PAPER

Feature Description with Feature Point Registration Error Using Local and Global Point Cloud Encoders

Kenshiro TAMATA[†a)], ***Nonmember and*** **Tomohiro MASHITA**[†], ***Member***

SUMMARY A typical approach to reconstructing a 3D environment model is scanning the environment with a depth sensor and fitting the accumulated point cloud to 3D models. In this kind of scenario, a general 3D environment reconstruction application assumes temporally continuous scanning. However in some practical uses, this assumption is unacceptable. Thus, a point cloud matching method for stitching several non-continuous 3D scans is required. Point cloud matching often includes errors in the feature point detection because a point cloud is basically a sparse sampling of the real environment, and it may include quantization errors that cannot be ignored. Moreover, depth sensors tend to have errors due to the reflective properties of the observed surface. We therefore make the assumption that feature point pairs between two point clouds will include errors. In this work, we propose a feature description method robust to the feature point registration error described above. To achieve this goal, we designed a deep learning based feature description model that consists of a local feature description around the feature points and a global feature description of the entire point cloud. To obtain a feature description robust to feature point registration error, we input feature point pairs with errors and train the models with metric learning. Experimental results show that our feature description model can correctly estimate whether the feature point pair is close enough to be considered a match or not even when the feature point registration errors are large, and our model can estimate with higher accuracy in comparison to methods such as FPFH or 3DMatch. In addition, we conducted experiments for combinations of input point clouds, including local or global point clouds, both types of point cloud, and encoders.

key words: *3D environment reconstruction, point cloud, registration, machine learning*

1. Introduction

A model of 3D real environment can be applied in many cases, such as augmented reality applications that add virtual interaction to the real environment, visiting houses with a head-mounted display, and reconstructing indoor structures for creating floor plans of buildings. A model of the real 3D environment can be created by scanning the environment with a sensor and then merging the sensor data into an entire model of the target environment.

The point cloud is a data format for representing a 3D shape on a computer, and a 3D model of the real environment can generally be created by aligning the positions of multiple point clouds. For example, some methods for creating 3D models such as Kinectfusion [1], DVO-SLAM [2], ElasticFusion [3], and BundleFusion [4] have been proposed. Current 3D real environment modeling applications such as Kinectfusion are based on the assumption that captured frames are temporally continuous. In other words, the differences in camera angles and camera positions between frames are assumed to be small, and the initial positions between the point clouds are assumed not to differ greatly. However in an actual use of 3D scanning, some kinds of scanning error may occur, including a quick and large rotation of a sensor which cannot be assumed as a continuous scan or termination and restarting of the scanning due to an application error. In this case, the assumption of a temporally continuous scanning are not satisfied, and the alignment of the point clouds will fail. Then the user have to scan the target environment again from the beginning. In addition, a procedure of parallel scanning with multiple sensors and creating a 3D model by merging the scans is impossible in the condition of the temporally continuous scanning.

Some methods to align point clouds that have greatly different initial positions have been proposed such as 3D Match [5] or PPF Net [6], [7]. These methods describe features that express geometric structures which the detected points and their surrounding points composed of. Then, these point clouds are aligned by matching positions of the detected points using the described features. However, detecting the same position in the real world from the two frames is difficult because detecting a point accurately in a point cloud is not as feasible as detecting corners from RGB images. This problem arises from the difference in the distributions of a point cloud and the missing parts due to occlusion or sensing errors. The density of a point cloud is basically dependent on the observation angle and distance to the target object. Moreover, sensing noise, which is dependent on the characteristics of the sensor and object's surface, also affects the distribution of a point cloud and missing parts. Thus, a point pair for starting the stitching process between two point clouds should ideally be at the same point, but this is not guaranteed even when taken from the same position. Moreover, some parts of the environment cannot be observed by the sensor due to occlusion or reflection. In this case, some of the corresponding feature points may not exist because the missing parts of two point clouds are different. In this paper, we use "feature point registration error" to describe the error in a pair of corresponding feature points caused by sensing error or feature point detection error.

To achieve feature descriptions robust to feature point registration error, we designed a feature description model that uses local information around the feature points and

Manuscript received April 12, 2021.
Manuscript revised July 30, 2021.
Manuscript publicized October 11, 2021.

[†]The authors are with Osaka University, Toyonaka-shi, 560–0043 Japan.
a) E-mail: tamata.kenshiro@lab.ime.cmc.osaka-u.ac.jp

DOI: 10.1587/transinf.2021EDP7082

global information from the entire point cloud. A pair of local areas around a feature point with error sometimes differ significantly due to occlusion and camera angle. In this case, an inference using local areas only for feature description fails. In addition, since there are usually many similar local shapes but at different locations, the inference using local areas only is not suitable. Feature description using only the global information is greatly affected by the angle and position of the camera since the overall shapes of the point clouds varies by the angle and position of the camera. We therefore design a feature descriptor robust to the errors of point clouds measured temporally and discontinuously by combining local and global information.

2. Related Work

Point cloud matching can be classified into coarse matching and fine matching [8]. In a coarse matching, a transformation matrix for a rough registration of two point clouds is estimated from feature points detection, description and matching. In a fine matching, a transformation matrix for the detailed alignment is estimated after a coarse matching. Iterative closest point algorithm [9] is often used for fine matching.

3D hand-crafted descriptors using local geometric structures of point clouds including FPFH [10], PFH [11], SHOT [12], Spin Images [13], and USC [14] have been proposed. FPFH describes the features by searching neighbor points in a certain range for each feature point and creating histograms about the relationship of angles and distances to each neighboring point. SHOT describes the features by dividing the region around a feature point into 32 regions and for each region, computing the inner products of the normal vectors of the points in the region and the normal vector of the feature point.

3D learning based descriptors including 3DMatch [5], PPFNet [6], perfect Match [15], and FCGF [16] have been proposed. In order to learn point cloud data, point cloud encoders including PointNet [17], PointNet++ [18], DGCNN [19], Spidercnn [20], and Shell Net [21] also have been proposed. For example, PointNet classifies or segments the input point clouds. In those kinds of tasks, PointNet describes feature vectors for each point using a parameter shared model. In a classification task, PointNet describes a class vector by describing a global vector from feature vectors using max pooling or average pooling and inputting the global vector to MLP layers. In a segmentation task, PointNet describes category vectors for each feature vector by concatenating each feature vector and the global vector and inputting them to parameter shared MLP layers.

Basically the above mentioned studies on feature description do not take into account sensing error and feature point detection error, and they are designed based on the assumption that a matched pair of two feature points should be at the same or close location. However in the case of an actual point cloud, a corresponding pair of feature points may includes the previously mentioned registration errors. To deal with those sensing error and feature point detection error, in this study, we introduce a feature descriptor that is robust to the feature point registration error in a point cloud matching.

3. Method

We propose a method of feature description that accepts feature point registration error for aligning point clouds measured with temporal discontinuity. Figure 1 is the concept of our feature descriptor. We design the feature descriptor so that the distance between feature vectors will be close in a feature space if the distance of the feature points is within τ_d cm of each other in the real environment, and the distance between feature vectors will be far in the feature space if the distance of the feature points is further than τ_d cm from each other in the real environment. To achieve this goal, we apply metric learning to train the feature descriptor. We therefore design the feature descriptor, which takes the local area point cloud around the feature point and the entire point cloud as input and train the descriptor by a metric learning using contrastive loss [22] or triplet loss [23].

3.1 Feature Description Using Local and Global Area

We consider that local information may not be enough to estimate whether the feature point pair is close enough to be considered a match or not because the local shape may be similar, but the location may be different in the real environment. For example, the local shape of a flat surface is very common in indoor environments and to find a correct pair of feature points using only local flat surface information is very difficult. On the other hand, the point cloud of a global area generally has rich and unique information but its shape is highly depending on the sensor's position, orientation and so on. To achieve accurate feature description, we combine both local and global information. The global area information has the geometric structure of the whole point cloud, which can solve the problem that the local shape may be similar, but the location may be different in the real environment. In addition, the local area infor-

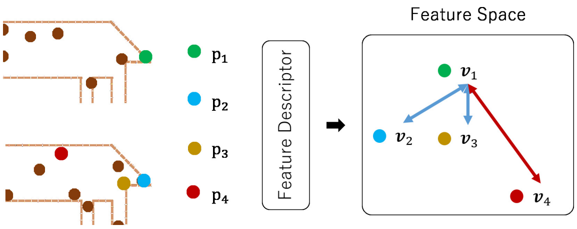
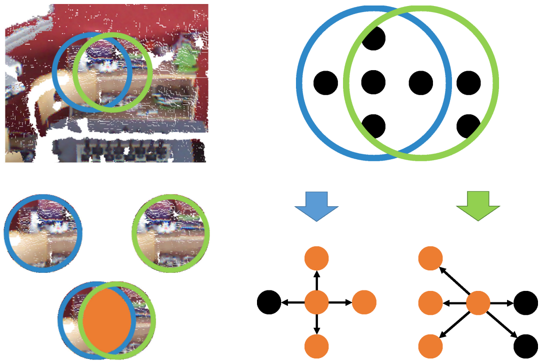


Fig. 1 An example of the feature point registration error. The green feature point $\boldsymbol{p}_1$ sampled from the upper point cloud, the corresponding point at the same location in the lower point cloud is $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$ is a feature point registration error at the same location. $\boldsymbol{p}_4$ is a different position point as $\boldsymbol{p}_1$ in the real environment. Our goal is to make a feature descriptor that describes that the distances between the feature vectors of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2, \boldsymbol{p}_3$ are close in the feature space, and the distance between the feature vectors of $\boldsymbol{p}_1$ and the different position point $\boldsymbol{p}_4$ is far.



(a) A local area overlapping with the feature point registration error

(b) Differences of local features arising from the feature point registration error

Fig. 2 (a) and (b) are properties of a local area point cloud pair with the feature point registration error. The orange area and circles show overlap, and the black circles show non-overlap due to the feature point registration error. The black arrows show the relationships between a feature point and its surrounding points. In (a), blue and green local areas have the feature point registration error, and the appearance varies by the error. However, they have a large overlapping area. (b) shows that relationships between a feature point and its surrounding points vary significantly due to the feature point registration error even when the local area point cloud pair has a large overlapping.

mation does not vary by a difference of a camera position significantly, which can solve the problem that the shape of the global area point cloud varies by the camera position difference. In the feature description using local and global area information, we consider that the local area information should have detailed geometric structure in order to recognize a feature point registration error. On the other hand, the global area information should have a rough geometric structure of the entire point cloud in terms of computational cost and recognition accuracy. As an input point cloud for describing features, we consider that a dense point cloud is appropriate for a local area point cloud to recognize the detailed local geometric structure, and a sparse point cloud is appropriate to recognize the geometric structure of the entire point cloud roughly. Therefore, we define the local area point cloud as the dense point cloud sampled by n_l points existing within τ_l cm from the feature point and the global region point cloud as the sparse point cloud sampled by n_g points from the entire point cloud.

3.2 Local Feature Description

Because the local area point cloud has a higher resolution of local geometric structures compared to the global area point cloud, the local area point cloud is more affected by the feature point registration error than the global area point cloud. Figure 2 shows the influence of the feature point registration on the local area point cloud pair. The appearances of the local area point clouds vary with the feature point registration error, and the relationships between the feature points and the surrounding points vary significantly with the error. However, the local area point cloud pair has a large overlapping area even when the pair has feature point registration error. Therefore, in order to describe a local area feature that is robust to the feature point registration error, we consider describing the whole geometric structure of the local area to be appropriate, instead of describing a relationship between a feature point and surrounding geometric structure. Then, local area feature description robust to the feature point registration error will be possible by detecting the overlap of geometric structures between the local area point clouds.

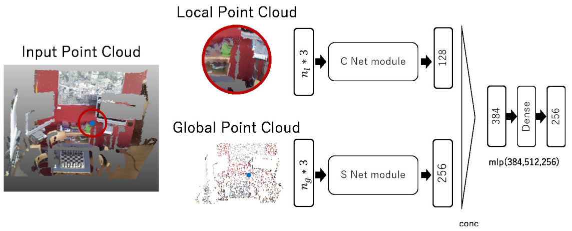


Fig. 3 Our model uses both the local area points around the feature point and the global points sampled from an original point cloud for feature description. In this paper, we call the local area encoder C-Net module and the global area encoder S-Net module, and we call the model combining these two models CS-Net.

3.3 Global Feature Description

The global area point cloud has a lower resolution compared to the local area point cloud and has a whole geometric structure of a point cloud. In addition, because the global area point cloud is sampled over a wide range, the global area point cloud is less affected by the feature point registration error than the local area point cloud. The global area feature should have information about where the feature point is located in the whole point cloud in order to correctly estimate the feature point pairs that have the same local geometric structure but different locations in the real environment as we mentioned in Sect. 3.1.

3.4 Overall Model

Our model consists of two encoding blocks, the local area encoder for a point cloud around the feature point and the global area encoder for a point cloud as shown in Fig. 3. The local area encoder describes a 128-dimensional feature vector, and the global encoder describes a 256-dimensional feature vector. After encoding the local and global area point clouds, these output vectors are combined and input to the fully connected layer to describe a feature vector of the feature point. The fully connected layer consists of input, intermediate, and output layer and they have 384, 512, 256 nodes, respectively.

We designed the local area encoder based on a model that performs a classification task. The classification task model is based on machine learning models such as PointNet [17], PointNet++ [18] and DGCNN [19]. It describes a class vector by describing the feature vectors of the relationships between each input point and other points and summarizing the feature vectors using max pooling. This

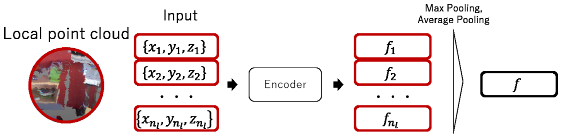


Fig. 4 C-Net module takes a local area point cloud around the feature points from the original point cloud and describe a local area feature. The local area point cloud has a higher resolution compared to the global area point cloud. C-Net module, which is designed based on a classification model, describes a local area feature using the local area point cloud.

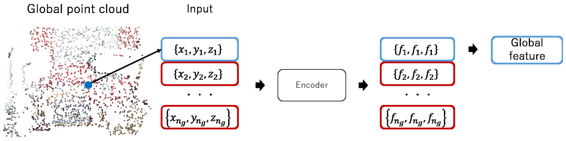


Fig. 5 S-Net module takes the feature point and a global area point cloud sampled a certain number of points from the entire original point cloud. The global area point cloud has a lower resolution compared to the local area point cloud, however it has the entire geometric structure information. S-Net module describes features with relationships between the global area point cloud and feature point.

architecture describes the information of the whole geometric structure of the input point cloud. Therefore, we use the architecture of the classification task model in order to encode the local area point cloud, as shown in Fig. 4. We employed DGCNN to encode the local area point clouds.

We designed the global area encoder based on a model that performs a segmentation task. Generally, a model for a segmentation task based on machine learning describes category vectors for each input point and inputs category vectors to parameter-shared, fully connected layers. For each point, this architecture describes the relationship between the point and the whole geometric structure of the input point cloud. Therefore, we use the architecture of a model for the segmentation task in order to encode the global area point cloud, as shown in Fig. 5. We employed DGCNN for encoding the global area point clouds, as well as the local area encoder.

In this paper, we call the local area describing block, the global area describing block, and the combined model C-Net module, S-Net module, and CS-Net, respectively. We aim to make a feature descriptor robust to feature point registration error by inputting feature point pairs including the errors into CS-Net as training data.

4. Experiments

We evaluate how our method CS-Net describes feature vectors robust to feature point registration error. To check if CS-Net can presume that the pairs of feature point are positive or negative correctly even if the distances of the errors of the positive pairs increase, we evaluate CS-Net using the keypoint matching benchmark [5], [24], [25] on multiple sets of pairs of feature points with different sizes of the feature point registration error of positive pair respectively. The keypoint matching benchmark is defined as

$$\frac{1}{N}\sum_{i=1}^{N} I(f(x_i^n) - f(y_i^n)), \tag{1}$$

$$I(x) := \begin{cases} 1, & \text{if } \|x\| < \theta \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where N is the number of negative feature point pairs, the function $f(\cdot)$ is a point cloud feature descriptor, x_i^n and y_i^n are the negative feature point pair, and θ is a threshold at which the recall of the positive pairs is 95%.

Keypoint matching benchmark calculates a false-positive error rate, and the lower the false-positive ratio, the better the performance.

4.1 Experiment Setup

We prepared datasets for training and evaluation using 3DMatch RGBD benchmark [5]. The datasets in the 3D Match RGBD benchmark consists of Analysis-by-Synthesis [26], 7-Scenes [27], SUN3D [28], RGB-D Scenes v.2 [29] and Halber *et al.* [30]. They contain RGB images, depth images and camera positions obtained from 62 real environments. The dataset we prepared was divided into a training dataset and an evaluation dataset. These datasets have triplets consisting of a base, positive and negative anchors. The training dataset has 30,000 triplets, and the evaluation has 10,000 triplets. The base, positive and negative anchors have a local area point set and a global area point set. We set the number of local area points to 2048, sampling radius threshold to 30 cm, and number of global area points to 2048, represented by n_l, τ_l, and n_g, respectively. CS-Net is implemented with pytorch. We evaluated the both contrastive loss [22] and triplet loss [23], which are generally used in metric learning, to train CS-Net. We train CS-Net for 250 epochs using ADAM [31] with the learning rate of 0.001.

4.2 Comparison Results

To confirm that our method can describe feature vectors robust to feature point registration error, we evaluated our method with a keypoint matching benchmark [5] on the four datasets with different feature point registration errors of the positive pairs. The four datasets have positive pair errors of 0-5 cm, 5-10 cm, 10-15 cm, and 15-20 cm, and the errors of the negative pairs are more than 20 cm for both training and evaluation datasets. We compared our method with handcrafted methods, including FPFH [10], SHOT [12], and Spin Image [13] and a learning based method, 3DMatch [5], using the datasets. The result of this evaluation is shown in Table 1, which summarizes the error rate calculated by Eq. (1) of each feature description model for each size of feature point registration error. The results show that our CS-Net model trained with contrastive loss performed best, and CS-Net trained with triplet loss performed second best. 3D Match, the learning-based feature description method, was better than hand-crafted features such as FPFH, SHOT,

Table 1 We evaluate CS-Net and related works using a keypoint matching benchmark for each dataset that has positive pairs with different sizes of the feature point registration error. This table summarizes the error rate calculated by Eq. (1). Our method is more accurate than related studies and prevents a significant decrease in accuracy due to the errors.

	0-5 cm	5-10 cm	0-15 cm	15-20 cm
CS-Net(triplet)	13.3	14.4	15.6	15.4
CS-Net(cont)	**10.7**	**11.4**	**11.9**	**11.5**
3DMatch(triplet)	49.7	49.9	53.3	57.8
3DMatch(cont)	52.1	55.6	59.2	64.6
FPFH	76.3	79.0	81.1	81.6
SHOT	90.3	91.8	91.8	91.8
Spin Images	92.6	93.2	92.5	93.8

and Spin Image. In addition, the accuracy of 3D Match decreases significantly with an increase of the feature point registration error, whereas the accuracy of CS-Net only decreased marginally. These results show that CS-Net can describe features robust to feature point registration error.

4.3 Combination Study

As we mentioned in Sect. 3, we designed CS-Net based on two assumptions. One is the assumption that the point cloud feature descriptor should use both local and global area information as inputs, and the other is the assumption that C-Net module is suitable for local area encoding and S-Net module is suitable for global area encoding. To confirm these two assumptions, we evaluated the combinations of the different inputs and models. Moreover, we confirm the performances of two loss functions for training CS-Net. The results are shown in Table 2.

4.3.1 Local and Global Inputs

CS-Net consists of a C-Net module to encode a local area and an S-Net module to encode a global area. In this section, we confirm that whether we need to use both modules by comparing C-local-Net, the model for only the local area and S-global-Net, the model for only the global area. Table 2 shows that performance of both inputs of local and global area are better than the performances of C-local-Net and S-global-net with contrastive and triplet loss. In addition, S-global-Net that uses only global areas performs better than a C-local-Net that uses only local areas. Therefore the models using only local area information do not have the ability to describe the features in the matching of point clouds measured discontinuously, and global area information is very important. We confirmed that the use of global area information improves the feature description and the use of local area information together with global area shows better performance. However, the feature description from only a local area is inefficient.

Table 2 We investigate the performance of the proposed method CS-Net with different combinations of local area and global area point clouds and C-Net module and S-Net module of encoders. This table also summarizes the error rate calculated by Eq. (1), as well as Table 1.In the case of the both inputs of global and local point cloud, the model name is described in the order of the model describing the features of the local area and the model describing the features of the global area. For example, CS-Net means a combination of C-Net module for the local feature description and S-Net module for the global feature description. In the case of the both inputs of global and local point cloud, the model name is described in the order of the feature descriptor and used input. For example, C-local-Net means the model consisting only C-Net module for the local feature description. The experimental results shows that CS-Net achieved the highest performance and describing the global area point clouds by S-Net module is significant for the feature description.

	0-5 cm	5-10 cm	0-15 cm	15-20 cm
CS-Net(triplet)	13.3	14.4	15.6	15.4
CS-Net(cont)	**10.7**	**11.4**	**11.9**	**11.5**
CC-Net(triplet)	40.3	42.7	44.8	47.7
CC-Net(cont)	44.8	46.6	48.5	52.0
SS-Net(triplet)	16.0	16.5	17.8	19.2
SS-Net(cont)	13.4	13.8	14.0	14.4
SC-Net(triplet)	34.7	37.8	42.4	47.5
SC-Net(cont)	48.4	51.0	52.7	55.2
C-local-Net(triplet)	41.4	43.7	47.1	49.5
C-local-Net(cont)	44.9	45.8	48.0	50.8
S-local-Net(triplet)	62.8	65.2	65.8	68.9
S-local-Net(cont)	49.3	51.9	53.3	57.6
C-global-Net(triplet)	95.1	95.1	95.1	95.0
C-global-Net(cont)	94.9	95.0	95.1	95.0
S-global-Net(triplet)	17.7	18.7	18.6	17.7
S-global-Net(cont)	21.9	20.2	22.8	21.0

4.3.2 C-Net Module for a Local Area and S-Net Module for a Global Area

In CS-Net, C-Net module describes features of the local area and S-Net module describes features of the global area. In this section, we confirm which module is more suitable for local and global area encoding by comparing the performance of C-local-Net and S-local-Net, and the performance of C-global-Net and S-global-Net. Then we compare the performances of all combination of two encoders in the case of the both area.

Table 2 shows that C-local-Net performs better than S-local-Net, and S-global-Net performs better than C-global-Net unlike the case with only local areas. Therefore when using either one of the two areas, C-Net module is suitable for the local area feature description and S-Net module is suitable for the global area feature description. In the case of a feature description using both local and global areas, CS-Net, SS-Net, SC-Net, and CC-Net perform better in that order when using triplet loss, and CS-Net, SS-Net, CC-Net, and SC perform better in that order when using contrastive loss. Therefore when using the both local and global areas, C-Net module is suitable for the local area feature description and S-Net module is suitable for the global area feature description. The performances of both CC-Net and SC-Net, which uses C-Net module to describe the global area, are lower than that of CS-Net and SS-Net because C-Net module can not learn the global area information. In addition, the global information is supposed to have a very significant role as discussed in Sect. 4.3.1, even when both areas are used because CS-Net and SS-Net have better performance than SC-Net and CC-Net. We therefore conclude that C-

Net module is suitable for the local area feature description because CS-Net performs better than SS-Net.

Compared with CS-Net, SS-Net and S-global-Net, which uses the S-Net module for global feature description, C-local-Net and S-local-Net, which use only local area for feature description, tend to have a higher error rate as the feature point registration error increases. This suggests that using the information in the global area is important for the feature description robust to the feature point registration error because the information in the local area changes significantly with feature point registration errors, and the information in the global area is less affected by that error.

4.3.3 Triplet Loss vs. Contrastive Loss

We used triplet loss and contrastive loss for training the models. Table 2 shows that CC-Net, SC-Net, C-local-Net, S-global-Net perform better with triplet loss, while CS-Net, SS-Net, S-local-Net perform better with contrastive loss. C-global-Net has little difference in performance due to the loss functions. The performances have no tendency to be better with contrastive or triplet loss when using only local area, only global area, and the both areas. In addition, the difference in the number of trainable parameters does not affect the evaluation performance regularly depending on the loss function used. Therefore, it is not possible to systematically determine the performance of triplet loss or contrastive loss due to differences in model structure, differences in input data, and differences in the number of trainable parameters.

5. Conclusion

In order to align real environmental point clouds with different initial camera positions, we proposed a feature descriptor called CS-Net that is robust to feature point registration error in point cloud matching. We confirmed that CS-Net can discriminate between pairs of feature points that are close to each other and those that are distant with higher accuracy than the other methods compared in this paper. We also confirmed that CS-Net can estimate whether the feature point pair is close enough to be considered a match or not, even when feature point registration error is large. Moreover, in the combination study, we confirmed that CS-Net, which uses both local and global area point clouds as inputs, outperformed C-local-Net and S-global-Net, which exclusively use local or global areas, by a margin of 34.0% error and 6.8% error on average, respectively. We also found that the C-Net module is suitable for local area encoding and the S-Net module is suitable for global area encoding.

Acknowledgments

This work is supported by DAIKIN INDUSTRIES, Ltd. Thank you to Jason Orlosky for checking the manuscript.

References

[1] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," 2011 10th IEEE International Symposium on Mixed and Augmented Reality, pp.127–136, IEEE, 2011.

[2] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2100–2106, IEEE, 2013.

[3] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," The International Journal of Robotics Research, vol.35, no.14, pp.1697–1716, 2016.

[4] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," ACM Transactions on Graphics (ToG), vol.36, no.4, p.1, 2017.

[5] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1802–1811, 2017.

[6] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.195–205, 2018.

[7] H. Deng, T. Birdal, and S. Ilic, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," Proc. European Conference on Computer Vision (ECCV), pp.602–618, 2018.

[8] Y. Díez, F. Roure, X. Lladó, and J. Salvi, "A qualitative review on 3d coarse registration methods," ACM Computing Surveys (CSUR), vol.47, no.3, pp.1–36, 2015.

[9] P.J. Besl, N.D. McKay, and P.S. Schenker, "A method for registration of 3-d shapes," Sensor fusion IV: control paradigms and data structures, pp.586–606, International Society for Optics and Photonics, 1992.

[10] R.B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," 2009 IEEE International Conference on Robotics and Automation (ICRA), pp.3212–3217, IEEE, 2009.

[11] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3384–3391, IEEE, 2008.

[12] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," Computer Vision and Image Understanding, vol.125, pp.251–264, 2014.

[13] A.E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," IEEE Trans. Pattern Anal. Mach. Intell., vol.21, no.5, pp.433–449, 1999.

[14] F. Tombari, S. Salti, and L. Di Stefano, "Unique shape context for 3d data description," Proc. ACM Workshop on 3D Object Retrieval, pp.57–62, 2010.

[15] Z. Gojcic, C. Zhou, J.D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.5545–5554, 2019.

[16] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," Proc. IEEE International Conference on Computer Vision, pp.8958–8966, 2019.

[17] C.R. Qi, H. Su, K. Mo, and L.J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.652–660, 2017.

[18] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," Advances in Neural Information Processing Systems, pp.5099–5108, 2017.

[19] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, and J.M.

Solomon, "Dynamic graph cnn for learning on point clouds," ACM Transactions On Graphics (ToG), vol.38, no.5, pp.1–12, 2019.

[20] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," Proc. European Conference on Computer Vision (ECCV), pp.87–102, 2018.

[21] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics," Proc. IEEE International Conference on Computer Vision, pp.1607–1616, 2019.

[22] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp.1735–1742, IEEE, 2006.

[23] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1386–1393, 2014.

[24] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.1, pp.43–57, 2010.

[25] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A.C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3279–3286, 2015.

[26] J. Valentin, A. Dai, M. Nießner, P. Kohli, P. Torr, S. Izadi, and C. Keskin, "Learning to navigate the energy landscape," 2016 Fourth International Conference on 3D Vision (3DV), pp.323–332, IEEE, 2016.

[27] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.2930–2937, 2013.

[28] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," Proc. IEEE International Conference on Computer Vision, pp.1625–1632, 2013.

[29] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3d scene labeling," 2014 IEEE International Conference on Robotics and Automation (ICRA), pp.3050–3057, IEEE, 2014.

[30] M. Halber and T. Funkhouser, "Fine-to-coarse global registration of rgb-d scans," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1755–1764, 2017.

[31] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

Kenshiro Tamata graduated from Osaka University in 2018 and completed the M.E. in 2020. He is currently a doctor student in Osaka University. His interests includes Computer Vision and Pattern Recognition.

Tomohiro Mashita graduated from Osaka University in 2001 and completed the M.E. and doctoral program in 2003 and 2006, respectively. He was a postdoctoral fellow at Osaka University from 2006 to 2008. He was a senior research fellow at Graz University of Technology from 2012 to 2013. He is currently an associate professor at Cybermedia Center, Osaka University. His research interests includes Computer Vision and Pattern Recognition. He is a member of the IEICE, the IPSJ, and the IEEE.