The Japan Society for Precision Engineering
19th International Conference on Precision Engineering
Nov.28(Mon.)–Dec.2(Fri.) , 2022 Nara , Japan

C155

Improving Multi-sensor Point Cloud Fusion via Color Information and Point Cloud Partitioning

Shixun YAN*, Sarthak PATHAK** and Kazunori UMEDA**

Abstract: Point cloud fusion is an important task for applications such as 3D modeling, environment inspection, and digitization. Recently, deep learning has also started to deal with point cloud fusion. However, most methods in literature are suitable only for identical or nearly identical point clouds. In practice, the sizes of point clouds vary depending on the sensors, capture distances, measurement methods, and many other factors. For example, when acquiring point clouds with a wide range FARO FOCUS laser scanner, a large, but sparse point cloud of the entire environment is obtained. On the other hand, when acquiring point clouds with an RGB-D camera like the Intel RealSense, a small but dense point cloud of a specific location is obtained. For applications such as inspection, it is usually necessary to have dense point clouds of specific target regions in addition to the structure of the whole environment. The fusion of point clouds with different sizes is therefore essential. Accordingly, this paper focuses on the fusion of point clouds with different sizes.

Key words: point cloud, registration, deep learning

1. Introduction

Distance image sensors are useful for tasks such as 3D measurement, 3D modeling and robot navigation. However, all types of distance image sensors have their own limitations depending on their measurement method. It is therefore effective to fuse multiple distance image sensors when, for example, the dense point clouds of specific target regions in addition to the structure of the whole environment is needed. Therefore, it is important to register point clouds obtained from different distance image sensors robustly and precisely. In recent years, many deep learning-based approaches for point cloud registration have been explored. PointNetLK[1] uses a deep learning model[2] to extract the features of the entire point cloud and to obtain a rigid transformation matrix that to make the features of the target point clouds like each other. MaskNet[3] is a model for estimating matching regions between one point cloud and the other. However, these methods cannot handle point clouds with very different sizes or orientation. PointpartNet[4] handles this situation by partitioning point clouds via nearest neighbor estimation, matching region search using partial features, followed by correspondence between point clouds and registration. In this paper, we improve PointpartNet[4] using color information and propose a model that can register with higher accuracy.

2. Proposed Method

In this study, two types of point clouds are assumed: 1. **full** : wide area point clouds acquired by e.g. LiDAR; 2. **part:** more detailed point cloud acquired by small area scanners such as RGB-D cameras. A sample point cloud is shown in Fig. 1 respectively.

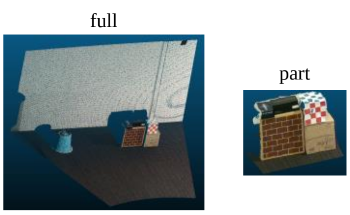

Fig. 1 Example of point clouds registererd in this research.

The outline of proposed method consists of the following steps:

1. global translation estimation by partial feature extraction and matching region search
2. global rotation estimation via a neural network.
3. precise local registration via color information.

Steps 1 and 2 are the same as in PointpartNet[4]. An image of step 3 is shown in Fig. 2.

2.1 Matching Region Search and Global Rotation Estimation

First, downsampling is performed to achieve the same density in full and part. Then, **full** is partitioned to extract partial features. Specifically, for each point in **full**, *n* nearest-neighbour points are selected to create a sub-point set **fullgroup**, where *n* is the number of points in **part**. **fullgroup** and **part** are then input to PointNet$_1$[2] to extract the global features, respectively. The global features are input to MultiLayer Perceptron (MLP$_1$) and the matching likelihood **Score** is calculated. $Score_i$ approaches 1 if $\mathbf{fullgroup}_i$ and **part** are similar. The highest score $Score_H$ is selected and the corresponding $\mathbf{fullgroup}_H$ is judged to be the matching region with **part**. The positional relationship between $\mathbf{fullgroup}_H$ and **part** is denoted by global translation $\mathbf{t_G}$. Global rotation $\mathbf{R_G}$ is estimated close to the true value as possible as quickly as possible using DirectNet[5]. The global features of $\mathbf{fullgroup}_H$ and **part** are used as input to MLP$_2$ to calculate the $\mathbf{R_G}$[1]. Here, global features for global registration are calculated using PointNet$_2$[2].

2.2 Local Registration

In this paper, we focus mainly on improving the accuracy of local registration between point clouds, as shown in Fig. 1, via the

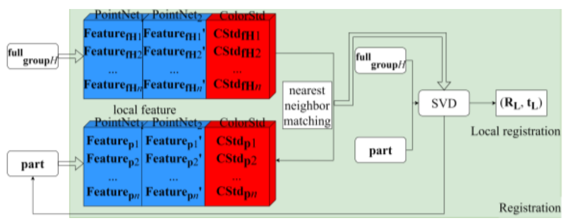

Fig. 2 Local registration architecture.

* Graduate School of Science and Engineering, Chuo University, (karasu@sensor.mech.chuo-u.ac.jp)
** Faculty of Science and Engineering, Chuo University ({pathak, umeda}@mech.chuo-u.ac.jp)

use of color information. Local registration is a process for registering point clouds that are close in position and orientation after global registration has been carried out. 3D point cloud coordinates can be used as features, however there are cases where it is difficult to find the corresponding points based on the structure alone Therefore, color information is also used as input information when finding the corresponding points. The proposed model combines the local features of **fullgroup**$_H$ and **part** obtained during the matching region search as well as color information, using them as features for the local registration process. Local features are the features of each point in the point cloud. It is generally difficult to calculate color information in sparse, 3D point clouds, unlike in dense 2D images. Moreover, color information acquired from different range image sensors is often different. Thus, instead of using color directly, the standard deviation of color around the neighborhood of each point was chosen to represent the amount of discriminative. A higher standard deviation around a point indicates the presence of discriminative texture information, which enables point cloud matching with higher confidence. Here, the color information used is the standard deviation of the n/32 nearest neighbor points for each of the RGB channel. Finally, the closest point between **fullgroup**$_H$ and **part** in the feature space is determined as the corresponding point. The calculated point correspondences and 3D point coordinates are input to SVD to calculate the rigid body transformation matrix. Local registration is repeated until the rigid-body transformation converges.

2.3 Loss Function

The model trains two networks simultaneously: matching region search to select the highest score, and the global registration. As a result, the loss function is divided into two parts: matching loss $loss_m$ and global registration loss $loss_{gr}$.

The matching loss $loss_m$ is the negative log-likelihood loss often used in classification problems. The global positioning loss $loss_{gr}$ minimizes the difference between the rigid body transformation matrix $\mathbf{G_G} = \{[\mathbf{R_G}, \mathbf{t_G}]\}$ and its true value $\mathbf{G_{gt}}$. It is therefore expressed using Mean Square Error (MSE) as follows.

$$loss_m = -log\left(\frac{exp(Score_{label})}{\sum_{i=1}^{g} exp(Score_i)}\right) \quad (1)$$

$$loss_{gr} = \left\|(\boldsymbol{G_G})^{-1} \cdot \boldsymbol{G_{gt}} - \boldsymbol{I}_4\right\|_F \quad (2)$$

Here, *label* is the index of the centre point $\mathbf{f}_{gt}$ and is the true value.

3. Experiments

The experiments were described on the public dataset ShapeNet[6], a 3D model dataset of objects with color information. **full** was a point cloud resampled from the data of ShapeNet[6]. The scale of **full** was 0.400m to 1.000m. **part** was the point cloud from which a quarter of **full** was cut off, with added noise from a normal distribution with mean 0 m and variance 0.05 m^2. The true values of the rigid body transformation matrix during training were randomly generated with rotations [0, 90] ° and translations [0, 1.57] m. The true values of the rigid-body transformation matrix during testing were randomly generated with rotations [0, 180] ° and translations [0, 3.14] m. Training was done for 200 epochs with a batch size of 16. The number of points in **full** was 256 and the number of points in **part** was 64. During the testing, the number of points in **full** was 1024; the number of points in **part** was 256. Comparison experiments with PointpartNet[4] were also carried out. Table 1 shows the results of the comparison between the proposed model and PointpartNet[4]. Experimental results include the estimation success rate and mean error of the rotation and translation estimation when matching is successful.

The success rates for rotation and translation estimation error were improved from 84.07% and 87.01% to 84.45% and 87.57%, respectively, as compared to PointpartNet[4]. The mean errors for rotation and translation estimation were reduced from 23.136 ° and 0.063 m to 21.898 ° and 0.059 m, respectively, as compared to PointpartNet[4]. Thus, it can be said that local registration with color information was effective in increasing performance.

Table 1. Experimental results for each method.

Rotation	success error<10 °	mean error	mean error on success
PointpartNet[4]	84.07%	23.136 °	0.222 °
Proposed Method	**84.45%**	**21.898 °**	**0.224 °**
Translation	**success error<0.1 m**	**mean error**	**mean error on success**
PointpartNet[4]	87.01%	0.063 m	0.002 m
Proposed Method	**87.57%**	**0.059 m**	**0.002 m**

4. Conclusion

In this paper, we modified PointpartNet[4], a deep learning-based positioning method that performs registration for point clouds of different sizes, to improve the estimation accuracy. This method successfully registers point clouds of different sizes by adding color information to the local registration. Compared to conventional methods, this method proved to be more robust and accurate. In the future, it will be verified if point cloud positioning on real data is feasible and color information will be input to a neural network for more precise registration. Also, a better way of representing point cloud color information will be considered.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 19H04191.

References

1) Y. Aoki, H. Goforth, R. A. Srivatsan and S. Lucey: PointNetLK: Robust & Efficient Pointcloud Registration Using PointNet, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), (2019) 7156.
2) C. R. Qi, H. Su, K. Mo and L. J. Guibas: Pointnet: Deep learning on point sets for 3d classification and segmentation, Proc. Computer Vision and Pattern Recognition (CVPR), (2017) 77.
3) V. Sarode, A. Dhagat, R. A. Srivatsan, N. Zevallos, S. Lucey and H. Choset: MaskNet: A Fully-Convolutional Network to Estimate Inlier Points, 2020 International Conference on 3D Vision (3DV), (2020) 1029.
4) S. Yan, S. Pathak, and K. Umeda: PointpartNet: 3D Point-cloud Registration via Deep Part-based Feature Extraction, Advanced Robotics, (accepted).
5) A. Kurobe, Y. Sekikawa, K. Ishikawa and H. Saito: CorsNet: 3D Pointcloud Registration by Deep Neural Network, in IEEE Robotics and Automation Letters, 5, 3, (2020) 3960.
6) A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi and F. Yu: Shapenet: An information-rich 3d model repository, arXiv preprint arXiv:1512.03012 (2015).