VIEWPOINT CALIBRATION METHOD BASED ON POINT FEATURES FOR POINT CLOUD FUSION

A. Liang Zhang, B. Xiao Zhang, C. Juan Song, D. Peiyi Shen, E. Guangming Zhu, F. Shaokai Dong

School of Software Engineering, Xidian University, Xian, Shanxi Province, China

ABSTRACT

Point cloud fusion is a significant process in many computer vision applications such as multi-robot maps building or multi-SLAM. When robots capture maps in different places, the maps suffer from large angle difference between respective viewpoints of the robots and spatial misalignment. These problems bring great challenges to point cloud fusion. In order to overcome such difficulty, in this study, we propose a 3D viewpoint calibration method. The method relies on the novel combination of 3D-SIFT (scale-invariant feature transform) keypoints and FPFH (fast point feature histogram) features, which are used for feature matching. Based on the feature correspondences, we compute the transformation to resolve the difference of the viewpoints. The experiments demonstrate that (1) the keypoints and features used in our method are distinctive and robust to camera viewpoint change; and (2) using viewpoint calibration can reduce the iterations of registration algorithms, and transforming different viewpoints to a close one could save computation time and improve accuracy.

***Index Terms*—** Point cloud fusion, viewpoint, 3D-SIFT, feature matching, FPFH

1. INTRODUCTION

With recent advances in stereo camera and depth sensing devices, applications of 3D point cloud data have become popular in computer vision. It is easy to obtain a 3D point cloud model via algorithms like RGBD-SLAM [1][2], Structure from Motion [3]. In computer vision applications such as multi-robot maps building and multi-SLAM, each robot builds its own map in different places. Therefore, multiple sweeps of a single sensor or multiple viewpoints of a same scene exist in different maps. In this situation, different maps bring great difficulties to robot location and navigation. So, in order to get a full map for location and navigation, it becomes very necessary to spatially align these different point cloud maps. /par Many registration methods have been developed over the past few years such as the Iterative Closest Point (ICP)[4], and the Normal Distributions Transform(NDT)[5]. However, ICP registration method needs good initial guess to ensure its convergence and suffers from expensive computation complexity. Therefore, the method can not handle large difference in viewpoints; and many iterations are needed resulting in high computational complexity.

In order to find a good initial guess for ICP registration algorithm to ensure the good results of point cloud fusion in multiple viewpoints, many researchers have focused on various feature-based approaches to compute an robust initial guess by feature matching. Gelfand extracted volumetric feature from surfaces of the input point clouds [6], but this feature is not adapted to the changes of viewpoints. Zou considered Gaussian curvature of surface geometry for feature extraction [7]. However, with the viewpoint changing, the same object in the point cloud will show different surfaces, which will cause feature matching inaccurately. Rusu proposed a persistent feature histograms (PFH) and fast point feature histogram (FPFH) in [8][9]. PFH involves the geometric information in the query point and its k nearest neighbors, which is more robuster compared to volumetric features and Gaussian curvature. Based on PFH, FPFH is more faster by simplifying the calculation process. However, when PFH or FPFH is used to find feature matching to compute an initial transformation, mismatch always exist because of viewpoint difference. Besides, it takes too much time to compute these features in millions of points. SIFT(scale-invariant feature transform) is well known due to its good performance under changes of viewpoint, scale and rotation. In [10], the author extracted SIFT features in 2D image to normalize 3D viewpoint. However, SIFT in image drops quickly when substantial viewpoint changes happened. Flint has extended the ideal of SIFT to the 3D case in[11]. Ref [12] has shown that in point cloud, using scale space extremum to compute 3D-SIFT keypoints can ensure the robustness to substantial viewpoint changes.

In this paper, a viewpoint calibration method is proposed for point cloud fusion, which makes use of the combination of advantages between FPFH and 3D-SIFT. In our method, firstly, 3D-SIFT keypoints are extracted, and then in order to find feature matching, FPFH features are computed on these keypoints. Our method has some advantages. For one thing 3D-SIFT keypoints are distinguishing among millions of points even in different viewpoints, for another we compute FPFH features at these keypoints which to a great extent can avoid feature matching errors.

2. VIEWPOINT CALIBRATION

The whole process of the proposed algorithm is shown in Fig. 1. Different point clouds from multiple viewpoints are processed by viewpoint calibration method to reduce the gap of multi-views. Then the transformed point cloud and the source point cloud fuse into a global one via ICP registration algorithm.

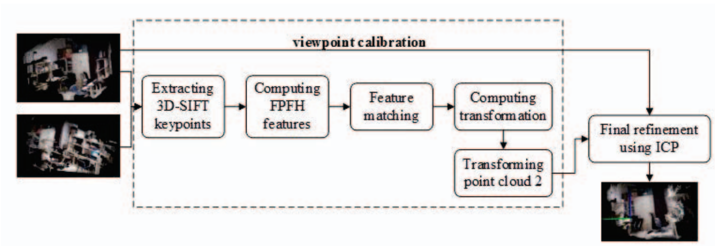

Fig. 1. Flowchart of the whole process.

The stage of feature matching in viewpoint calibration is introduced to find FPFH correspondences at the 3D-SIFT keypoints. FPFH feature is described by 33 dimensional information. In order to find feature correspondences, we use kdtree which is a space-partitioning data structure and is useful for nearest neighbor searches. The target features are used to build a kdtree, and then we traverse the kdtree to find k features similar to the source points feature as candidates. Then, we select the nearest one as the target by comparing Euclidean distance. After that, we can get a series of matching features.

In this section, we will introduce the main procedures of viewpoint calibration.

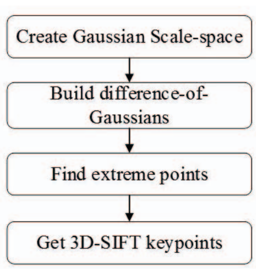

Fig. 2. Major steps of extracting 3D-SIFT keypoints.

2.1. Extracting 3D-SIFT keypoints

In the past, 2D-SIFT feature has been used in many different applications from image registration, switching, to object recognition. Compared to 2D-SIFT,3D-SIFT keypoint has stronger adaptability to viewpoint changes, and can be accurately identified at different positions . The major steps are shown in Fig. 2.

The Gaussian scale-space is computed according to downsampling point cloud with voxel grid of different sizes. And each point cloud called octave is divided into some layers with different scales. In every layer we use kdtree space to perform a radius search. In each layer, the scale is defined as follows:

$$\sigma(s) = \sigma_0 * 2^{(s-1)/S} \tag{1}$$

Where σ_0 is the base scale, s is the index layer, and S is the number of layers in each octave. For each two adjacent layers, a new DoG (Difference of Gaussian) is computed. If a point has the highest or lowest DoG value compared with k nearest neighbors, it is considered as an extreme point. Finally, extreme points with lower contrast than minimum will be removed. And the rests will be considered as 3D-SIFT keypoints. Fig. 3 shows the results of 3D-SIFT keypoints in our scene.

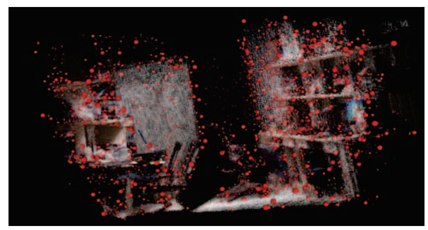

Fig. 3. Example of 3D-SIFT keypoints (red points) in point cloud.

2.2. Computing FPFH features

After extracting 3D-SIFT keypoints, FPFH features at these keypoints are computed. FPFH (fast point feature histogram) feature contains the relative orientation of normal between point pairs falling within the spherical neighborhood of a query point. The query point and its k neighbors form point pairs in a point cloud. Fig. 4 shows an example of point pair (p_s, p_t) and the coordinate system is built between them. In the figure, u, v, w are three coordinate axes, n_s is the normal of p_s, and n_t is the normal of p_t. Three feature angles α, φ, and θ represents the orientation of normal and coordinate system. The references [8][9] define the three feature angles in each point as SPFH. After that, for each point a procedure of redetermine its k neighbor is performed. Then the neighboring SPFH is used to compute FPFH as described in formula (2), where ω_i represents the distance between query point p and a neighboring point p_i.

$$FPFH(p) = SPFH(p) + 1/k * \sum_{i=1}^{k} SPFH(P_i)/\omega_i \tag{2}$$

2.3. Computing transformation

After finding the matched features, a list of possible point correspondences are obtained. We set the two 3-D point sets as $P = \{p_1, p_2, p_3, ..., p_n\}$ and $Q = \{q_1, q_2, q_3, ..., q_n\}, p_i, q_i \in$

Table 1. Comparison of our method with others in time consumption and accuracy. Our method is faster than others thats because we only process 3D-SIFT keypoints which saves much time.

	Dataset1		Dataset2		Dataset3	
	Time(s)	Accuracy	Time(s)	Accuracy	Time(s)	Accuracy
ICP	185	7.52	failed	2.51	failed	1.22
NDT	165	7.38	failed	2.43	failed	2.58
FPFH+ICP	154	7.65	286	6.42	455	5.76
FPFH+NDT	133	8.02	298	5.16	568	5.25
3D-SIFT+FPFH+ICP(ours)	102	9.08	209	8.89	287	8.88
3D-SIFT+FPFH+NDT(ours)	85	8.92	215	8.76	326	8.64

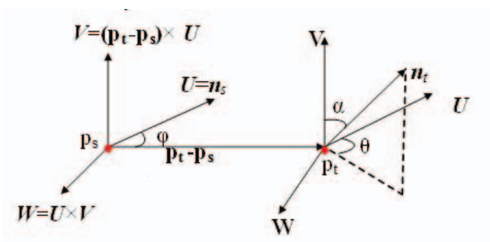

Fig. 4. Coordinate system for a point pair (p_s, p_t) in red.

$R^3 i = 1, 2, ..., n$. Here we consider P as the source point set, and Q as the target point set. The two point sets are in different coordinate systems. In this section, rotation matrix $R \in R^{3*3}$ and a translation vector T that transform the source to the target are computed. We use the Least-Squares estimation proposed in [13] to calculate R and T through computing the mean squared error defined as follows:

$$\xi^2 = \sigma_q^2 - tr(DS)^2/\sigma_p^2 \tag{3}$$

where σ_q^2 is the variance of point set $\{p_i\}$ defined as $\sigma_q^2 = 1/n * \sum_{i=1}^{N} \|q_i - q\|^2$. The covariance matrix of two point sets is $Cov_{PQ} = 1/n * \sum_{i=1}^{N} (q_i - q)(p_i - p)^T$ where p and q represent the centroid of the two point sets respectively. Let a singular value decomposition (SVD) of Cov_{PQ} be

$$UDV^T = SVD(Cov_{pq}) \tag{4}$$

R and T are computed according to formula (5) and (6)

$$R = USV^T \tag{5}$$

$$T = q_i - Rp_i \tag{6}$$

S must be chosen as formula (7)

$$S = \begin{cases} I, if det(U)det(V) = 1 \\ diag(1, 1, ...1, -1), if det(U)det(V) = -1 \end{cases}, \tag{7}$$

After calculating R and T, we use the matrix to transform the source point cloud to reduce the viewpoint gap. In Fig. 5, we show the results of transforming different viewpoints using viewpoint calibration method.

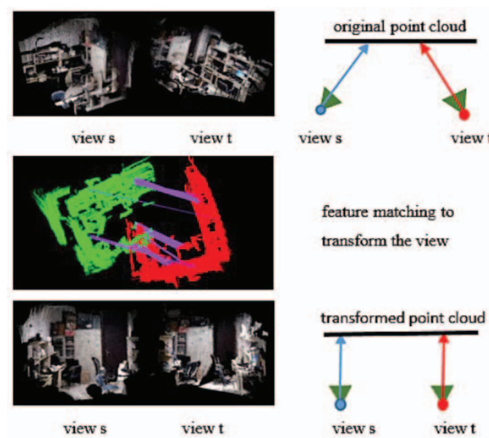

Fig. 5. In the first row, we obtained the original point clouds from different viewpoints. In middle row, we show the feature matching by purple lines. In the bottom, pictures show the transformed viewpoints.

3. EXPERIMENTAL RESULTS

In our experiment, three different point cloud datasets are used to verify the advantage of proposed algorithm. Two of them are the scene of our lab, which are obtained by Xtion Pro Live Kinect and the 3rd one [14] is the scene of a part of the city obtained by LiDAR. The dataset1 exists a small gap in horizontal direction between source and target. The dataset 2 and 3 exist large gap in space. We use C++ and matlab on the computer of Inter core i5 CPU 4G memory. ICP[4] and NDT[5] are adapted as the baseline. In order to evaluate the performance of different methods, we use transformation accuracy (T_a) computed by formula (8) and (9)

$$T_a = N / \sum_{i=1}^{N} \|q_i - q_i^{'}\|^2 \tag{8}$$

$$q_i^{'} = R * p_i + T \tag{9}$$

N is the number of target point cloud, q_i is a 3D point in the target point cloud, p_i is the source point. R and T are computed as mentioned above. We divide the transformation

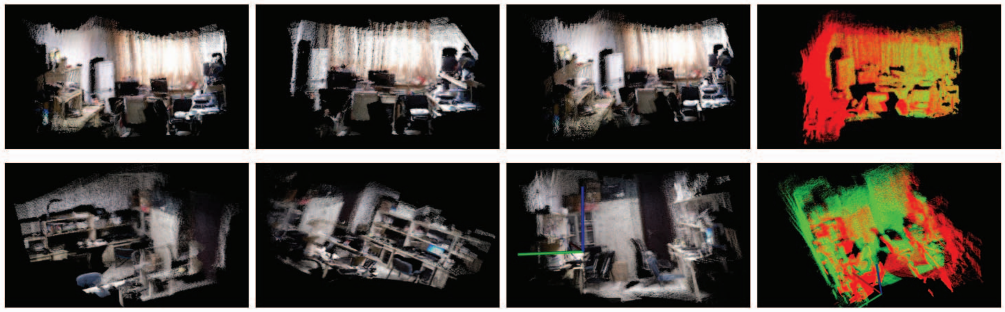

Fig. 6. The 1st and 2nd column represent the source point cloud and target point cloud. The 3rd and the 4th column represent our method results. The 1st row is the first dataset and the 2nd row is dataset2.We use different colors to show that the source and target point clouds fuse together.

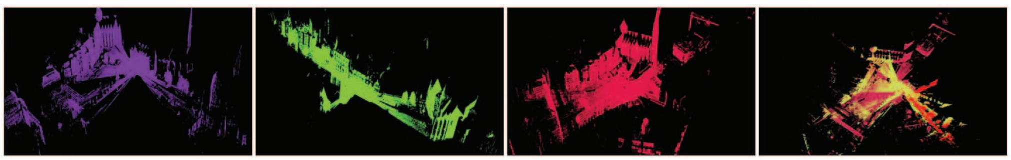

Fig. 7. The 1st and 2nd columns represent source point clouds. The 3rd column represents target point cloud. The 4rd column represents the result of three point clouds fusion

accuracy of different methods into 10 intervals, and number from 1 to 10 represents the level of accuracy. The number is bigger the result is better.

Table 1 shows the time consumption and accuracy comparison. It is easy to see that ICP and NDT fail to perform point cloud fusion in dataset 2 and 3, however, the proposed algorithm can fuse quite well. Thats because extracting 3D-SIFT can ensure the robustness to viewpoint changes. The relationship between iteration and accuracy using different methods is shown in Fig. 8 and Fig. 9. It can be seen when using the same iterations of ICP or NDT, our method is more accurate than the other solutions.

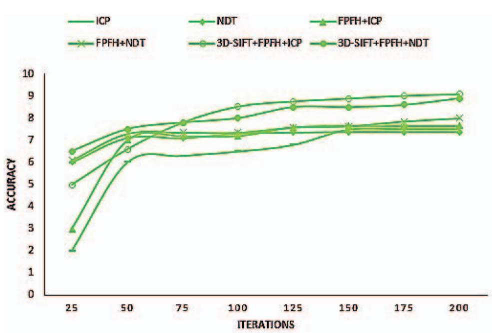

Fig. 8. The relationship of iterations and accuracy in dataset1.

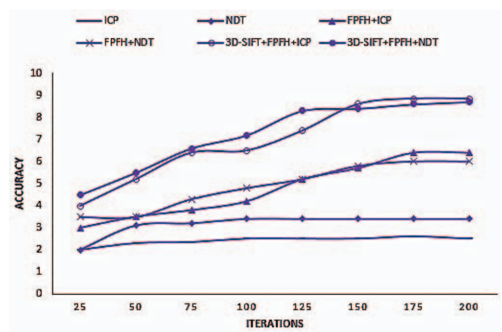

Fig. 9. The relationship of iterations and accuracy in dataset2.

4. CONCLUSION

In this paper, we propose a viewpoint calibration method to solve the problem of large gap in multiple viewpoints in point cloud fusion. The experiment shows that our method based on 3D-SIFT keypoints and FPFH features can improve the accuracy and efficiency in point cloud fusion.

5. ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their valuable comments. This work is partially supported by the China Postdoctoral Science Foundation (Grant No. 2016M592763), the Fundamental Research Funds for the Central Universities (Grant NO. JB161006, JB161001), the National Natural Science Foundation of China (Grant NO. 61401324, 61305109).

6. REFERENCES

[1] Hu G, Huang S, Zhao L, et al. A robust RGB-D SLAM algorithm[C]// Ieee/rsj International Conference on Intelligent Robots and Systems. IEEE, 2012:1714-1719.

[2] Endres F, Hess J, Sturm J, et al. 3-D Mapping With an RGB-D Camera[J]. IEEE Transactions on Robotics, 2014, 30(1):177-187.

[3] Snavely N, Seitz S M, Szeliski R. Photo Tourism: Exploring Photo Collections In 3D[J]. Acm Transactions on Graphics,2006, 25(3):835-846.

[4] Besl P J, Mckay N D. A method for registration of 3-D shapes[J]. IEEE Transactions on Pattern Analysis MachineIntelligence, 1992, 14(3):239-256.

[5] Magnusson M. The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection[D]. Oebro universitet, 2009.

[6] Gelfand N, Mitra N J, Guibas L J, et al. Robust Global Registration[C]// Eurographics Symposium on Geometry Processing. 2005:197.

[7] Zou G, Hua J, Lai Z, et al. Intrinsic geometric scale space by shape diffusion.[J]. IEEE Transactions on Visualization Computer Graphics, 2009, 15(6):1193.

[8] Rusu R B, Blodow N, Marton Z C, et al. Aligning point cloud views using persistent feature histograms[C]// Ieee/rsj International Conference on Intelligent Robots and Systems, September 22-26, 2008, Acropolis Convention Center, Nice, France. 2008:3384-3391.

[9] Rusu R B, Blodow N, Beetz M. Fast Point Feature Histograms (FPFH) for 3D registration[C]// IEEE International Conference on Robotics and Automation. IEEE Press, 2009:1848-1853.

[10] Y. Cao, M. Yang, and J. McDonald. Robust Alignment of Wide Baseline Terrestrial Laser Scans via 3D Viewpoint Normalization. In Workshop on App. of Comp. Vision. IEEE,2011.

[11] Flint A, Dick A, Hengel A V D. Thrift: Local 3D Structure Recognition[C]// Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications. IEEE Computer Society, 2007:182-188.

[12] Hasch R, Hellwich O, Weber T. Comparison of 3D Interest Point Detectors and Descriptors for Point Cloud Fusion[C]// Pcv. 2014:57-64.

[13] Arun K S, Huang T S, Blostein S D. Least-squares fitting of two 3-d point sets[J]. IEEE Transactions on Pattern Analysis Machine Intelligence, 2009, PAMI-9(5):698-700.

[14] https://sourceforge.net/projects/pointclouds/files/PCD

[15] Peng F, Wu Q, Fan L, et al. Street view cross-sourced point cloud matching and registration[C]// 2014 IEEE International Conference on Image Processing. 2014:2026-2030.

[16] Zheng Z, Li Y, Wang J. LiDAR point cloud registration based on improved ICP method and SIFT feature[C]// IEEE International Conference on Progress in Informatics and Computing. IEEE, 2015:588-592.

[17] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91—110.

[18] Rusu R B, Cousins S. 3D is here: Point Cloud Library (PCL)[C]// IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May. DBLP, 2011:1-4.