

A Parallel Delaunay Algorithm Applied in Lunar Rover Stereo Vision System

Wang Zhe

Beijing Institute of Tracking and Telecommunication
Technology
Beijing, China
wzhe06@gmail.com

Gao Sanhong

Beijing Institute of Tracking and Telecommunication
Technology
Beijing, China
gsh@sina.vip.com

Li Lichun

Beijing Aerospace Control Center
Beijing, China
lichunmail@163.com

Abstract—A parallel Delaunay algorithm is implemented to accelerate terrain feature points triangulation procedure in lunar rover stereo vision system. The algorithm uses divide-and-conquer approach and is optimized by employing a new strategy of the merge step. Furthermore, we parallel the algorithm using OpenMP library on a multi-core computer and compare the parallel Delaunay algorithm with traditional one using real terrain feature point data and simulation data which size ranges from 100,000 terrain points to 1,000,000. This paper presents design and structure of the parallel Delaunay algorithm and analyzes its performance.

Keywords—Delaunay; parallel; triangulation; OpenMP; lunar rover; stereo vision

I. INTRODUCTION

In the second stage of Chinese lunar exploration program, Chang'e-3 lunar rover will land on the moon and then execute a series of exploration tasks. The lunar rover must know surrounding terrain in real time to complete navigation and exploration tasks in unstructured and strange lunar surface environment. In the procedure, binocular stereo vision cameras installed on the lunar rover collect stereo image pairs of surrounding environment and then transmit them to ground station to reconstruct lunar 3D terrain environment.

112 stereo image pairs are collected at one site and terrain data increases rapidly as the rover explore more unknown area. Large data size means it needs more time to generate visual environment from the stereo image pairs, on the other hand, real-time performance is a key factor of lunar rover teleoperation system because teleoperators must know current environment state of lunar rover as soon as possible and make corresponding command. However, processing time of traditional sequential system rapidly increases in the complexity of $O(N^2)$ as terrain data accumulating. In this paper, we designs a new Delaunay algorithm called Parallel Fast Merge Delaunay Algorithm by optimizing merge step of traditional divide and conquer approach.

II. RELATED WORK

A. Architecture of Lunar Rover Stereo Vision Terrain Reconstruction System

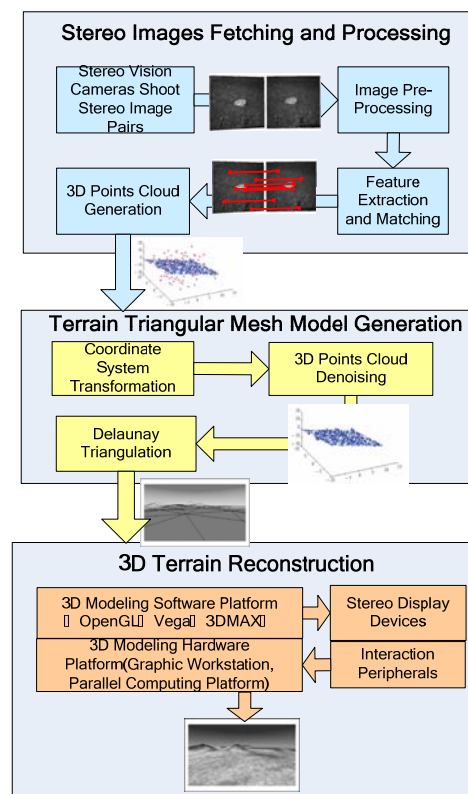


Figure 1. Architecture of Lunar Rover Stereo Vision Terrain Reconstruction System

The parallel fast merge Delaunay algorithm is designed in this paper to accelerate virtual terrain generation speed in lunar rover stereo system. In facts, many other kinds of Delaunay algorithms have been successfully applied in many projects such as American MER (Mars Exploration Rover)^[1], Planetary Surface Exploration Project of ESA^[2], as well as

recently launched Curiosity Mars Rover^[3]. As new wave of lunar and Mars exploration rising, many institutes and universities conduct research on stereo vision 3D terrain reconstruction project including CMU (Carnegie Mellon University), Tsinghua University, HIT (Harbin Institute of Technology) etc^{[5][4][5]}.

Based on Marr vision theory, classic stereo vision system consists of five parts systems, camera calibration, feature extraction, matching, triangulation and visualization. System designed in this paper is exactly based on Marr vision framework. Figure 1 shows architecture of the system. Delaunay algorithm is the key part of the second stage which generates terrain triangular mesh making use of 3D points cloud generated in the first stage.

B. Delaunay Triangulation Algorithm

To get virtual 3D terrain, discrete feature points should be organized into DEM (digital elevation model). For irregular points cloud, Delaunay triangular mesh is the “best” and appropriate DEM. There has been significant previous work on Delaunay algorithms^[6], while two classic approaches are described as follows.

The first approach for Delaunay meshes involves incremental insertion (using the Bowyer-Watson kernel^{[7][8]}). The first step is to find a polygon encircling all points and to build initial triangular mesh in it, then insert other points in to the mesh one by one. Use LOP (Local Optimization Procedure) algorithm to ensure the mesh is Delaunay mesh in the insertion procedure.

The second approach is that of divide-and-conquer. The mesh is (recursively) partitioned in two regions, with each partition triangulated by itself. The border between the regions is constructed separately then joint two adjacent regions when they finish triangulation itself until all regions are merged together. Aggarwal^[9] described a triangulation algorithm which constructed the border by joining the regions after they were built. Chen^[10] described a 2D algorithm which assigned certain points to both regions. The key of divide-and-conquer method is joint procedure of two regions which is research focus of previous works.

Complexity of sequential Delaunay algorithm ranges from $O(N \log N)$ to $O(N^2)$, while it means big time consuming for large data size thus the procedure usually become a time bottleneck of the stereo vision terrain reconstruction system. In this paper, we realize parallel Delaunay algorithm based on divide-and-conquer approach using OpenMP framework on a multi-core shared-memory machine to break the bottleneck of triangulation procedure.

C. OpenMP

OpenMP is a parallel programming model for shared memory pioneered by SGI and developed in collaboration with other parallel computer vendors. OpenMP is fast becoming the de facto standard for parallelizing applications^[11]. OpenMP programs are presented in Fortran, C, and C++ and support Windows, Linux and Unix operating system. Shared memory machine based OpenMP framework is adopted in the system because latency of inter processors communication can be avoided compared with distributed

memory system and OpenMP has already been proven a mature and efficiency parallel computing framework.

III. PARALLEL DELAUNAY ALGORITHM

Delaunay algorithm is the most important one in the second stage of the terrain reconstruction system. Divide-and-Conquer approach described in section II.B is one of classic Delaunay algorithms and appropriate to realize parallelization so it is chosen as the algorithm framework and optimized by improving merge step. The following sections successively describe algorithm framework, fast merge strategy, LOP procedure and parallel Delaunay algorithm on OpenMP computing platform.

A. Algorithm Framework

Algorithm framework is designed based on classic Divide-and-Conquer method: recursively divide points cloud into two partitions according to point space position while boundary condition is that a sub partition has only one point, then merge sub regions bottom up following merge strategy until all points are included in the mesh. The optimization step guarantees the mesh is Delaunay triangular one.

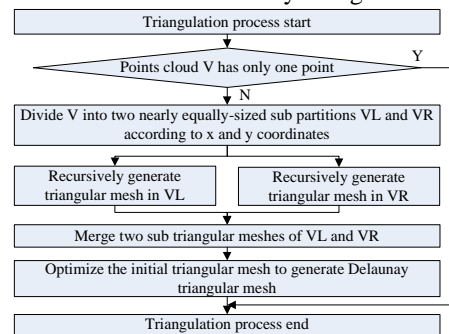


Figure 2. Parallel Delaunay Algorithm Framework

B. Fast Merge Strategie

Merging sub triangular meshes is the most complex step in the algorithm and it consumes more than 60% time in triangulation algorithm so optimization of this step is critical. To merge two meshes, it needs to find common tangents of the tow meshes firstly, and then divide the concave polygon surrounded by the two meshes and common tangents. The two steps is separated in classic merge strategy, while a fast merge strategy which combine the two steps is designed as follows that can save 25%-30% time of merge procedure.

Figure.3 (a) describes algorithm flow and Figure.3 (b) takes a simple point set as example. L and R represent left sub mesh and right sub mesh which have already finished its triangulation procedure. Bounds of the two meshes are stored as double linked list so it is easily to find right-most point of left bound and left-most point of right bound. Link the two points and start to divide the space between two meshes towards both up and down direction until find the up common tangent and down common tangent. The fast merge algorithm ends at same time of finding common tangents.

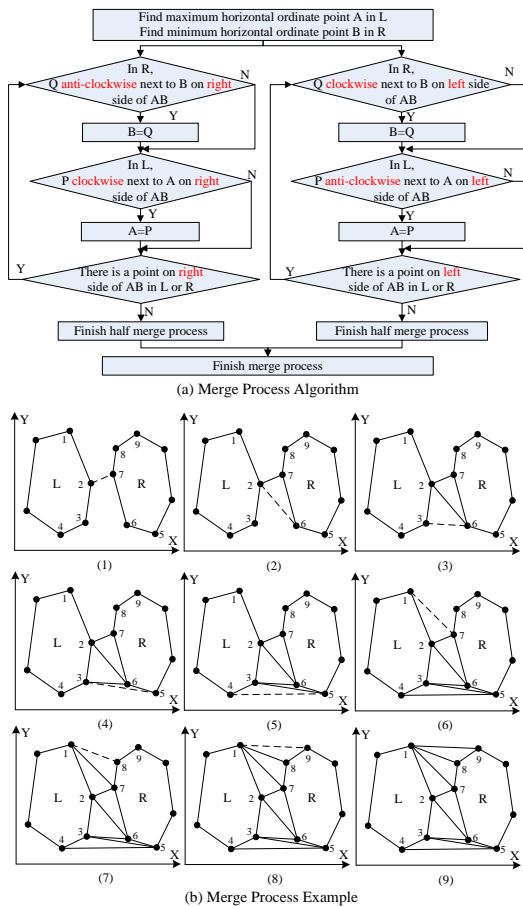


Figure 3. Fast Merge Algorithm and Example

C. LOP Procedure

Initial merged triangular mesh is not Delaunay triangular one, so the mesh should be optimized by LOP (Local Optimal Procedure) algorithm and finally form Delaunay triangular mesh. The procedure applies empty circumscribed circle property of Delaunay triangular mesh: check quadrilateral formed by two triangles which share common edge, if circumscribed circle of one triangle contains the fourth vertex, then exchange two diagonals of the quadrilateral. Recursively check quadrilaterals in the mesh until all quadrilaterals meet empty circumscribed circle property. The algorithm is frequently used in triangulation procedure, while the algorithm is also redesigned in the system using angle relation to avoid complex evolution operation.

D. Parallel Algorithm on OpenMP Computing Platform

Divide and Conquer framework described in section III.A decides the algorithm is extraordinary appropriate for parallel computing platform. The whole points cloud is divided into several regions according to the number of processors of the platform then assign each region to each

processor. Each processor finishes the triangulation procedure of its own data then recursively merges two triangular meshes of adjacent processors. OpenMP is actually a C++ library, we include the library to modify sequential fast merge Delaunay algorithm. However, when the mesh is very small compared to the number of processors operating on it, there is relatively big time waste of thread establish, communication, switch and destroy compared with multi-core compute time saving. So the program divides points cloud into the number of regions corresponding to the number of processors to guarantee the highest speedup of the system.

IV. EXPERIMENTATION

A. Dataset and Experiment Environment

There are 10 stereo image pairs collected from Chinese lunar terrain simulation field for experimentation. The simulation field simulates terrain, illumination and soil condition of lunar surface. The stereo vision system extracts 5,120 feature points to generate feature points cloud. To test the performance of the system in several different size of dataset, we random generates datasets range from 100,000 terrain points to 1,000,000 points according to real experiment data collected from Chinese lunar terrain simulation ground.

The hardware environment used for experiment is a 4-core computer, Lenovo M8200t with 3.2-GHz Intel i5 CPU (4 cores), 2.0 Gb RAM. The operating system is Windows Server 2008. Our code is written in C++ with OpenMP library to provide parallel functionality.

B. Delaunay Triangular Mesh Precision

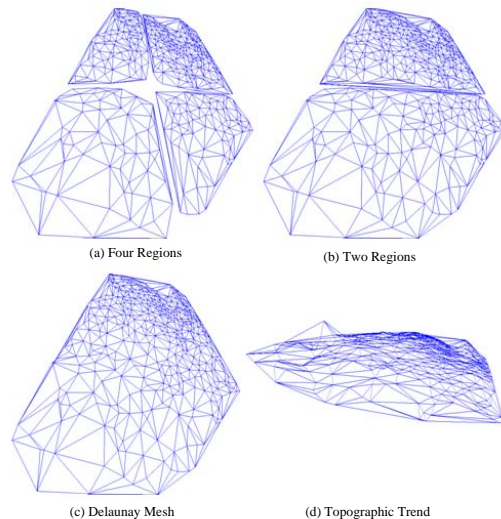


Figure 4. Parallel Delaunay Triangulation Procedure Using 4 Processors

Figure 4 shows a true triangulation procedure which the points cloud is divided into 4 regions and 4 processors complete parallel process. Terrain data is one part of

simulation field data which contains 512 feature points. The final Delaunay triangular mesh is perfect correct which totally conform to Delaunay attributes. Experiments in large dataset also show that precision of Delaunay triangulation procedure reaches 100%.

C. Performance of Parallel Fast Merge Delaunay Algorithm

We run Parallel Fast Merge Delaunay Algorithm (PFMDA) on different datasets and take Sequential Fast Merge Delaunay Algorithm (SFMDA), Traditional Divide and Conquer Delaunay Algorithm (TDCDA) and Traditional Static Delaunay Algorithm (TSDA) as comparisons. Different dataset 100,000, 200,000, 300,000, 400,000, 500,000, 600,000, 700,000, 800,000, 900,000, 1,000,000 is used for experiment and each dataset is tested 10 times. The results take average value of 10 experiments.

TABLE I. EXPERIMENT RESULTS

Dataset	TSDA(s)	TDCDA(s)	SFMDA(s)	PFMDA(s)
100,000	27.596	1.295	1.123	0.624
200,000	109.418	2.496	2.246	1.264
300,000	242.612	3.822	3.370	1.809
400,000	430.951	5.039	4.508	2.418
500,000	710.421	6.349	5.616	3.042
600,000	1010.231	7.550	6.723	3.588
700,000	1352.421	8.704	7.893	4.196
800,000	1727.346	10.515	9.110	4.836
900,000	2135.462	11.419	10.172	5.429
1,000,000	2678.275	13.089	11.216	6.053

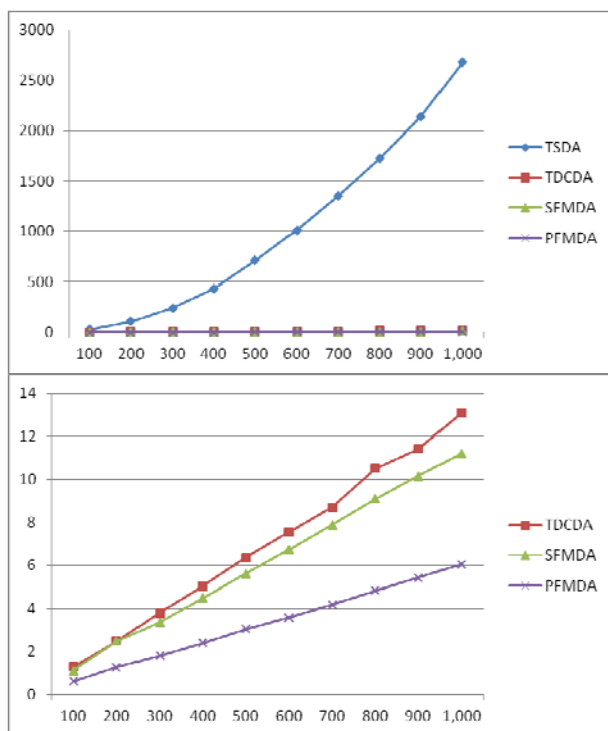


Figure 5. Comparison of Different Algorithms

The result shows speedup of TDCDA is 20.4 to 204.6, speedup of SFMDA is 1.132, and speedup of PFMDA is 1.832.

V. CONCLUSION

This paper designs a lunar rover stereo vision system and applies it on an elastic parallel computing platform to break the bottleneck of 3D terrain reconstruction procedure. Experiment data is from lunar terrain simulation ground and computer simulation. Parallel Delaunay algorithm running on elastic parallel computing platform has the best performance compared with sequential one and traditional parallel system.

REFERENCE

- [1] Jet Propulsion Laboratory. Mars Exploration Rover [J].NASA Facts, 2005.
- [2] Vergauwen,M, M.Pollefeys, L.Gool. A Stereo Vision System for Support of Planetary Surface Exploration [J]. Machine Vision and Applications, 2003, 14(1):5-14.
- [3] Volpe R. Rover Technology Development and Mission Infusion beyond MER[C]. IEEEAC, January 23, 2005.
- [4] Singh S, B.Digney. Autonomous Cross-Country Navigation Using Stereo Vision[C].CMU-RI-TR-99 -03, 1999:1-74.
- [5] Maurette M. Mars Rover Autonomous Navigation [J]. Autonomous Robots, 2003, 14(2/3):199-208.
- [6] Daniel K.B, E.B. Guy, C. Kadow. Engineering a Compact Parallel Delaunay Algorithm in 3D [C]. SCG'06, Sedona, Arizona, USA, June 5-7, 2006.
- [7] T. Okusanya and J. Peraire. 3D parallel unstructured mesh generation[C]. AMD, 220:109–115, 1997.
- [8] N. Chrisochoides and F. Sukup. Task parallel implementation of the Bowyer-Watson algorithm[C]. In Proceedings of the Fifth International Conference on Numerical Grid Generation in Computational Fluid Dynamic and Related Fields, 1996.
- [9] A. Aggarwal, B. Chazelle, L. Guibas, C. O'Dunlaig, and C. Yap[J]. Parallel computational geometry. Algorithmica, 3:293–327, 1998.
- [10] M. Chen, T. Chuang, and J. Wu. Efficient parallel implementations of 2D Delaunay triangulation with high performance Fortran[C]. In Proceedings of 10th SIAM Conference on Parallel Processing for Scientific Computing, 2001.
- [11] H.T. Ma, Research on Analysis and Optimization for OpenMP Program[D]. PLA Information and Engineering University, 2009