PART-IN-WHOLE TYPE 3D PARTIAL SHAPE RETREIVAL BASED ON CONNECTED FACES WITH POINTNET FEATURES

Masaki Aono* and Wataru Iwabuchi†
Computer Science and Engineering, Toyohashi University of Technology, Aichi, Japan
E-mail: *aono@tut.jp, †iwabuchi@kde.cs.tut.ac.jp

***Abstract*—In this paper, we focus on "part-in-whole" type 3D partial shape retrieval, where we propose a new partial shape representation based on the idea of *connected Faces* accompanied with the features extracted from PointNet, given 3D shapes data as a solid model with Boundary Representation (BREP). The idea of *connected Faces* is inspired by the analogy with N-gram in Natural Language Processing.**

We conducted experiments with an ABC 3D dataset to compare several candidates of partial shape features, and to confirm that *connected Faces* approach has an advantage over traditional partial shape representations. Specifically, we compared the *connected Faces* approach of PointNet features with SNH (Surface Normal Histogram) as well as PFH (Point Feature Histogram), and found that *connected Faces* with PointNet outperformed other approaches in terms of several evaluation measures.

I. Introduction

In recent years, 3D models have been used in various fields such as manufacturing, medical care, architecture, education, and entertainment. Accordingly, the number of 3D models is increasing, and a method to search with a high accuracy is required. In the manufacturing industry represented by automotive parts manufacturers, it is expected to shorten the manufacturing process by searching the past processing results from the database. In typical 3D CAD systems, 3D data have been usually created and saved by "solid models" including Constructive Solid Geometry (CSG) [3] and Boundary Representation (BREP) [11]. They are then tessellated into a collection of triangular (or polygonal) meshes for a variety of processes such as Finite Element Method in CAE (Computer Aided Engineering) [25] and real-time rendering in MAR (Mixed and Augmented Reality) [12].

In this paper, we assume that 3D shapes are represented by "Boundary Representation (BREP)", keeping the topological structures, without reducing them to a collection of triangular meshes.

Our proposed method extracts a partial shape model from the 3D model target database prior to the search request. With respect to the query for the search request, we also create a collection of partial shape models for the query at the time of the search, and perform the search as a shape matching problem of the partial shapes between the query and the target.

It should be noted that there are two types of 3D partial shape retrieval: "whole-to-whole" and "part-in-whole" types. The detail on these two types will be elaborated on and surveyed in the next section. Here, we emphasize that our research is classified as the "part-in-whole" type.

In the next section, we elaborate on two types of 3D partial shape retrieval and survey the previous work. In Section III, we describe the overall flow of our proposed system, and then we list some candidates for partial shape representations together with our proposed method based on *connected Faces* in analogy with N-gram in Natural Language Processing. In Section IV, we describe our experiments focusing on comparisons of several aspects including normalization and partial shape representations with or without *connected Faces*.

II. Related Work

As we introduced in the previous section, 3D partial shape retrieval can be roughly classified into two types: "part-in-whole" and "whole-to-whole" retrieval types. Since our proposed method is a "part-in-whole" methods, here we summarize related work on 3D partial shape retrieval that falls into "part-in-whole" type.

Liu et al [18] surveyed research on 3D partial shape retrieval and classified the research into two types as shown in Fig. 2.

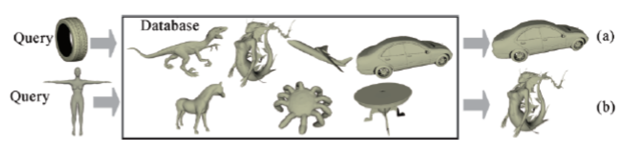

Fig. 2. Illustration of two types of 3D partial shape retrieval. (a) part-in-whole retrieval(b) whole-to-whole retrieval by partial similarity. The task of (a) is to match and search a tire to a car in a database, while the task of (b) is to match and search a woman-like shape to a mermaid in the database. This figure is cited from [18]

"Part-in-whole" type 3D partial shape retrieval has been studied by many researchers including Furuya et al. who proposed Randomized Sub-Volumes Partitioning (RSVP) [7], and Tran et al. who employed a composite approach with local and global features to partial 3D shape retrieval [22]. These approaches are examples that assume a partial shape as input is explicitly given by users.

On the other hand, there are approaches that take advantage of the information of 3D shape hierarchy and/or 3D segmented data. Bai et al [2] falls into the 3D shape hierarchy approach to partial shape retrieval, assuming that 3D shapes are given

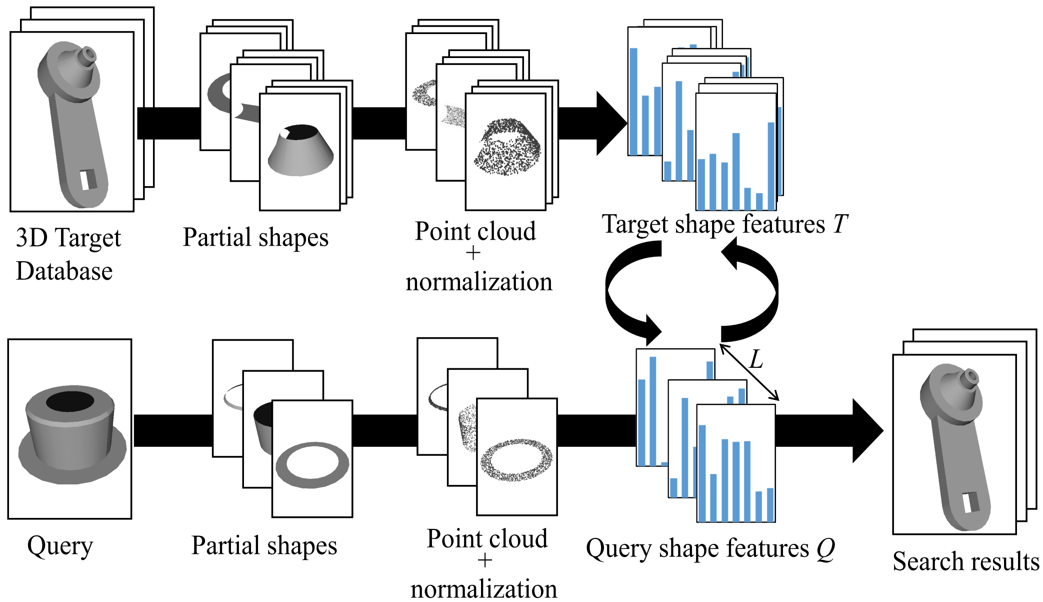

Fig. 1. Flow of Partial Shape Retrieval in Our System

by BREP. They prepare a "library" to register partial shape for design reuse. Given query and the target database, they construct reusable sub-part hierarchy so that the search problem is reduced to a shape matching problem between then query and sub-part. Theologou et al's approach falls into segment-based approach to partial shape retrieval [28], where each segment needs to have a "label". Lupinetti et al [19] surveyed CAD assembly model retrieval from both global similarity and partial similarity perspectives.

Onodera et al [5] proposed a graph-based approach to 3D partial shape retrieval, assuming that the data is represented by BREP. The node in their graph corresponds to *Face*, and the arc corresponds to *Edge* in BREP. Their method first computes all the geometric parameters such as surface areas and perimeters for the query and the target data. Subsequently, for all combinations of query faces and the target data faces, they compute the similarities. Depending on the similarities, they remove *Faces* from the graph. From this process, their method decomposes the original graph into a collection of separate connected graphs. Then, they compute topological similarities between separate connected graphs. Unfortunately, as the 3D shape models become complex, the number of connected graphs grow, so that even with an initially moderate number of 3D shape models, the combination of topological matching becomes huge, Thus, the computational time of their partial shape retrieval becomes intractable.

Kobayashi et al [16] used SHREC 2015 Range Scans [9] and SHREC 2016 Partial [22] datasets, and conducted evaluation experiments of partial shape retrieval. Since we employ their feature as one of the candidates for partial shape definition, the detail of their methods will be described in the next Section. Here we mention that Koyatashi et al's method is based on the observation that output of 3D scanners such as KINECT [1] only covers the visible part of the 3D shape object from the scanner. They define a partial 3D shape as the collection of visible surface meshes.

Recently, deep neural network approach to 3D partial shape retrieval gains attention. In our previous approach using 3D CNN to partial shape retrieval [13], we trained 3D CNN with local features defined per voxel, and applied our method to the SHREC16 Partial dataset [22]. However, the input to the partial shape retrieval in our previous approach has only had limited success for 3D objects that have been scanned like antique earthware where no *connected Face* information was available. Furuya et al [8] introduced "Part-Whole Relation Embedding network" (PWRE-net). Their method is attractive in the sense that they achieved high search performance with general 3D model datasets such as P-ModelNet, P-SH11NR, ShapeNet Core55, and ObjectScans. Unfortunately, no 3D CAD dataset was tested with their method. Moreover, there is no explicit decomposition of a query at run time.

III. Proposed Method

Figure 1 illustrates the overall flow of our 3D partial shape retrieval system. In this section, we will describe each step in the flow diagram. Section III-A describes the extraction of partial shapes, Section III-B focuses on 3D point cloud generation, Section III-C describes normalization, Section III-D focuses on our features for partial shapes. and Section III-E discusses feature matching.

A. Partial Shape Extraction

Since we assume that 3D shapes are represented by Boundary Representation (BREP), it is natural to define partial shapes by *Faces* in BREP of a 3D shape. Figure 3 demonstrates an example of *Faces* in real 3D shape data. In addition to the extraction of an individual *Face*, we define *connected Faces*, in analogy with *N*-gram in Natural Language Processing, based on the adjacency relationship of *Faces* in BREP. For example, we can define *two connected Faces* by extracting a group of *Faces* connected to each other by their shared *Edges*. In general, we can define *N connected Faces* by extracting a group of *Faces* that can be traversed from a given *Face* to adjacent *Faces N* times with shared *Edges*. The advantage of this approach to partial shape definition includes robustness against geometric information such as surface orientation, spatial position, and size.

In our approach, we assume that if two *Faces* have a shared *Edge*, they are supposed to be adjacent to each other. We then define the *connected number* as the number of contiguous *Faces*, here denoted by Cn, a parameter to a partial shape to be extracted from a given 3D shape model. Examples of $Cn = 2$ and $Cn = 3$ are shown in Figure 4. It should be noted that the partial shape extraction is applied to both the target models in the database and the query model. Sample extracted partial shapes are depicted in Figure 5. After partial shapes are extracted, we apply tessellation to the partial shapes, converting them into 3D meshes.

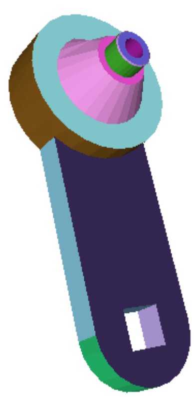

Fig. 3. An example of *Faces* in BREP of a 3D shape object, where a different color corresponds to a different *Face*.

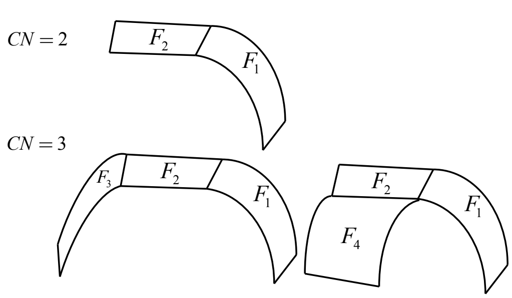

Fig. 4. Examples of $Cn = 2$ and $Cn = 3$. *Faces* can be flat or curved. Adjacent *Faces* have a shared *Edge*.

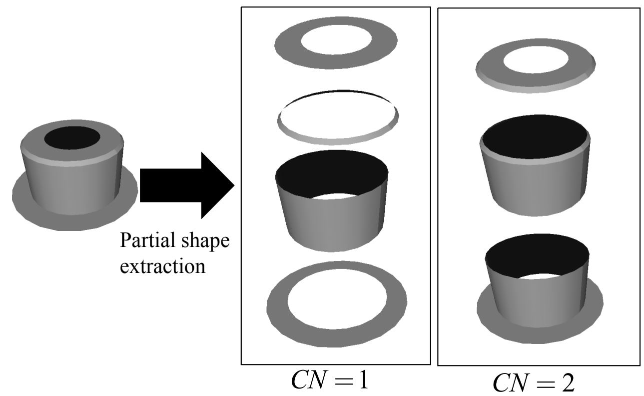

Fig. 5. $Cn = 1$ and $Cn = 2$ examples of partially extracted shapes from a 3D shape model on the left.

B. 3D Point Cloud Generation

In the last stage of 3D partial shape extraction, we have 3D (triangular) meshes converted from *connected Faces*. Then, we perform point cloud generation based on Osada et al's method [21]. Assume that the number of points is represented by H_{Osada}. Given three vertices of an arbitrary triangle as $\mathbf{a}$, $\mathbf{b}$and $\mathbf{c}$. Then, we generate a random point $\mathbf{p}$ on the triangle as below:

$$\mathbf{p} = (1 - \sqrt{r_1})\mathbf{a} + \sqrt{r_1}(1 - r_2)\mathbf{b} + \sqrt{r_1}r_2\mathbf{c},$$

where r_1 and r_2 represent random variables ranging from 0.0 to 1.0. For each point $\mathbf{p}$, we associate the normal vector defined as the normalized perpendicular vector to the 3D triangle. The random points generated with ordinary pseudo random number generator have an inherent bias. There are several options to remedy the bias. For example, we could use quasi-random number generators such as Sobol sequences [15] in high dimensional space. Here we employ Farthest Point Sampling (FPS) [4]. In FPS, a point cloud made of H_{Osada} points is gone through sampling to H_{FPS} points. Figure 6 demonstrates the application of FPS.

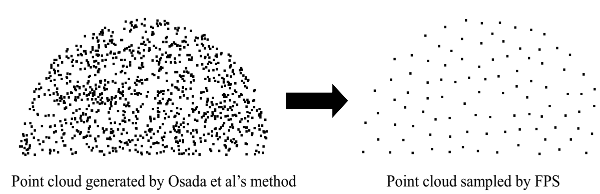

Fig. 6. FPS application example for a point cloud generated by Osada et al's method

C. Normalization

In "part-in-whole" type 3D partial shape retrieval, it is essential to find 3D shape objects having locally high similarity within the target 3D database, regardless of the position, orientation, and size of a given query. Since our proposed features elaborated on in the next section include properties influenced by the position, orientation, and the size, we apply normalization before feature matching. For this purpose, we first translate the centroid $\mathbf{g}$ to the origin.

$$\mathbf{g} = \frac{1}{H_{\mathrm{FPS}}} \sum_{i=1}^{H_{\mathrm{FPS}}} \mathbf{p}^{(i)},$$

where $\mathbf{p}^{(i)}$ is the coordinate of i-th point. Then, we apply PointSVD and NormalSVD for pose normalization [27]. Under the current circumstances, the point cloud position matrix P is represented by the following formula:

$$P = \begin{pmatrix} p_x^{(1)} - g_x & p_x^{(2)} - g_x & \dots & p_x^{(H_{\mathrm{FPS}})} - g_x \\ p_y^{(1)} - g_y & p_y^{(2)} - g_y & \dots & p_y^{(H_{\mathrm{FPS}})} - g_y \\ p_z^{(1)} - g_z & p_z^{(2)} - g_z & \dots & p_z^{(H_{\mathrm{FPS}})} - g_z \end{pmatrix}$$

On the other hand, the point cloud surface normal matrix N is represented by the following formula:

$$N = \begin{pmatrix} n_x^{(1)} & n_x^{(2)} & \dots & n_x^{(H_{\mathrm{FPS}})} \\ n_y^{(1)} & n_y^{(2)} & \dots & n_y^{(H_{\mathrm{FPS}})} \\ n_z^{(1)} & n_z^{(2)} & \dots & n_z^{(H_{\mathrm{FPS}})} \end{pmatrix}$$

In Point SVD, point cloud position matrix P is decomposed with singular value decomposition as below:

$$P = U \Sigma W^{\mathrm{T}},$$

where U is a 3×3 orthonormal matrix, Σ is a diagonal 3×3 matrix having singular values, and W^{T} is a $3 \times H_{\mathrm{FPS}}$ orthonormal matrix. To obtain rotation matrix R, we normalized each singular vector of U, which we denote $\hat{U}$, then take a transpose of the normalized left singular matrix as follows:

$$R = \hat{U}^{\mathrm{T}}$$

In a nutshell, the normalized point cloud matrix and surface normal matrix are represented by the following equations:

$$P' = \hat{U}^{\mathrm{T}} P = RP$$
$$N' = \hat{U}^{\mathrm{T}} N = RN$$

Finally, we divide every transformed point by the largest distance from the new origin for scale normalization.

D. Extracted Features

In this section, we focus on the features extracted from 3D partial shapes. We define three different features; (1) Surface Normal Histograms (SNH)(one of our earlier proposed features of partial 3D shapes suitable for 3D digital scanners [16]), (2) Point Feature Histograms (PFH) [26], and (3) PointNet Features extracted from intermediate layer of PointNet [24].

1) Surface Normal Histogram (SNH): In the first feature SNH which is our previous approach, we first compute the surface normal at each point. Subsequently, we set up a "grid" covering each partial 3D shape as illustrated in Figure 7.

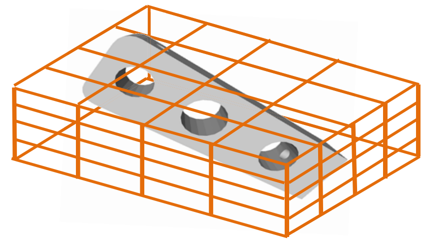

Fig. 7. A "grid" covering a 3D shape in SNH feature extraction

For each grid, we project surface normal of each point in the grid on three planes as illustrated in Figure 8. Then, for the collection of all the surface normals, we compute the distribution of xy, yz, and zx projected planes, thereby obtaining each histogram, and finally concatenate three histograms to generate SNH as depicted in Figure 9.

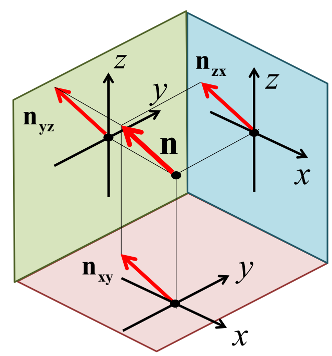

Fig. 8. Surface normal of each point in the grid is projected onto three perpendicular planes

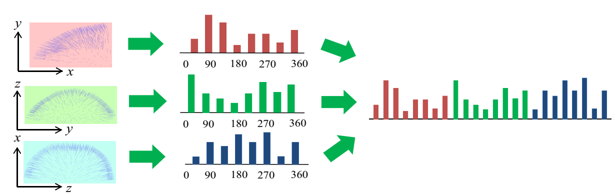

Fig. 9. Surface Normal Histogram (SNH) generation from three projected distributions of surface normals in a grid

Kobayashi et al's SNH features are targeted for 3D shape objects for general categories, and are suitable for the objects with rounded surfaces varying surface normals. SNH features are susceptible to the "grid" definition. For instance, if x, y, and z axis definitions are slightly altered, the resulting SNH may differ significantly. This is especially a problem if the 3D shape objects have numerousf flat surfaces similar to typical 3D mechanical CAD models, where some of them are slanted with respect to the three projected faces (i.e. XY-, YZ-, ZX-planes). In order to alleviate the above problem, we have added 45 degree rotations around x and y axes after pose normalization.

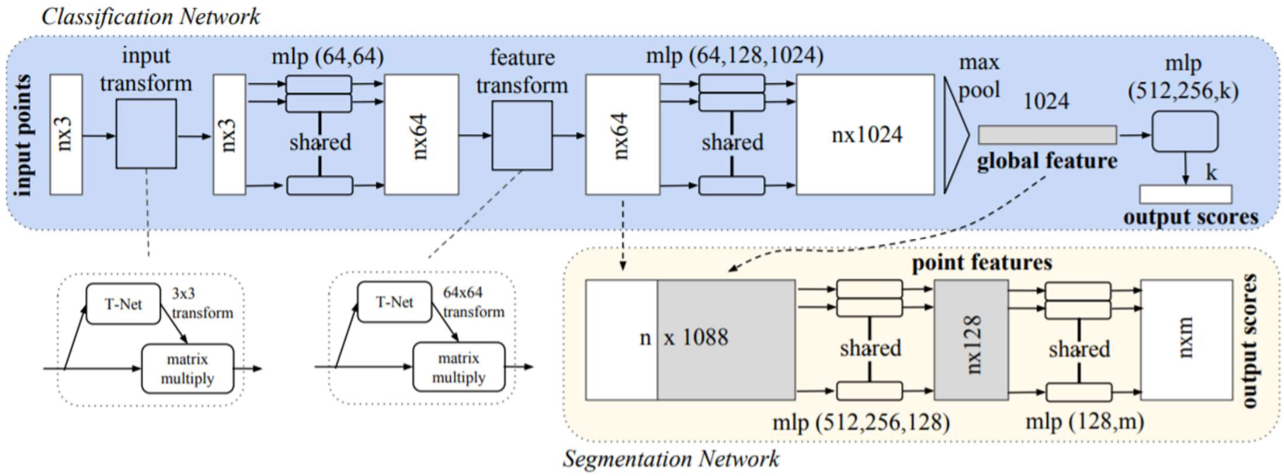

Fig. 10. PointNet architecture [24]

2) Point Feature Histogram (PFH): The second feature we adopt here as one of the partial shape features is PFH (Point Feature Histogram) [26]. The PFH is extracted from the angular relationship between the point position vector and the surface normal vector. First, we extract a pair of two points $\mathbf{p}_s$ and $\mathbf{p}_t$. These points have the associated surface normal vectors $\mathbf{n}_s$ and $\mathbf{n}_t$. We define the following uvw coordinate system as:

$$\begin{aligned}\mathbf{u} &= \mathbf{n}_s \\ \mathbf{v} &= \mathbf{u} \times \frac{\mathbf{p}_t - \mathbf{p}_s}{\|\mathbf{p}_t - \mathbf{p}_s\|_2} \\ \mathbf{w} &= \mathbf{u} \times \mathbf{v}\end{aligned}$$

By taking advantage of the uvw coordinate system, we compute the following three angular features based on normal vectors $\mathbf{n}_s$ and $\mathbf{n}_t$:

$$\begin{aligned}\alpha &= \mathbf{v} \cdot \mathbf{n}_t \\ \phi &= \mathbf{u} \cdot \frac{\mathbf{p}_t - \mathbf{p}_s}{d} \\ \theta &= \arctan(\mathbf{w} \cdot \mathbf{n}_{\mathrm{t}}, \mathbf{u} \cdot \mathbf{n}_{\mathrm{t}})\end{aligned}$$

Here, $d = \|\mathbf{p}_t - \mathbf{p}_s\|_2$. The above angular features are computed for each pair of points and their normals. By repeating the computation of the angular features for all the pairs, we generate a histogram.

3) PointNet Feature: The third features are extracted from the intermediate layer of PointNet. PointNet is a deep learning architecture having point cloud as its input [24]. The architecture of PointNet is shown in Figure 10. As a 3D data representation, point cloud based methods need to produce the output invariant under the sequence of generated points. PointNet has achieved this invariance by its max pooling before obtaining "global feature" as shown in Figure 10. In this paper, we extract the features from the intermediate layer, by employing pre-trained PointNet with ModelNet40 [29]. The intermediate layer where we extract features is one layer before the last (output) layer.

E. Feature Matching

Let a set of features extracted from a given query be denoted by $Q = \{\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_L\}$, and a set of features extracted from a target database be denoted by $T = \{\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_M\}$, where L is the number of partial shapes extracted from a query, and M is the number of partial shapes extracted from the target database. Then, we define the similarity $s(Q, T)$ between Q and T as follows:

$$s(Q,T) = \frac{1}{L} \sum_{\mathbf{q} \in Q} \max_{\mathbf{t} \in T} \mathbf{q} \cdot \mathbf{t}$$

We compute all the combinations of partial shapes between the query and the target database, and sort in descending order to generate ranked search results. In the above equation, we need to search the largest cosine similarity $\mathbf{t} \in T$ for $\mathbf{q} \in Q$. Usually, this process is computationally expensive. We alleviate this expensive computation by Fast Library for Approximate Nearest Neighbors (FLANN) [20], as one of approximate nearest neighbor search methods. It is noted that FLANN is known to automatically select optimized parameters using either Randomized kd-tree or Hierarchical k-means tree.

IV. Experiments

In this section we describe experimental results using features described in the previous section. Section IV-A describes the dataset, Section IV-B refers to methods for comparison, Section IV-C describes the evaluation measures, and Section IV-D describes experimental results.

A. Dataset

The dataset we have used in 3D partial shape dataset is a subset of ABC dataset [17]. ABC dataset provides a variety of 3D data representation including Parasolid [6] and STEP [23], which allow BREP (or Boundary Representation).

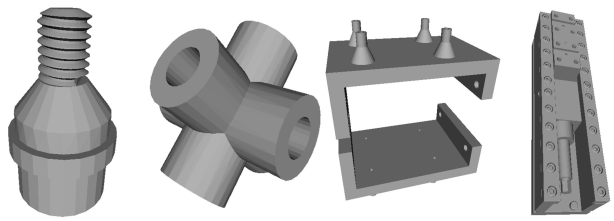

Fig. 11. Sample 3D shapes in our target 3D database from ABC dataset

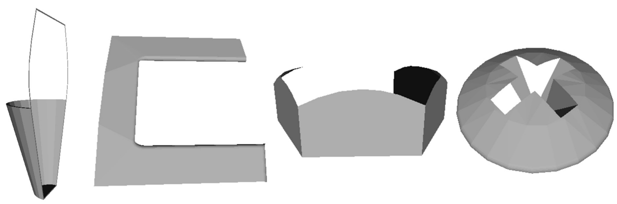

Fig. 12. Sample query objects

In our experiments, we have adopted a STEP format of ABC dataset, where we have extracted 9,960 3D shape models for our target database as exemplified in Figure 11. For the query dataset, we have extracted 4 to 10 *connected Faces*, i. e., $Cn = 4$ to $Cn = 10$ from BREP data (or STEP) of the subset. As a result, the number of 3D partial shapes of the query turns out to be 44,278 (as illustrated in Figure 12). Since our partial shape retrieval is a "part-in-whole" type, for each *connected Face*, we need to have "part" as a query and "whole" as the searched objects. Preparing the *relevant* dataset for each query is highly difficult. In our experiments, we set up only the "part" as *relevant* if it is extracted from the original "whole" objects for automatic evaluation. Since the above setting allows only one answer to each query, the search performance is proportional to how early we can find the "whole" object in the search ranking. Of course the search performance depends on what features we use and what kind of *connected Faces* we adopt, which are revealed in this section.

B. Methods for Comparison

As methods for comparison, we have selected Kobayashi et al's method (i.e. a method with SNH features) for partial shape retrieval [16]. For Kobayashi et al's method, we set the number of partial shape models extracted from target database to 18, considering computational time. The parameters for SNH includes the number of grids ($4 \times 4 \times 4$) and the number of bins (for each projected plane) to be 8. Thus, the feature dimension of SNH is 1,536 (= $4 \times 4 \times 4 \times 8 \times 3$).

C. Evaluation Measure

For evaluation measure, we have adopted Nearest Neighbor (NN) (or P@1), Recall, Normalized Discounted Cumulative Gain (NDCG).NN is

$$\text{Nearest Neighbor(NN)} \quad = \quad rel(1),$$

where $rel(x)$ denotes the number of *relevant* data in the top x ranked results for a query. Recall is expressed by the following:

$$\text{Recall} \quad = \frac{rel(K_{\text{Recall}})}{C},$$

where C denotes the number of *relevant* data in the target database, and $K_{\text{Recall}} = 10$. In our experiments, $C = 1$. NDCG is a measure of ranking quality as to how early the *relevant* data is found. The earlier the *relevant* data is searched, the better. It is expressed by the following equation:

$$\text{DCG(i)} \quad = \quad \begin{cases} G(1) & (i = 1) \\ \text{DCG(i} - 1) + \frac{\text{G(i)}}{\log_2(\text{i})} & \text{(otherwise)} \end{cases}$$

$$\text{NDCG@K}_{\text{NDCG}} \quad = \quad \frac{\text{DCG(K}_{\text{NDCG}})}{1 + \sum_{j=2}^{K_{\text{NDCG}}} \frac{1}{\log_2(j)}},$$

where i denotes the ranking in the search result, $G(i)$ is a gain returning 1 when the i-th search result is *relevant*; otherwise returning 0, and K_{NDCG} represents the number of target 3D models. In our experiments, $K_{\text{NDCG}} = 9,960$.

D. Experimental Results

Table I demonstrates the results of the significance of pose normalization. As the table shows, NormalSVD turns out to be better in search result. We conjecture that the ABC dataset we used has 3D shapes with flatter planar surfaces, whereas when applying PointSVD to planes, there is a tendency that face normals randomly flip.

TABLE I
COMPARISON OF POSE NORMALIZATION

	NN	Recall@10	NDCG@9960
SNH + PointSVD ($Cn = 2$)	0.55	0.76	0.69
SNH + NormalSVD ($Cn = 2$)	**0.70**	**0.92**	**0.83**

In the next experiment, we focus on the number of *connected Faces*. Due to the computational time, we only conducted experiments with $Cn = 1$ and $Cn = 2$, keeping SNH as the features of partial 3D shapes, fixed with NormalSVD from the previous experiment. Table II demonstrates the result. As we anticipated, $Cn = 2$ has better search performance than $Cn = 1$. Even though experiments with $Cn = 3$ are not feasible in terms of computational time, We list the number of partial shape models with $Cn = 1$, $Cn = 2$, and $Cn = 3$ in Table III. From this table, unless we develop some innovative methods, it may be easily understood that experiments with $Cn = 3$ are not feasible under the current circumstances.

TABLE II
COMPARISON WITH RESPECT TO THE NUMBER OF *connected Faces*

	NN	Recall@10	NDCG@9960
SNH + NormalSVD ($Cn = 1$)	0.50	0.72	0.66
SNH + NormalSVD ($Cn = 2$)	**0.70**	**0.92**	**0.83**

We then conducted experiments in terms of features described in Section III-D. The result is shown in Table IV. Among three different features, PointNet features resulted in

TABLE III
THE NUMBER OF PARTIAL SHAPE MODELS EXTRACTED

	the number of query partial shape models L				the number of target partial shape models M			
	Maximum	Minimum	Average	Median	Maximum	Minimum	Average	Median
$Cn = 1$	10	4	6.82	5.00	43974	1	233.37	41.00
$Cn = 2$	36	7	16.58	9.00	143330	1	750.46	134.00
$Cn = 3$	124	9	34.72	29.00	65974770	1	37983.57	535.00

the best performance. We conjecture that PointNet is robust against the noise in pose normalization for sampled points. PointNet incorporates a construct called *T-Net* inspired by spatial transformer network [14], which is a network that applies Affine transformation to input point clouds.

TABLE IV
COMPARISON BETWEEN FEATURES

	NN	Recall@10	NDCG@9960
SNH without *connected Faces*	0.07	0.17	0.21
SNH + NormalSVD ($Cn = 2$)	0.70	0.92	0.83
PFH ($Cn = 2$)	0.66	0.92	0.81
PointNet + NormalSVD ($Cn = 2$)	**0.76**	**0.96**	**0.87**

After the above experiments, we fix PointNet as feature extraction, $Cn = 2$ and NormalSVD as pose normalization. With this in mind, we demonstrate a successful "part-in-whole" partial retrieval example, and compare the result with a simple SNH method without *connected Faces* as shown in Figure 13.

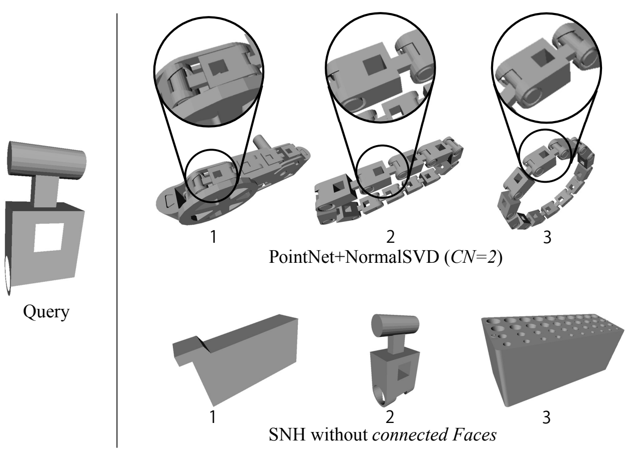

Fig. 13. Successful "part-in-whole" search example

E. Partial Feature Dimensions Adopted by Our System

In our experiments, we compare partial shape features taken by the three different features representations discussed so far. Table V summarizes our experimental settings in terms of partial feature dimension. Apparently, PFH's approach requires only 125 dimension, while PointNet requires 256 dimension, and SNH requires 1,536 dimension for representing each partial shape. Obviously when 3D partial shapes are kept in 3D database, SNH not only consumes memory, but the 3D partial shape search needs most time-consuming similarity computation, regardless of applying FLANN as approximate nearest neighbor described in *Feature Matching* section.

TABLE V
COMPARISON OF FEATURE DIMENSIONS

	SNH	PFH	PointNet
partial feature dimension	1,536	125	256

F. System for "Part-in-Whole" 3D Shape Retrieval

Based on the previous experiments, we have constructed an interactive "part-in-whole" type of 3D partial shape search system, where PointNet features are adopted, and NormalSVD is chosen as pose normalization. Sample system outlooks are shown in Figures 14 and 15. The system is programmed in Python with Flask [10] as a Web application. For the interactive selection of a group of *Faces*, the color of chosen *Faces* is turned into "Red" so that users can easily tell what parts are selected as the query. It should be noted that the query can be any number of *connected Faces*, while the target partial shapes are limited to $Cn = 2$. Partially similar shapes are displayed on the right as the top nine most similar search results. Each 3D shape model in the search result window, as well as the query window, can be independently Affine transformed (i.e. rotated, scaled, and translated) to see if there is any part in the object similar to the red portion of the query 3D object.

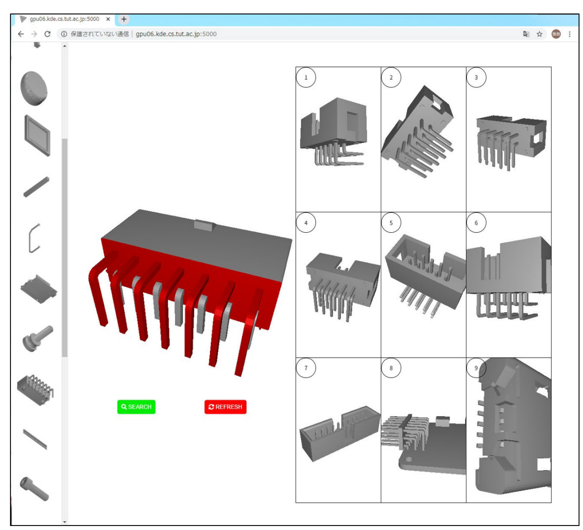

Fig. 14. An outlook of our 3D partial shape retrieval system. The red part was interactively selected by user.

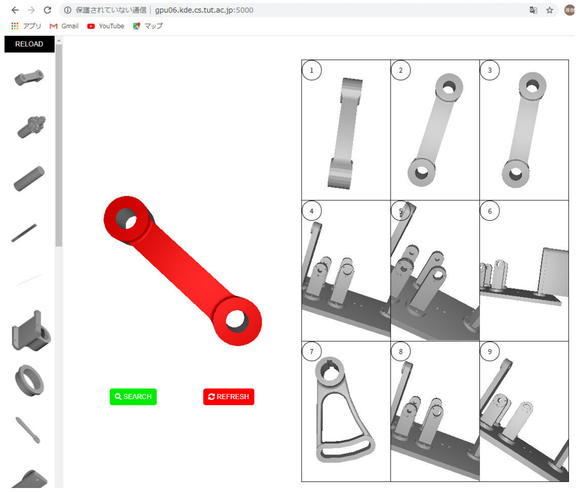

Fig. 15. Another sample scene of our 3D partial shape retrieval system.

V. Conclusion

In this paper, as one of 'the 'part-in-whole" type 3D partial shape retrieval, we propose a new partial shape representation based on the idea of *connected Faces* accompanied with the features extracted from PointNet, assuming that 3D shapes are given by BREP. The idea of *connected Faces* is inspired by the analogy with N-gram in Natural Language Processing. We compared *connected Faces* approach of PointNet features with SNH (Surface Normal Histogram) as well as PFH (Point Feature Histogram), and found that *connected Faces* with PointNet outperformed other approaches in terms of NN (Nearest Neighbor), Recall@10, and NDCG@9960, where 9,960 data points were used from the ABC dataset. We also conducted a comparison between PointSVD and NormalSVD as pose normalization, and found that NormalSVD performs better when flatter surfaces are ubiquitous as in 3D mechanical CAD data such as ABC dataset. Finally, we demonstrate our system incorporating the *connected Faces* where users can interactively select partial shapes as a query with arbitrary number of *connected Faces* and undergo the "part-in-whole" type partial shape retrieval.

For future work, we will investigate how to cope with "part-in-whole" partial shape retrieval with three or more *connected Faces* in a target 3D shape database. Furthermore, we will test our system with another dataset other than the ABC dataset.

Acknowledgment

A part of this research was carried out with the support of the Grant-in-Aid for Scientific Research (B) (issue number 17H01746).

References

[1] Kinect. https://en.wikipedia.org/wiki/Kinect.

[2] Jing Bai, Shuming Gao, Weihua Tang, Yusheng Liu, and Song Guo. Design reuse oriented partial retrieval of cad models. *Computer-Aided Design*, 42(12):1069 – 1084, 2010.

[3] Tao Du, Jeevana Priya Inala, Yewen Pu, Andrew Spielberg, Adriana Schulz, Daniela Rus, Armando Solar-Lezama, and Wojciech Matusik. Inversecsg: Automatic conversion of 3d models to csg trees. *ACM Trans. Graph.*, 37(6):213:1–213:16, December 2018.

[4] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, Sep. 1997.

[5] Makoto Onodera et al. Development of automatic reusable mesh generation using partially similar shape retrieval (in japanese). *Proceeding of Japan Society of Mechanical Engineers*, 83(853):17–00073–17–00073, 2017.

[6] Parasolid XT Format. Unigraphics solutions, inc. 2008.

[7] T. Furuya, S. Kurabe, and R. Ohbuchi. Randomized sub-volume partitioning for part-based 3d model retrieval. In *Proceedings of the 2015 Eurographics Workshop on 3D Object Retrieval*, 3DOR, pages 15–22, Aire-la-Ville, Switzerland, Switzerland, 2015. Eurographics Association.

[8] Takahiko Furuya and Ryutarou Ohbuchi. Learning part-in-whole relation of 3d shapes for part-based 3d model retrieval. *Computer Vision and Image Understanding*, 166:102 – 114, 2018.

[9] A. Godil, H. Dutagaci, B. Bustos, S. Choi, S. Dong, T. Furuya, H. Li, N. Link, A. Moriyama, R. Meruane, R. Ohbuchi, D. Paulus, T. Schreck, V. Seib, I. Sipiran, H. Yin, and C. Zhang. Range scans based 3d shape retrieval. In *Proceedings of the 2015 Eurographics Workshop on 3D Object Retrieval*, 3DOR, pages 153–160, Aire-la-Ville, Switzerland, Switzerland, 2015. Eurographics Association.

[10] M. Grinberg. *Flask Web Development, 2nd Edition*. O'Reilly Media, Incorporated, 2018.

[11] Christoph M. Hoffmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.

[12] Information technology Computer graphics, image processing and environmental data representation Mixed and augmented reality (MAR) reference model. Standard, International Organization for Standardization, Geneva, CH, March 2019.

[13] Wataru Iwabuchi and Masaki Aono. 3d cnn based partial 3d shape retrieval focusing on local features. *APSIPA*, pages 1523–1529, 2018.

[14] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015.

[15] Stephen Joe and Frances Y. Kuo. Remark on algorithm 659: Implementing sobols quasirandom sequence generator. *ACM Trans. Math. Softw.*, 29(1):4957, March 2003.

[16] Y. Kobayashi and M. Aono. Three-dimensional model retrieval method and three-dimensional model retrieval system, March 2017. Japanese Patent Application No.2017-029425, (in Japanese).

[17] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big CAD model dataset for geometric deep learning. *CoRR*, abs/1812.06216, 2018.

[18] Zhen-Bao Liu, Shu-Hui Bu, Kun Zhou, Shu-Ming Gao, Jun-Wei Han, and Jun Wu. A survey on partial retrieval of 3d shapes. *Journal of Computer Science and Technology*, 28(5):836–851, Sep 2013.

[19] Katia Lupinetti, Jean-Philippe Pernot, Marina Monti, and Franca Giannini. Content-based cad assembly model retrieval: Survey and future challenges. *Computer-Aided Design*, 113:62 – 81, 2019.

[20] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

[21] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, October 2002.

[22] I Pratikakis, MA Savelonas, Fotis Arnaoutoglou, G Ioannakis, Anestis Koutsoudis, T Theoharis, MT Tran, VT Nguyen, VK Pham, HD Nguyen, et al. Shrec16 track: Partial shape queries for 3d object retrieval. *Proc. 3DOR*, 1(8), 2016.

[23] Michael J. Pratt. Introduction to iso 10303 - the step standard for product data exchange. pp. *J. Comput. Inf. Sci. Eng.*, 1:102–103, 2001.

[24] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016.

[25] Singiresu S. Rao. *The Finite Element Method in Engineering: Fifth Edition*. Elsevier Inc., November 2010.

[26] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI - Künstliche Intelligenz*, 24(4):345–348, 2010.

[27] Atsushi Tatsuma and Masaki Aono. Multi-fourier spectra descriptor and augmentation with spectral clustering for 3d shape retrieval. *Vis. Comput.*, 25(8):785804, June 2009.

[28] Panagiotis Theologou, Ioannis Pratikakis, and Theoharis Theoharis. Part-based 3d object retrieval via multi-label optimization. *Computer Vision and Image Understanding*, 160:148 – 157, 2017.
[29] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.