RAPID EXTRACTING PILLARS BY SLICING POINT CLOUDS

LUO De-an, WANG Yan-min

Dept. of Surveying and Mapping, Beijing University of Civil Engineering and Architecture, No.1 Zhanlanguan Road, Xicheng District, Beijing, PRC - (Luodean, Wangyanmin)@bucea.edu.cn

Commission III, WG III/3

KEY WORDS: Terrestrial Laser Scanner, Point Cloud , Cultural Heritage , Feature , Extraction

ABSTRACT:

Terrestrial Laser scanner (TLS) has been widely used in our recent architectural heritage projects and huge quantity of point cloud data was gotten. In order to process the huge quantity of point cloud data effectively and reconstruct their 3D models, more effective methods should be developed based on existing automatic or semiautomatic point cloud processing algorithms. Here introduce a new algorithm for rapid extracting the pillar features of Chinese ancient buildings from their point cloud data, the algorithm has the least human interaction in the data processing and is more efficient to extract pillars from point cloud data than existing feature extracting algorithms. With this algorithm we identify the pillar features by dividing the point cloud into slices firstly, and then get the projective parameters of pillar objects in selected slices, the next compare the local projective parameters in adjacent slices, the next combine them to get the global parameters of the pillars and at last reconstruct the 3d pillar models.

1. INTRODUCATION

The existing ancient buildings are our important cultural heritages and they represent our ancient civilization. Wooden structures are their most essential part in these buildings, and can be damaged easily due to natural disasters and time elapsing. In order to permanently preserve these cultural heritages, different technologies are employed to construct their digital archives. TLS has been widely used in our recent architectural heritage projects (e.g. the 3D reconstruction project for the Forbidden City in Beijing) and huge quantity of point cloud data was gotten. Till now point cloud data can be gotten easily and cheaply and how to reconstruct 3D models from these point clouds becomes a critical problem. In order to process the huge quantity of point cloud data effectively and reconstruct their 3D models, more effective algorithms should be developed based on existing automatic or semiautomatic point cloud processing algorithms.

Here introduce a new algorithm for rapid extracting the pillar features of Chinese ancient buildings from their point cloud data, and the algorithm has the least human interaction in the data process procedure and is more efficient to extract pillars from point cloud data than existing feature extracting algorithms. With this algorithm we identify the pillar features by dividing the point cloud in slices firstly, and then get the projective parameters of pillar objects in selected slices, the next compare the local projective parameters in adjacent slices and at last combine them to get the global parameters of the pillars. After getting the global parameters 3d pillar models can be reconstructed easily.

2. LITERATURE REVIEW

Due to the fact that the acquired raw data does not provide any structure or feature information, therefore the extraction of feature from 3D point clouds is an important topic in the field of reconstructing 3D models from point cloud.

The feature extraction (e.g. the plane, cylinder) is a common problem encountered in many areas of geometry related computer science. Over the years a vast number of algorithms have been proposed, of which only a few of them have the ability to automatic or semiautomatic detect the cylinder features (wooden pillars). They are the Hough Transform, the Random Sample Consensus (RANSAC) paradigm and the Template Matching.

The Hough Transform is a voting scheme that extracts a parameterized shape primitive from a discretized parameter space. It's useful and effective for extracting the features with a few parameters (e.g. the 3D plane). But for the cylinder which can be described by five parameters, it is very time and memory consuming and unreliable to detect the feature directly (Illingworth, 1998). To reduce the dimension of the parameter space, the cylinder detection can also be split into two parts: the detection of the cylinder axis direction (2 parameters) and the detection of a circle in a cylinder (3 parameters), and the improved algorithm leads to a higher efficiency than before. However, no matter how the original algorithm or its improved algorithm the computation expense is costly to extract cylinder features from the huge quantity of points.

The RANSAC paradigm was originally developed for robust fitting of a parametric model to data that may contain a high degree of noise and outliers.The robustness against noise and outliers renders the RANSAC paradigm as a suitable choice when performing shape detection on real-world scanned data, which often is of low quality. The RANSAC paradigm is effective for detect all kinds of basic geometry features (e.g. plane, cylinder, sphere). A major drawback of the traditional RANSAC is the fact that for the detection of small shapes in large point-clouds, the number of required shape hypothesis is prohibitively large and a very efficient scoring mechanism is need to test each point for proximity to a given shape hypothesis, and these two procedures becomes expensive operations. An improvement algorithm can improve the efficiency of feature detection to some extent (Schnabel, 2006).

The Template Matching can automatically extract CAD models (e.g. cylinder, sphere) from the point cloud. The algorithm consists of three steps: translating each possible transformation of the object on the acquired data, calculating a similarity metric, and finding the points where the metric exceeds a certain threshold. The computational requirements of such an implementation are unacceptably high (Rabbini, 2006). As a result, most of the literature concerned with this algorithm focuses on efficiency improvements (Ramapriyan, 1976).

For all above algorithms, huge of time and memory expenses are their same characteristics to detect related features from huge of point cloud data. For any accident building all scanned points can add up to 100~200 million, any of the above algorithms can not work well to extract pillar features from point cloud. New algorithm should be developed to extract pillars from huge of point cloud data.

3. THE FRAME OF PRESENTED ALGORITHM

In order to reconstruct the whole 3d model of an accident building different scans are gotten standalone. After registering all points are joined to a geo-referencing coordinate system, for three coordinate axes in the geo-referencing coordinate system, two of them are parallel to the horizontal plane and the other is the perpendicular to the horizontal plane. So in most case we can safely hypothesis that all wooden pillars are perpendicular to the horizontal plane after all scans were registered into a geo-referencing coordinate system.

3.1 The basic idea

Based on the above hypothesis we divide the original point cloud into slices along with the Z axis firstly, and then identifying local pillar geometric features (e.g. the projective circle and the radius) in selected slices with mathematical morphology, the next compare the gotten geometric features in adjacent slices and find the same or similar features, next to combine the same information to extract the global pillar features and at last reconstruct 3D pillar models. Different from other algorithms based on normal vector this algorithm extracts geometric features and reconstructs their 3D models directly, and has higher computation efficiency and less time and memory expense.

3.2 The flow of pillar feature extraction

The basic data processing procedure of the algorithm is listed:

(1) Register different scans into a uniform geo-reference coordinate system, smooth the noise and remove outliers;

(2) Cut the point cloud into slices along Z axis according to selected interval;

(3) Project all points belong to the selected slice onto the slice plane, identify projective geometric information of pillars (e.g. the projective circles, centre points and the radiuses) from the selected slice plane with mathematical morphology;

(4) Identify the same or similar pillar feature on adjacent slices and match their parameters to get global parameters of the same pillar;

(5) Reconstruct related 3D pillar models from the above gotten parameters.

3.3 Implementation of the algorithm

3.3.1 Pre-process: In most cases, single scan is not enough to reconstruct an accident building's 3D model, the building will to be scanned from different viewpoints in order to completely reconstruct its 3D model. Because each scan has its own local coordinate system, all local point clouds must be transformed into a common coordinate system. This procedure is usually referred as registration and all acquired points will be transformed into a uniform geo-reference coordinate system. Moreover, noise, outliers and overlapped point data are widely exist in the registered point cloud data, and they should be smoothed or removed by all sorts of algorithms. Raw acquired point cloud data cannot be directly used to extract features, and a series of works (e.g. registering, noise smooth and data simplification) will be done before.

3.3.2 Slice the point cloud: According the above hypothesis, all wooden pillars are perpendicular to the horizontal plane after different scans were registered into a geo-referencing coordinate system. So Z axis will be parallel to pillars and we will slice the entire point cloud into sections from bottom to top along the Z axis. To choose a plane to be perpendicular to the Z axis, and to move it along the same axis with a designed step interval. Each time the plane moves a step further, a slice is created, which consists of the points which are located within a threshold (e.g. half of the step interval). At the end of this process, the entire point cloud will be divided into slices.

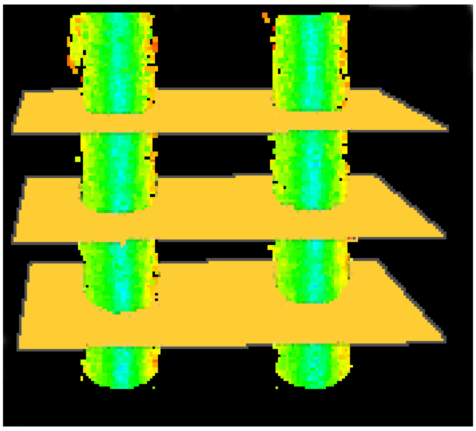

Figure 1. Slice the point cloud (partial)

3.3.3 Get the projective geometric information: A slice gap consists of all points near the slice plane and the distance within half step interval. After slicing all points of the selected slice will be projected onto the slice plane, and then raster the slice plane to a binary image, at last mathematical morphology will be employed to extract the projective circles and get the parameters (e.g. radius and centre point) of pillars from the image. Figure 2 show the projective point distribution of a slice and figure 3 show the binary image of the projective points.

Figure 2. The projective point distribution of a slice (partial)

Figure 3. The binary image of the projective points (partial)

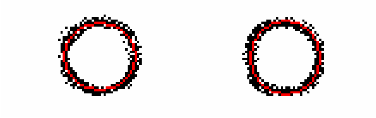

Figure 4. Extract the circle frames in the image (red)

3.3.4 Match adjacent slices and get the global pillar parameters

For each slice we can get a group of projection parameters of pillars. Here we need to match the parameters of each slice to parameters on the adjacent slices. By this way, we will be able to monitor how the feature changes from one slice to the next. If we have located a circle on one slice, and another circle on the next, with the second circle being the same or a bit different in size than the first, this would give us a hint that the object has a surface that forms some kind of a cylinder or a cone.

By comparing parameters on the adjacent slices we can find the change rule and judge the object type. For a cylinder object the projective circle centre and its radius between adjacent slices will be the same from bottom to top in theory; for a cone object the projective circle centre will be the same between adjacent slices and its radius between adjacent slices will keep a constant change from bottom to top in theory. However the projective parameters of a cylinder object will not be the same in practice, and a random difference will be exist between adjacent slices, while a trend difference will exist between two adjacent slices of a cone object.

Based on the above description we can easily get the global parameters from all the parameters of slices. For all pillars perpendicular to the horizontal plane, check the radiuses and the centre points of the projective circles, if they random change within a given threshold, we can classify it as cylinder. The global parameters of this type cylinder include: ① Radius, which can be defined as the average of all projective radiuses; ② Cylinder axis vector, which can be constructed by the average position of all projective circle centres and Z axis vector;③Extent, check all slices from bottom to top and find the first slice and the last slice which includes a cylinder, and the extent of a cylinder can be defined as from the first to the last slice. Due to the step interval of slice more precision extent can be gotten by slicing the first and last slice with a shorter step interval.

For those cylinders which not perpendicular to the horizontal plane, their projective ellipses will be gotten in each slice and the centre point, the semiminor axis and semimajor axis will be gotten too. If the difference of the semiminor axis between two adjacent slices is less than given threshold and changes in random, and the centre of adjacent projective ellipses changes within a constant, we can classify it as cylinder too. Here the global parameters of this type cylinder can be defined as: ① Radius, which can be defined as the average of all the semiminor axes of projective ellipse;②Cylinder axis vector, which can be constructed by any two ellipse centres which is positioned by x, y and z in theory, the average vector of all constructing cylinder axis vectors will be the best choice;③ Extent, same as the previous extent definition.

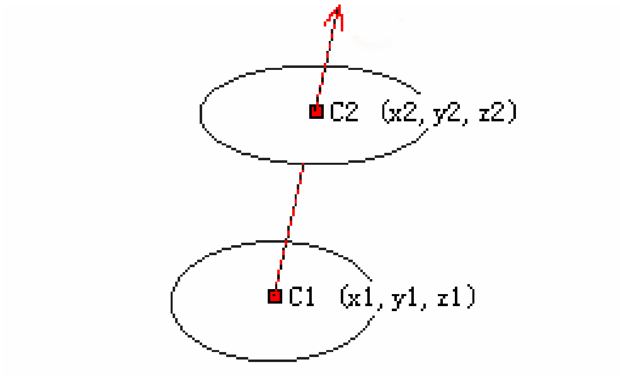

Figure 5. Get the cylinder axis vector (not perpendicular)

3.3.5 Reconstruct 3D pillar models

After getting all the global parameters of pillars it's easy to reconstruct the 3D pillar models by any 3D model tools (e.g. AutoCAD or 3DMax).

4. ALGORITHM IMPLEMENTATION AND TEST

For the implementation of the presented algorithm, a simple application was developed based on the OpenGL graphics library with C++, and all the related functions (e.g. Slice the point cloud, project related points onto the slice plane, extract the projective parameters, find the global parameters of a pillar and reconstruct 3D pillar models) have been developed, and each function can be executed alone.

In order to test the feasibility and the practicable of the presented algorithm and the developed application, raw acquired data of an accident building in the Forbidden City in Beijing was selected as the sample data.

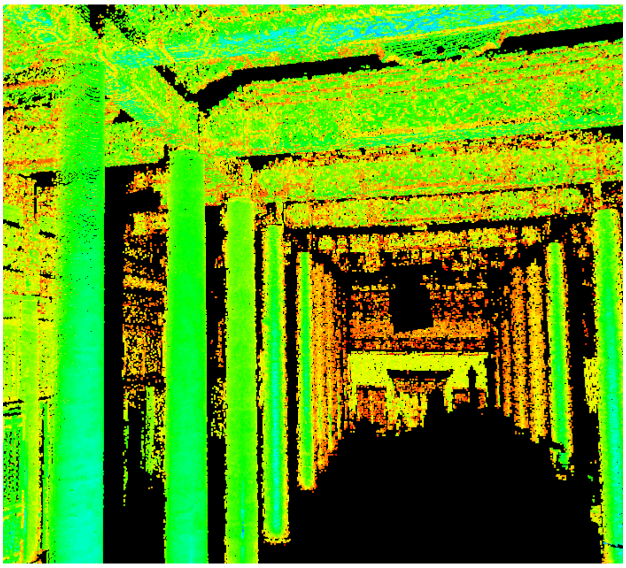

Figure 6. The original point cloud of an accident building

From figure 6 we can see that the point cloud consists mainly of the scanned points of the pillars, the walls and the crossbeams, the point cloud are combined and merged with six scans by registering. However, noise and outliers exist in the point cloud extensively and robust algorithms are needed to smooth or remove them before further point cloud segmentation and feature extraction.

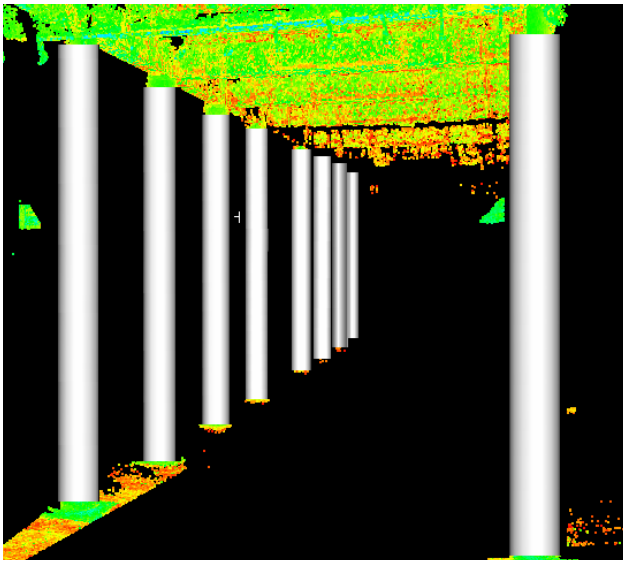

Figure 7. The reconstructed 3D pillars.

Figure 7 shows the reconstructed results of pillars based on our presented algorithm, and we can see that the quality of reconstructed 3D pillars is high. Here points of the walls, part of the crossbeams and some of other objects were removed firstly. Moreover two other feature detecting algorithms were employed to make a control test with the same sample data and in the same computation environment (e.g. same hardware). The test results list in table 1:

algorithm	time expense
the Hough Transform	>35 Sec
the RANSAC paradigm	>39 Sec
the presented algorithm in the paper	<5 Sec

Table 1. Feature extracting comparision

5. CONCLUSIONS

For a special kind of scanned object—wooden pillars, we have developed an algorithm which can rapidly extract pillar features from huge quantity of point cloud data by slicing the point cloud into sections. In this paper, the basic theory, the flow of data processing, the feature extracting test and its control tests about our presented algorithm are described in detail.

Compare to the Hough Transform and the RANSAC paradigm, our presented algorithm is more efficient and more practicable for extracting pillars from scanned point cloud, reduces much more time and memory expenses. However, the presented algorithm in this paper is not a common algorithm, it’s useful and efficient for extracting pillar features from the point cloud, but for other kind of cylinder objects it will not always keep its efficiency and practicable operation.

REFERENCES

Illingworth J., Kittler J., 1988. A survey of the hough transform. Computer Vision Graph. Image Process, 44(1), pp. 87–116.

Fischler M. A., Bolles R. C, 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), pp. 381–395.

Ramapriyan, H. K., 1976. A multilevel approach to sequential detection of pictorial features. IEEE Transactions on Computers, 25(1), pp. 66–78.

Schnabel R., Wahl R., Klein R., 2006. Shape Detection in Point Clouds. Tech. Rep. CG-2006-2, Universität Bonn, Germany.

Rabbani T.Shah, Automatic Reconstruction of Industrial Installations Using Point Clouds and Images. PHD thesis in NCG,2006, http://www.ncg.knaw.nl/Publicaties/Geodesy/pdf/ 62Rabbani.pdf.

Kimme, C., Ballard, D.H., and Sklansky, J., 1975. Finding Circles by an Array of Accumulators. Communications of the ACM, 18 (2), pp. 120-122.

ACKNOWLEDGEMENTS

The presented work has been done within in the scope of the junior research group “Key technologies research on constructing 3D digital models for accident buildings”. The project is funded by the Beijing Municipal Education Commission Science and Technology Foundation (Code: KM 200610016009) and the Beijing Nature Science Foundation (Code: 8063027).