Algorithm Design of Global Point Cloud Registration Based on BSP Model

Song Liu
College of Computer Science & Information
Guizhou University
Guiyang 550025, China
e-mail: songliu@vip.163.com

Xiaoyao Xie
Key Laboratory of Information & Computing Science
Guizhou Normal University
Guiyang 550001, China
e-mail: xyx@gznu.edu.cn

Abstract—Registrations of multi-angle point cloud involve enormous point cloud data, complicated and heavy computation. Previous non-parallel method is strong computing resources demanding and low performance of point cloud global registration. Bulk Synchronous Parallel Computing Model (BSP) processes the data with a parallel computing method, which can be applied to huge computation such as point cloud data processing. We design a point cloud global registration algorithm base on BSP model and build a Hama parallel computing cluster with average PCs. The results of four engineering examples show that the registration algorithm base on BSP model reduces computing resources requirements, accomplishes global registration with acceptable accuracy and improves the efficiency of registration. The Hama computing cluster we built is implementation simplicity and ease of point cloud processing. The improvement thought and implementation method can extend to other point clouds processing like filters, rendering, modeling and so on.

Keywords-Point cloud; Registration; Parallel computing; BSP model; Hama

I. INTRODUCTION

As ground objects have complex geometry and cover large area, normally point cloud data obtained by laser 3D scanner need multi-angle scanning and registrations to represent whole surface of an object. Registrations of multi-angle point cloud involve enormous point cloud data, complicated and heavy computation. Seeking efficient registration processing method is one of the important applied researches in the 3D point cloud.

In recent years, point cloud data processing technology researches, mainly focus on improving the efficiency of point cloud data processing, including representative results such as papers [1] ~ [10]. These results are of value and reference to improving efficiency of point cloud data registration processing. However, these studies did not give a feasible and practical parallel computing method of multi-angle scanning point cloud data registration processing. Paper [1] [2] [3] [4] discussed improvements of two point clouds registration computing. Paper [5] [6] [7] [8] studied the registration computational problems of multi-angle scanning point cloud in a non-parallel processing mode. Paper [9] [10] mentioned the use of parallel computing methods to solve multi-angle scanning point cloud global registration, but the proposed method is slightly complicated, demanding of computing resources.

The Bulk Synchronous Parallel Computing (BSP) Model was developed by Leslie Valiant and Bill McColl, intend to build an extendable parallel computing theoretical model independent of specific architectures [11]. BSP model computation proceeds in a series of global supersteps, which separate communication and synchronization. This method makes parallel program structure more clearly and significantly lowers the difficulty of coding and debugging. Parallel computing systems like Pregel [12] from Google and Hama [13] from Apache both are efficient implementations of BSP model.

We will design a point cloud registration parallel algorithm according to BSP technology thought. We will build a parallel cluster composed of average PCs, to solve the problem that general computer has difficulty processing multi-angle point cloud registration for increasing registration efficiency.

II. ANALYSIS OF REGISTRATION PROCESSING

The registration of two point clouds uses iterative closest point (ICP) algorithm [14], which mainly includes searching for correspondences, estimating rigid transformation and updating source point cloud. For multi-angle point cloud, the method of two point clouds registration has difficulty to fulfill global registration. Global registration with this method, selects two related point clouds for registration, then pick another related to the last registration result point cloud to registration sequentially until all point clouds register into a same coordinates.

On the assumption that an object has q multi-angle point clouds $P_1, P_2, \ldots, P_q, (q > 2)$. In these point clouds, there are two related point clouds $P_i, P_j (1 \le i, j \le s)$ and the threshold value of mean square error is τ.

Step 1: Select two related point clouds $P_i, P_j (1 \le i, j \le s)$ and set $P_i = [P_{i1}, P_{i2}, \ldots, P_{im}]$ to be source cloud M. The other point cloud $P_j = [P_{j1}, P_{j2}, \ldots, P_{jn}]$ is target cloud N. Through k $(k > 0)$ times iteration, source cloud M updates to M_k.

Step 2: Utilize the points in k times updated source cloud M_k, search for correspondences in target cloud N, and pair them into correspondence point set $< M_k, N_{k\min} >$.

Step 3: According to the correspondence point set $< M_k, N_{k\min} >$, calculate rigid transformation matrix $H_k = \begin{pmatrix} U_k & V_k \\ 0 & 1 \end{pmatrix}$ and mean square error $e(H_k)$.

Step 4: Utilize rigid transformation matrix H_k to update source cloud M_k, get $M_{k+1} = H_k M_k$.

Step 5: Compare the error with No. $k-1$ registration. If $|e(H_k) - e(H_{k+1})| \le \tau$, this round registration finished, go to Step 6, otherwise $k = k+1$, go to Step 2.

Step 6: If all point clouds have registered, global registration accomplished, output the result. If not, set this round result as target cloud, pick an unregistered point cloud as source cloud according to the registration relationship, then go to Step 1 for next round registration.

This method of point clouds gradual registration, is time-consuming and has complicated, and heavy computation, especially there are more point clouds involve registration. To improve registration computation efficiency, based on parallel computing model, the improvements of registration as follow are made.

Multi-angle scanning point cloud registration will be calculated in parallel computing cluster consisted of several computing nodes. Each point cloud is assigned to a point cloud computing nodes for processing. All registration calculations will be divided into local and global two types of calculations. Local computation involves only one point cloud, completed through the computing node assigned to this point cloud. Global calculation takes into account the relationship between the point clouds. In global calculation progress, information and results of local computation phase process are exchanged to each node.

According to the technology thought of BSP, process of multi-angle point clouds global registration is divided into for global barrier synchronous supersteps. Each superstep consists of three components:

- Concurrent computation: all computing nodes finish the computing task.
- Communication: Send and receive messages between computing nodes.
- Barrier synchronization: detect the messages send and receive between computing nodes, ensure the messages send to the destination nodes.

III. Design of Registration Algorithm Base on BSP

On the assumption that there are q unregistered point clouds $P_1, P_2, \cdots, P_q, (q > 1)$. One of these clouds is target cloud $P_t, t \in \{1, 2, \cdots, q\}$ corresponding to target computing node U_t. The rest $q-1$ clouds are source clouds $P_s = \{P_1, P_2, \cdots, P_{t-1}, P_{t+1}, \cdots, P_q\}$ corresponding to source computing nodes $U_s = \{U_1, U_2, \cdots, U_{t-1}, U_{t+1}, \cdots, U_q\}$.

Superstep1

Concurrent computation 1-1: All computing nodes calculate keypoint sets $K_1, K_2, \cdots K_q$ base on SIFT (Scale Invariant Feature Transform) algorithm [15]. Then calculate keypoint feature descriptor sets $F_1, F_2, \cdots, F_q$ base on PFH (Point Feature Histograms) [16] [17]. Structure feature point sets $L_1, L_2, \cdots, L_q$ with keypoint sets and feature descriptor sets. Among $L_1, L_2, \cdots, L_q$, feature point sets $L_i = [K_i, F_i], i \in \{1, 2, \cdots, q\}$.

Communication 1-2: All computing nodes $U_1, U_2, \cdots U_q$ send feature point sets $L_1, L_2, \cdots L_q$ to source computing node $U_i \in U_s$.

Superstep2

Concurrent computation 2-1: Source computing node U_i calculates first correspondence sets of source cloud P_i with rest point clouds $P_1, P_2, \cdots, P_{i-1}, P_{i+1}, \cdots, P_q$. The correspondence sets are $C_i = \{C_{i1}, \cdots, C_{i(i-1)}, C_{i(i+1)}, \cdots, C_{iq}\}$.

Communication 2-2: Source node U_i sends the contained pairs number $n_{im} = num(C_{im})$ and relationships r_{im} of first correspondence sets $C_{im} \in C_i$ to the target computing node U_t.

Superstep3

Concurrent computation 3-1: After target node U_t received n_{im}, r_{im}, calculate topology relationship R centers on target cloud P_t, according to rules of “each point cloud has a related point cloud” and “two point clouds with registration relationship have the most correspondence pairs possible”.

Communication 3-2: Target node U_t send topology relationship R to each source node.

Superstep4

Concurrent computation 4-1: Source node U_i dumps non-related correspondence sets, gains correspondence sets $C_{ij}, i, j \in \{1, 2, \cdots, q\}, i \ne j$. The $C_{i,j}$ represent the correspondence sets of P_i with related P_j. According to C_{ij}, iteratively calculate local rigid transformation H_{ij}.

Communication 4-2: Source node U_i sends H_{ij} to all source nodes U_s.

Superstep5

Concurrent computation 5-1: If source cloud P_1 is not directly related to target cloud P_t, transmit local rigid transformation H_{ij} to global transformation H_{it} base on topology relationship R. Register each source cloud P_i to target cloud P_t with rigid transformation H_{it}.

Communication 5-2: Source node U_i sends final registration report to Target node U_t.

Flow diagram of multi-angle point clouds global registration algorithm base on BSP as shown in Fig .1.

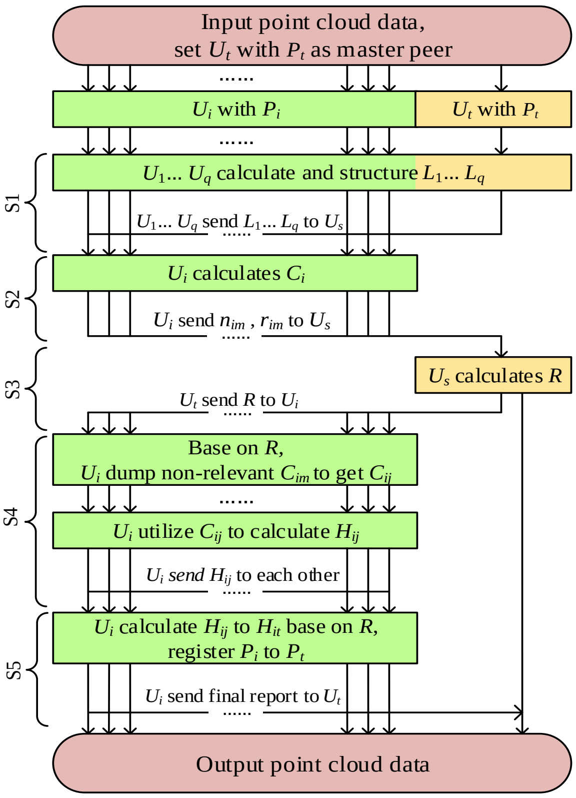

Figure 1. Flow diagram of global registration algorithm base on BSP

IV. Computation Analysis

A. *building computing cluster*

The Hama cluster we built consists of 24 PCs and one server. The server contains 3.2GHz quad-core CPU, 4GB memory and 500GB hard drive. The server manages file system namespace, task assignment of all computing nodes and management of barrier synchronous. The PCs each contains 1.3GHz dual-core CPU, 2GB memory and 70GB hard drive. The PCs are used for point cloud data storage, processing and calculating.

The cluster set up in the LAN segment range from 192.168.1.100 to 192.168.1.124. Each computer installed Ubuntu 12.04 32bit operating system, Hadoop 1.2.1 and Hama 0.62. The programming language is C++. The structure chart of cluster as shown in Fig .2.

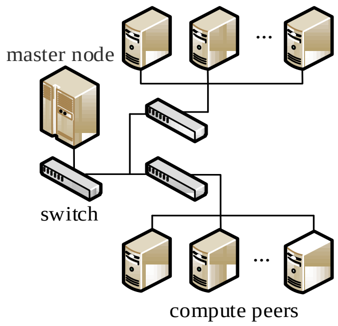

Figure 2. The structure of Hama computing cluster

B. *Computing result*

The example is from Guizhou province science and technology major special project “Research and application of fast stability assessment of expressways and high slopes base on the theory of digital graphics medium” (QKH GZ [2012]3017). The example contains point clouds data of one tunnel of Qingzhen. In Fig .3, P_1, P_2, P_3 are source clouds, respectively contain 282358, 366314, 226427 points. P_4 is target cloud contains 328785 points. After SuperStep1, global topology relationship tree shown in Fig .4. Through SuperStep2, 3 and 4, the output of the final registration shown in Fig .5 with consuming time 26 minutes in total.

Point cloud P_1 Point cloud P_2 Point cloud P_3 Point cloud P_4

Figure 3. Point cloud scanning data from a tunnel of Qingzhen

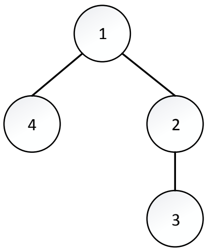

Figure 4. Topology relationship tree of multi-angle point clouds

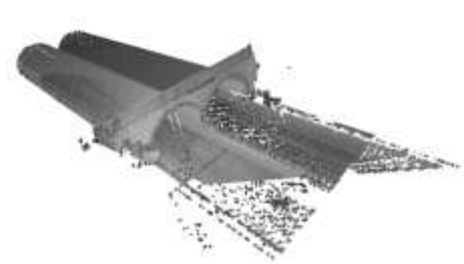

Figure 5. Registration result of a tunnel of Qingzhen

The following example is the construction site of Hailar Stadium in 2010. This example contains point clouds of 18 angles with about an average of 300,000 points in each point cloud.

The feature descriptors of P_8 and P_9 extracted by SuperStep2 has shown in Fig .6. The global topology relationship R has shown in Fig .7.

TABLE 1 shows the standard deviation of each registration result. Fig .8 and Fig .9 show the final registration result with/without colors.

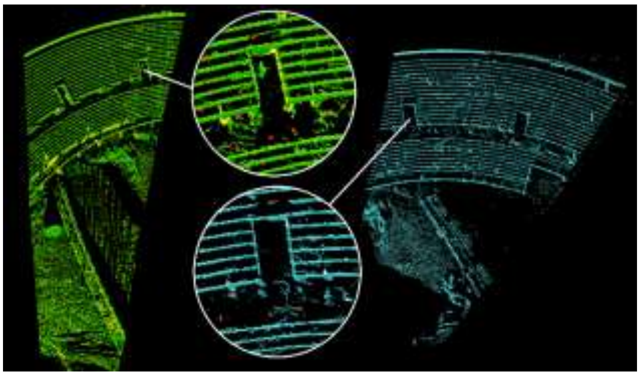

Figure 6. Feature descriptors of P_8 and P_9

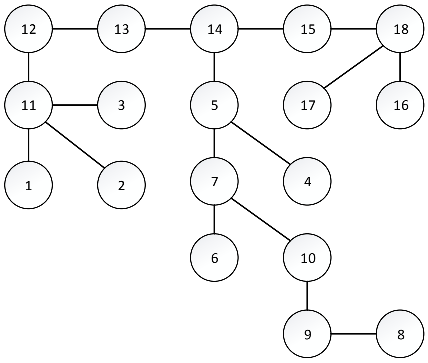

Figure 7. Global topology relationship R

TABLE I THE STANDARD DEVIATION OF EACH REGISTRATION RESULT

R	***StdDev***	***R***	***StdDev***
P_1,P_{11}	0.030632	P_9,P_{10}	0.012505
P_2,P_{11}	0.045736	P_{11},P_{12}	0.016032
P_3,P_{11}	0.04924	P_{12},P_{13}	0.028185
P_4,P_5	0.068198	P_{13},P_{14}	0.061181
P_5,P_7	0.034104	P_{14},P_{15}	0.039218
P_5,P_{14}	0.201595	P_{15},P_{18}	0.029727
P_6,P_7	0.09104	P_{16},P_{18}	0.030448
P_7,P_{10}	0.022	P_{17},P_{18}	0.018242
P_8,P_9	0.00717		

Figure 8. Final registration result in different colors

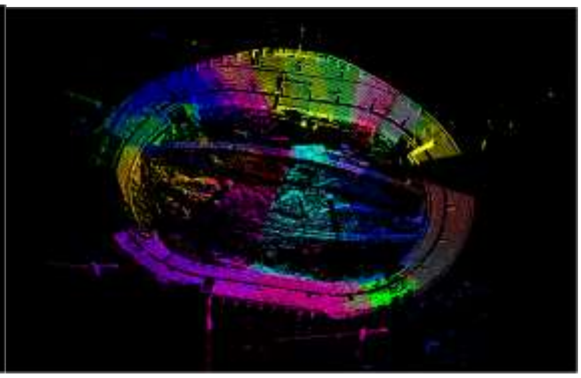

Figure 9. Final registration result of Hailar Stadium

In addition, as shown in TABLE 2, examples' results of Jiaxiulou historic building and Hongfeng Bridge have shown that, multi-angle scanning point clouds are precisely registered by global registration algorithm base on BSP on Hama computing cluster.

TABLE II THE RESULTS OF MULTI-ANGLE SCANNING POINT CLOUDS EXAMPLES

Project cases	Number of point clouds	Number of points in total	Consuming time	StdDev
Tunnel of Qingzhen	4	1203884	26 mins	0.042971
Jiaxiulou historic building	7	869901	34 mins	0.084278
Hongfeng Bridge	5	2142718	58 mins	0.106539

V. CONCLUSION

Point clouds registration algorithm base on BSP, has the feature of parallel computing, can utilize average PCs to accomplish global registration and reduces the requirements of computing resources. Several examples show that, under the premise in registration accuracy, the algorithm can improve the efficiency of registration. In addition, the Hama computing cluster we built is implementation simplicity and ease of point cloud processing. The improvement thought and implementation method can extend to other point clouds processing like filters, rendering, modeling and so on.

ACKNOWLEDGEMENT

Guizhou province science and technology major special project “Research and application of fast stability assessment of expressways and high slopes base on the theory of digital graphics medium” (QKH GZ [2012]3017). Guizhou province informationalization special foundation “Research on parallel realistic real-time rendering technology of massive scale point cloud model” (QKH GZ [2012] 1158).

REFERENCES

[1] Torre-Ferrero, Carlos, et al. "3D point cloud registration based on a purpose-designed similarity measure." EURASIP Journal on Advances in Signal Processing 2012.1 (2012): pp1-15.

[2] Gressin, Adrien, et al. "Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge." ISPRS Journal of Photogrammetry and Remote Sensing 79 (2013): pp240-251.

[3] Lin, Hongbin, Bin Liu, and Yucun Zhang. "Research on robust registration algorithm for point clouds based on kernel density estimation." Zhongguo Jixie Gongcheng(China Mechanical Engineering) 23.14 (2012): pp1726-1731.

[4] Weinmann, Martin, and Boris Jutzi. "Fast and accurate point cloud registration by exploiting inverse cumulative histograms (ICHs)." Urban Remote Sensing Event (JURSE), 2013 Joint. IEEE, 2013.

[5]Han, J-Y., N-H. Perng, and H-J. Chen. "LiDAR Point Cloud Registration by Image Detection Technique." Geoscience and Remote Sensing Letters, IEEE10.4 (2013): pp746-750.

[6] Liu, Jun, Guohua Geng, and Mingquan Zhou. "Sequence Iterative 3D Registration of Multi-view Point Cloud for Cultural Site Scenes." Computer Science & Service System (CSSS), 2012 International Conference on. IEEE, 2012.

[7] Xiao, Junhao, Bejamin Adler, and Houxiang Zhang. "3D point cloud registration based on planar surfaces." Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on. IEEE, 2012.

[8] Chaudhury, Kunal N., et al. "Global registration of multiple point clouds using semidefinite programming." arXiv preprint arXiv:pp1306.5226 (2013).

[9] Martinez, Jorge L., et al. "Using multicore processors to parallelize 3D point cloud registration with the Coarse Binary Cubes method." Mechatronics (ICM), 2013 IEEE International Conference on. IEEE, 2013.

[10] Wei, Qi Ming, and Wei Yong Wu. "Unorganized 3-D Points Data Registration with Local Geometric Feature Based on CUDA." Advanced Materials Research433 (2012): pp4725-4729.

[11] Valiant, Leslie G. "A bridging model for parallel computation." Communications of the ACM 33.8 (1990): pp103-111.

[12] Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

[13] Seo, Sangwon, et al. "Hama: An efficient matrix computation with the mapreduce framework." Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on. IEEE, 2010.

[14] Besl, Paul J., and Neil D. McKay. "Method for registration of 3-D shapes." Robotics-DL tentative. International Society for Optics and Photonics, 1992.

[15] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): pp91-110.

[16] R.B. Rusu, N. Blodow, M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, May 12-17, 2009.

[17] R.B. Rusu, A. Holzbach, N. Blodow, M. Beetz. Fast Geometric Point Labeling using Conditional Random Fields. In Proceedings of the 22nd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), St. Louis, MO, USA, October 11-15, 2009.