IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 19, 2019, accepted October 24, 2019, date of publication November 7, 2019,
date of current version November 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2952157

3D Point Cloud Retrieval With Bidirectional Feature Match

NARANCHIMEG BOLD[1], (Student Member, IEEE), CHAO ZHANG[2], (Member, IEEE), AND TAKUYA AKASHI[1], (Member, IEEE)

[1]Department of Design and Media Technology, Graduate School of Engineering, Iwate University, Iwate 020-8551, Japan
[2]Graduate School of Engineering, Information Science, University of Fukui, Fukui 910-8507, Japan

Corresponding author: Takuya Akashi (akashi@iwate-u.ac.jp)

This work was supported in part by the JSPS KAKENHI under Grant JP16K01647, Grant JP16KK0069, and Grant JP19K11515, and in part by the Mongolia–Japan Higher Engineering Education Development Project (MJEED) under Grant J14C16.

ABSTRACT In the recent decade, the development of 3D scanners brings the expansion of 3D models, which yields in the increase of demand for developing effective 3D point cloud retrieval methods using only unorganized point clouds instead of mesh data. In this paper, we propose a meshing-free framework for point cloud retrieval by exploiting a bidirectional similarity measurement on local features. Specifically, we first introduce an effective pipeline for keypoint selection by applying principal component analysis to pose normalization and thresholding local similarity of normals. Then, a point cloud based feature descriptor is employed to compute local feature descriptors directly from point clouds. Finally, we propose a bidirectional feature match strategy to handle the similarity measure. Experimental evaluation on a publicly available benchmark demonstrates the effectiveness of our framework and shows it can outperform other alternatives involving state-of-the-art techniques.

INDEX TERMS Point cloud retrieval, 3D shape retrieval, bidirectional feature match.

I. INTRODUCTION

With the development of 3D data acquisition technologies, point clouds can be generated faster with low cost [1], [2], which leads to rapid growth of the 3D point data stored in databases. As such, automatically organizing and retrieving the 3D models from databases becomes essential to various of applications in diverse fields such as computer graphics and computer vision. There has been a surge of interest in methods for retrieval of 3D models [3]–[9] from databases. Most of the existing methods employed feature descriptors (global or local) to describe 3D objects, where the 3D objects are usually represented in the form of triangular/polyhedral meshes. Although creating meshes from point clouds is a well-studied topic [10], [11], sometimes it can be quite complex due to the lack of connectivity of higher-level information about the underlying surface [11], [12], especially when the input point cloud data has missing parts. Therefore, it is necessary to develop 3D shape retrieval methods which can conduct retrieval directly with unorganized point clouds (i.e., without surface reconstruction).

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

Measuring the similarity between 3D objects is an essential and fundamental task in 3D shape retrieval. A common strategy in shape similarity assessment is to evaluate the similarity score between the shapes in terms of distances with associated feature descriptors. Various similarity/distance measures [13], [14] have been proposed and widely applied in 3D object retrieval tasks, such as the L_p distance, the Hausdorff distance, and the earth mover's distance. Meanwhile, the term dissimilarity is often conceived in terms of similarity [3], [15]. Despite the success of applying similarity measures in mesh retrieval tasks, the evaluation of the effectiveness in 3D point cloud retrieval has been sparsely treated so far.

In this paper, we tackle the problem of 3D point cloud retrieval by extending the bidirectional similarity measure [16] to build a meshing-free framework. At first, an effective keypoint detection procedure is conducted, where each point cloud is normalized by using PCA [17] and keypoints for feature extraction are detected by thresholding the local similarity of normals. Then, we utilize point cloud based RoPS [18] descriptor to compute the feature descriptor directly from point clouds. Finally, as the main contribution, we estimate the similarity between point clouds by a bidirectional similarity measurement, which is extended from the

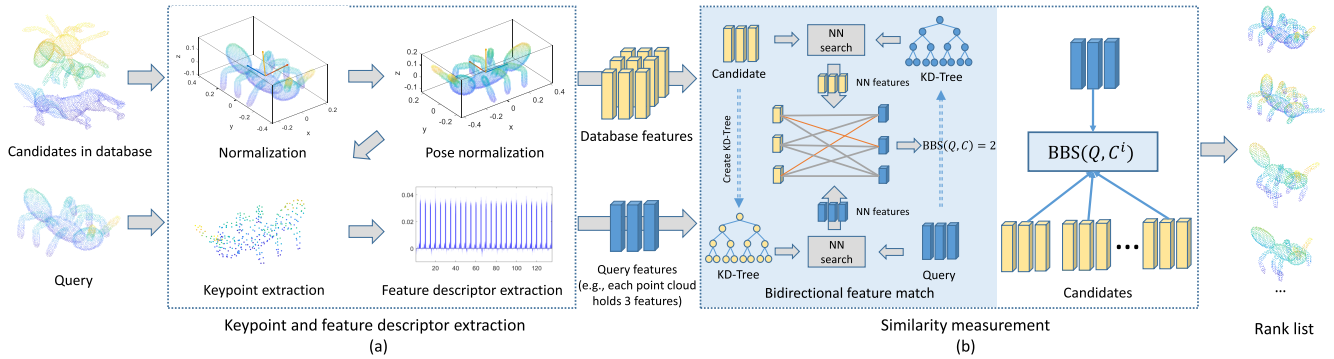


FIGURE 1. A general overview of our 3D point cloud based retrieval framework. (a) we illustrate the pipeline for keypoint selection and RoPS feature extraction, where each point cloud's coordinates are normalized by the diagonal of the point cloud's bounding box and aligned using PCA based pose normalization. Keypoints for feature extraction are then detected by thresholding local similarity of normals (i.e., points with a high variation of normals within a local area are selected as keypoints). The RoPS feature descriptor is computed at each keypoint. (b) The shaded part presents a bidirectional feature match mechanism: first, KD-trees are created from candidate and query features respectively, and search of nearest neighbor (NN) features over KD-trees for each keypoint is conducted for the query and each candidate. Second, the bisimilarity of NN features between candidates and the query are evaluated to define the best-buddies similarity (BBS) score. Finally, candidates are ranked in descending order according to the BBS score.

distance measurement of best-buddies similarity (BBS) [16] for similarity measurement in the feature domain of 3D point clouds. Figure 1 illustrates the overall procedure of our 3D point cloud retrieval framework. We demonstrate in the experiment that our 3D point cloud retrieval framework achieves competitive performance compared against methods based on alternative similarity measures and the state-of-the-art descriptors. It is important to note that our proposed framework is an efficient, easy-to-apply point cloud retrieval method rather than a learning-based method which is data-dependent.

II. RELATED WORK

Giving a 3D object as a query, a content-based 3D object retrieval system analyzes the query and retrieves ranked 3D objects from the collections of a 3D dataset according to measured similarity. The key components here is to design a discriminative feature descriptor and a similarity measure for 3D objects, which have been widely studied [3], [4], [19]. Global or local features can be employed to measure the similarity between two 3D objects. Global features have the ability to encode an entire object with a single vector, and we refer to [3] as a survey article for an overview of 3D retrieval methods based on global features. However, such methods ignore the shape details and the research efforts in recent years have focused on developing more discriminative local features which are distinctive and robust against occlusion and clutter [19]. The local features are based on local geometric features extracted from interest points/keypoints, which hold rich information that allow for effective description and matching. The similarity search is usually performed in the extracted feature domain. In this section, we review the most related two key components in 3D object retrieval: 3D local feature descriptor and similarity measure.

A. 3D LOCAL FEATURE DESCRIPTOR

The local feature descriptors are closely related to 3D keypoint detectors, which identify appropriate points that are distinctive. A number of 3D keypoints detectors have been proposed [20], [21]. Although various feature descriptors have been proposed according to specific applications such as 3D object recognition, 3D retrieval, it is difficult for the users to choose an appropriate feature descriptor for specific purposes [19]. In general, a good feature descriptor should be highly descriptive to provide a comprehensive and predominant representation of local surface, and also should be computationally efficient, compact and robust to a common nuisances such as noise and varying point cloud resolution [18], [19]. A large variety of 3D local feature descriptors have been proposed, including spin images [22], fast point feature histograms (FPFH) [23], signature of histograms of orientation (SHOT) [24], rotational projection statistics (RoPS) [18], and TriSI [25]. Comprehensive evaluations of local feature descriptors can be found in [19], [26]. Among these, RoPS descriptor is reported to be robust against a set of variations including noise, clutter, occlusion, and can further achieve superior performance for feature matching in terms of precision and recall [18], [19]. On the other hand, the bag-of-words (BOW) approach is popular for processing the raw features, which demonstrates superior retrieval performance for both articulated and rigid objects [27], [28]. Such method extracts local feature descriptors from 3D objects which are later grouped in clusters (e.g., using k-means) and represent them into histograms which become the feature vector of the 3D objects. BOW framework combined with RoPS descriptor outperforms competitive methods in the SHREC'17 contest [7] on non-rigid 3D point cloud retrieval task.

B. SIMILARITY MEASURE

The goal of feature matching in feature-based 3D object retrieval is to find keypoint correspondences across 3D dataset for given keypoints and the according local feature descriptors. The performance mainly depends on the similarity measure used to compute the distance between pairs

of descriptors [3], [15]. The basic similarity measures are the Manhattan/L1 and Euclidean/L2 distance [29], mostly due to their simplicity and computational efficiency. Other distance measures have also been widely applied in 3D object retrieval, such as the Hausdorff distance applied to [30], the earth mover's distance applied to [31], and bipartite graph matching applied to [9]. Investigations have also been made to measure the dissimilarity between non-rigid 3D models. For instance, Memoli and Sapiro [32] introduce a theoretical framework to compare 3D shapes based on the Gromov-Hausdorff distance directly. Moreover, Johnson and Hebert [22] use a linear correlation coefficient to match the spin images of the scene with the spin images of the models. Hetzel *et al.* [33] match multidimensional histogram features of two 3D models using the chi-squared test to recognize 3D object.

Dekel *et al.* [16] introduce a robust, parameter-free, bidirectional similarity measure for template matching and achieved state-of-the-art performance on template matching. BBS measures the similarity between two image patches by counting best-buddies pairs (BBPs) which are a pair of patches from each image, if two patches are the nearest neighbor of each other.

Benefiting from RoPS feature descriptor and inspired by BBS, our 3D point cloud retrieval framework takes account both accuracy and computational efficiency.

III. METHODOLOGY

As it has also been concluded in Fig. 1, our 3D point cloud retrieval framework includes three main parts: (1) pose normalization, (2) keypoint and feature descriptor extraction, (3) bidirectional feature match, which will be introduced in detail respectively in this section.

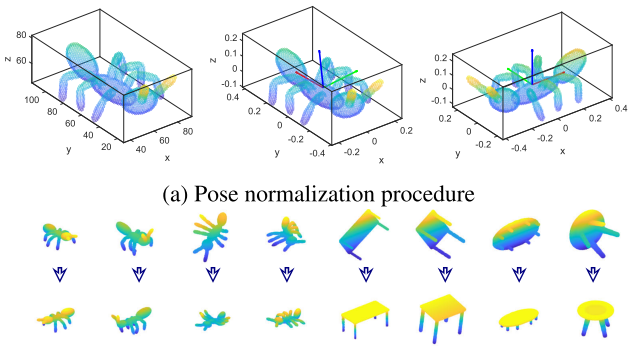
A. POSE NORMALIZATION

The goal of this procedure is to place the point clouds into a canonical coordinate to alleviate the variation concerning transformations such as rotation, translation, and scaling of the objects. Specifically, we first normalize the coordinates of the points by setting the length of the diagonal of the point cloud's bounding box to 1. After the normalization, the eigenvectors are calculated to identify the dominant direction which holds the largest possible variance and the point clouds are aligned with the principal axes. Figure 2a shows the pose normalization procedure and Fig. 2b shows some examples after pose normalization.

B. KEYPOINT AND FEATURE DESCRIPTOR EXTRACTION

Instead of using randomly selected points or grid sampled points as keypoints for feature extraction, we design a simple technique to select distinctive keypoints directly from point clouds, where the variation of local normals defines the "distinctiveness".

Let $P = \{p_1, \ldots, p_i, \ldots, p_n\}$ be points of a certain point cloud, and the normal $\vec{n}_i$ at p_i is calculated using k-neighborhood of p. For each p_i, we calculate the variation of local



(a) Pose normalization procedure

(b) The result of the pose normalization

FIGURE 2. (a) An input point cloud (left), is scaled using the diagonal of the point cloud's bounding box, centering at the origin (middle), and rotated according to the principal axes to have consistent orientation for the same category (right). Vectors in red, blue, and green are the eigenvectors of the covariance matrix. (b) Example point clouds of ants (articulated) and tables (non-articulated) in different initial poses (top row). The normalized results (bottom row).

normals V_i within a sphere (local space) with support radius r. The variation is defined by averaging the $cos\theta_j$ between p_i and each neighbour point p_j, which can be easily calculated from the inner product of normal vectors,

$$V = \frac{1}{\sum_{m=1}^{n} \mathbb{1}\left(d_j \leq r\right)} \sum_{j:d_j \leq r} cos\theta_j, \tag{1}$$

where smaller V indicates larger variation, $d_j = \|p_j - p_i\|_2$ and $cos\theta_j = \vec{n}_i \cdot \vec{n}_j$. $\mathbb{1}(.)$ is an indicator function to turn true/false into 1/0. Keypoints are selected by setting a threshold th on the variation,

$$p_i = \begin{cases} \text{is keypoint}, & \text{if } V_i < th \\ \text{is not keypoint}, & \text{otherwise.} \end{cases} \tag{2}$$

Furthermore, it is essential to prevent object-specific features from filtering out (e.g., although the flat surface of a chair or a table has a low variation of local normals, it is a distinctive feature for both objects). For this reason, we add a constraint that besides Eq. 2, if $\vec{n}_i$ dominates the local space, we select p_i as a keypoint. An example of keypoint selection with/without the additional constraint is shown in Fig 3. It is worth pointing out that there is a trade-off between the retrieval accuracy and computational cost with respect to the number of extracted keypoints. To balance it, we uniformly sample the remaining points shown in Fig. 3 with a grid sampling parameter $grid\_size$.

We utilize point cloud based RoPS [18] descriptor to calculate the feature descriptor directly from point clouds. The RoPS descriptor is based on unique, repeatable, and robust local reference frame (LRF). The idea is, at first, an LRF is constructed for each keypoint. Points within the local space (with support radius equals to r_{rops}) are aligned with the LRF to achieve invariance against rigid transformations (i.e., rotations and translations). The points within the local space are then rotated d_{rops} times around the three coordinate axes respectively. For each rotation, the points are further

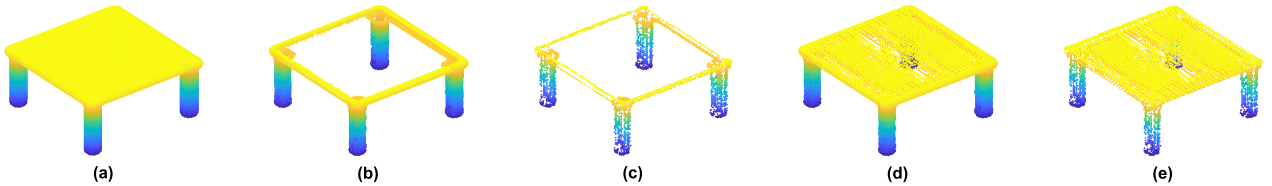


FIGURE 3. An example of keypoint extraction with/without constraint. Support radius r is set to 0.013. (a) An example of point cloud of table. Without constraint: (b) the threshold of V, $th = 0.8$, (c) $th = 0.2$. With constraint: (d) $th = 0.8$, (e) $th = 0.2$.

projected onto three 2D planes (i.e., xy, yz and xz). Then each plane is divided into some bins (bin_{rops}) and five statistics including low-order central moments and entropy are calculated from each histogram. Finally, the RoPS descriptor is generated by concatenating all these statistics of all rotations and projections. Influence of the parameters will be discussed in Section IV.

C. BIDIRECTIONAL FEATURE MATCH

The goal of this section is to calculate the similarity between features of a query and a candidate point cloud by exploiting best-buddies similarity (BBS) [16]. The difference is, our method counts the best-buddies pairs (BBP) bidirectionally between features of query and candidate in terms of the RoPS descriptors. The BBS is originally designed for 2D template matching.

Let $Q = \{q_n\}_{n=1}^{N}$ denoting the features of the query point cloud with N keypoints and $C = \{c_m\}_{m=1}^{M}$ be features of a candidate point cloud with M keypoints, where $q_n, c_m \in \mathbf{R}^{3\times3\times5\times d_{rops}}$. KD-trees denoted by T_Q and T_C are created from the features of the query and the candidate point cloud respectively. Then for each q_n and c_m, its NN can be found by searching T_C and T_Q respectively. q_n and c_m form one BBP if and only if c_m is the nearest neighbor of q_n in the KD-Tree T_C, and vice versa. Formally,

$$\begin{aligned}&\mathrm{BBP}(q_n, c_m, T_Q, T_C)\\ &= \begin{cases}1, & \mathrm{NN}(q_n, T_C) = c_m \wedge \mathrm{NN}(c_m, T_Q) = q_n\\ 0, & \text{otherwise},\end{cases}\end{aligned} \tag{3}$$

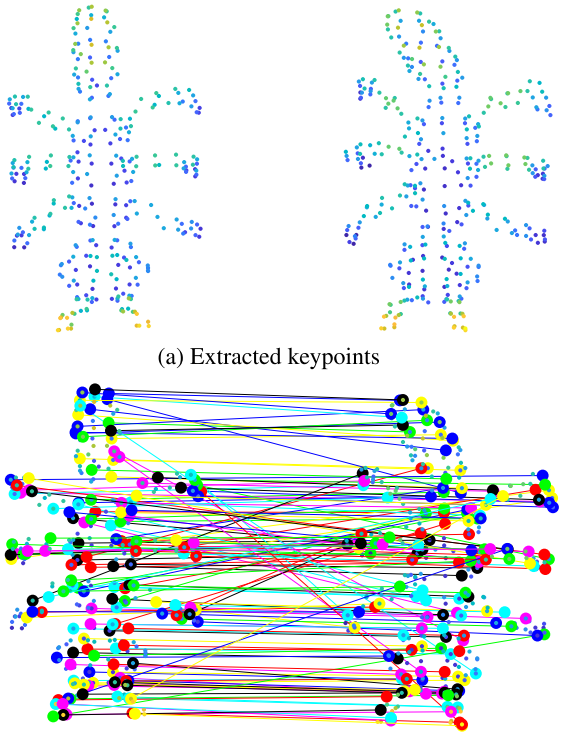
where $\mathrm{NN}(q_n, T_C)$ returns the NN of q_n in KD-tree T_C. The similarity function BBS can be further defined as

$$\mathrm{BBS}(Q, C) = \sum_{n=1}^{N}\sum_{m=1}^{M}\mathrm{BBP}(q_n, c_m, T_Q, T_C). \tag{4}$$

The BBS score is computed for each candidate C in the database by counting the number of BBPs between Q and C. Finally, the candidates are ranked in descending order according to their BBS scores. Figure 4 illustrates an example of BBPs between a query and a candidate point cloud. Despite the non-rigid deformation of the object, BBPs include many inlier matches to show the effectiveness.

IV. EXPERIMENT

In this section, we comprehensively discuss the effect of parameters and the performance of the proposed framework compared to other alternatives. Conventional similarity measures and competitive descriptors such as SHOT [24], SI [22] and TriSI [25] are taken into account.



(a) Extracted keypoints

(b) Bidirectional feature matching

FIGURE 4. (a) Keypoints extracted from query (left) and candidate (right) point cloud. (b) The BBPs between the query and candidate point cloud. Each BBP is illustrated by a line in a random color.

A. EXPERIMENTAL SETTING

We evaluate our method with different settings on McGill 3D shape dataset [34], which is a well-known non-rigid 3D shape benchmark built for retrieval purpose. The McGill dataset contains 456 objects from 19 classes (classes with articulating parts such as 'ants', 'crabs', 'humans' etc., and classes without articulation such as 'tables', 'cups', 'birds' etc.,). We randomly select three objects from each class to form a query set of 57 objects in total. For performance evaluation, we utilize mean average precision (mAP) and average retrieved time. mAP is computed by first sorting point clouds in descending order of relevance for each query, and then averaging average precision (AP) calculated from each query. Mean retrieval time is the average elapsed time taken to extract the features (keypoint + descriptor) and compute the relevance

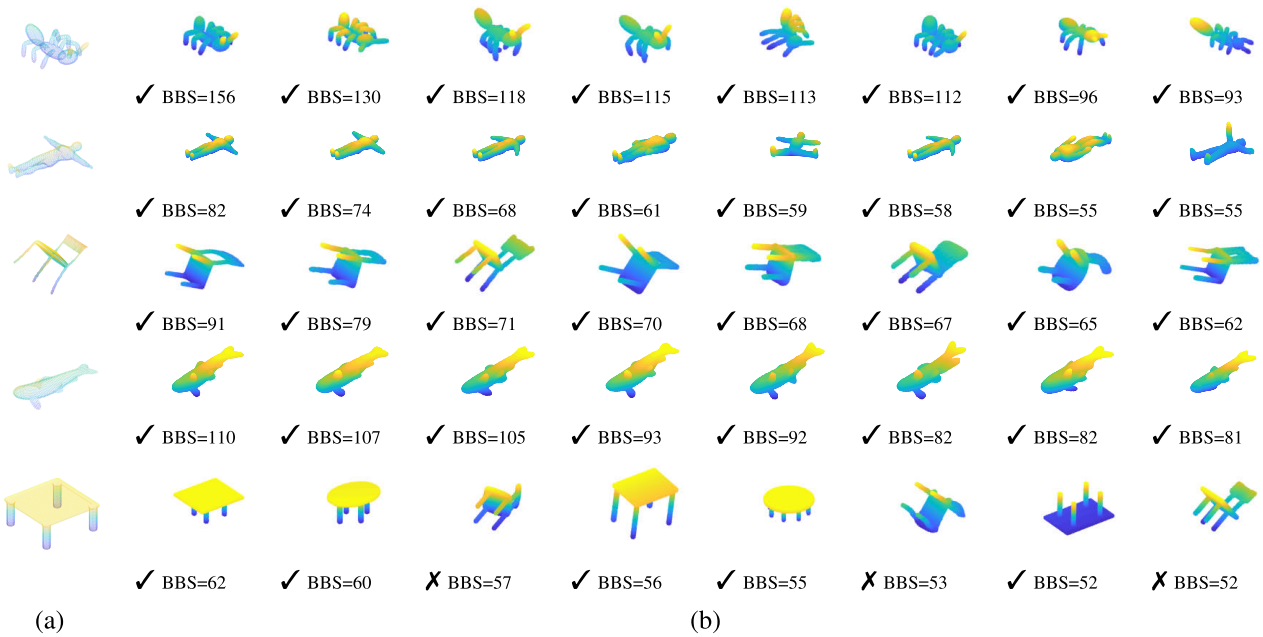


FIGURE 5. Retrieval examples from the McGill dataset. (a) Input queries. (b) Top-8 results retrieved by the proposed method. The retrieved objects are ranked from left to right decreasingly in terms of BBS score, which is presented in the bottom of each result. Cross and check marks show the correctness of the results. A retrieval result is treat to be failed if its class label differs from the query.

of per query. For evaluation, we implement a prototype of our framework in MATLAB. As to the RoPS descriptor, we modify the MATLAB code[1] to compute RoPS descriptors from point clouds directly. The MATLAB code of SHOT descriptor is provided by the authors and the MATLAB code of SI, TriSI are available online [2][3]. All experiments are conducted on a computer with an Intel Core i7 2.9 GHz and 20GB of RAM.

Our framework depends on several major parameters: the support radius r, the threshold th on V, and the $grid\_size$ which are used for keypoint selection and simultaneously the RoPS descriptor parameters: the support radius r_{rops}, the number of rotations d_{rops}, and the number of partition bins bin_{rops}. We set $r = 0.013$, threshold $th = 0.8$, $grid\_size = 0.05$, $r_{rops} = 15$, $d_{rops} = 3$, and $bin_{rops} = 3$ by default in this paper which balances the effectiveness and efficiency well, according to the analysis in Section IV-E.

B. QUALITATIVE ANALYSIS

To justify the effectiveness of the proposed method, in this section, we visually demonstrate the results of the bidirectional feature match. First, we show some retrieval results in Fig. 5. As it can be observed, the retrieved top-8 objects all belong to the same class of their corresponding queries except ''tables'', which is likely to be mistaken for ''chairs''.

Furthermore, we analyze the BBPs between the objects in the same/different classes in Fig. 6. As it can be observed clearly that the number of BBPs between the same class (top row) is larger than the number of BBPs between different class (bottom row), which well demonstrates the ability of bidirectional feature match to distinguish object in the class of the query from candidates in other classes.

C. QUANTITATIVE ANALYSIS

In Table 1, we report the quantitative results of our method. By applying our keypoint extraction, the mAP improves by $0.6\% \sim 0.8\%$ compared to the keypoint extraction based on random sampling (5% of the points are randomly selected to be keypoints) or grid sampling ($grid\_size = 0.05$). The analysis of keypoint extraction further discussed in Section IV-E. Furthermore, bidirectional feature matching achieves the best performance compared to other similarity measures.

To further analyze the retrieval results in detail, a confusion matrix of the retrieval results is visualized in Fig. 7. Based on the APs of three queries per class, the confusion matrix shows the mAP of each class with respect to all classes. It can be observed that most of the diagonal cells achieve the highest values compared to other cells in the same row, showing the correctness of the retrieval. The highlighted off-diagonal cells correspond to incorrectly retrieved results. As an example of failure, the mAP of query class 'octopuses' (5th row) with respect to the retrieval results of 'spiders' (9th column) is 35.3%, which is greater than the ground truth 'octopuses' (5th column)(26.2%). Also, the mAP of 'snakes' (7th row), 'dinosaurs' (15th row), and 'tables' (last row) differ from the ground truths. In the case of 'dinosaurs', the mAP of 'four-limbs' is greater than the ground truth by 20%. This may be attributed to the fact that the dinosaurs also have four-limbs.

[1] http://yulanguo.me/img/RoPS.rar
[2] http://www.csse.uwa.edu.au/~ajmal/code/calcSpinImages.m
[3] http://yulanguo.me/img/TriSI.rar

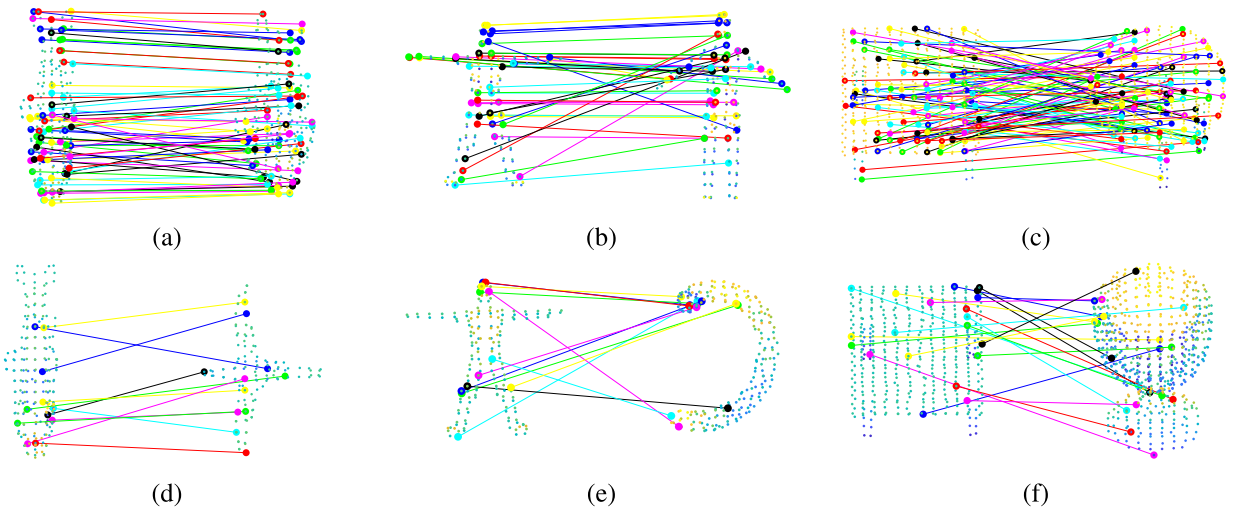


FIGURE 6. Illustration of BBPs between objects in same/different classes. For each pair, query and candidate point clouds are presented side-by-side and connected by lines representing BBPs. In the top row, query and candidate belong to the same class. (a) fishes, (b) humans and (c) tables. In the bottom row, objects are in different classes. (d) fishes vs. birds, (e) humans vs. snakes, and (f) tables vs. cups.

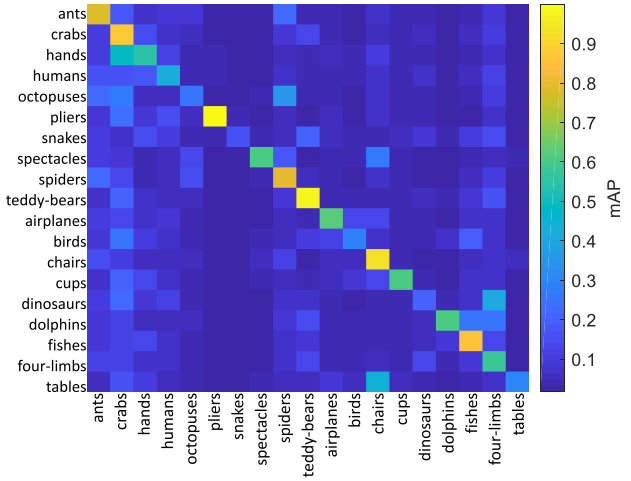


FIGURE 7. Confusion matrix of the retrieval results by our method. Each cell shows the mAP of each query class from the top row to the bottom row, which is calculated by averaging the APs of the three queries per class. The class specified statistics of the retrieval results are presented in each column.

D. COMPARISON AGAINST OTHER SOLUTIONS

We design comparative alternatives by combining several state-of-the-art feature descriptors with classic single directional similarity measures. For feature descriptor, we studied SHOT [24], spin images [22] and TriSI [25]. Table 1 presents a comprehensive performance evaluation of the proposed framework, compared to other solutions. For SHOT, spin images and TriSI, we utilize triangle meshes constructed from the original point clouds for feature extraction, the support radius setting follows the RoPS descriptor.

We can make several observations from Table 1. First, bidirectional distance achieves the best performance compared to other similarity measures for all descriptors. However, the calculation of bidirectional distance takes longer time compared to single directional similarity measures. Second, solutions with SI and SHOT descriptors are the most efficient while solutions with RoPS and TriSI cost more time. Lastly, BBS combined with RoPS descriptor and our keypoint detection clearly outperforms other solutions in terms of mAP. Despite that, the proposed method costs more mean retrieval time mainly because of the extraction of query's RoPS descriptor.

E. EFFECT OF MAJOR PARAMETERS

In this section, we explore the effect of each parameter brings to the performance of our retrieval framework. Parameters r, th, $grid\_size$ of keypoint extraction procedure, r_{rops}, d_{rops}, bin_{rops} of RoPS descriptor extraction are studied. The results are concluded in Fig. 8 and Fig. 9 respectively. In Fig. 8, besides the proposed keypoint extraction, we also consider both random sampling with a fixed number of keypoints and grid sampling with parameter $grid_{size}$. From the first column of Fig. 8, we can see that different settings of the support radius r and the threshold value th affect the performance to a certain extent. The best performance achieved when $r = 0.013$ and $th = 0.8$. From the second column of Fig. 8, we can see that our keypoint extraction method improves the performance of the proposed framework while keeping efficiency. It decreases the mean retrieval time by 15 seconds when the $grid\_size = 0.03$. It is worth pointing out that the number of keypoints directly affects the effectiveness and efficiency of the proposed framework. Moreover, as decreasing the $grid\_size$ will cost more computational resources and make the method impractical, we set the $grid\_size = 0.05$ in this paper for practical comparison. By observing the third and the fourth columns of Fig. 8, we can see that the BBS significantly outperforms classic measures combining either of the keypoint sampling methods and descriptors.

The parameter tuning of RoPS descriptor can also affect the whole performance. We compare the proposed framework

TABLE 1. Performance evaluation of the proposed framework against comparative methods with respect to mAP and mean retrieval time. Techniques utilized in our framework are highlighted in blue. The computational time is mainly cost in the stage of extracting the feature descriptor of the query.

Keypoint extraction	Feature descriptor	Similarity measure					
		L1		L2		BBS	
		mAP	time (s)	mAP	time (s)	mAP	time (s)
Random sampling	RoPS	0.42	29.27	0.48	28.87	**0.60**	66.66
Grid sampling ($grid\_size = 0.05$)	RoPS	0.40	7.75	0.47	9.09	0.59	12.29
Our keypoint extraction	RoPS	0.41	8.21	0.47	8.18	**0.60**	12.68
Our keypoint extraction	SHOT	0.16	7.74	0.26	7.37	0.42	10.56
Grid sampling ($grid\_size = 0.05$)	SHOT	0.16	6.59	0.25	6.54	0.41	10.10
Grid sampling ($grid\_size = 0.05$)	Spin Image	0.12	4.31	0.12	4.29	0.31	4.72
Grid sampling ($grid\_size = 0.03$)	Spin Image	0.12	31.41	0.13	32.70	0.51	34.12
Our keypoint extraction	TriSI	0.12	185.44	0.18	187.88	0.56	210.68
Grid sampling ($grid\_size = 0.05$)	TriSI	0.12	191.41	0.18	193.19	0.56	211.35

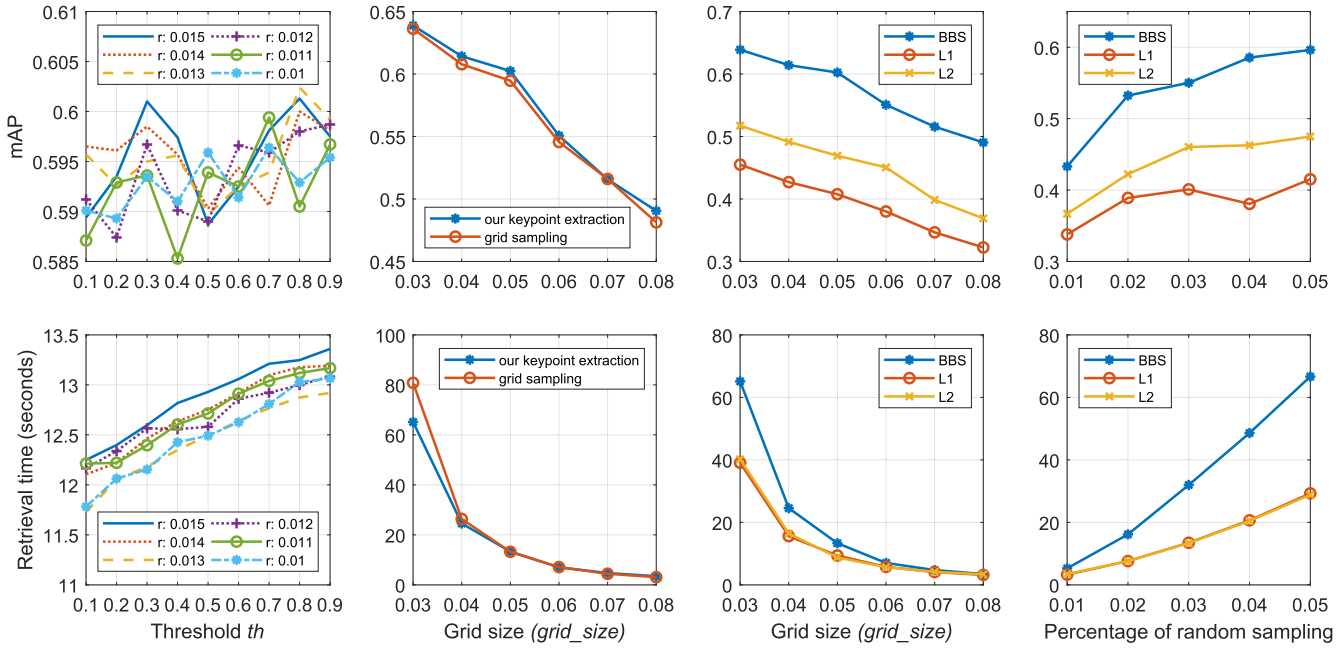


FIGURE 8. Effects of the parameters of keypoint extraction on mAP (top row) and mean retrieval time (bottom row). In the first column, *r* varies from 0.01 to 0.015 and *th* varies from 0.1 to 0.9. In the second column, the proposed keypoint extraction method is compared to grid sampling, with *grid_size* varies from 0.3 to 0.8. In the third and the fourth columns, similarity measures are compared by combing with different keypoint sampling methods. The percentage of random sampling method varies from 0.01 to 0.05.

with different settings in terms of mAP and mean retrieval time. First, we study the effect of support radius (r_{rops}), which determines the amount of local space that is encoded by the RoPS descriptor [18]. More specifically, we test the performance by varying support radius (from 5 to 30 point cloud resolution) while the other parameters remain fixed: $d_{rops} = 3$ and $bin_{rops} = 3$ for the RoPS descriptor. The result in Fig. 9(a) shows that mAP of the proposed framework improved significantly when the support radius of RoPS is increased from 5 to 15, and the performance degraded when the support radius of RoPS is further increased from 15 to 30. Moreover, a large support radius enables the RoPS descriptor to provide more information (descriptiveness) of the object, but suffer from high computational cost. In Fig. 9(a), it can be seen that the mean retrieval time of the proposed framework with large RoPS support radius ($r_{rops} = 30$) is 2.5 times greater than the one with small support radius ($r_{rops} = 5$). Second, we test the performance of the proposed framework by verifying the number of rotations (d_{rops}), while keeping the other parameters fixed ($r_{rops} = 15$, $bin_{rops} = 3$) for RoPS extraction. The results are shown in Fig. 9(b), where the mAP of the proposed framework increases when the d_{rops} is increased from 1 to 3 and decreases when $d_{rops} = 4$, and increased again from 5 to 6. Same as the support radius, increasing the number of rotations encodes more information of local space into the RoPS descriptor, which will cost more computational resources. Lastly, we tune bin_{rops} which is another important parameter of RoPS descriptor while the others are set to $r_{rops} = 15$ and $d_{rops} = 3$. Contrast to r_{rops} and the d_{rops}, bin_{rops} does not affect the cost of the proposed framework as shown in Fig. 9(c). As shown in Fig. 9 (c), the performance of the proposed framework degrade steadily when bin_{rops} increases from 3 to 13. We therefore choose the support radius $r_{rops} = 15$, the number of rotations $d_{rops} = 3$,

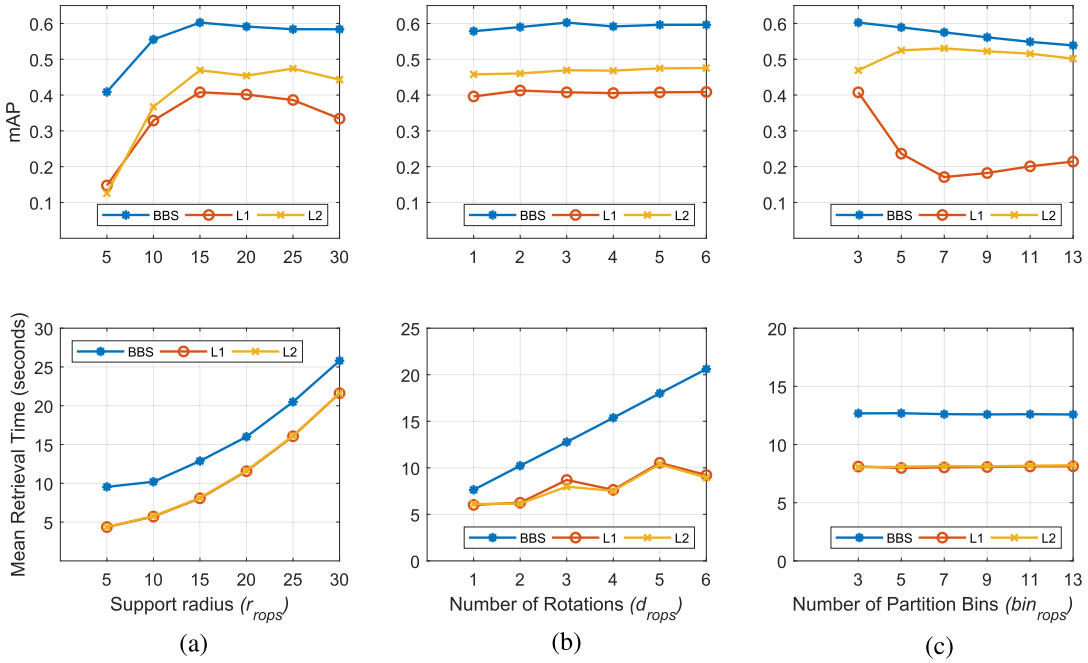


FIGURE 9. Effects of the parameters of RoPS descriptor on mAP (top row) and mean retrieval time (below row). The support radius r_{rops} varies from 5 to 30, the number of rotations d_{rops} varies from 1 to 6, and the number of partition bins bin_{rops} varies from 3 to 13.

and the number of partition bins $bin_{rops} = 3$ to balance the trade-off between the effectiveness and efficiency.

V. CONCLUSION

We presented a meshing-free 3D point cloud retrieval framework based on the bidirectional feature match, which is to extend the best buddies similarity measurement to the feature domain of 3D point clouds. Moreover, we introduced effective keypoint extraction by using PCA-based pose normalization and thresholding local-normal-similarity. Besides, we exploited point cloud based RoPS feature descriptor to encode 3D point clouds. The entire pipeline, including keypoint selection, feature computation, and similarity measurement, is highly effective and makes our framework practical for robust 3D point cloud retrieval tasks. The experimental results demonstrate the effectiveness and validate that the proposed framework can outperform other alternatives involving state-of-the-art techniques. As the computational cost of our proposed framework mainly lies in the feature descriptor extraction, it is important to develop more efficient yet effective 3D feature descriptor in the future.

ACKNOWLEDGMENT

(Naranchimeg Bold and Chao Zhang contributed equally to this work.)

REFERENCES

[1] Y. Guo, F. Wang, and J. Xin, "Point-wise saliency detection on 3D point clouds via covariance descriptors," *Vis. Comput.*, vol. 34, no. 10, pp. 1325–1338, 2017.

[2] A. Haleem and M. Javaid, "3D scanning applications in medical field: A literature-based review," *Clin. Epidemiol. Global Health*, vol. 7, no. 2, pp. 199–210, 2018.

[3] J. W. H. Tangelder and R. Veltkamp, "A survey of content based 3D shape retrieval methods," in *Proc. Shape Modeling Appl.*, 2004, pp. 145–156.

[4] A. Del Bimbo and P. Pala, "Content-based retrieval of 3D models," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 2, no. 1, pp. 20–43, 2006.

[5] F. Chen, B. Li, and L. Li, "3D object retrieval with graph-based collaborative feature learning," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 261–268, Jan. 2019.

[6] Y. Liu, X.-L. Wang, H.-Y. Wang, H. Zha, and H. Qin, "Learning robust similarity measures for 3D partial shape retrieval," *Int. J. Comput. Vis.*, vol. 89, nos. 2–3, pp. 408–431, 2010.

[7] F. Limberger, R. Wilson, M. Aono, N. Audebert, A. Boulch, B. Bustos, A. Giachetti, A. Godil, B. Le Saux, B. Li, Y. Lu, H.-D. Nguyen, V.-T. Nguyen, V.-K. Pham, I. Sipiran, A. Tatsuma, M.-T. Tran, and S. Velasco-Forero, "Point-cloud shape retrieval of non-rigid toys," in *Proc. 3DOR*, 2017, pp. 75–84.

[8] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. Van Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. Porikli, M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen, "A comparison of methods for non-rigid 3D shape retrieval," *Pattern Recognit.*, vol. 46, no. 1, pp. 449–461, 2013.

[9] Y. Gao, Q. Dai, M. Wang, and N. Zhang, "3D model retrieval using weighted bipartite graph matching," *Signal Process., Image Commun.*, vol. 26, no. 1, pp. 39–47, 2011.

[10] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," *Comput. Graph. Forum*, vol. 36, no. 1, pp. 301–329, 2017.

[11] Q. Wang and M.-K. Kim, "Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018," *Adv. Eng. Inform.*, vol. 39, pp. 306–319, Jan. 2019.

[12] C. Mura, G. Wyss, and R. Pajarola, "Robust normal estimation in unstructured 3D point clouds by selective normal space exploration," *The Vis. Comput.*, vol. 34, nos. 6–8, pp. 961–971, 2018.

[13] R. C. Veltkamp, "Shape matching: Similarity measures and algorithms," in *Proc. IEEE Int. Conf. Shape Modeling Appl. (SMI)*, May 2001, pp. 188–197.

[14] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, p. 1, 2007.

[15] R. Gregor, A. Lamprecht, I. Sipiran, T. Schreck, and B. Bustos, "Empirical evaluation of dissimilarity measures for 3D object retrieval with application to multi-feature retrieval," in *Proc. 13th Int. Workshop Content-Based Multimedia Indexing (CBMI)*, Jun. 2015, pp. 1–6.
[16] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman, "Best-buddies similarity for robust template matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2021–2029.
[17] I. Jolliffe, *Principal Component Analysis*. Berlin, Germany: Springer, 2011.
[18] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3D local surface description and object recognition," *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, 2013.
[19] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, 2016.
[20] F. Tombari, S. Salti, and L. Di Stefano, "Performance evaluation of 3D keypoint detectors," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 198–220, 2013.
[21] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "NARF: 3D range image features for object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) Workshop Defining Solving Realistic Perception Problems Personal Robot.*, vol. 44, Oct. 2010, pp. 1–3.
[22] 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, May 1999.
[23] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2009, pp. 3212–3217.
[24] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 356–369.
[25] Y. Guo, F. Sohel, M. Bennamoun, J. Wan, and M. Lu, "A novel local surface feature for 3D object recognition under clutter and occlusion," *Inf. Sci.*, vol. 293, pp. 196–213, Feb. 2015.
[26] X.-F. Hana, J. S. Jin, J. Xie, M.-J. Wang, and W. Jiang, "A comprehensive review of 3D point cloud descriptors," 2018, *arXiv:1802.02297*. [Online]. Available: https://arxiv.org/abs/1802.02297
[27] A. M. Bronstein, M. M. Bronstein, L. J. Guibas, and M. Ovsjanikov, "Shape Google: Geometric words and expressions for invariant shape retrieval," *ACM Trans. Graph.*, vol. 30, no. 1, p. 1, Jan. 2011.
[28] Z. Lian, A. Godil, and X. Sun, "Visual similarity based 3D shape retrieval using bag-of-features," in *Proc. Shape Modeling Int. Conf.*, Jun. 2010, pp. 25–36.
[29] G. L. López, A. P. P. Negrón, A. D. A. Jiménez, J. R. Rodríguez, and R. I. Paredes, "Comparative analysis of shape descriptors for 3D objects," *Multimedia Tools Appl.*, vol. 76, no. 5, pp. 6993–7040, 2017.
[30] Y. Gao, M. Wang, R. Ji, X. Wu, and Q. Dai, "3-D object retrieval with Hausdorff distance learning," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2088–2098, Apr. 2014.
[31] O. A. Zemzami, H. Aksasse, M. Ouanan, and B. Aksasse, "3D models retrieval using earth mover's distance," *Int. J. Open Problems Comput. Sci. Math.*, vol. 238, no. 1395, pp. 1–10, 2013.
[32] F. Mémoli and G. Sapiro, "A theoretical and computational framework for isometry invariant recognition of point cloud data," *Found. Comput. Math.*, vol. 5, no. 3, pp. 313–347, 2005.
[33] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D object recognition from range images using local feature histograms," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Dec. 2001, pp. 1–2.
[34] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," *Mach. Vis. Appl.*, vol. 19, no. 4, pp. 261–275, 2008.

NARANCHIMEG BOLD received the B.E. degree from the National University of Mongolia, Mongolia, in 2007, and the M.E. degree from Chonbuk National University, South Korea, in 2010. She is currently pursuing the Ph.D. degree with in design and media technology with the Graduate School of Engineering, Iwate University, Japan. Her research interests include feature matching and fusion, machine learning, and multimodal learning.

Chao Zhang received the Ph.D. degree from Iwate University, Japan, in March 2017. He is currently a full-time Assistant Professor with the Faculty of Engineering, University of Fukui, Japan. His research interests include computer vision and graphics, mainly focused on feature matching and vision-based optimization problems. He is a member of the IEEE Computer Society, the IEEE Signal Processing Society, the ACM, and the IEICE.

TAKUYA AKASHI received the Ph.D. degree in system design engineering from the University of Tokushima, in 2006. In 2015, he was a Visiting Associate with the California Institute of Technology. He is currently an Associate Professor with Iwate University, where he has been with the Department of Electrical Engineering, Electronics and Computer Science, since April 2009. His research interests include evolutionary algorithms, image processing, and human sensing. He is a member of the RISP, IEICE, and IEEJ.