BASED ON GPU FOR STRIP ADJUSTMENT ALGORITHM OF LIDAR DATA

Bo Song[1], Guoqing Zhou[1], Yanling Lu[1], Xiang Zhou [1,2,*], Peng Liang[1]

[1] Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin University of Technology, No. 12 Jian'gan Road, Guilin, Guangxi 541004, China
[2] College of Mechanical and Control Engineering, Guilin University of Technology, No. 12 Jian'gan Road, Guilin, Guangxi541004, China

KEY WORDS: GPU, strip adjustment, Iterative Closest Point, Octree, Hashing function

ABSTRACT:

In order to solve the problem that the source of LiDAR data error needs to be adjusted and the data volume is large, the adjustment speed between the voyages is slow and cannot be automatically adjusted. Based on the iterative nearest point (ICP) algorithm, this paper proposes an improved iterative closest point (ICP) algorithm based on GPU parallel octree. The algorithm quickly constructs the octree of LiDAR nautical belt data in the GPU, uses the octree to quickly find the overlapping area of the nautical band, and then uses the ICP algorithm in the overlapping area to solve the adjustment parameters R and T quickly. Then the entire flight belt is quickly adjusted. Experiments with example data show that this method can quickly and automatically adjustment a large number of LiDAR data, and the adjustment precision can meet the precision requirements of the production.

* Corresponding author: Xiang Zhou; Email: zqx0711@glut.edu.cn

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-3-W10-229-2020

1. INTRODUCTION

The airborne LiDAR system is a complex system consisting of Laser Range Finder (LRF), Inertial Measurement Unit (IMU), Global Positioning System (GPS) (Zhou, 2009; Zhou, 2014; Zhou, 2015). It is mainly used for rapid surveys of buildings, forestry, beaches, coastlines, islands, and seabed topography. It can directly map three-dimensional data of topography and landforms, and can generate high-precision topographic maps after simple processing. Our research team won an Innovative Development Grand Grant in Guangxi in 2018. The purpose of this special project is to develop an airborne laser high precision three-dimensional seabed measuring instrument (LiDAR). The principle structure of the instrument is shown in Figure 1.

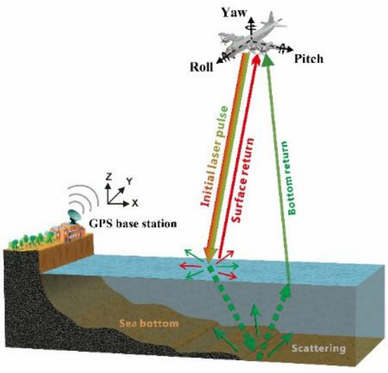

Figure 1. Dual LiDAR working diagram

The integration between different systems inevitably leads to systematic errors. Even after good calibration, there may still be residual systematic errors in the airborne LiDAR system (Zhou, 2018). When these errors are finally reflected in the point cloud coordinates acquired by different strips, the coordinates of the same-named points calculated by different stripe differences will be different. The general algorithm is to use the same-name feature of the strip overlap region or the ground control point data to establish the observation equation, and eliminate the influence of the error by the band adjustment. Strip adjustment (Liying W, 2013) is to adjust the strip deformation by using the difference of the same name between the overlapping areas of the belt. In the adjustment system, the strip deformation parameters must be defined first, and the deformation parameter value is obtained by the least squares adjustment theory. Adjust the point cloud coordinates. Domestic and foreign scholars ((Kilian J, 1996) and (Vosselman G, 2001)) have carried out a lot of research on strip adjustment. They are mainly divided into three categories: feature points, feature lines and feature surfaces; from the dimension of adjustment, they are mainly divided into two categories: One-dimensional, three-dimensional.

The purpose of the strip adjustment is to eliminate the systematic offset between the overlapping regions by the adjustment to generate a seamless product, and the elimination of the systematic offset between the strips means the coincidence of features between the strips. Therefore, the strip adjustment can be regarded as an adjustment problem. In view of the fact that the main purpose of the strip adjustment is to improve the relative accuracy of the data between different strips rather than the absolute precision, the strip adjustment of the airborne LiDAR data can be realized by the uncontrolled three-bit surface adjusting technique. Among the three-dimensional surface adjusting algorithms, the most widely used algorithm is the Iterative Closest Point (ICP) method proposed by Besl and Mackay in 1992 (Besl P J, 1992). The essence of the ICP algorithm is to find the optimal rigid body transformation matrix by using the least squares principle based on the principle that the distance from the nearest point of the three-dimensional surface is the closest to the nearest square. There are three limitations to this algorithm: (1) Requiring two adjusted points to focus on one point set is a subset of another point set; (2) At the same time, in order to avoid falling into local optimum, the real position of the two point sets is not much different; (3) When dealing with massive point clouds, the algorithm is inefficient and cannot be applied. For these three points, the characteristics of the onboard LiDAR data itself have already met the second point. The position of the point cloud in the overlapping area is not much different, only the other two points need to be solved. This paper proposes an improved iterative nearest point algorithm based on GPU parallel octree. For the first point, the LiDAR point cloud is encoded by the octree, and the overlapping regions of different strips are searched, and the subset relationship of different stripe point clouds is ensured in the overlapping area; For the third point, this paper introduces GPU parallel computing (Zhou K, 2010; Jiqiang W, 2018) to quickly process massive point cloud.

2. ALGORITHM

According to the requirements of the ICP algorithm for the initial position of the point cloud and the large amount of LiDAR point cloud data. In this paper, the minimum bounding box is constructed for the LiDAR data of two adjacent strips, and two octrees are constructed in parallel in the GPU. A hash function is introduced in the process of construction to implement coding management of point cloud data. Two overlapping regions of the point cloud are obtained, and then the adjustment is performed using the improved ICP algorithm in the overlapping region. The algorithm flow is shown in Figure 2. The specific process is as follows：

(1) The two bounding strips to be adjusted are constructed with a minimum bounding box.

(2) Construct the octree separately, and introduce a hash function to encode the point cloud data during the construction process.

(3) The intersection of the two strips is obtained by the intersection operation of the octree, and the corresponding point pairs are searched by the ICP algorithm in the overlap region.

(4) The error point pair is removed by using the threshold of the angle of the direction vector between the corresponding point pairs.

(5) Solving the adjustment parameters R and T using the quaternion method.

(6) Using the adjustment parameter to adjustment between the two strips, obtaining the adjustment error, and judging whether the threshold requirement is met, if it is satisfied, the iteration is terminated; otherwise, proceeding to step 4, continuing the iteration.

(7) The final solved adjustment parameters R and T are applied to the strip data to complete the adjustment.

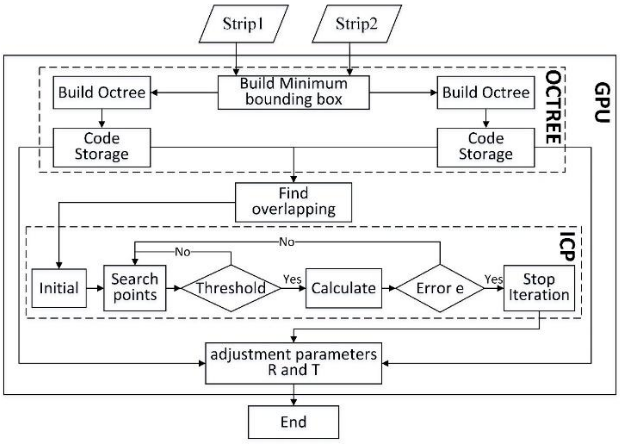

Figure 2. The process of algorithm

2.1 Find Overlapping Areas

Dr Hunter first proposed a quadtree structure in 1978 (Hunter G M, 1978). The octree structure is an extension of the quad-tree structure in three dimensions. Due to its spatial segmentation, adaptability to set operations and spatial query, it has great advantages in point cloud management and can be used for spatial change detection between multiple point clouds. The octree structure can be divided into 2n×2n×2n by cyclic recursion. The geometric entities after division often have the following characteristics: (1) Each subspace has a minimum division unit of equal size; (2) The position of each leaf node in the octree is known. We only need to encode the nodes to quickly access any leaf node in the octree structure. This article builds the octree in parallel directly in the GPU. First, the minimum bounding box is constructed for the point cloud, and then the hash function is introduced in the process of tree construction, which can greatly save the memory and traversal consumption. This way no longer stores the parent and child node pointers, but instead stores a unique index on each node, called position encoding. Each octant in this position-coded octree will get a number from 0-7, which is 3 bits in size. The digital code depends mainly on the relative position of its parent node. The octree code is shown in Figure 3.

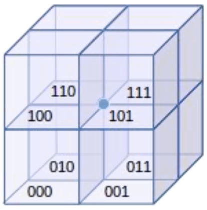

Figure 3. Octal coding

Therefore, the position coding of each tree seed node can be connected through a top-down recursive relationship, and all nodes' octant positions are coded to express their position in the octree. In order to obtain the depth of the tree in which the node is located according to the position coding, it is also necessary to set a flag bit to indicate the end of the position coding. Without such a flag bit, it is difficult to distinguish between 001 and 0000001. By using 1 bit to mark the end of the sequence, 1001 can easily distinguish 10001. A fast search of the point cloud can be achieved based on the uniqueness of the position coding of each node and the coordinate information of each point in the point cloud. Secondly, the unique node position coding and bounding box can also be used to quickly detect the overlapping area between the two clouds. The overlapping area searching process is shown in Figure 4.

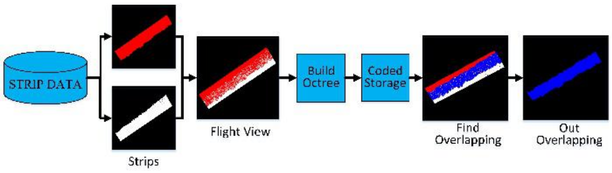

Figure 4. The process of find overlapping area

2.2 ICP Algorithm

The idea of ICP algorithm proposed by Besl and Mckay is to find the closest point in the model point cloud Q for the adjustment point cloud P (Besl P J, 1996; Zexiao X, 2010). According to the principle of least squares, the adjustment parameters are continuously iteratively obtained: the rotation matrix R and the translation parameter T, and the test index is to obtain the root mean square error RMS.

$$RMS = \sqrt{\frac{1}{n}\sum_{k=1}^{n} q_k - (Rq_k + T)^2} \tag{1}$$

When solving R and T, the main steps of solving the unit quaternion method are as follows:

Find the cross-covariance matrix S of the point set:

$$\left.\begin{aligned} S &= \frac{1}{n}\sum_{1}^{n}\left[\left(p_i - c_p\right)\left(q_i - c_q\right)\right] \\ c_p &= \frac{1}{n}\sum_{k=1}^{n} p_i \\ c_q &= \frac{1}{n}\sum_{k=1}^{n} q_i \end{aligned}\right\} \tag{2}$$

Where c_p、 c_q are the centroids of the point set, and n is the number of corresponding point pairs.

The symmetric matrix N is constructed by the cross-covariance matrix S:

$$\mathrm{N} = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & S_{yy} - S_{xx} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & S_{zz} - S_{yy} - S_{xx} \end{bmatrix} \quad (3)$$

Calculate the eigenvector corresponding to the largest eigenvalue of N: get R and T according to q_R:

$$\left.\begin{array}{ccc} & R = & \\ q_0^2 + q_x^2 + q_y^2 + q_z^2 & 2\left(q_x q_y - q_0 q_z\right) & 2\left(q_x q_z + q_0 q_y\right) \\ 2\left(q_y q_x + q_0 q_z\right) & q_0^2 - q_x^2 + q_y^2 - q_z^2 & 2\left(q_y q_z - q_0 q_x\right) \\ 2\left(q_z q_x - q_0 q_y\right) & 2\left(q_z q_y + q_0 q_x\right) & q_0^2 - q_x^2 - q_y^2 + q_z^2 \\ & T = c_q - R \cdot c_p & \end{array}\right\} (4)$$

The two strips can be adjusted using the calculated R and T.

2.3 Threshold of Direction Vector

The corresponding points obtained by the nearest point search still contain some error points, which seriously affects the accuracy of strip adjustment. In this paper, the fixed value of the angle of the direction vector of the corresponding point is selected according to this problem to judge whether it is the correct corresponding point, so as to improve the precision of strip adjustment.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-3-W10-229-2020 | © Authors 2020. CC BY 4.0 License.

a) normal vector estimation

For any point P in the point cloud, estimating the normal vector is equivalent to calculating the normal vector of the tangent plane that the point fits with its K neighbors. The domain covariance analysis method used in this paper is used to obtain the normal vector of points(Xin W, 2011). The principle is as follows:

$$C_V = \frac{1}{k}\sum_{i=1}^{k}(P_i - P_0)(P_i - P_0)^T \qquad (5)$$

Where P_0 is the centroid of the K neighborhood, and the eigenvector corresponding to the minimum eigenvalue of the covariance matrix C_V is the normal vector of the point P.

The normal vector we get here may be the opposite, we need to adjust it to the same direction. For this problem, you only need to direct all normal directly to the viewpoint.

b) direction vector angle

According to the above steps, the normal vector of each point has been obtained, and the normal vector of each point is unitized, and the angle of the normal vector between the corresponding points is calculated. In theory, each pair of corresponding points in the point cloud adjustment is correctly corresponding. In the point cloud after the initial adjustment in the experiment, there are still some error points between the corresponding points. We set a certain threshold t between the corresponding points to determine whether the point is the correct point. If the angle between them is less than t, then identify it as the correct point and keep it; On the other hand, if the angle between them is greater than t, it will be identified as the wrong point and deleted, in order to improve the precision of the adjusting.

3. EXPERIMENT AND ANALYSIS

This article uses C++ to use the PCL open source library for GPU programming in a Windows environment. The hardware environment is Intel Xeon W-2145CPU, memory is 64G, and NVIDIA GTX 1080Ti GPU with 11GB of memory. Partial LiDAR data scanned by Jyväskylä, a city in central Finland, acquired by Blom in 2011. The data area is about 500×2000 meters, the strip overlap is about 70%, the number of point clouds is about 19 million points, and the average scan point density is 17 points/square meter. The data is typically representative. The upper right part of the area is a forest-covered mountainous area, the lower right part is a flat urban area, and the lower part is a sea area. The types of land objects are also diverse, including buildings of different sizes, different heights of vegetation, Sea, all kinds of ports and so on. These are shown in Figure 5.

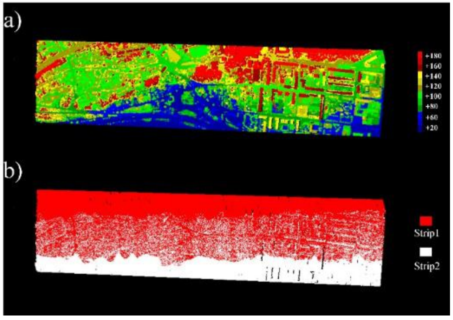

Figure 5. a) Data is colored and displayed by elevation.
b) Data is colored and displayed by flightline

The data is point cloud data that has been systematically overhauled, but there is still systematic error. Intercepting part of the section to see the local details before splicing is shown in Figure 6, as follows:

(1) There is a large elevation error on both sides of the strip, and the height difference at the edge of the strip can reach 0.5m.

(2) There is a relatively obvious horizontal offset between the strips, the east-west offset is close to 0.8, and the north-south shift is nearly 0.6m.

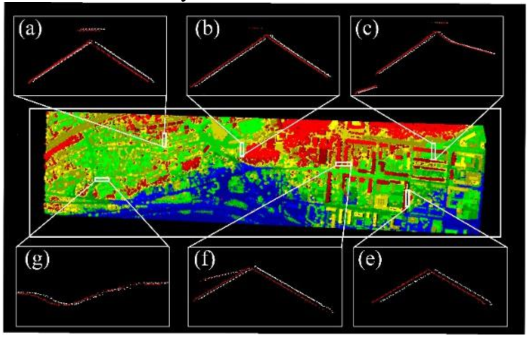

Figure 6. Detail view of the detail before adjustment

Experiments were carried out using the algorithm of this paper. The local detail effect of the experimental area is shown in Figure 7. The experimental results are shown in Table 1: The 3D mismatch is 0.01256, the difference in the Z direction is 0.014 m, and the RMS value is 0.0091. It can be seen from Fig. 7 that the relative offset of the same name between different strips is eliminated, and the point cloud coincides well.

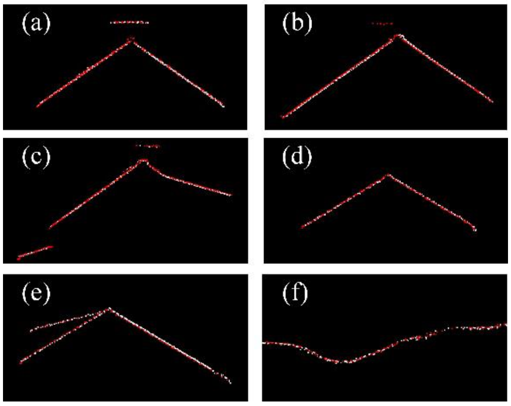

Figure 7. Local detail section after adjustment

In order to verify the accuracy of the algorithm, this data was used to perform experiments again using the feature line in the commercial software Terrasoild. The experimental results show that the 3D adjusting error is 0.01455, the difference in the Z direction is 0.016 m, and the RMS value is 0.020. The experimental results show that the algorithm results are better than the Terrasoild software. Terrasoild software requires a lot of manual intervention when looking for feature lines. The accuracy of this algorithm is comparable to that of Terrasoild software, even better, with reduced manual intervention.

Table 1 algorithm results comparison

	3D mismatch	Dz (m)	RMS
This paper	0.01256	0.014	0.0091
Terrasoild	0.01455	0.016	0.020

4. CONCLUSION

Strip adjustment is the elimination of LiDAR system error, and its accuracy directly affects the quality of the latter products. Its research is of great significance. Therefore, for the LiDAR data of massive point cloud, this paper introduces the ICP algorithm in the 3D surface adjusting algorithm and improves it. An improved iterative nearest point algorithm based on GPU parallel octree is proposed. The actual data is used to verify the results. The results show that the proposed algorithm can shorten the adjusting precision and save the adjusting time, which is suitable for LiDAR data adjusting. This algorithm only performs adjustment work for two strips. It needs repeated iteration for adjustment multiple strips, and the adjustment time will be greatly increased. This is also a limitation of the algorithm in this paper.

ACKNOWLEDGEMENT

This paper is financially supported by the National Natural Science of China under Grant numbers 41431179,41961065; Guangxi Innovative Development Grand Grant under the grant numbers GuikeAA18118038, GuikeAA18242048; the National Key Research and Development Program of China under Grant number 2016YFB0502501 and the BaGuiScholars program of of Guangxi (Guoqing Zhou).

REFERENCES

Besl P J， Mckay N D. 1992: A method for registration of 3-dshapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2): 239-256.

Guoqing Zhou, Xiang Zhou, Jiazhi Yang, Yue Tao, Xueqin Nong, Oktay Baysal, 2015: Flash LiDAR Sensor using Fiber Coupled APDs, *IEEE Sensor Journal*, 15(9), 758-4768.

Guoqing Zhou, Xiang Zhou, 2014: Seamless Fusion of LiDAR and Aerial Imagery for Building Extraction, *IEEE Transactions on Geoscience and Remote Sensing*, 52(11), 7393 – 7407.

Guoqing Zhou, Ming Xie, 2009: GIS-based Three-dimensional Morphologic Analysis of Assateague Island National Seashore from LIDAR Series Datasets, *Journal of Coastal Research*, vol. 25, no. 2, 435–447.

Zhou,G., & Zhou,X, 2018: *Technology and Applications for Array LiDAR Imager (Chinese)*. Wuhan Univ. Press, 978-7-307-19683-4.

Hunter G M, Steiglitz K. 1979: Operations on images using quad trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2): 145-153.

Jiqiang Wang, Shiyu Jia. 2018: Parallel octree generation algorithm based on GPU. *Journal of Qingdao University*, 31(04): 69-75.

Kilian J, Haala N, Englich M. 1996: Capture and evaluation of airborne laser scanner data. International Archives of Photogrammetry and Remote Sensing, 31: 383-388.

Liying Wang, Weidong Song. 2013: Comparison of Belt-to-Rate Adjustment of Airborne LiDAR Data and Adjustment of Strip Region Network. *Science of Surveying and Mapping*, 38(02), 32-34.

Vosselman G, Maas H. Adjustment and filtering of raw laser altimetry data. //Proceedings of OEEPE Workshop on Airborne Laser scanning and Interferometric SAR for Detailed Digital Terrain Models, Stockholm, Sweden. 2001.

Xin Wei, Pu Jiexin. 2011: Point cloud integration base on distances between points and their neighborhood centroids. *Journal of Inage and Grarhics*.16(5): 886-891.

Zexiao X, Xu Shang, Lixu Yong. 2010: A high-accuracy men-thod for fine registration of overlapping point clouds. *Image and Vision Computing*, 28(4): 563-570.

Zhou K, Gong M, Huang X, 2010: Data-parallel octrees for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 17(5): 669-681.