3-Dimensional Human Head Reconstruction Using Cubic Spline Surface on CPU-GPU Platform

Normi Abdul Hadi
Ibnu Sina Institute for Scientific and Industrial Research
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
+607-5536075
normi@fskm.uitm.edu.my

Norma Alias
Ibnu Sina Institute for Scientific and Industrial Research
Universiti Teknologi Malaysia
Johor Bahru, Malaysia
+607-5536075
normaalias@utm.my

ABSTRACT

This paper presents a CPU-GPU architecture for 3-dimensional human head reconstruction. The CT scan images of human head are processed to get the surface points, before spline surface are fitted to the points. In the process, a step is chosen to be executed on the GPU which is the cumulative sum calculation. This is because this step executes high processing time in the CPU. The developed architecture is tested on various numbers of surface points between 20 to 5000 points. Then, the performance of the developed architecture is measured based on execution time, speedup and granularity. The result shows that 3D human head reconstruction gives better performance by employing GPU in the process as compared to CPU, where the GPU execution time is 3.08 faster than the CPU. In terms of granularity, the computation time is 41.69 greater than the communication time. Therefore, the developed CPU-GPU architecture is suitable to be implemented for 3-dimensional human head reconstruction.

CCS Concepts

• Computing methodologies→Parallel computing methodologies→Parallel algorithms.

Keywords

Spline; CUDA; Parallel; CAGD.

1. INTRODUCTION

Real image reconstruction technique is continuously been improved due to the advancement of technology, and the demand of world. Medical image such as Computerized Tomography (CT) image requires high resolution of the rendered image, to ensure the related analysis can be done precisely. Since this type of image involved human or animal, the image will be complicated and consists of thousands of pixels. Therefore, huge memory, powerful processor, fast computational process and limited communication cost are required to handle this image [1, 2]. The burden of a processor can also be reduced by redesigning the algorithm for the whole process. One of the suitable design is by transferring some of sequential algorithms into parallel algorithms [3].

The aim of this paper is to design a CPU-GPU architecture in 3D human head reconstruction using spline and perform the computation on the developed architecture. Spline is chosen due to its properties which can produce smooth and accurate curve and surface. Few works can be found on parallelizing 3D image reconstruction. In [4] and [5], beta-spline is employed to generate the surface independently in several processors. The data can be easily distributed without worrying about the smoothness on the generated surface since cubic beta-spline has preserved G^2 continuity.

GPU is Graphics Processing Unit, originally developed to accelerate graphical applications, but nowadays it also helps in computation. Modern GPUs consist of hundreds of cores can process thousands of threads [6]. The chosen GPU is Compute unified device architecture (CUDA) by NVIDIA introduced in 2006, which allows users to use a GPU for general computing [7, 8]. Although GPU is a powerful device, using GPU alone for a process may produce high execution time due to the communication time between threads. Therefore, CPU-GPU platform is preferred with task distribution according to the specific capabilities. Generally, serial portions are executed on the CPU, while parallel portions are executed on the GPU to optimize the performance [9].

The illustration of GPU is given in Figure 1 [10].

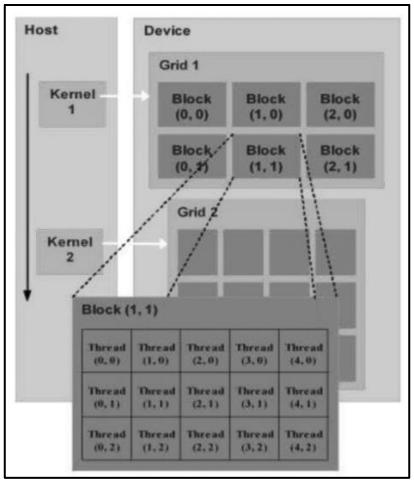


Figure 1. GPU illustration.

The host in the Figure 1 is the CPU, and GPU is the device. A GPU consists of grids, blocks and threads with their own ID. The

ICIIT '19, February 20–23, 2019, Da Nang, Viet Nam
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6633-5/19/02…$15.00
DOI: https://doi.org/10.1145/3321454.3321476

thread can communicate with other thread in a block by high-speed shared memory, and other blocks by global memory [11].

Parallelization system carried out in this paper is based on a conceptual framework proposed by [1] named as V-cycle as shown in Figure 2.

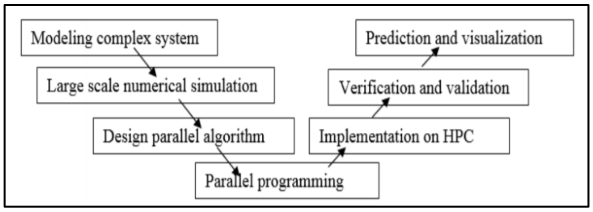

Modeling complex system
Prediction and visualization
Large scale numerical simulation
Verification and validation
Design parallel algorithm
Implementation on HPC
Parallel programming

Figure 2. Conceptual Framework.

Based on the framework, the first process is the modelling complex system. In this paper, the modelling process is the preprocessing stage to extract the image outline from the original CT scan image as shown in Figure 3.

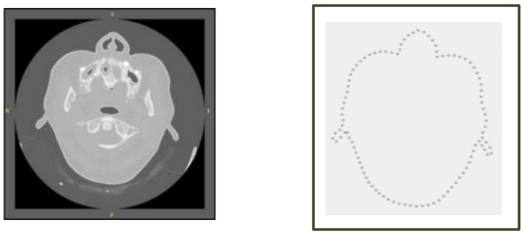


Figure 3. CT scan image and the extracted outline.

The surface points for image visualization are extracted from the outline in Figure 3. This is where the second step of V-cycle involved which is large scale numerical simulation. Each slice of CT scan images consists of hundreds of outline points. Thus, thousands of points are involved for the whole set of CT scan images. Figure 4 shows six samples of stacked CT scan images.

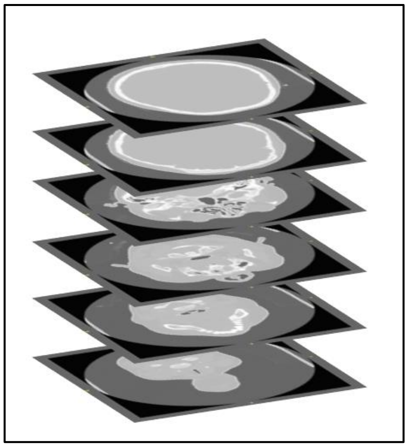


Figure 4. Stacked CT scan images.

All points for each slice are significant to be considered in image visualization to avoid misinterpretation of the image. Thus, large enough surface points must be extracted. The process for these huge amount of surface data is designed for parallel system, as in steps three to five in V-cycle. For verification and validation process for the developed CPU-GPU platform, it is based on execution time, speed up and granularity which is discussed in the result section. Finally, the visualization process is by rendering 3D image of human head.

The rest of this paper is organized as follows. Section 2 will discuss reconstruction process of 3D model of human head. Then in Section 3, the implementation of CPU-GPU algorithm will be elaborated. Result and performance analysis are presented in Section 4. This paper ends with conclusion in Section 5.

2. 3-DIMENSIONAL HUMAN HEAD RECONSTRUCTION PROCESS

This section discusses on the human head reconstruction process starting from extracted 2D image outline. All points are fitted with cubic spline curves, $F(\mathrm{t})$ where

$$F(\mathrm{t}) = \sum_{i=0}^{n} V_i B_i(t) \qquad \textbf{(1)}$$

Curve equation in (1) is the standard spline equation for degree n with control points V_i, and basis function $B_i(t)$, and $0 < t < 1$. One of suggested spline to be used is beta-spline. This curve is built on G2 continuity condition that preserves the smoothness of the curves [12].

Then, required curve points are extracted from (1). This project considers 10 curve points per segment, and extracted as

$$\begin{gathered} p_i = F(t_i) = \sum_{i=0}^{n} V_i B_i(t_i) \\ \text{where } t_i = 0{,}0.1{,}0.2, \ldots ,1 \end{gathered} \qquad \textbf{(1)}$$

Next step is calculation for distance between points, p_i. This step is important to extract the surface points afterwards. The distance, D_i is calculated as,

$$D_i = \| p_i - p_{i+1} \| \qquad \textbf{(2)}$$

Consequently, the distance from starting point p_0 to the ith point $p_i$given as, $\sum_{i=0}^{i} p_i$ called as cumulative distance. Finally, surface points, S_i are assigned when its cumulative sum satisfies，

$$\begin{gathered} \sum D_i < h_j \\ \text{where } h_j = \frac{j * \sum D_i}{\text{total number of surface points (TTS)}}. \end{gathered} \qquad \textbf{(3)}$$

Total number of surface points (TTS) must be big enough to ensure the accuracy of the produce 3D image. This project considers 20, 50, 100, 500, 1000, 1500 and 5000 surface points for comparison. It is expected that 5000 surface points gives the best 3D image, but high execution time. Therefore, GPU is employed as the platform to reduce the time. The process is discussed in the next section.

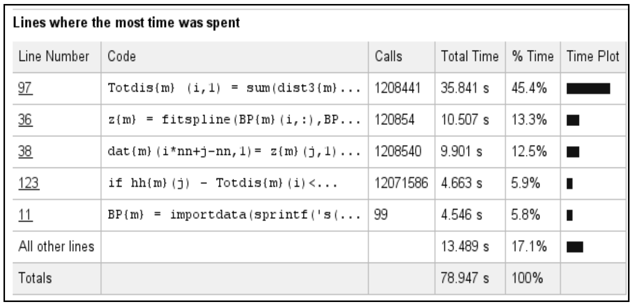
3. 3-DIMENSIONAL HUMAN HEAD RECONSTRUCTION ON CPU-GPU PLATFORM

CPU-GPU platform is the collaboration of CPU and GPU in running the whole process. This collaboration is better than CPU-alone or GPU-alone, because the task is divided based on the capabilities of the CPU and GPU. the hardware used in this project are listed in Table 1.

Table 1. Hardware specification

Hardware	Specification
CPU	Intel (R) XEON (R) (2.10GHz) 2 processors
GPU	NVIDIA Tesla K20c Max Thread Block Size = [1024 1024 64]
OS	Windows 10 64-bit
Software	MATLAB and C++

There are two elements of software used for computation part. Generally, the command code is written in MATLAB. But the GPU kernel is written using C++ language, and executed in MATLAB. The task that will assigned to the GPU is decided based on its execution time on the CPU. Figure below shows the five highest execution time for 20 number of surface points, given by MATLAB profiler.



Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
97	`Totdis{m} (i,1) = sum(dist3{m}...`	1208441	35.841 s	45.4%	■
36	`z{m} = fitspline(BP{m}(i,:),BP...`	120854	10.507 s	13.3%	■
38	`dat{m}(i*nn+j-nn,1)= z{m}(j,1)...`	1208540	9.901 s	12.5%	■
123	`if hh{m}(j) - Totdis{m}(i)<...`	12071586	4.663 s	5.9%	■
11	`BP{m} = importdata(sprintf('s(...`	99	4.546 s	5.8%	■
All other lines			13.489 s	17.1%	■
Totals			78.947 s	100%	

Figure 5. Execution time for five (5) codes for 20 surface points given by MATLAB profiler.

Figure 5 shows that the cumulative distance calculation needs the highest execution time which is 45.4% of total time, followed by the curve fitting process with 13.3%. Other three lines does not involve computation process. Therefore, this cumulative distance step will be executed in the GPU to reduce the execution time. The architecture of CPU-GPU is shown in the following figure.

Figure 6 is the extended version of Figure 1, with specific steps. Step 1 until 3 are executed on the CPU. After that, the calculated distance, D_i is passed to the GPU by the kernel function to calculate the cumulative sum. The calculated cumulative sum is then passed back to the CPU for surface points extraction and rendering process in steps 5 and 6 respectively. The data passing process (CPU-GPU-CPU) must be also considered in the analysis, since it requires communication time. The analysis is given in the following section.

4. RESULTS AND PERFORMANCE ANALYSIS

The developed architecture in Figure 6 is applied to the whole CT images. In this paper, total of 99 slices are considered in image rendering. The rendered 3D images for 20, 50, 100, 500, 1000, 1500 and 5000 surface points per slice are given in the following Figure 7.

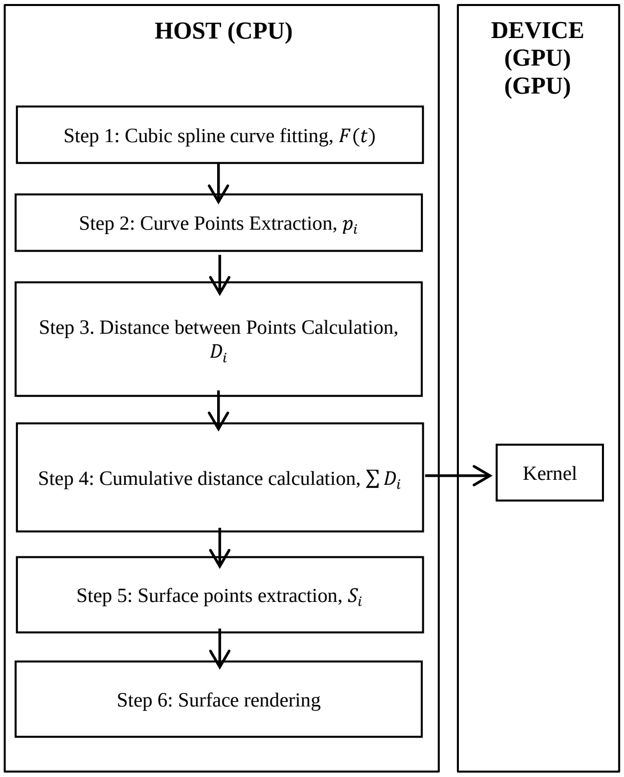
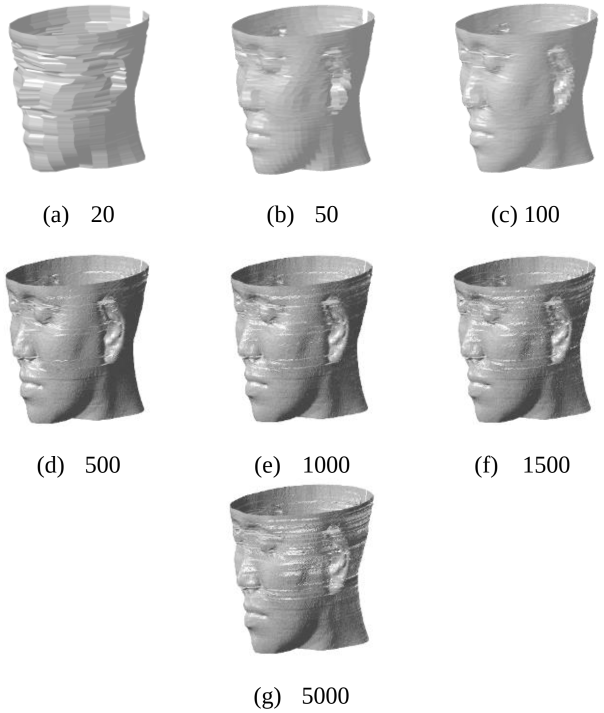

HOST (CPU)
DEVICE (GPU) (GPU)
Step 1: Cubic spline curve fitting, $F(t)$
Step 2: Curve Points Extraction, p_i
Step 3. Distance between Points Calculation, D_i
Step 4: Cumulative distance calculation, $\sum D_i$
Kernel
Step 5: Surface points extraction, S_i
Step 6: Surface rendering

Figure 6. The CPU-GPU architecture for six (6) processes with one kernel.



(a) 20 (b) 50 (c) 100

(d) 500 (e) 1000 (f) 1500

(g) 5000

Figure 7. Rendered 3D images with different number of surface points.

It is to be highlighted that the stated surface points are for one slice. A set of human head has 99 slices, therefore the total amount of considered surface points is as minimum as 20 × 99 points and as maximum as 5000 × 99 points. For small amount of surface points which are 20, 50, 100, the improvement of the quality of generated 3D image can be observed. The features of human head especially the face becomes clearer as the number of surface points are increased. For 500 surface points and above, the improvement can be observed on the ear area. Human's ear is a very complicated part to be generated since it involved many curves. Figure 8 shows an example of human ear area of a CT image slice.

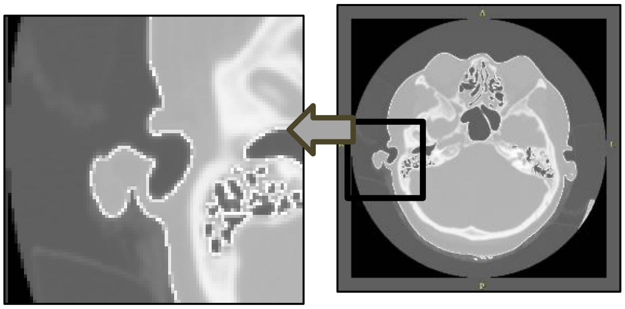


Figure 8. Example of human ear area from a CT image.

The ear area in the figure has a lot of curves. Thus, the number of surface points of human head must be big enough to represent each point of the ear.

The developed method is also been analyzed for its parallel performance. In this paper, the analysis of execution time, speed up and granularity is presented. The execution time for each number of surface points is given in the following figure.

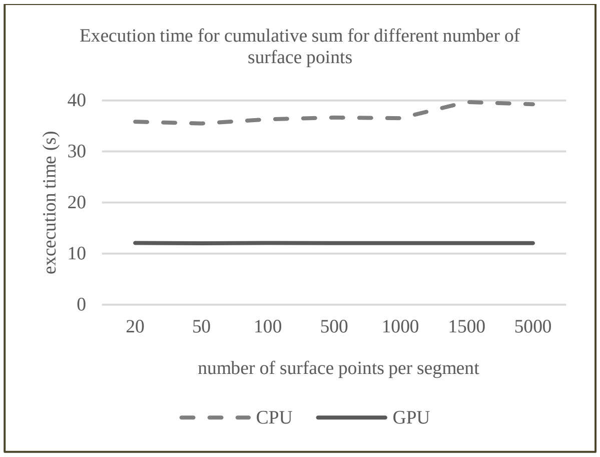
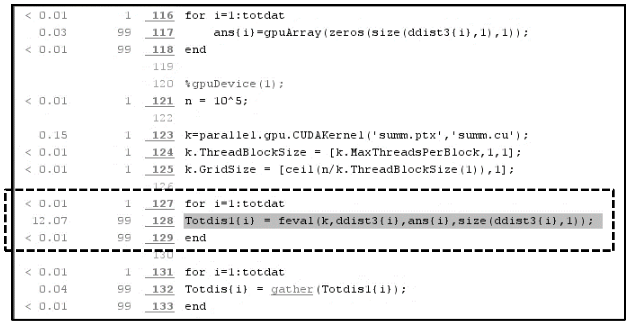


Figure 9. Execution time for cumulative sum for different number of surface points.

From Figure 9, the execution time is stable for all numbers of surface points with average 37.11s for CPU and 12.05 for GPU. Thus, the average speed up given as CPU time per GPU time is 3.08 which means that with the assistance of GPU, the execution is 3.08 time faster.

As mentioned before, the data passing between CPU and GPU requires communication time. It is an important factor and related to the dependencies between objects [13]. Communication time for cumulative distance process is given by Figure 10.



```
< 0.01    1  116 for i=1:totdat
  0.03   99  117     ans{i}=gpuArray(zeros(size(ddist3{i},1),1));
< 0.01   99  118 end
             119
             120 %gpuDevice(1);
< 0.01    1  121 n = 10^5;
             122
  0.15    1  123 k=parallel.gpu.CUDAKernel('summ.ptx','summ.cu');
< 0.01    1  124 k.ThreadBlockSize = [k.MaxThreadsPerBlock,1,1];
< 0.01    1  125 k.GridSize = [ceil(n/k.ThreadBlockSize(1)),1];
             126
< 0.01    1  127 for i=1:totdat
 12.07   99  128 Totdis1{i} = feval(k,ddist3{i},ans{i},size(ddist3{i},1));
< 0.01   99  129 end
             130
< 0.01    1  131 for i=1:totdat
  0.04   99  132 Totdis{i} = gather(Totdis1{i});
< 0.01   99  133 end
```

Figure 10. Execution time for each line for GPU for 20 surface point.

Figure 10 shows time for cumulative summation calculation using GPU. Command lines in the dashed box show the computation time. Other lines are communication time whether data passing CPU-GPU or otherwise. Communication and computation time give the granularity of the GPU which is computation time per communication time. The granularity is 41.69, thus computation time is 41.69 higher than the communication time. In designing CPU-GPU architecture, the granularity must be big enough to ensure it is worth to use the GPU.

5. CONCLUSION

The CPU-GPU architecture for 3D human head reconstruction is presented in this paper. The result shows that by employing the GPU, the performance can be improved in terms of speedup and granularity. In this human head case, CPU still can avoid as much as 5000 × 99 of surface points. However, it might be out of memory for a larger amount of data. Thus, the purpose of GPU is not only to speed up the execution process, but also to handle large amount of data that cannot be afforded by CPU alone. For future improvement, more kernel will be included in the architecture. Additionally, multiple GPU devices can also be used for the whole process.

6. ACKNOWLEDGMENTS

The author would like to express sincere gratitude to the ICIIT 2019 for the invitation for this conference. To Ministry of Education, Malaysia, Universiti Teknologi MARA and Universiti Teknologi Malaysia for funding this study.

7. REFERENCES

[1] Alias, N., Musa, H., Riyadh Sergey, V., Hamzah, N. and Mugahed Al-rahmi, W. 2017. Nanotechnology Theory Used For Simulation Of Emerging Big Data Systems On High Performance Computing: A Conceptual Framework. *Journal of Theoretical and Applied Information Technology*. 95, (2017), 22.

[2] Alias, N., Sahnoun, R. and Malyshkin, V. 2017. High-Performance Computing And Communication Models For Solving The Complex Interdisciplinary Problems On DPCS. *ARPN Journal of Engineering and Applied Sciences*. 12, 2 (2017).

[3] Alias, N., Nofri, N., Suhari, Y., Farhah, H., Saipol, S., Dahawi, A.A., Saidi, M.M., Hamlan, A., Rahim, C. and Teh, C. 2016. Parallel Computing Of Numerical Schemes And Big Data Analytic For Solving Real Life Applications. *Jurnal Teknologi*. 78, 8–2 (2016), 151–162.

[4] Abdul Hadi, N., Abd Halim, M.S., Ibrahim, A., Sulaiman, H., Yahya, F. and Md Ali, J. 2014. Dyadic Segmentation for

Parallel Beta-Spline Surface Reconstruction from Multi-slice Images. *Fifth International Conference on Intelligent Systems, Modelling and Simulation* (2014), 524–528.

[5] Syawal, M., Halim, A., Hadi, N.A. and Sulaiman, H. 2017. An Algorithm for Beta-Spline Surface Reconstruction from Multi Slice CT Scan Images using MATLAB pmode. *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)* (2017), 1–6.

[6] Raju, K. and Chiplunkar, N.N. 2018. A survey on techniques for cooperative CPU-GPU computing. *Sustainable Computing: Informatics and Systems*. 19, (2018), 72–85. DOI= https://doi.org/10.1016/j.suscom.2018.07.010.

[7] Kim, J.S. and Lee, M. 2018. Image Blending Techniques Based on GPU Acceleration. (2018), 106–109.

[8] Saidi, M.M. and Alias, N. 2016. High Speed Computing of Ice Thickness Equation for Ice Sheet Model. *Jurnal Teknologi*. 78, 8–2 (2016), 143–149.

[9] Nickolls, J. and Dally, W.J. 2010. The GPU computing era. *IEEE Micro*. 30, 2 (2010), 56–69. DOI= https://doi.org/10.1109/MM.2010.41.

[10] Alias, N., Hidayad, M. and Kamal, A. 2017. Integration Of A Big Data Emerging On Large Sparse Simulation And Its Application On Green Computing Platform. 12, 12 (2017), 3817–3826.

[11] Alias, N., Mohamad Mohsin, H., Nadirah Mustaffa, M., Hafilah Mohd Saimi, S. and Reyaz, R. 2016. Parallel Artificial Neural Network Approaches For Detecting The Behaviour Of Eye Movement Using Cuda Software On Heterogeneous CPU-GPU Systems. *Jurnal Teknologi*. 78, 12–2 (2016), 77–85.

[12] Halim, S.A., Halim, M.S.A. and Hadi, N.A. 2018. Surface reconstruction from computed tomography (CT) image of human head with the effect of noise. *AIP Conference Proceedings*. 2013, (2018). DOI= https://doi.org/10.1063/1.5054216.

[13] Elkhani, N., Muniyandi, R.C. and Zhang, G. 2018. Multi-Objective Binary PSO with Kernel P System on GPU. 13, June (2018), 323–336.