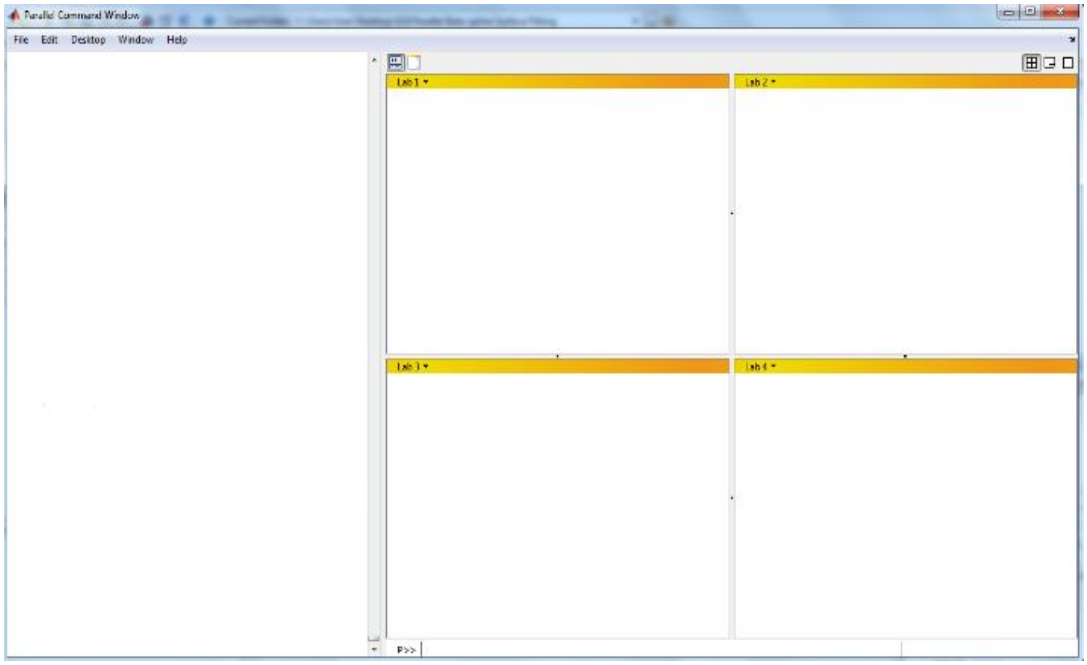


Fig. 4. MATLABParallel Command Window.

From Figure 4 above, each lab represents the processors involved. Since this study focus on the parallel computation of 4 processors, it consists of Lab 1, Lab 2, Lab 3 and Lab 4. The MATLAB coding was inserted at the bottom of the command window with the sign of 'P>>'. Then the coding runs simultaneously for the 4 nodes and the results were stored in each of the node. In order to obtain the results from the nodes and transfer it to the client pc, the following command was used in the same window, pmode lab2client (variable name to transfer)(lab number) (variable name received)

Now, all the patches that that have been calculated simultaneously were stored in the client pc. Then, the plotting process was done in the client pc by using MATLAB GUI. In developing the interface using GUI, 'guide'command was used in the command window. GUIDE Quick Start was appeared as shown in Figure 5 below.

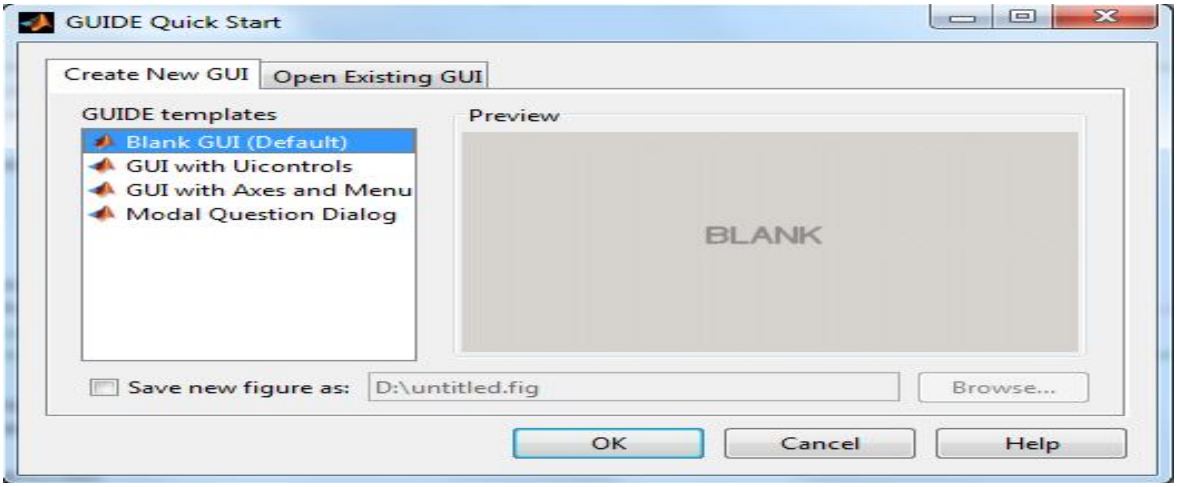


Fig. 5. GUIDE Quick Start.

A blank interface was appeared and ready for interface design process. Since this study is to reconstruct surface and display the time taken of parallel and sequential process, the interface must consists of objects of axes, push button, edit box and text box. Axes object is to display the reconstructed surface of the human head by using Beta-spline using the developed algorithm. Push button object is the action button which processes the code. Meanwhile, the edit box displays the time taken after the code runs. The interface design that has been developed in GUIDE mode is shown in Figure 6 below.

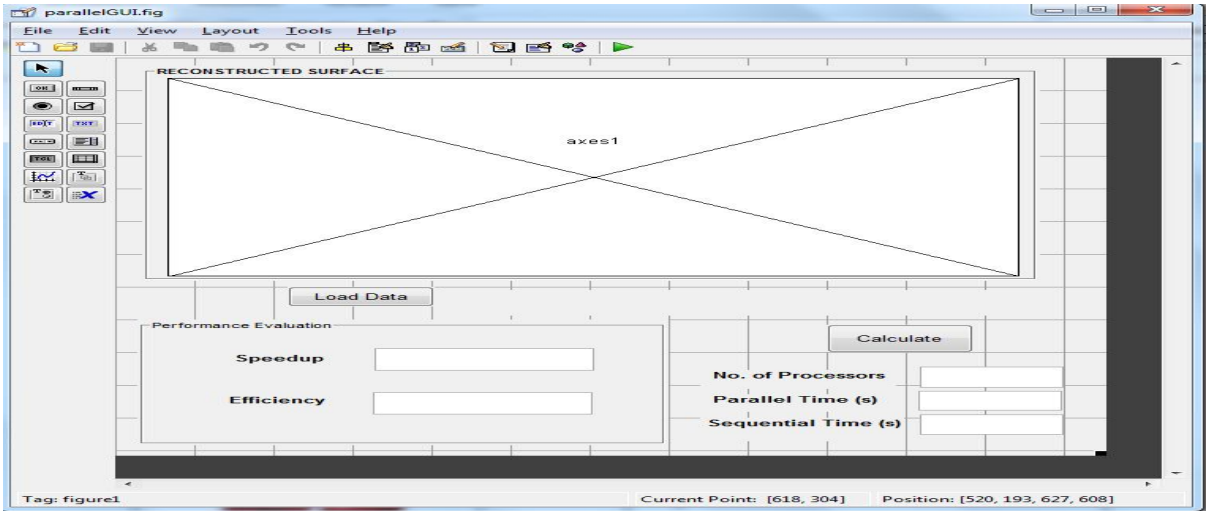


Fig. 6. The Developed GUI to Reconstruct Surface in MATLAB.

V. RESULTS AND DISCUSSION

Figure 7 below shows the generated surface from multi slice CT scan images by using 2 and 4 processors. The surface was analyzed to determine whether the surface is smooth and all the patches continuously connected to each other.

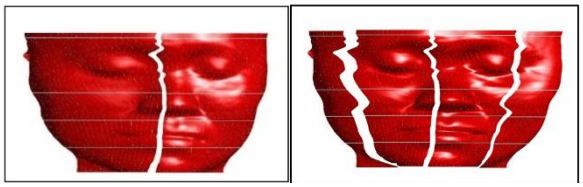


Fig. 7. Reconstructed Surfaces using 2 and 4 Processors.

In this study, the surface was in horizontal segmentation. Control points can be segmented horizontally or vertically without making the surface fail to be generated successfully. It is because the Beta-spline technique is guaranteed G^2 continuous [4].

In the implementation section, the results for each node were moved to the client pc. Thus, a GUI using MATLAB was develop and used to reconstruct the surface and display the results with computation time. This GUI is shown in Figure 8.

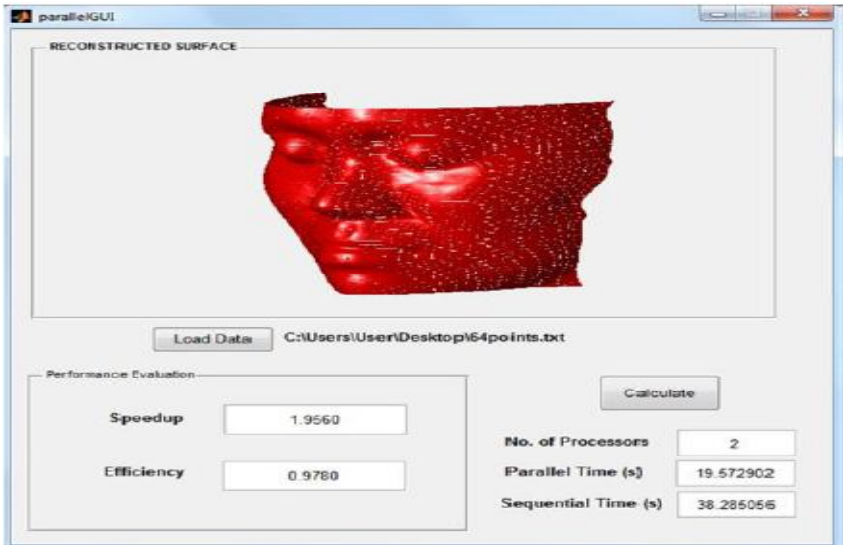


Fig. 8. GUI for Parallel Beta Spline Surface Reconstruction.

The processing time, speedup and efficiency were measured in order to evaluate the performance of the cluster. The measurement involves the evaluation of time elapsed using sequential and parallel algorithms with different number of surface points. The time computation was computed and recorded from the GUI. Then, the results weretabulated in Table I below.

Num. of Processor	Num. of Surface point, N				
	2	10	20	30	40
1	36.0447	36.5838	37.1269	37.6421	38.2851
2	18.9218	19.0713	19.2219	19.3261	19.5729
4	9.6651	9.7206	9.8049	9.8893	9.9855

Table: 1 Processing Time

Results from Table I presents that there is a significant difference of processing time between sequential and parallel computation with different processors. The results show that by increasing the number of processor, the time processing is decreasing. Parallel computation by using two processors decreases the processing time by almost half of the sequential processing time. It is also observed that the processing time by using four processors is faster than by using two processors. The number of N surface points should be at least 40 to make the surface smoother without any hole. It can be observed that by increasing the number of N, the computation time has also increased. But the time can be reduced by using parallel computation.

Table I above shows the average processing time in second (s) for the ARS PCS with 1, 2 and 4 processors. Each processor consists of the value 2, 10, 20, 30 and 40 surface points. The time taken is actually an average from 4 processors actual reading of the processing time.By using the result in Table I, the speedup and efficiency of the ARS cluster were calculated and the result is shown on Table II below.

N	Number of processor(s)					
	Speedup			Efficiency		
	1	2	4	1	2	4
2	1.0000	1.9049	3.7294	1.0000	0.9525	0.9323
10	1.0000	1.9183	3.7635	1.0000	0.9591	0.9409
20	1.0000	1.9315	3.7866	1.0000	0.9657	0.9466
30	1.0000	1.9477	3.8064	1.0000	0.9739	0.9516
40	1.0000	1.9560	3.8341	1.0000	0.9780	0.9585

Table: 2 Analysis of Parallel Performance

The numerical results show that by increasing the number of processor, the speedup also increased. That means the parallel computation is capable of making calculation faster than the sequential process. It also shows that by increasing the number of N surface points, the speedup also increased.

In observing efficiency of data, 100% utilization of the processor was used for the case of 1 processor. Meanwhile, when using 2 and 4 processors, the efficiency start to decrease to around 96% and 94% respectively. This indicates that the computation took another 4% and 6% as the communication time between the processors.

CONCLUSION

Beta spline technique is guaranteed G^2 continuous and the computation uses independent data points. Therefore, parallel computation can be done by dividing the data points and assign each processor with the separate data and can be successfully combined together without considering the continuity problem. The surface of the human head by using beta-spline was successfully generated as shown in results and discussion section because it is identical to the work done by Hadi [2].

This study highlights the uses of parallel computation to solve high computations in beta spline surface fitting using a low-cost self-develop parallel laboratory [5]. The computation successfully been done on MATLAB "pmode" platform and the results was shown on the developed GUI.

ACKNOWLEDGEMENT

We would like to thank Universiti Teknologi MARA (UiTM) for giving us facilities and financial support in conducting this study.

REFERENCES

[1] Faria, L. N., Mascarenhas, N. D., Morón, C. E., Saito, J. H., Rosa, R. R., & Sawant, H. S. (2007). A parallel application for 3D reconstruction of coronal loops using image morphing. Image and Vision Computing, 25(1), 95-102.

[2] Hadi, N. A., Ibrahim, A., Yahya, F., & Ali, J. M. (2012). Beta-spline Surface Fitting to Multi-slice Images. Journal of World Applied Science and Technology, 1326-1331.

[3] Nordqvist C. (February 5th 2016), CT Scan (CAT Scan): How Do They Work? Available: http://http://www.medicalnewstoday.com/articles/153201.php.

[4] Barsky, B. A., & DeRose, T. D. (1989). Geometric continuity of parametric curves: three equivalent characterizations. IEEE Computer Graphics and Applications, 9(6), 60-68.

[5] Sulaiman, H., Said, N. M., Ibrahim, A., & Alias, N. (2013, March). High performance visualization of human tumor detection using WTMM on Parallel Computing System. In Signal Processing and its Applications (CSPA), 2013 IEEE 9th International Colloquium on (pp. 205-208). IEEE.

[6] Hadi, N. A., Halim, M. S. A., Ibrahim, A., Sulaiman, H., Yahya, F., & Ali, J. M. (2014, January). Dyadic Segmentation for Parallel Beta-Spline Surface Reconstruction from Multi-slice Images. In 2014 5th International Conference on Intelligent Systems, Modelling and Simulation (pp. 524-528). IEEE.

[7] Matczak, K., Romanowska, A. B., & Smith, J. D. (2011). Dyadic polygons.International Journal of Algebra and Computation, 21(03), 387-408.

[8] Sulaiman, H., Said, N. M., Ibrahim, A., & Alias, N. (2013, March). High performance visualization of human tumor detection using WTMM on Parallel Computing System. In Signal Processing and its Applications (CSPA), 2013 IEEE 9th International Colloquium on (pp. 205-208). IEEE.

[9] Yahya, F., Ali, J. M., Majid, A. A., & Ibrahim, A. (2008, September). Automatic G1 surface reconstruction from serial cross-sectional images. In International Conference on Advances in Visual Information Systems (pp. 96-99). Springer Berlin Heidelberg.

Singh, K., Kwoh, L. K., & Lim, H. (1998, July). A stochastic stereo matching technique based on beta-splines. In Geoscience and Remote Sensing Symposium Proceedings, 1998. IGARSS'98. 1998 IEEE International (Vol. 5, pp. 2354-2356). IEEE.

[10] Sharma, G., & Martin, J. (2009). MATLAB®: a language for parallel computing. International Journal of Parallel Programming, 37(1), 3-36.

[11] Luszczek, P. (2009). Parallel programming in MATLAB. International Journal of High Performance Computing Applications, 23(3), 277-283.

★★★