Massive Learning Algorithm in Surface Reconstruction

Wai Pai Lee [1], Shafaatunnur Hasan [1, 2], Siti Mariyam Shamsuddin [1, 2]

[1] Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia
[2] Big Data Centre, Universiti Teknologi Malaysia (UTM), Malaysia
pailee.wai@gmail.com, shafaatunnur@utm.my, mariyam@utm.my

Abstract. The evolution of 3D scanning devices are rapid and the size of point cloud from those devices is becoming large. In fact, large scale point cloud surface reconstruction consumes a lot of time for processing. In this paper, Massive Learning architecture self-organizing map (SOM) is designed to solve large scale point cloud surface reconstruction problem. In addition, part of SOM algorithm is processed on NVIDIA GPU. By doing this, surface reconstruction time for large scale point cloud will be reduced. The final output point cloud object is not completely perfect due to the limitation of SOM. The lack of sides' relationship leads to imperfect object. In the performance experiments, Massive Learning architecture SOM is better than single layer SOM while the size of point cloud increases. The difference between them become significant when the point cloud size increases.

Keywords: Point cloud, self-organizing map, massive learning, surface reconstruction, parallel computing

1 Introduction

Recently, Big Data has become one of the famous issues. Digital information has grown nine times in the year 2011 compared with the digital information in the year 2006 and the amount of it will reach 25 trillion gigabytes following the trend [1]. Due to this Big Data trend, most of the solution related to data forced to be modified or changed including 3D scan point cloud.

In recent years, the number of 3D data processing concerned applications has increased because of the emergence of the 3D sensor and scanning devices [2]. Therefore, there are a lot of new methods are invented in order to solve the slow performance problem of large scale 3D point cloud data processing. For instance, Artificial Neural Network (ANN), Genetic algorithm, simulated annealing, particle swarm optimization and differential evolution are used to solve surface reconstruction problem [3]. Besides new inventions, modification of existing approach is also a solution such as Screened Poisson Surface Reconstruction [4].

On the other hand, Massive Learning is always used for large scale complicated data processing. In fact, this is because it will classify data and reduce the dataset complexity in the early layers. Basically, Massive Learning is a set of machine learning technique that learns multiple levels [1]. Furthermore, parallel computing is very suitable for Massive Learning processing.

In this paper, multilayer approach with the GPU platform is chosen in this research. Although there are many phases while rendering an object from the point cloud, 3D surface reconstruction phase will be the only concentration. Besides that, only NVIDIA GPU will be used in this research in order to standardize the program algorithm. For GPU implementation, GPUMLib [5] SOM will be the base SOM algorithm and additional layer will be implemented to achieve multilayer approach.

There are several objectives in this research. At the beginning, fundamental SOM will be implemented to test the suitability of that algorithm in surface reconstruction. Other than that, multilayer concepts and parallel computing approach will be implemented with the basic SOM. This step mainly improves the performance of basic SOM in large scale 3D surface reconstruction process. Lastly, the performance speed while reconstruction 3D object will be compared and analysed. The comparisons are between multilayer SOM and basic SOM and between parallel approach and CPU sequential approach.

This research aims to solve large scale 3D surface reconstruction problem, but the contributions are not limited to 3D objects only. For instance, 2D maps or images can also be reconstructed. The exponential growth of dimensions data decreases the performance drastically. Thus, this research will bring benefits to N dimensional representation and construction.

2 Problem Background

3D surface reconstruction are becoming more details and demanding these years. The evolution of scanning devices and technology make this happen. However, the explosion of control point details causes problems on performance of 3D surface reconstruction and representation.

Currently, there are many solutions to do 3D surface reconstruction such as screened poison surface reconstruction, artificial neural network and so on. However, these solutions do not optimize the points before doing surface reconstruction. Therefore, this paper propose using additional layer for optimizing the points.

3 Methodology

In this paper, the final output will be the reconstructed point cloud and the time performance for that process. There are three main phases conducted in this research. Basically, the first phase will be the preparation and design of the research, the second phase is the implementation phase and lastly the analysis phase.

In the first phase, data source has been identified. The data source is the data set from Point Cloud Library (PCL) which is one of the open source point cloud data source. There are a lot of point cloud data there, selection of data is done by taking only the ply format files. Besides data source selection, hardware specification is set and fixed throughout the whole research.

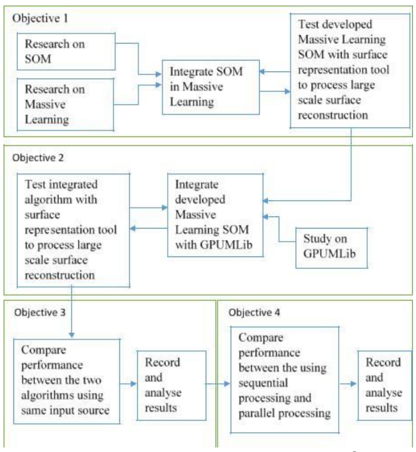

Figure 2.1 Implementation Workflow

Next, the implementation workflow is started as shown in figure 2.1. At the beginning, a sequential approach Multilayer SOM is implemented by modifying the base basic SOM. After that, parallel computing approach is implemented in order to enhance the performance of completed Multilayer SOM. Lastly, several datasets are used as input and the results are analysed.

4 Proposed Multilayer SOM

In basic SOM, there are several components which are initialization, competition, cooperation, and adaptation [5]. In proposed Multilayer SOM, each layer of it does the same process like basic SOM. The flow of the Multilayer SOM is shown in figure 3.1.

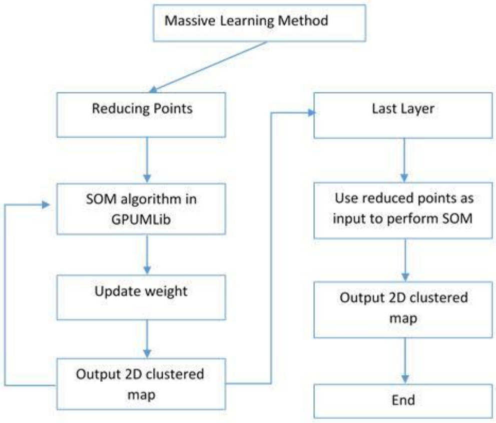

Figure 3.1 Multilayer SOM flow

In the first component, the weight of each SOM layer is initialized with the minimum value of 0 and the maximum value of 1024. This is because the input point cloud coordinates are rescaled to fit a size of 1024 cube in order to standardize the data. Next, competition process is run for finding the winning node in the map. After that, the cooperation process which finds the topological neighbourhood by their distances. Lastly, the weight of each topological neighbourhood nodes is updated in the adaptation process.

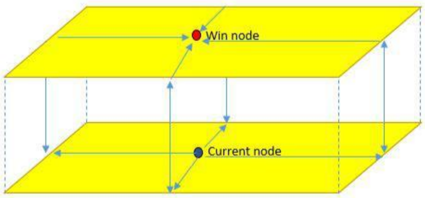

Figure 3.2 Dual Layer Closed SOM

In this paper, a modified SOM is used as multilayer SOM's layer for getting the sides relationship in the topological map. This modified SOM improves the relationship of the sides' node but not completely solve the limitation of sides' node relationship in basic SOM. The idea is linking the sides by giving double layer of the topological map as shown in figure 3.2.

5 Result and Discussion

Figure 4.1 shows the output model of the surface reconstruction experiment.

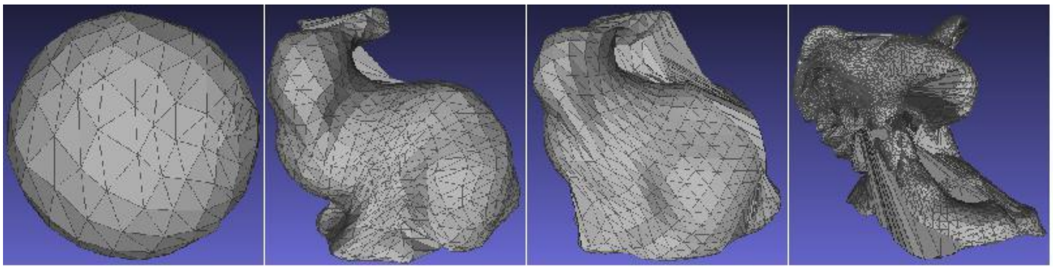

Figure 4.1 Surface Reconstruction Model Output

From the result, it shows that a well-structured point cloud without concave structure like sphere object can be reconstructed nearly perfectly using the proposed Multilayer SOM. Because of the well-organized points, the relationship between points can be identified easier without large conflicts. However, inaccurate result happens when the concave structure exists.

In the second and third object from Figure 4.1, the concave part which is the bunny's ear is not able to be reconstructed accurately. This is because there is no range comparison in the competition of SOM. Thus, the points located at another side will be considered as the neighbour of certain points. In fact, the output of eagle point cloud faced the same issue.

Table 4.1 Comparison between Single Layer SOM and Multilayer SOM

	Single Layer SOM				Multilayer SOM			
Model	Sphere	Bunny1	Bunny2	Eagle	Sphere	Bunny1	Bunny2	Eagle
Points	422	8171	35947	796825	422	8171	35947	796825
Iteration	1000	1000	1000	500	1575	1575	1575	1575
MapX	10	25	25	100	20	40	40	100
MapY	20	40	50	200	40	60	60	200
Time (s)	4.84	124.528	619.12	52560	10.08	60.192	75.21	4392

Table 4.1 shows the performance comparison between single layer SOM and Multilayer SOM. The results show that single layer SOM is better in term of time taken when the data is small. This is expected because the minimum map size in Multilayer SOM is bigger than single layer SOM in order to fit the map size decreasing behaviour after each layer. Large map size requires higher processing power. Therefore, Multilayer SOM is not performing well in small scale data.

The situation changes significantly when the data size increases. There is a huge gap in the eagle result. From that result, Multilayer SOM is faster than single layer SOM around 12 times. In Multilayer SOM, the initial iteration number for reduction of point cloud data is just 25 times but it is 500 times in single layer SOM. After the reduction, the input points of second layer SOM in Multilayer SOM become 20000 (100 x 200) points which are just around 2.5% of the original points. Thus, the high iteration number in Multilayer SOM does not limit the performance.

Table 4.2 Performance Comparison between CPU and GPU using Multilayer SOM

	CPU				GPU			
Model	Sphere	Bunny1	Bunny2	Eagle	Sphere	Bunny1	Bunny2	Eagle
Points	422	8171	35947	796825	422	8171	35947	796825
Iteration	1575	1575	1575	1575	1575	1575	1575	1575
MapX	20	40	40	100	20	40	40	100
MapY	40	60	60	200	40	60	60	200
Time (s)	4.30	64.70	87.32	9540	10.08	60.192	75.21	4392

Table 4.2 shows the performance comparison between CPU and GPU using Multilayer SOM. From the results, CPU approach is faster when reconstructing small data like the sphere but it becomes slower when the data size increases. This is because the map size is too small for utilizing all of the threads and blocks in GPU. Thus, GPU approach is slightly slower than CPU approach in small data case.

Actually, the main reason is not caused by the size of point cloud data but the map size. Parallel computing in Multilayer SOM is implemented when calculating the winning node and updating the weights. Hence, there is no a huge difference changed between Bunny1 and Bunny2. In fact, the map size depends on the points of data. When the details of points increase, map size should increase in order to remain most of the important points. Therefore, data size affects the performance differences between CPU and GPU indirectly.

6 Conclusion

Multilayer SOM has been proven that it is able to reduce the time taken for large scale surface reconstruction. However, there is a trade-off whereby the efficiency of surface reconstruction process in small scale data is not well according to the research result. Furthermore, the output of Multilayer SOM is depending on the first layer output. If the output is not very accurate, the final output will be wrong. Therefore, a good map size and number of iteration are needed in order to have a good result.

Also, the modified SOM map structure in this research has been proven that it is able to improve closed surface structure point cloud object such as the sphere. However, the limitation for concave structure still exists as shown in the results. The limitation happens when the point cloud object has concave structure. The inaccurate relationship leads the result imperfect.

Besides that, parallel computing platform (NVIDIA GPU) has been used to improve the performance while calculating winning node and updating the map grid weights. It has been proven that parallel computing platform is a good solution for large scale data but not small scale data. This is because GPU threads and blocks are not fully utilized when dealing with small data.

References

1. Chen, X. W. and Lin, X. (2014). Big Data Massive Learning: Challenges and Perspectives. IEEE Access. 2, 514–525.
2. Morell, V., Orts, S., Cazorla, M. and Garcia-Rodriguez, J. (2014). Geometric 3D point cloud compression. Pattern Recognition Letters. 50, 55–62.
3. Lim, S. P. and Haron, H. (2012). Surface reconstruction techniques: a review. Artificial Intelligence Review. 42(1), 59–78.
4. Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. TOG. 32(3), 1–13.
5. Hasan, S., Shamsuddin, S. M. and Lopes, N. (2014). Machine learning big data framework and analytics for big data problems. Int. J. Advance Soft Compu. Appl. 6(2).
6. Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE. 78(9), 1464–1480.