Computer Vision and Image Understanding 166 (2018) 102–114

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Learning part-in-whole relation of 3D shapes for part-based 3D model retrieval

Takahiko Furuya*, Ryutarou Ohbuchi

University of Yamanashi, 4-3-11 Takeda, Kofu-shi, Yamanashi-ken 400-8511, Japan

ARTICLE INFO

Keywords:
Part-based 3D model retrieval
Part-in-whole retrieval
Deep learning

ABSTRACT

Given a query that specifies partial 3D shape, a Part-based 3D Model Retrieval (P3DMR) system finds 3D shapes whose part or parts matches the query. An approach to P3DMR is to partition or segment whole models into sub-parts and performs query-part-to-target-parts matching. Whatever the definition of part, e.g., a rectangular volume in Euclidean space or a part segmented on a mesh manifold, the computation will be very costly. The part-whole matching must account for, for each 3D whole shape in a database, varying position, scale and orientation of the segmented sub parts. Another approach, in an attempt to make part-whole matching efficient, tries to approximate part-whole inclusion test with a single comparison between a pair of features, one representing the part-based query and the other representing the whole shape. Aggregation of local geometrical features of parts into a feature per whole 3D shape, e.g., via Bag-of-Features approach, is an example. This approach so far suffered from inaccuracy as the aggregation is not optimized for part-whole inclusion test of 3D shapes. This paper proposes a novel P3DMR algorithm called *Part-Whole Relation Embedding network* (*PWRE-net*) that effectively and efficiently performs part-whole inclusion test via learned embedding into a common feature space. Using deep neural network, the PWRE-net learns, from a large number of part-whole shape pairs, a common embedding of partial shapes and their associated whole shapes. For the training, training datasets containing part-whole shape pairs are created automatically from unlabeled 3D models. Experimental evaluation shows that PWRE-net outperforms existing algorithms both in terms of retrieval accuracy and efficiency.

1. Introduction

As number of three-dimensional (3D) shape models grows at an explosive rate, technology for shape-similarity based 3D Model Retrieval (3DMR) has been gaining attention. The majority of 3DMR algorithms focus on Whole-based 3D Model Retrieval (W3DMR) in which whole 3D shape is given as a query to the retrieval system. This form of 3DMR has been studied for over a decade. For a class of application scenarios and databases, certain W3DMR algorithms achieve retrieval accuracy and efficiency for practical use.

Another class of 3DMR is called Part-based 3D Model Retrieval (P3DMR) or part-in-whole retrieval (Liu et al., 2013), which would retrieve a list of whole 3D shapes given a partial 3D shape as a query. P3DMR has many practical applications in such areas as industrial product design, archaeology, medicine, or drug screening. Unlike W3DMR, however, P3DMR has not been studied well in the past. P3DMR is technically more challenging than W3DMR. In P3DMR, we don't know which model in a database contains a shape specified by a part-based query. We also don't know, a priori, at which position, scale, and orientation the partial shape of the query is included in the 3D model. A brute force search through the configuration space and the database incurs very large computational cost. We thus think that the challenge central to P3DMR is that of computational cost. Retrieval accuracy is of course important, even under 3D geometric transformation (i.e., translation, scaling, and rotation in 3D space) and/or global deformation of the partial and whole shapes.

The existing P3DMR algorithms can be classified into one of the two approaches; Part-to-Parts Matching (PPsM) and Part-to-Whole Matching (PWM). The PPsM approach is adopted by the majority of existing P3DMR algorithms (e.g., Attene et al., 2011; Furuya et al., 2015; Kanezaki et al., 2010; Shalom et al., 2008). This approach regards a whole 3D shape as a set of (potentially overlapping) sub-parts. The whole 3D shape is segmented into multiple (e.g., tens to thousands of) sub-parts, and each sub-part is described by a shape feature. Alternative definitions of “sub-part” exist, e.g., a part defined on mesh manifold generated via mesh segmentation, or a part in 3D Euclidean space generated by a spherical- or a rectangular (Furuya et al., 2015) sub-volume. A similarity between a part-based query and a whole 3D shape

* Corresponding author.
E-mail address: takahikof@yamanashi.ac.jp (T. Furuya).

https://doi.org/10.1016/j.cviu.2017.11.007
Received 21 April 2017; Received in revised form 15 November 2017; Accepted 19 November 2017
Available online 21 November 2017
1077-3142/

is computed by comparing the feature of the part-based query with features of *ALL* the sub-parts of the whole 3D shape. The PPsM-based algorithm achieves certain level of retrieval accuracy via laborious matching between the query and sub-parts of the whole 3D shapes. However, PPsM suffers from high temporal- and spatial-cost. Temporal cost is high since a feature of the query must be compared against features of every sub-part of every 3D shape in the database. Spatial cost for storing features of all the sub-parts of all the 3D shapes is also very high.

In PWM, each of the partial shape of the query and the whole 3D shape is described by single feature. The feature of the whole 3D shape is expected to represent its constituent sub-parts so that the inclusion relationship between the part-based query and the whole 3D shape can be tested. To approximate the inclusion test, typical PWM algorithms (e.g., Liu et al., 2006; Savelonas et al., 2014) aggregate a set of sub-part features into a single feature vector per whole 3D shape, often by using Bag-of-Features (Csurka et al., 2004) or its variants (e.g., Perronnin et al., 2010; Zhou et al., 2010). Thanks to the aggregation, scores of approximated part-whole inclusion test can be computed efficiently by comparing a pair of features. However, the approximation of the inclusion test via aggregation of sub-part features is imperfect, resulting in failure of the test. Consequently, PWM-based algorithms tend to have lower accuracy than PPsM-based ones. It is also difficult for PWM to precisely localize a part in matched whole shapes.

In this paper, we propose a P3DMR algorithm called *Part-Whole Relationship Embedding network*, or *PWRE-net* (pronounced ``power net"). Our approach is close to that of PWM, as we try to efficiently compare a part-based query with a whole 3D model by a single comparison of a pair of features in a common embedding space. The algorithm learns to embed features of sub-part 3D shapes and features of whole 3D shapes into a common feature space so that part and whole can be quickly compared. The embedding is computed by using two feature transformation pipelines (Fig. 1), one for sub-part 3D shape features and the other for whole 3D shape features, realized by using *Deep Neural Network* (*DNN*). A handcrafted low-level local geometrical feature with built-in invariance to 3D rotation, translation, and scaling is used as input to the pipelines. In P-block, a low-level feature for a part is refined and then transformed to become a feature in the part-whole common feature space. In W-block, a set of low-level features describing parts of a whole 3D model is refined individually, aggregated by averaging, and then transformed to become a feature in the common feature space. An initial ranked list of retrieval results is found simply by comparing embedded features of whole 3D shapes with the embedded features of the query part. A PPsM-based post-processing is performed on the whole 3D shapes included in the initial ranked list for more precise ranking and for localization of the part in the matched whole.

We create two benchmark datasets to quantitatively evaluate PWRE-net algorithm. The benchmarks include both training data (i.e., part-whole shape pairs) and test data. As with many DNN-based algorithms, training of PWRE-net relies on quantity and quality of training dataset consisting of part-whole shape pairs. No such dataset exists, and manually creating one large enough (e.g., 1M pairs) for the DNN training is not feasible. We thus devise a simple yet effective algorithm to generate part-whole shape pairs from a set of unlabeled whole 3D models. Experimental evaluation using these datasets shows that the proposed PWRE-net produces retrieval accuracy superior to previous PPsM-based and PWM-based algorithms.

Contributions of this paper can be summarized as follows;

- Proposed and evaluated a part-based 3D model retrieval (P3DMR) algorithm called PWRE-net that employs common embedding of partial shape and whole shape via deep-learning based feature transformation.
- Created two benchmark datasets for P3DMR consisting of a training set and a test set.

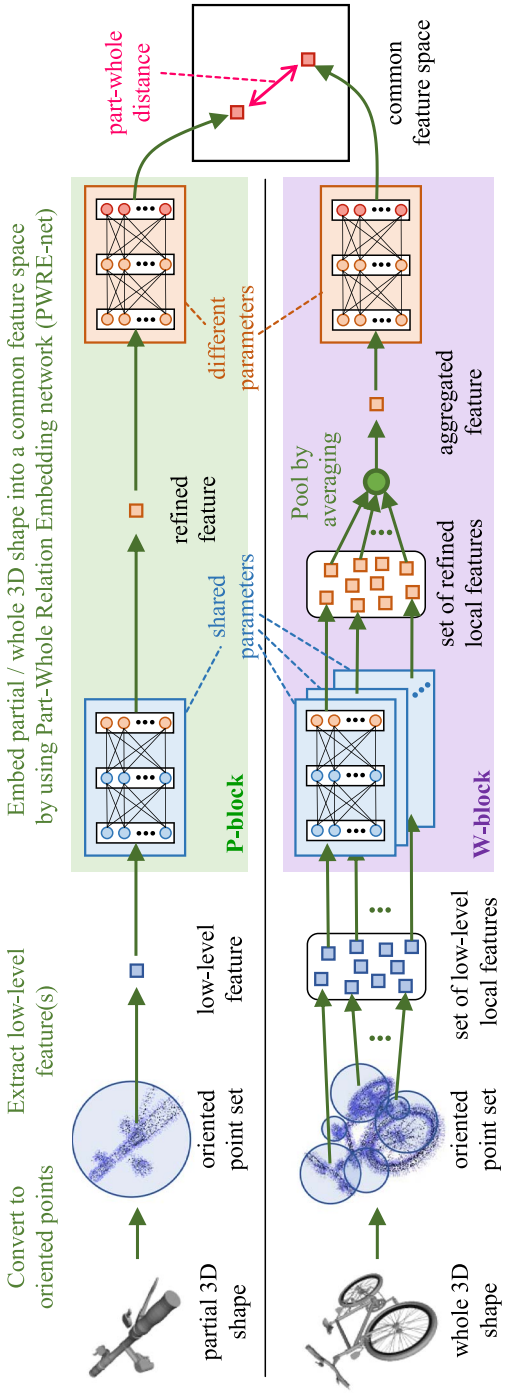


Fig. 1. The Part-Whole Relation Embedding network (PWRE-net) effectively and efficiently compares the partial 3D shape against the whole 3D shape in their common, salient feature space. The PWRE-net learns diverse part-in-whole relations of 3D shapes by using numerous pairs of partial/whole 3D shapes. Training pairs of 3D shapes are automatically generated from a collection of unlabeled whole 3D shapes at very low cost.

- Experimentally evaluated proposed PWRE-net algorithm by using the benchmarks datasets and demonstrated that the proposed deep learning-based approach is effective for the challenging P3DMR task.

This paper is organized as follows. We review related work in the next section. In Section 3, the proposed algorithm is described. Experiments using new benchmark databases and their results are presented in Section 4. We discuss limitation of the algorithm in Section 5, followed by conclusion and future work in Section 6.

2. Related work

2.1. Approaches to P3DMR

The *Part-to-Parts Matching* (*PPsM*) compares a part-based query with parts of a whole 3D shape. Sliding sub-volume search (e.g., Kanezaki et al., 2010; Song and Xiao, 2014) is a representative PPsM-based algorithm. Kanezaki et al. (2010) and Song and Xiao (2014) generate numerous rectangular sub-volumes having diverse scales at every position of 2.5D indoor scene, and features extracted from these sub-volumes are compared against a feature of the query. Another class of PPsM-based algorithms (e.g., Attene et al., 2011; Ferreira et al., 2010; Ip and Gupta, 2007; Shalom et al., 2008; Suzuki et al., 2005) segments the whole 3D shapes into sub-parts for comparison. For example, (Shalom et al., 2008) hierarchically segments a whole 3D shape represented as a manifold mesh into multiple meaningful sub-parts. A tree structured graph, whose nodes correspond to parts of the whole 3D shape, is generated and bipartite graph matching is performed to compare part-based query against parts of the whole 3D shape. (Furuya et al., 2015) employs randomized sub-volume partitioning of the whole 3D shape, and each sub-volume is described by a compact binary 3D geometric feature having invariance against rotation of 3D shapes.

These PPsM-based algorithms effectively identify, or localize, parts similar to the part-based query from whole 3D shapes. Also, to speed up retrieval, several algorithms adopted index structure (Attene et al., 2011; Shalom et al., 2008) or hashing (Furuya et al., 2015) of sub-volume features. However, they suffer from a large memory footprint for a database containing a significant number of 3D models.

Alternatively, the *Part-to-Whole Matching* (*PWM*) approach describes each of the part-based query and the whole 3D shape as a feature, and similarity between the part and the whole is efficiently computed by a comparison among the pair of features. Most of the PWM approach (e.g., Liu et al., 2006; Savelonas et al., 2014; Sfikas et al., 2013) adopts local features and their aggregation to compute features for the whole 3D shapes. (Savelonas et al., 2014) is a P3DMR algorithm for 3D pottery objects. It extracts a set of local 3D geometric features from an entire shape of 3D pottery, and they are aggregated into a single feature per 3D model for efficient comparison against the feature for the query which is a pottery fragment.

Retrieval using the PWM approach is usually more efficient than that using the PPsM approach. That is, only one feature per whole 3D shape need to be stored and compared against a feature of the part-based query. Also, the PWM approach works well if the partial query is sufficiently large (e.g., more than 50% of the entire 3D shape). However, if a smaller part is specified as a query, feature matching using an existing PWM approach would fail since the aggregated features is not optimal for P3DMR. Furthermore, unlike the PPsM approach, the PWM approach has difficulty localizing parts within the whole 3D shape that match the query since positional information of local features is discarded during aggregation. Our approach thus combines PPsM with PWM, in which the latter is used for re-ranking for accuracy and for localization.

2.2. Deep learning for 3D model retrieval

Recently, 3D model retrieval has seen application of deep learning. For W3DMR, in which an entire 3D shape is given as a query, a number of algorithms use Deep Convolutional Neural Network (DCNN) (e.g., Bai et al., 2016; Furuya and Ohbuchi, 2016b; Masci et al., 2015; Su et al., 2015; Wu et al., 2015). They train DCNN by using a set of labeled 3D shapes to build accurate feature detector for 3D shapes. Feature vectors describing 3D models are extracted from a layer close to the final classification layer of the DCNN after training of the DCNN as a classifier is completed.

Deep learning also shows its remarkable accuracy on cross-modal 3D model retrieval. For example, for sketch-based 3DMR, in which a hand-drawn sketch is given as query, Wang et al. (2015) and Zhu et al. (2016) embed both feature of a hand-drawn sketch and feature of a 3D shape into a common latent space by using DCNN. Similarly, (Li et al., 2015) learned a latent feature space shared by both 2D natural images and 3D shapes to perform 3DMR queried by natural images.

Note that successes of these deep learning-based 3DMR algorithms relies heavily on large-scale, labeled datasets (e.g., Deng et al., 2009; Eitz et al., 2012; Wu et al., 2015) to train deep neural networks. However, no dataset large enough to be useful for deep learning of P3DMR existed. This lack of dataset probably explains why deep learning hasn't been applied to P3DMR yet. To deal with this issue, we employ automatic generation of training dataset for P3DMR by using a large set of unlabeled whole 3D shapes.

2.3. Deep learning for part-based 2D image retrieval

Recent progress of deep learning technique for 2D images enables effective and efficient part-based 2D image retrieval (e.g., Mohedano et al., 2016; Salvador et al., 2016; Tolias et al., 2016). These studies employ 2D DCNN to extract accurate visual features from the part-based query image and the retrieval target images. At the convolutional layer in the DCNN, a feature map, or neuron activations computed by convolution, of the query is compared against local regions of feature map of the retrieval target. Matching the feature maps is performed at every convolutional layer in the DCNN to detect 2D object(s) having diverse scales.

To improve P3DMR, a possible approach is to extend the 2D DCNN such as those used in (Mohedano et al., 2016; Salvador et al., 2016; Tolias et al., 2016) to 3D. That is, voxel representation of partial 3D shape and whole 3D shape are fed into the 3D CNN and their 3D feature maps are used for matching. However, we don't employ this 3D CNN-based approach since voxel-based 3D CNN generally generates feature that is not invariant to rotation of objects. As we mentioned in the introduction, P3DMR algorithm should be robust against 3D geometric transformation including 3D rotation of the partial and whole shapes. We thus employ carefully designed 3D geometric feature having invariance to 3D rotation of 3D shape as input to DNN.

3. Proposed algorithm

3.1. Overview of the algorithm

To effectively and efficiently compare partial 3D shapes against whole 3D shapes, we try to learn part-in-whole inclusion relation between partial shapes and whole shapes by means of their embedding into common feature space. Using the embedding, a partial shape and a whole shape are in part-in-whole relation if their features are close in the common feature space. A deep embedding network called *Part-Whole Embedding Network* (*PWRE-net*) is used to map low-level shape features into the common feature space (see Fig. 1) To make the embedded features robust against 3D geometric transformation of 3D shape, the PWRE-net takes as its input a handcrafted, low-level 3D geometric feature having invariance against translation, scaling, and

rotation. Details of the PWRE-net will be described in Section 3.2.

Learning a common embedding feature space for diverse part-whole pairs requires a large number of training samples. Here, a sample is a pair of partial 3D shape and whole 3D shape having a proper part-in-whole relationship. To our knowledge, no dataset containing large enough number of such part-whole pairs exists. Manually creating such a large dataset, e.g., by hand-editing whole shape to generate parts, is impractical. We thus propose a simple yet effective algorithm to automatically generate large number of diverse part-whole shape pairs at low cost from a set of *unlabeled* 3D models found in large-scale 3D shape repositories (e.g., Wu et al., 2015). Section 3.3 describes detailed procedures for generating part-whole pairs and for training the PWRE-net by the using generated part-whole pairs.

Once the PWRE-net is properly trained, part-based 3D model retrieval is performed in two-stage cascade of PWM followed by PPsM. In the first, PWM stage, similarity from the partial shape query to all the retrieval targets (i.e., whole 3D shapes) in the database are computed in the common embedding feature space. In the second stage, PPsM between the query and the top-ranked retrieval results from the first stage is performed to improve ranking results and to localize the part in the target 3D model. Section 3.4 describes details of retrieval.

3.2. Architecture of PWRE-net

The PWRE-net consists of two sub neural networks (or blocks) running in parallel, that are, P-block for embedding partial 3D shapes and W-block for embedding whole 3D shapes (see Fig. 1). The PWRE-net resembles to Siamese network (Chopra et al., 2005), which consists of two deep embedding networks having identical structure. The P-block and the W-block of the PWRE-net, however, have different structures. The P-block is designed to extract salient 3D geometric features of the *partial* 3D shapes. On the other hand, the W-block is designed to extract salient features of the *whole* 3D shapes considering their local, detailed 3D geometric features.

3.2.1. Architecture of P-block

***Input to P-block*:** We feed a low-level 3D geometric feature that represents a partial 3D shape into the P-block. Given a partial 3D shape **P**, which is defined as a polygonal 3D model, **P** is first converted into a set of oriented points. We use an algorithm by (Osada et al., 2002) to convert a polygonal model to an oriented point set. Specifically, the algorithm uses the following equation to sample a point **p** on a triangle surface of 3D model.

$$\mathbf{p} = (1 - \sqrt{r_1})\mathbf{t}_1 + \sqrt{r_1}(1 - r_2)\mathbf{t}_2 + \sqrt{r_1}r_2\mathbf{t}_3 \quad (1)$$

In the equation, $\mathbf{t_1}$, $\mathbf{t_2}$, and $\mathbf{t_3}$ are vertices of the triangle, and r_1 and r_2 are quasi-random number sequences generated by using Sobol's algorithm (Press et al., 1992). Compared to pseudo-random number sequence, quasi-random number sequence samples the surface more uniformly, reducing variance. Given a total number of points per 3D model N_p, the number of points for each triangle is computed in proportion to the area of the triangle. Each point **p** is associated with the normal vector **n** of the triangle on which the point is sampled. We sample $N_p = 1000$ oriented points on a partial 3D shape. Oriented point set of the 3D shape is uniformly scaled to fit a sphere having diameter 1.

The oriented point set is then described by a low-level 3D geometric feature that is invariant against translation, scaling, and rotation in 3D space. We use Surflet-Pair-Relation Histograms (SPRH) (Rusu et al., 2009; Wahl et al., 2003) as a feature for the oriented point set. For each pair of two oriented points **p** and **p'**, associated with their normal vectors **n** and **n′**, we compute three angular statistics, i.e., α, β, and γ, which are calculated as follows;

$$\alpha = \arctan(\mathbf{w}\cdot\mathbf{n}', \mathbf{u}\cdot\mathbf{n}')$$
$$\beta = \mathbf{v}\cdot\mathbf{n}'$$
$$\gamma = \mathbf{u}\cdot(\mathbf{p}' - \mathbf{p})/\delta \quad (2)$$

where $\mathbf{u} = \mathbf{n}$, $\mathbf{v} = \mathbf{u} \times (\mathbf{p}' - \mathbf{p})/\delta$, $\mathbf{w} = \mathbf{u} \times \mathbf{v}$, and $\delta = \|\mathbf{p}' - \mathbf{p}\|_2$

We compute these angular statistics for all the pair of oriented points sampled on the partial 3D shape. The set of triplets, each of which consists of α, β, and γ per oriented point pair, is voted to a three-dimensional joint histogram to form a SPRH feature for the partial 3D shape. (Rusu et al., 2009) uses 5 bins for each statistic resulting in $5 \times 5 \times 5 = 125$-dimensional SPRH feature vector. In this paper, we use more bins to extract richer low-level 3D geometric feature and leave refinement of the feature to the neural network that follows. Specifically, we use 9 bins each to extract $9 \times 9 \times 9 = 729$-dimensional SPRH feature from the oriented point set.

Feature embedding by using P-block: Given a SPRH feature for the partial 3D shape, the P-block embeds the SPRH feature into the salient feature space shared with the W-block. The P-block is a fully-connected neural network with 6 layers including the input and the output (or embedding) layer. The number of neurons for each layer is set to 729, 1024, 1024, 1024, 512, and 128, respectively. 729 neurons at the first layer corresponds to the number of dimensions of the SPRH feature and 128 neurons at the output layer is the dimensionality of the common embedding feature space. To non-linearly transform features, we use ReLU (Krizhevsky et al., 2012) as activation function at all the layers except for the input and output layers. Neuron activations of the output layer is L2-normalized to form the embedded feature of the partial 3D shape.

Before entering the P-block, each SPRH feature is normalized, by using ZCA-whitening algorithm (Bell and Sejnowski, 1996), so that it has zero mean, unit variance, and no correlation among all the pair of dimensions in the SPRH feature vector. Whitening of input features is used commonly to accelerate convergence of deep neural network training.

3.2.2. Architecture of W-block

Input to W-block: The W-block takes as its input a set of local, low-level 3D geometric features extracted from local regions of a whole 3D shape. Given a whole 3D shape **W** represented as a polygonal mesh, it is converted to an oriented point set by using the algorithm described in Section 3.2.1. We sample $N_p = 16{,}000$ oriented points per whole 3D shape, as opposed to $N_p = 1000$ for a partial shape, so that each local region of the whole 3D shape has sufficient number of points for feature extraction. Each local region is defined by a sphere. Center of the sphere coincides with a randomly chosen oriented points of **W**, and radius of the sphere is randomly chosen from the range [0.01, 0.4]. We sample 500 local regions per whole 3D shape.

Each local region is then described by another low-level shape feature Point Feature Histogram (PFH) (Rusu et al., 2009). The PFH is essentially localized SPRH, so the computation is quite similar to the SPRH. That is, triplets of the angular statistics (α, β, γ) are computed from all the pair of oriented points within the local region and these triplets are voted to form a joint histogram. Similar to the P-block, we use 9 bins to extract a 729-dimensional low-level PFH feature per local region. We obtain 500 PFH features per whole 3D shape.

We used a sphere having random location and radius to define a local region for simplicity and efficiency. Alternatives are possible, e.g., automatic mesh-manifold-based segmentation of whole 3D shape into parts (e.g., Shalom et al., 2008), or cutting off cuboids having random position, orientation, and aspect ratio (Furuya et al., 2015). We avoided these alternatives as mesh-manifold-based segmentation can be computationally expensive and random cuboids (Furuya et al., 2015) are often very thin so that they often won't match query 3D shape.

We used a set of local shape features, as opposed to a global shape feature (e.g., Wahl et al., 2003), to describe the whole 3D shape. We think that a global shape feature won't capture enough local geometry of the whole shape for part-whole matching. When refined, aggregated, and then refined again for embedding, a set of local features would better represent partial geometry of a whole 3D shape.

Algorithm 1
Automatic part-whole pair generation.

Input: · a set of N_w pairs of unlabeled whole 3D shape $\mathbf{W}$ and its 3D shape feature $\mathbf{f}$ (e.g., DkSA-POD): $S = \{(\mathbf{W}_i, \mathbf{f}_i)\}$, $i = 1, \ldots, N_w$
· number of nearest neighbors: k
· total number of training pairs: N_t
Output: · a set of $N_t / 2$ positive pairs: $Pos = \{(\mathbf{P}_i, \mathbf{W}_i)\}$, $i = 1, \ldots, N_t / 2$
· a set of $N_t / 2$ negative pairs: $Neg = \{(\mathbf{P}_i, \mathbf{W}_i)\}$, $i = 1, \ldots, N_t / 2$

```
Initialize sets of training pairs: Pos ← ∅, Neg ← ∅;
while | Pos | + | Neg | < N_t do
    Randomly pick up (W_a, f_a) from S;
    Cut off a partial shape P_a from W_a by using a sphere having random
    position and random radius;
    Compute k nearest neighbors of W_a, i.e., kNN(W_a), by comparing f_a with
    all the 3D shape features in S;
    Append all the k positive pairs to Pos:
    Pos ← Pos ∪ {(P_a, W_i) | W_i ∈ kNN(W_a)}, i = 1, ..., k;
    Append randomly selected k negative pairs to Neg:
    Neg ← Neg ∪ {(P_a, W_i) | W_i ∈ S \ kNN(W_a)}, i = 1, ..., k;
end
return Pos, Neg;
```

Feature embedding by using W-block: The W-block is also a fully-connected neural network. But it differs from the P-block in that the W-block has the aggregation layer which pools a set of (refined) local features into a single feature per whole 3D shape for further refinement and embedding. Given a set of local features (i.e., PFH features) for the whole 3D shape, each PFH feature is first fed into the W-block independently for individual refinement. Then, in the aggregation layer placed at the middle of the W-block, the set of refined local features is pooled, by averaging, into a single feature per whole 3D shape. The aggregated feature is embedded into the common feature space via the latter half of the W-block to be compared against the embedded features of a partial 3D shape.

The W-block has 7 layers, all fully connected except for the aggregation layer placed in the middle. The aggregation layer is put after the third fully connected layer. Numbers of neurons for the layers, not including the aggregation layer, are 729, 1024, 1024, 1024, 512, and 128.

As with the P-block, each PFH feature is ZCA-whitened prior to entering the W-block. Output of the W-block, embedded feature for a whole 3D shape, is L2-normalized and placed in the common feature space to be compared against features of partial 3D shapes.

Note that the first three layers of the P-block and W-block share their parameters, i.e., weights for the edges connecting adjacent layers. This is because they both perform local feature refinement, and sharing makes training easier by reducing total number of parameters of the PWRE-net. It also regularizes training, and reduces memory requirements of the neural network.

3.3. Effective training of PWRE-net

To train the PWRE-net, we need two types of part-whole pairs, i.e., *positive* pairs and *negative* pairs. A positive pair consists of a partial 3D shape $\mathbf{P}_{\mathbf{pos}}$ and a whole 3D shape $\mathbf{W}_{\mathbf{pos}}$ that includes $\mathbf{P}_{\mathbf{pos}}$. A negative pair consists of a partial 3D shape $\mathbf{P}_{\mathbf{neg}}$ and a whole 3D shape $\mathbf{W}_{\mathbf{neg}}$ which does not include $\mathbf{P}_{\mathbf{neg}}$.

3.3.1. Automatic part-whole pair generation

We propose a simple algorithm for automatically generating numerous and diverse part-whole pairs for training. Fig. 2 shows the basic idea for the algorithm and its procedure is summarized in Algorithm 1. The algorithm requires a set S of (unlabeled) whole 3D shapes. We assume that each whole 3D shape in S is represented as an oriented point set. Given a whole 3D shape $\mathbf{W}_a$ picked up from S, we carve out a partial shape $\mathbf{P}_a$ from $\mathbf{W}_a$ by using a sphere having random position and random radius. $\mathbf{P}_a$ is a set of oriented points of the whole 3D shape enclosed by the sphere. We make a positive pair from $\mathbf{P}_a$ and $\mathbf{W}_a$ since $\mathbf{P}_a \subset \mathbf{W}_a$. A negative pair is formed from $\mathbf{P}_a$ and a whole 3D shape $\mathbf{W}_b$ sampled from S such that $\mathbf{P}_a \subset \mathbf{W}_a \neq \mathbf{W}_b$.

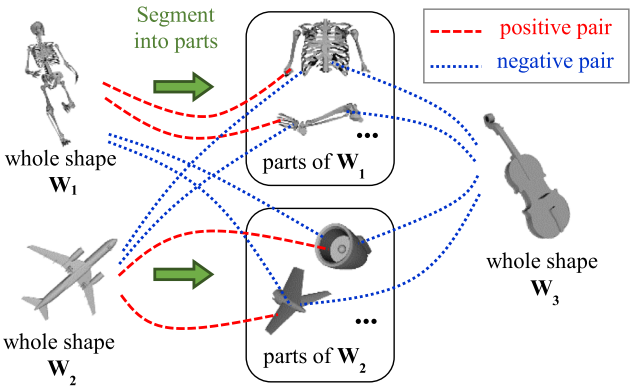


Fig. 2. Basic idea for automatic part-whole pair generation. Each positive pair consists of a whole 3D shape $\mathbf{W}$ and its part automatically segmented from $\mathbf{W}$. A negative pair is formed between $\mathbf{W}$ and a part of the other whole 3D shapes.

Nevertheless, if $\mathbf{W}_a$ and $\mathbf{W}_b$ happen to have very similar global shapes, they are expected to have similar parts. For example, two 3D shapes of car usually have tires, steering wheels, side view mirrors, etc. in common. Based on this observation, we make positive pairs from not only between $\mathbf{P}_a$ and $\mathbf{W}_a$ but also from $\mathbf{P}_a$ and other whole 3D shapes having global shape similar to $\mathbf{W}_a$. To do so, we first find kNN($\mathbf{W}_a$), that are, a set of k whole 3D shapes that are most similar to $\mathbf{W}_a$. The kNN() is performed in the feature space of 3D shape feature DkSA-POD (Furuya and Ohbuchi, 2016a) and k is set to 10 in the experiments. Note here that kNN($\mathbf{W}_a$) includes $\mathbf{W}_a$ itself. The set of positive pairs Pos_a and the set of negative pairs Neg_a for the partial 3D shape $\mathbf{P}_a$ are defined as follows;

$$\begin{aligned} Pos_a &= \{(\mathbf{P}_a, \mathbf{W}_i) | \mathbf{W}_i \in k\mathrm{NN}(\mathbf{W}_a)\} \\ Neg_a &= \{(\mathbf{P}_a, \mathbf{W}_i) | \mathbf{W}_i \in S \setminus k\mathrm{NN}(\mathbf{W}_a)\} \end{aligned} \tag{3}$$

Since we have a number (e.g., 3k) of whole 3D shapes in S, and the number of sub-parts per whole 3D model is significant (e.g., 1k), the number of all the possible positive and negative pairs is quite large. Using these all pairs for training of the PWRE-net is impractical. In this paper, we randomly sample 2M part-whole pairs, consisting of 1M positive pairs and 1M negative pairs, from the set of possible combinations and use the sampled pairs to train PWRE-net.

The algorithm described above runs a risk of generating noisy part-whole training pairs, due to randomness (of radius and position) in defining partial shape $\mathbf{P}_a$, inaccuracy of the shape features used in defining a set of k similar whole shapes, or the value k. In practice, however, the algorithm generated diverse training part-whole shape pairs in sufficient number with small enough noise for the training of deep neural network.

3.3.2. Training of PWRE-net

The PWRE-net is trained so that it embeds the feature of $\mathbf{P}_{\mathbf{pos}}$ and the feature of $\mathbf{W}_{\mathbf{pos}}$ in proximity to each other, and embeds the feature of $\mathbf{P}_{\mathbf{neg}}$ and the feature of $\mathbf{W}_{\mathbf{neg}}$ away from each other. Such an objective can be formalized as the contrastive loss function L (Chopra et al., 2005);

$$\begin{aligned} L = &\sum_{(\mathbf{P}_{\mathbf{pos}}, \mathbf{W}_{\mathbf{pos}}) \in Pos} d(f(\mathbf{P}_{\mathbf{pos}}), f(\mathbf{W}_{\mathbf{pos}})) \\ &+ \sum_{(\mathbf{P}_{\mathbf{neg}}, \mathbf{W}_{\mathbf{neg}}) \in Neg} \max(0, m^2 - d(f(\mathbf{P}_{\mathbf{neg}}), f(\mathbf{W}_{\mathbf{neg}}))) \end{aligned} \tag{4}$$

where f indicates feature embedding of a 3D shape computed by using

Algorithm 2
Training PWRE-net.

Input:
- embedding function, i.e., PWRE-net, parameterized by Θ: $f(\mathbf{x}; \Theta)$
- a training set of $N_t / 2$ positive pairs $Pos = \{\mathbf{P}_{\mathbf{pos}}, \mathbf{W}_{\mathbf{pos}}\}$
- a training set of $N_t / 2$ negative pairs $Neg = \{\mathbf{P}_{\mathbf{neg}}, \mathbf{W}_{\mathbf{neg}}\}$
- mini-batch size: N_b
- initial learning rate: η
- margin for contrastive loss function: m
- number of training epochs: N_e

Output:
- trained embedding function: $f(\mathbf{x}; \Theta)$

```
Randomly initialize parameters Θ by using the algorithm by He et al., 2015;
for i = 1 : N_e do
    for j = 1 : N_t / N_b do
        Form a mini-batch {(P, W)} consisting of N_b / 2 positive pairs and N_b / 2 negative pairs;
        Embed the mini-batch to obtain { (f(P), f(W)) };
        Compute loss by using Equation (4);
        Update parameters Θ by using Adagrad algorithm;
    end
end
return f(x; Θ);
```

the PWRE-net, d is a squared Euclidean distance between the two embedded features, and m is a margin that controls distances among embedded features of negative pairs. We fix m to 1.0 in the experiments. Parameters, or weights for edges, of the PWRE-net are randomly initialized by using the method proposed by (He et al., 2015). We use Stochastic Gradient Descent algorithm with mini-batch size = 32 for the training. To adaptively assign learning rate to each parameter, we employ Adagrad algorithm (Duchi et al., 2011) with initial learning rate = 0.1. Training is iterated for 10 epochs. Algorithm 2 describes the training procedure.

3.4. P3DMR by using the PWRE-net

Before the retrieval, features in the common embedding space of all the retrieval targets, that are, all the whole 3D shapes in the database, are computed by using the trained PWRE-net. The features are then stored in the database. Retrieval by using the proposed algorithm is performed in two stages. Algorithm 3 summarizes procedures for retrieval.

Algorithm 3
Two-stage retrieval by using trained PWRE-net.

Input:
- a part-based query: **P**
- a set of N pairs of retrieval target whole 3D shape and its embedded feature: $\{(\mathbf{W}_i, f(\mathbf{W}_i)\}$, i = 1, …, N
- trained embedding function, i.e., PWRE-net: $f(\mathbf{x}; \Theta)$
- number of top-ranked 3D shapes for re-ranking: N_r

Output:
- a ranked list of retrieval targets: L

```
First retrieval stage:
    Embed P into the common feature space to obtain f(P);
    Compute Euclidean distances among f(P) and {f(W_i)};
    Sort the N distances in ascending order to generate an initial ranked list L_init;
Second retrieval stage (re-ranking):
    Pick N_r top-ranked retrieval targets from L_init;
    Compute PPsM-based distances among P and N_r retrieval targets;
    Combine, by summing, distances computed at 1st stage and distances computed at 2nd stage;
    Re-order top-N_r retrieval targets in L_init by sorting the N_r combined distances to generate a final ranked list L;
return L;
```

At the first retrieval stage, a feature in the common embedding space of the given partial 3D shape query is computed by using the trained PWRE-net. Then, the feature of the query is compared against the features of the retrieval target whole shapes in the database by using Euclidean distance. The distances between the query and the retrieval targets are sorted in ascending order to yield initial ranking results.

The second retrieval stage attempts to further improve retrieval accuracy and to localize parts of the retrieval targets that match the query. To do so, we carefully compare the query and the top-ranked retrieval targets by using the PPsM approach. Specifically, we first pick $N_r = 20$ top-ranked retrieval targets from the initial ranking results and extract a set of low-level 3D geometric features from each of these N_r retrieval targets. We sample a set of 4000 oriented points per retrieval target and extract 300 PFH features from local regions segmented by using local spheres with random position and radius. The SPRH feature of the query is compared against the 300 PFH features of the retrieval target by using Euclidean distance. Minimum among these 300 distances becomes an overall PPsM distance between the query and the retrieval target. Final ranking of a whole shape included in the top N_r retrieval of the 1st stage is computed by summing, with equal weights, the distances from the 1st and 2nd stages.

Localizing parts of the retrieved whole 3D shapes that match the query is performed as follows. We assign a matching score to each part of the N_r whole 3D shapes. Matching score for each part is computed by inverting the distance between the query and the part obtained at the second retrieval stage. For each of the N_r whole 3D shapes, a partial shape having maximum score is localized as a match to the query shape.

We also visualize the result of localization by using the method described in (Furuya et al., 2015). That is, we color parts of the whole 3D shapes according to the similarities among the query and the parts of the whole 3D shapes. Each similarity value is normalized in the range [0, 1], and then, to visualize localization, the similarity is mapped to hue in HSV color space.

Retrieval using the proposed algorithm is efficient both in terms of time and memory footprint. The first retrieval stage only has to store low-dimensional embedded features that represent the whole 3D shapes (not the numerous sub-volumes of the whole 3D shapes) in the database. Distance between a pair of the query and the retrieval target is quickly obtained by comparing their embedded features once. Although the second retrieval stage performs PPsM, its computation is efficient since only $N_r = 20$ top-ranked retrieval targets are processed for re-ranking and localization of parts.

4. Experiments and results

4.1. Experimental setup

4.1.1. Benchmark databases

Since the existing benchmark databases for P3DMR (e.g., Dutagaci et al., 2009; Furuya et al., 2015; Pratikakis et al., 2016) do not include a set of training whole 3D shapes, we can't use these benchmarks to evaluate the proposed algorithm. We therefore create two new benchmark databases, i.e., P-ModelNet and P-SH11NR, each of which consists of a training set used to train the PWRE-net and a test set used to evaluate retrieval accuracy and efficiency. Fig. 3 shows examples of partial 3D shapes and whole 3D shapes contained in the benchmarks.

P-ModelNet: The P-ModelNet benchmark is built as a subset of the ModelNet dataset (Wu et al., 2015) which contains diverse rigid 3D shapes. The training set for the P-ModelNet includes 2832 whole 3D shapes classified into 16 object categories such as airplane, chair, car, piano, etc. The test set consists of a query set having 200 partial 3D shapes and a retrieval target set with 322 whole 3D shapes. The

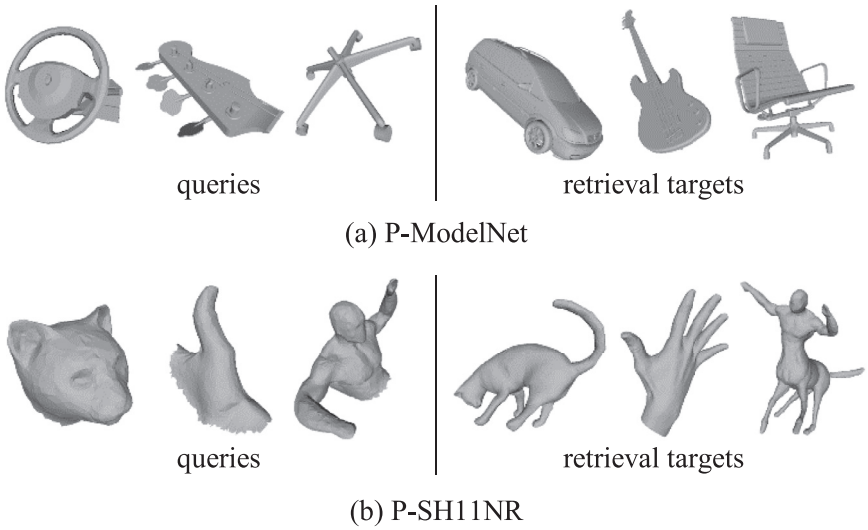


Fig. 3. Examples of part-based queries and retrieval target whole 3D shapes contained in the benchmark databases.

retrieval target set has the same object categories with the training set but these sets are disjoint each other. Each partial 3D shape was created by manually cutting off a part of the whole 3D shape in the retrieval target set. Ground truth, i.e., the set G_q of correct retrieval targets for the query $\mathbf{P_q}$, was defined as follows. First, the whole 3D shape, from which the query $\mathbf{P_q}$ was cut off, was added to G_q. We then visually checked if each of remaining retrieval targets contains similar part(s) to $\mathbf{P_q}$, and if so, we include the retrieval target in G_q. For example, G_q for a query shape of steering wheel contains whole 3D shapes of vehicle (e.g., car or bus) only if a steering wheel is defined inside these 3D shapes.

P-SH11NR: The P-SH11NR was originally created by (Furuya et al., 2015) based on non-rigid 3D shapes included in the SHREC 2011 Non-rigid (SH11NR) dataset (Lian et al., 2011). The original P-SH11NR has only the test set, i.e., the query set of 180 partial 3D shapes and the retrieval target set including 600 non-rigid 3D shapes classified into 30 object categories (e.g., bird, cat, human, octopus, etc). In the experiments, we randomly pick up 300 whole 3D shapes from the retrieval targets and group them into a training set. The remaining 300 whole 3D shapes are used as retrieval targets. Ground truth for each query 3D shape is the same as the original P-SH11NR benchmark.

Note that, both for the P-ModelNet and the P-SH11NR, we don't use the object category labels attached to the training 3D shapes during training of the PWRE-net. We use Nearest Neighbor (NN) [%], Mean Average Precision (MAP) [%], Recall-Precision curve as accuracy indices. NN is also referred as Precision@1 which means accuracy of a retrieved target ranked in the top of ranking results. Since training of the PWRE-net is essentially affected by randomness due to random initialization of parameters or automatic training pair generation, we conducted every experiment for 3 times and their average accuracy will be reported. We used a PC having two *Intel Xeon E5-2650V2* (8 cores, 16 threads each) CPUs, a *NVIDIA GeForce GTX 980* GPU, and 256GByte DRAM. Training of PWRE-net using 2M part-whole pairs took about 2–3 days.

Although P-ModelNet and P-SH11NR enable us to quantitatively evaluate PWRE-net, scales of these two benchmark datasets are quite small. In addition, 3D shapes in these datasets are “too clean” since they were created by using 3D modeling software. Ideally, we should also quantitatively evaluate PWRE-net under (1) larger-scale setting by using more diverse 3D shapes and (2) more realistic setting by using “noisy” 3D shapes generated with, for example, 3D range scanners. However, even if we could collect such 3D shapes, manually creating ground-truth for P3DMR is quite laborious. Therefore, we conduct qualitative evaluation by using two existing datasets of 3D models without ground-truth for P3DMR.

ShapeNet Core55: We use ShapeNet Core55 (Chang et al., 2015), which is one of the largest datasets of 3D CAD models, to evaluate scalability (i.e., learning capability) of PWRE-net. ShapeNet Core55 contains diverse rigid 3D shapes classified into 55 object categories. The training set and the test set of ShapeNet Core55 includes 35,764 and 10,265 whole 3D shapes, respectively. We use the training set to train PWRE-net and use the test set as retrieval targets. We use the same query set with P-ModelNet.

ObjectScans: We use a Large Dataset of Object Scans (Choi et al., 2016) to evaluate robustness of PWRE-net against various noise, such as variation in distance, holes, cracks, self-occlusions, and background clutter that occurs in range-scanned 3D models. We use a set of 401 polygonal 3D models classified into 10 object categories such as bicycle, chair, plant, etc. Each 3D model is reconstructed from a depth image sequence of real-world object(s) acquired with 3D range scanners. The set of 3D models is randomly split into a training set of 201 3D models and a test set (i.e., retrieval target) of 200 3D models. Prior to converting the 3D model into oriented point set, a floor on which objects lie is removed by using RANSAC algorithm (Fischler and Bolles, 1981) so that excessive number of points are not sampled on the floor. 45 part-based queries are created by manually cutting off the parts of the 3D models in the test set.

4.1.2. Competitors to proposed algorithms

We compare the proposed algorithm against three existing P3DMR algorithms (RSVP (Furuya et al., 2015), SV-PFH, and SV-DSIFT) and two baseline algorithms using deep learning (BF + DNN and FV + DNN).

RSVP: The RSVP algorithm is classified into PPsM approach which compares a query against numerous sub-volumes of retrieval targets. The RSVP segments each retrieval target into a set of sub-volumes, or cuboids, by using 3D grids having random interval and random orientation. Each sub-volume is then described by a compact binary 3D geometric feature. Distance between a pair of the query and the retrieval target is computed by comparing the binary feature of the query against all the binary features of the retrieval target. Hamming distance is used to quickly compare these binary features. We use the original parameter settings as (Furuya et al., 2015) for the experiments.

SV-PFH, SV-DSIFT: These algorithms are PWM approach which represents both the query and the retrieval target as a single feature per query/target. The SV-PFH (or SV-DSIFT) extracts a set of PFH features (or a set of DSIFT features (Furuya and Ohbuchi, 2009)) either from the query or the retrieval target. DSIFT is a local 2D image feature extracted from multi-view rendered images of the 3D shape. The set of local features is aggregated by using Super Vector coding (Zhou et al., 2010) to a feature per 3D shape for efficient comparison. The SV-PFH

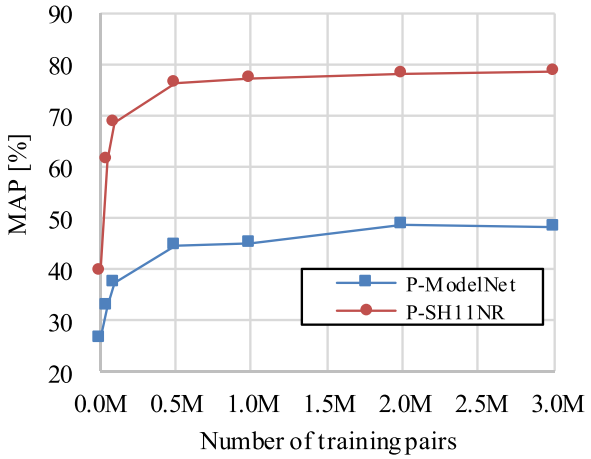


Fig. 4. Number of training pairs and retrieval accuracy.

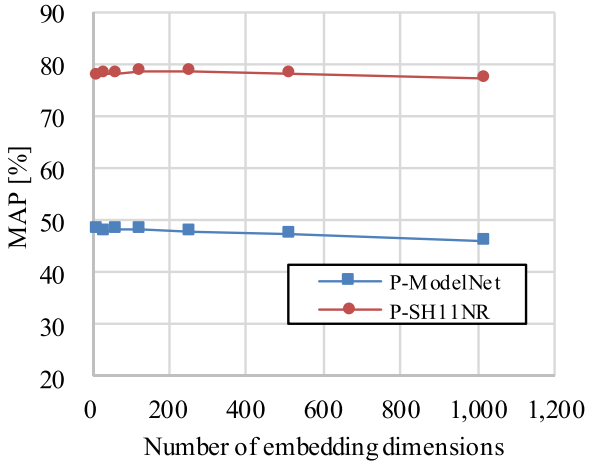


Fig. 5. Number of dimensions for embedded feature space and retrieval accuracy.

algorithm is quite similar to (Liu et al., 2006) and (Savelonas et al., 2014) in that they both employ local 3D geometric feature and their aggregation.

BF + DNN, FV + DNN: PWRE-net refines, aggregates, and embeds 3D shape features by using DNNs. To gauge importance of using DNNs for feature refinement and feature aggregation, we replace the first half of PWRE-net with the existing feature encoding algorithms. Specifically, BF + DNN uses Bag-of-Features (BF) algorithm (Csurka et al., 2004) while FV + DNN uses Fisher Vector (FV) coding algorithm (Perronnin et al., 2010) to refine and aggregate 3D shape features. A SPRH feature extracted from a part-based query is encoded by using either BF or FV. On the other hand, a set of PFH features representing a retrieval target is aggregated to a single feature by using BF or FV. These encoded/aggregated features are embedded into their common feature space via the embedding DNN (i.e., the latter half of the PWRE-net). We use codebook size of 4096 for BF and 16 for FV, respectively. Therefore, the embedding DNN takes as input feature vectors having 4096 dims. for BF + DNN and $2 \times 729 \times 16 = 23{,}328$ dims. for FV + DNN. The embedding DNN is trained under the same setting with PWRE-net as described in Section 3.3.

4.2. Experimental results

4.2.1. Hyper parameters for PWRE-net

Number of training pairs: Fig. 4 plots retrieval accuracies against the number of training pairs generated by using the proposed automatic part-whole pair generation algorithm. In the figure, "Number of training pairs" is sum of the number of positive pairs and the number of negative pairs, which are evenly sampled from the training set of whole 3D shapes. Obviously, increasing the number of training pairs improves retrieval accuracy. We can observe that 2M training pairs are sufficient to train the PWRE-net both for the P-ModelNet and the P-SH11NR benchmarks. These results suggest that learning numerous and diverse part-in-whole relation of 3D shape is effective for accurate P3DMR. We speculate that much larger number of training set pairs would be required for a large-scaler dataset having more diverse 3D shapes classified into larger number of object categories.

Number of embedding dimensions: Fig. 5 plots MAP scores against the number of dimensions for the embedded feature space shared by partial 3D shapes and whole 3D shapes. To conduct this experiment, we varied the numbers of neurons in the output layers for P-block and W-block, and 2M training pairs are used to train the PWRE-net. Interestingly, the numbers of embedding dimensions have only small impact on retrieval accuracy. Although slight peaks can be observed at around 100 dimensions, MAP scores are almost constant (over 40% for P-ModelNet and over 70% for P-SH11NR) from 4 to 1024 dimensions we have experimented. We speculate that the P-ModelNet and the P-SH11NR do not require high dimensional space for feature embedding since these datasets comprise relatively small number of object categories. In such datasets, relations among partial 3D shapes and whole 3D shapes are not complicated and hence the PWRE-net could find good feature embedding into the low-dimensional space.

We visualized the embedded feature space yielded by the PWRE-net. Fig. 6 shows embedded features of the partial 3D shapes and the whole 3D shapes in the test set of P-ModelNet. To visualize the features, we first embedded partial/whole 3D shapes into a 64-dimensional common feature space by using PWRE-net, and then, they were re-embedded into 2-dimensional space by using t-SNE algorithm (Maaten and

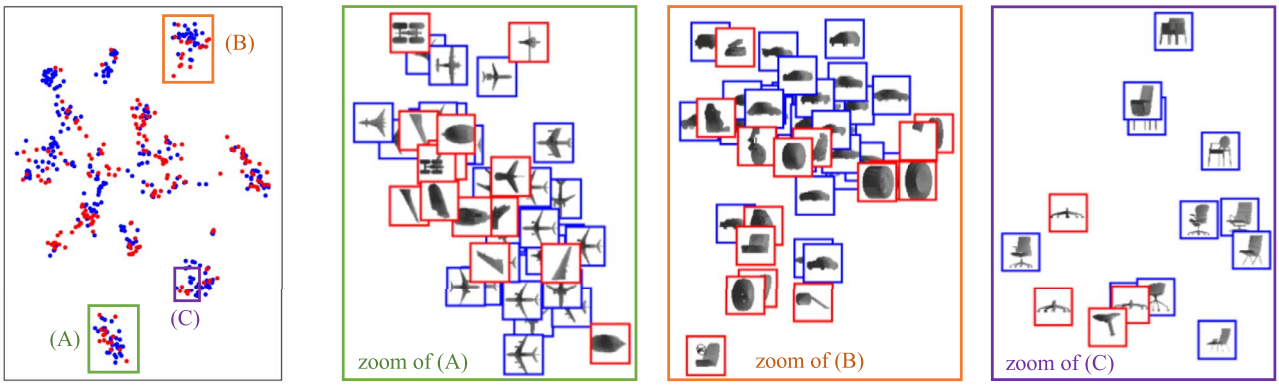


Fig. 6. t-SNE visualization of the embedded feature space. Left figure plots all the embedded features of part-based queries (red) and retrieval targets (blue) in the P-ModelNet. Right three figures are zooms of the embedded feature space. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Hinton, 2008). We can see that whole 3D shapes (e.g., airplanes) and their parts (e.g., wing, jet engine, and landing gear) are embedded nearby each other.

Input to PWRE-net: This subsection experimentally verifies that the combination of SPRH and PFH features is an appropriate choice for the input to PWRE-net. To this end, we substitute the other 3D shape features for SPRH and PFH. We use five 3D shape features, i.e., D2 (Osada et al., 2002), AAD (Ohbuchi et al., 2005), Spin Image (SI) (Johnson and Hebert, 1999), RoPS (Guo et al., 2013), and POD (Furuya and Ohbuchi, 2015). For the experiments, global 3D shape features i.e., D2 and AAD, are localized to extract a set of local features from a whole 3D shape. A set of local spheres is sampled from the whole 3D shape, and each local sphere is described by D2 or AAD. Similarly, local 3D shape features, i.e., SI, RoPS, and POD, are globalized to represent a global region of a partial 3D shape as a single feature vector. We also compare SPRH/PFH features having 9 bins with those having (original) 5 bins to show that using more bins contributes to higher retrieval accuracy. The hyper-parameters for training (i.e., number of training pairs, number of embedding dimensions, initial learning rate, etc.) are fixed throughout the experiments.

Table 1 compares retrieval accuracies of the seven combinations of 3D shape features input to PWRE-net. SPRH/PFH with 9 bins significantly outperforms the other features. We can also observe that accuracies of SI, RoPS, and POD suffer both in P-ModelNet and P-SH11NR. We presume low accuracies of these features would be due to failure of normalizing 3D rotation of spherical regions from which the features are extracted. SI, RoPS, and POD normalize rotation of the region by using either a normal vector at the center of the region or principal axes of points within the region. When we generate training part-whole pairs, we sample parts from a whole 3D shape by using spheres having random location and scale. Results of orientation normalization of such randomly chosen spheres are likely to be quite different from those of part-based queries. The discrepancy could have resulted in the low accuracy.

On the other hand, SPRH/PFH, as well as D2 and AAD, are inherently rotation invariant, without requiring rotation normalization. They achieve rotation invariance by using pairwise statistics of oriented points. In addition, via histogramming the statistics, they form similar feature vectors even if location and scale of the spherical regions vary somewhat.

4.2.2. Comparison against existing P3DMR algorithms

Retrieval accuracy: Table 2 compares retrieval accuracies of the seven P3DMR algorithms. In the table, "PWRE-net + reranking" performs both the first and the second retrieval stages of the proposed algorithm. That is, initial ranking results are generated by comparing the embedded features produced by the PWRE-net, and the top 20 retrieved whole 3D shapes are re-ranked by using the PPsM approach. "PWRE-net" performs only the first retrieval stage, i.e., the initial ranking results are used for evaluation.

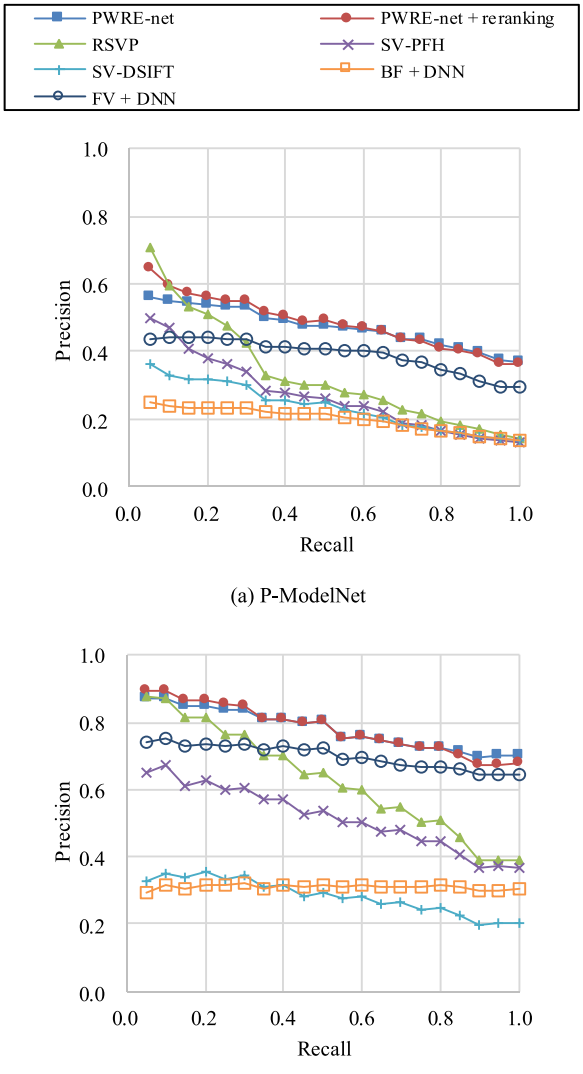
As shown in Table 2, the proposed algorithm significantly outperforms the existing P3DMR algorithms (except for NN in the P-ModelNet against RSVP). Re-ranking further improves retrieval accuracy, especially for NN score, due to careful matching among part-based query and sub-volumes of the top 20 whole 3D shapes in the initial ranking results. Comparison among the three PWM approaches (i.e., SV-PFH, SV-DSIFT, and PWRE-net) verifies the effectiveness of learning part-in-whole relations of 3D shapes; accuracies of PWRE-net exceed those of SV-PFH and SV-DSIFT with large margins. Fig. 7 compares Recall-Precision curves for the seven algorithms. The proposed algorithm keeps higher precisions at high recall, indicating that it can retrieve more relevant whole 3D shapes that have similar part(s) to the query. Fig. 8

Table 1
Comparison of 3D shape features fed into PWRE-net.

Input to PWRE-net		P-ModelNet	P-SH11NR
P-block	W-block	MAP [%]	MAP [%]
D2	localized D2	24.1	45.7
AAD	localized AAD	37.2	66.6
globalized SI	SI	14.9	11.3
globalized RoPS	RoPS	20.6	25.1
globalized POD	POD	27.8	25.3
SPRH (5 bins)	PFH (5 bins)	45.4	74.6
SPRH (9 bins)	PFH (9 bins)	48.4	78.3

Table 2
Comparison of retrieval accuracies [%].

Algorithms	P-ModelNet		P-SH11NR	
	NN	MAP	NN	MAP
RSVP	71.2	34.9	86.3	63.5
SV-PFH	45.5	28.6	60.0	52.5
SV-DSIFT	31.8	25.1	28.9	28.9
BF + DNN	19.3	21.2	25.9	31.4
FV + DNN	38.5	40.4	70.6	70.4
PWRE-net	53.0	48.4	85.7	78.3
PWRE-net + reranking	64.2	49.7	88.9	78.6



(a) P-ModelNet

(b) P-SH11NR

Fig. 7. Recall-Precision curves for the P3DMR algorithms.

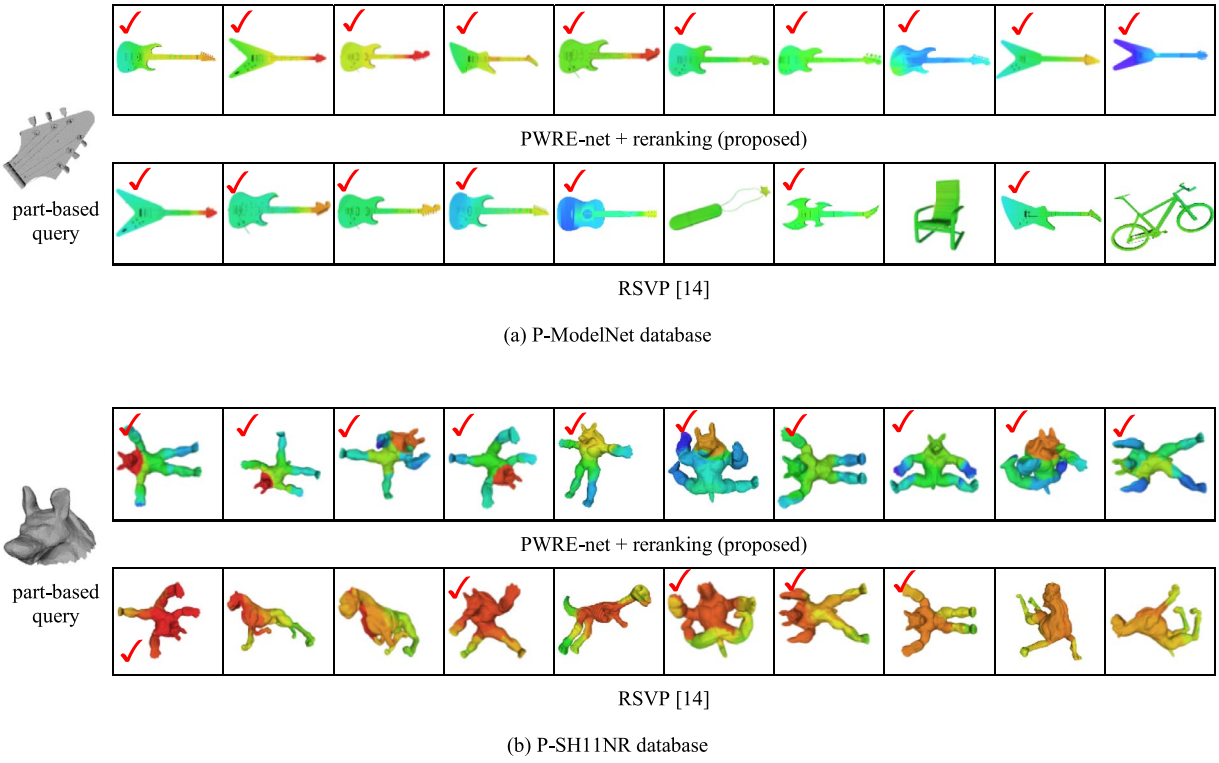


Fig. 8. Examples of retrieval rankings for the P-ModelNet test set (a) and the P-SH11NR test set (b). For each part-based query, the first row shows top 10 retrieved whole 3D shapes by the proposed algorithm and the second row is those by the RSVP (Furuya et al., 2015) algorithm. Checkmarks indicate "correct" results for the query. Colors on the surface of whole 3D shape depict similarity between the query and the part of the whole 3D shape (High similarity for red, low similarity for blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Comparison of retrieval efficiency for P-ModelNet.

Algorithms	Computation time [s] per query				Memory footprint [Mbytes]
	Feat.	Dist.	Rerank.	Total	
RSVP	2.298	0.002	–	2.300	42.3
SV-PFH	1.481	0.026	–	1.507	451.1
PWRE-net	0.136	0.002	3.441	3.579	13.5

shows examples of retrieval results, as well as localization of parts, produced by PWRE-net and RSVP. The PWRE-net clearly yields favorable ranking order than the RSVP.

Comparison among BF + DNN, FV + DNN, and PWRE-net shows the effectiveness of processing feature refinement and feature aggregation by DNN. Training by using BF-encoded features fails probably due to information loss caused by vector quantization of the input 3D shape features. FV + DNN performs better than BF + DNN since FV can encode richer information of the 3D shape features than BF. PWRE-net outperforms both of them by a large margin.

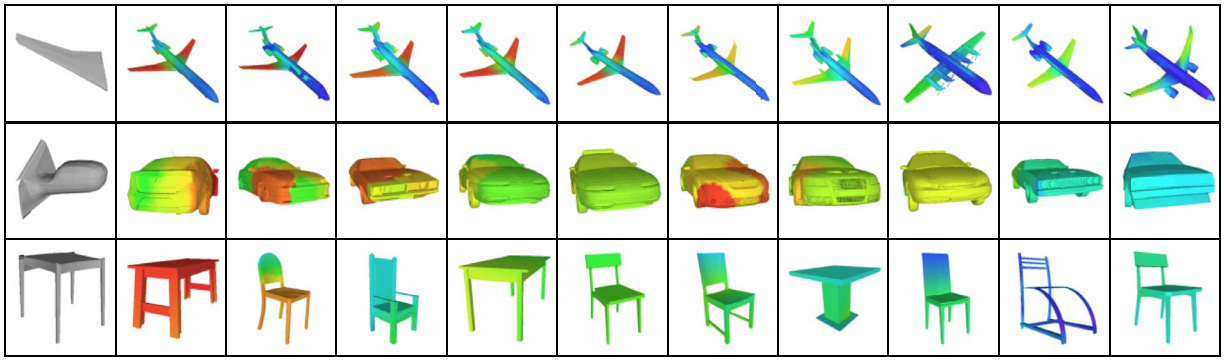
Retrieval efficiency: Table 3 compares efficiency for querying the P-ModelNet database. In Table 3, "feat." indicates time for feature extraction from a part-based query, "dist." is time for distance computation among the feature of the query and the features of 322 retrieval targets in the database, and "rerank." is time for re-ranking. "Memory footprint" shows spatial cost for the features of retrieval targets and the other data used for feature extraction (i.e., parameters for PWRE-net or SV codebook for RSVP and SV-PFH).

Although the PWRE-net with re-ranking is the slowest among the three algorithms we have compared, 3.5 seconds per query is acceptable for practical use. Note that computation time of the proposed algorithm is dominated by re-ranking. If we omit the re-ranking, the PWRE-net processes a query much faster than the RSVP and the SV-PFH. The PWRE-net is also memory-efficient; it requires only 13.5 Mbytes to store parameters of the PWRE-net and embedded features of the 322 retrieval targets. The PWRE-net could scale to larger databases since it represents each whole 3D shape as a compact (128-dimensional) feature which occupies only 512 bytes per 3D shape.

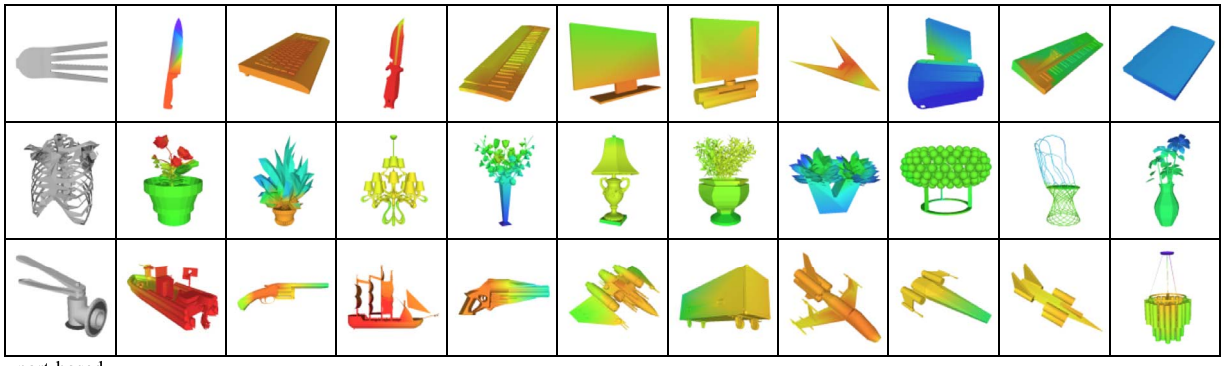
4.2.3. Evaluation under large-scale and realistic settings

Large-scale setting: To demonstrate learning capability of PWRE-net, we conducted retrieval experiment by using ShapeNet Core55 dataset. PWRE-net was trained by using 2M part-whole pairs sampled from 35,764 3D models in the training set. Fig. 9 shows examples of retrieval results. As shown in Fig. 9(a), querying airplanes, cars, and chairs by their partial shapes succeeds probably due to richness in number of these whole 3D models contained in the training set. On the other hand, as shown in Fig. 9(b), querying by partial shapes of fork, skeleton, or fire extinguisher fails since the training set contains few or no 3D models of these objects. Although we can't evaluate in a quantitative manner, these results suggest that PWRE-net could learn diverse part-whole relation of 3D shapes as long as sufficient number of 3D models can be used for training.

Realistic setting: We also demonstrate robustness of PWRE-net against various types of noise occurred in scanned 3D models such as holes, cracks, self-occlusions, and background clutter. PWRE-net was trained by using 2M part-whole pairs generated from 201 3D models in the ObjectScans training set. Fig. 10 shows examples of retrieval rankings for the ObjectScans test set. Although not perfect, PWRE-net produces good retrieval results queried by the partial shapes of plant, motorbike, and bicycle. PWRE-net would be able to learn part-in-whole relation of 3D shapes even if they contain geometric and topological noise and background clutter. This experimental result is encouraging since it suggests PWRE-net could be applied to part-based retrieval of not only synthetic 3D models but also more realistic 3D models generated with 3D range scanners. Further detailed and quantitative



(a) successful cases



(b) failure cases

Fig. 9. Examples of retrieval rankings for the large-scale dataset (ShapeNet Core55 test set). (a) PWRE-net can learn part-in-whole relation of diverse 3D object categories if sufficient number of 3D models are available for training. (b) However, for some object categories with very small number of training 3D models, learning part-in-whole relation fails and PWRE-net produces retrieval ranking of irrelevant 3D models.

evaluation under realistic setting should be considered, but we leave this issue to future work.

5. Discussion

In the previous section, we experimentally demonstrated superior retrieval accuracy and efficiency of PWRE-net to its competitors. However, PWRE-net has one significant drawback; Feature embedding by PWRE-net has difficulty in handling "multimodality" of partial 3D shape. Multimodality here means that a particular 3D shape exists as a part of multiple objects belonging to different categories. For example, a wheel can be included in diverse objects such as a car, bus, airplane, chair, piano, etc. Nevertheless, as illustrated in Fig. 6, PWRE-net embeds 3D shapes of wheel close to 3D shapes of cars, not close to airplanes and chairs. In such an embedding feature space, retrieval ranking queried by 3D shape of wheel would be occupied by 3D shapes of car, which is not necessarily desirable result.

The embedding feature space of Fig. 6 was formed probably because

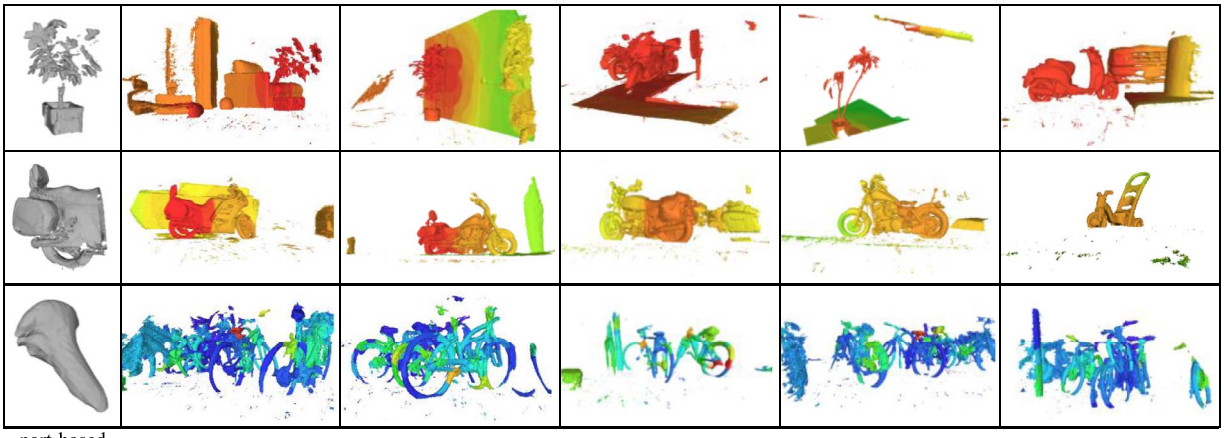


Fig. 10. Examples of retrieval rankings for the realistic dataset (ObjectScans test set). Although rank and localization are not perfect, PWRE-net produces favorable rankings even if both part-based queries and retrieval targets contain geometric and topological noise as well as cluttered backgrounds.

the automatic part-whole pair generation algorithm produced more wheel-car pairs than wheel-chair pairs and wheel-airplane pairs. The difference in number of part-whole pairs would depend on the proportion of volume of partial 3D shape to volume of whole 3D shape. Compared to wheels of a chair or an airplane, wheels of a car have larger proportion of volume in the whole 3D shape. Therefore, automatic part-whole pair generation algorithm, which samples local spheres at random position and random scale as partial shapes, is more likely to cut off wheels from cars than from chairs or airplanes. Consequently, most of the partial 3D shapes of a wheel are embedded close to whole 3D shapes of cars through the training that tries to minimize the contrastive loss function.

To solve the problem of multimodality of partial 3D shapes, we would require different network architecture and/or loss function from PWRE-net. One possible direction is to utilize the matching DNN (Han et al., 2015; Žbontar and LeCun, 2015). Instead of feature embedding, the matching DNN predicts a matching score, or a similarity, of two input data by using a combination of their features. The matching DNN concatenates two features of the input data at the middle of the DNN and the combined feature is further transformed through the subsequent network for prediction. The multimodality problem could be alleviated by using the matching DNN since it can simultaneously consider both partial and whole 3D shapes to predict their similarity. Properly trained matching DNN would yield high similarities for any input part-whole pairs of wheel-car, wheel-chair, and wheel-airplane. The matching DNN, however, also has a drawback; it suffers from high temporal cost for retrieval since all the pairs among the query and the retrieval targets need to be fed into the DNN at the retrieval stage to generate ranking list of retrieval targets.

6. Conclusion and future work

Part-based 3D Model Retrieval (P3DMR) is technically quite challenging. We don't know position, scale, and orientation of part(s) of 3D model that is specified by a part-based query shape. Also, we don't know which 3D model in a database contains a partial shape that matches to the query. Previous approaches to P3DMR suffer from inaccuracy or inefficiency to compare a part-based query shape against retrieval target whole 3D shapes in the database. Part-to-Parts Matching (PPsM) approach requires very high computational cost for matching the feature of part-based query to the features of all the sub-volumes of whole 3D shapes. On the other hand, Part-to-Whole Matching (PWM) approach, which approximates the inclusion test by using aggregation of local features extracted from the whole 3D shape, often suffers from low retrieval accuracy.

This paper proposed a novel P3DMR algorithm called Part-Whole Relationship Embedding network (PWRE-net). The algorithm learns, from a large number of part-whole shape pairs, a common embedding feature space that places two shapes together if one includes the other. Using the learned embedding space, part-whole inclusion can be tested very quickly by nearest neighbor search for an efficient P3DMR. The embedding is realized by using a pair of deep neural networks (DNNs) that transforms low-level 3D geometric features representing either part or whole into features in the common embedding space. A large number of diverse part-whole shape pairs necessary to train the DNNs are generated automatically from unlabeled 3D shapes. Experimental evaluation using newly created P3DMR benchmark datasets showed that the PWRE-net produced superior accuracy, speed, and spatial cost than previous P3DMR algorithms.

A possible future work is to quantitatively test the algorithm using a larger and more realistic 3D model database. Another is to improve quality and diversity of the part-whole shape pairs for training. This can be done, for example, by combining multiple definitions of locality for segmentation, e.g., using both sphere in 3D Euclidean space and circle on 2-manifold to generate parts from whole.

Acknowledgments

This research is supported by JSPS Grant-in-Aid for Young Scientists (B) #16K16055.

References

Attene, M., Marini, S., Spagnuolo, M., et al., 2011. Part-in-whole 3D shape matching and docking. Vis. Comput. 27 (11), 991–1004.
Bai, S., Bai, X., Zhou, Z., Zhang, Z., Latecki, L.J., 2016. GIFT: a real-time and scalable 3D shape search engine. In: CVPR 2016, pp. 5023–5032.
Bell, A., Sejnowski, T.J., 1996. Edges are the `independent components' of natural scenes. In: NIPS 1996.
Chang, A.X., Funkhouser, T., Guibas, L., et al., ShapeNet: an information-rich 3D model repository, arXiv, arXiv:1512.03012, 2015.
Choi, S. Zhou, Q-Y., Miller, S., et al., A large dataset of object scans, arXiv, arXiv:1602.02481, 2016.
Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In: CVPR 2005, pp. 539–546.
Csurka, G., Dance, C.R., Fan, L., et al., 2004. Visual categorization with bags of keypoints. In: ECCV 2004 Workshop on Statistical Learning in Computer Vision, pp. 59–74.
Deng, J., Dong, W., Socher, R., et al., 2009. ImageNet: a large-scale hierarchical image database. In: CVPR 2009, pp. 248–255.
Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. In: JMLR. 12. pp. 2121–2159.
Dutagaci, H., Godil, A., Axenopoulos, A., et al., 2009. SHREC'09 Track: querying with partial models. In: EG 3DOR 2009, pp. 69–76.
Eitz, M., Hays, J., Alexa, M., 2012. How do humans sketch objects. In: ACM TOG. 31 Article No.44.
Ferreira, A., Marini, S., Attene, M., et al., 2010. Thesaurus-based 3D object retrieval with part-in-whole matching. In: IJCV. 89. pp. 327–347.
Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Communications of the ACM, pp. 381–395.
Furuya, T., Kurabe, S., Ohbuchi, R., 2015. Randomized sub-volume partitioning for part-based 3D model retrieval. In: EG3DOR 2015, pp. 15–22.
Furuya, T., Ohbuchi, R., 2016a. Accurate aggregation of local features by using k-sparse autoencoder for 3D model retrieval. In: ICMR 2016, pp. 293–297.
Furuya, T., Ohbuchi, R., 2016b. Deep aggregation of local 3D geometric features for 3D model retrieval. In: BMVC 2016.
Furuya, T., Ohbuchi, R., 2009. Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features. In: CIVR 2009, Article No.26.
Furuya, T., Ohbuchi, R., 2015. Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval. In: Proc. ICMR 2015, pp. 171–178.
Guo, Y., Sohel, F., Bennamoun, M., et al., 2013. Rotational projection statistics for 3D local surface description and object recognition. In: IJCV. 105. pp. 63–86.
Han, X., Leung, T., Jia, Y., et al., 2015. Matchnet: Unifying feature and metric learning for patch-based matching. In: Proc. CVPR 2015, pp. 3279–3286.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: ICCV 2015, pp. 1026–1034.
Ip, C.Y., Gupta, S.K., 2007. Retrieving matching CAD models by using partial 3D point clouds. Comp. Aided Des. Appl. 4 (5), 629–638.
Johnson, A.E., Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. In: IEEE TPAMI. 21. pp. 433–449.
Kanezaki, A., Harada, T., Kuniyoshi, Y., 2010. Partial matching of real textured 3D objects using color cubic higher-order local auto-correlation features. Vis. Comput. 26 (10), 1269–1281.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. In: NIPS 2012, pp. 1097–1105.
Li, Y., Su, H., Qi, C.R., et al., 2015. Joint embeddings of shapes and images via CNN image purification. In: ACM TOG. 34 Article No.234.
Lian, Z., et al., 2011. SHREC'11 Track: shape retrieval on non-rigid 3D watertight meshes. In: EG 3DOR 2011, pp. 79–88.
Liu, Y., Zha, H., Qin, H., 2006. Shape topics: a compact representation and new algorithms for 3D partial shape retrieval. In: CVPR 2006, pp. 2025–2032.
Liu, Z.B., Bu, S.H., Zhou, K., et al., 2013. A survey on partial retrieval of 3D shapes. J. Comput. Sci. Technol. 28 (5), 836–851.
Maaten, L.V.D., Hinton, G.E., Nov 2008. Visualizing data using t-SNE. In: JMLR. 9. pp. 2579–2605.
Masci, J., Boscaini, D., Bronstein, M.M., et al., 2015. Geodesic convolutional neural networks on Riemannian manifolds. In: ICCV 3dRR, pp. 37–45.
Mohedano, E., McGuinness, K., O'Connor, N.E., Salvador, A., Marques, F., Giro-i-Nieto, X., 2016. Bags of local convolutional features for scalable instance search. In: Proc. ICMR 2016, pp. 327–331.
Ohbuchi, R., Minamitani, T., Takei, T., 2005. Shape-similarity search of 3D models by using enhanced shape functions. In: IJCAT. 23. pp. 70–85.
Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. In: ACM TOG. 21. pp. 807–832.
Perronnin, F., Sánchez, J., Mensink, T., 2010. Improving the fisher kernel for large-scale image classification. In: Proc. ECCV 2010, pp. 143–156.
Pratikakis, I., et al., 2016. SHREC'16 track: partial shape queries for 3D object retrieval. In: EG 3DOR 2016.
Press, W.H., et al., 1992. Numerical Recipes in C–The art of Scientific Computing. Cambridge University Press, Cambridge, UK, pp. 309–315.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: ICRA 2009, pp. 3212–3217.
Salvador, A., Giro-i-Nieto, X., Marques, F., Satoh, S., 2016. Faster R-CNN features for instance search. In: Proc. DeepVision: Deep Learning in Computer Vision Workshop at CVPR 2016.
Savelonas, M.A., Pratikakis, I., Sfikas, K., 2014. Fisher encoding of adaptive fast persistent feature histograms for partial retrieval of 3D pottery objects. In: EG 3DOR 2014, pp. 61–68.
Sfikas, K., Pratikakis, I., Koutsoudis, A., et al., 2013. 3D object partial matching using panoramic views, ICIAP 2013. In: LNCS. 8158. pp. 169–178.
Shalom, S., Shapira, L., Shamir, A., et al., 2008. Part analogies in sets of objects. In: EG 3DOR, pp. 33–40 2008.
Song, S., Xiao, J., 2014. Sliding shapes for 3D object detection in depth images. In: ECCV 2014, pp. 634–651.
Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3D shape recognition. In: ICCV 2015, pp. 945–953.
Suzuki, M.T., Yaginuma, Y., Yamada, T., et al., 2005. A partial shape matching method for 3d model databases. In: SEA 2005, pp. 389–394.
Tolias, G., Sicre, R., Jégou, H., 2016. Particular object retrieval with integral max-pooling of CNN activations. In: Proc. ICLR 2016.
Wahl, E., Hillenbrand, U., Hirzinger, G., 2003. Surflet-pair-relation histograms: a statistical 3D-shape representation for rapid classification. In: 3DIM 2003, pp. 474–481.
Wang, F., Kang, L., Li, Y., 2015. Sketch-based 3d shape retrieval using convolutional neural networks. In: CVPR 2015, pp. 1875–1883.
Wu, Z., Song, S., Khosla, A., et al., 2015. 3D ShapeNets: a deep representation for volumetric shape modeling. In: CVPR 2015, pp. 1912–1920.
Žbontar, J., LeCun, Y., 2015. Computing the stereo matching cost with a convolutional neural network. In: Proc. CVPR 2015, pp. 1592–1599.
Zhou, X., Yu, X., Zhang, T., Huang, T.S., 2010. Image classification using super-vector coding of local image descriptors. In: ECCV 2010, pp. 141–154.
Zhu, F., Xie, J., Fang, Y., 2016. Heat diffusion long-short term memory learning for 3D shape analysis. In: ECCV 2016, pp. 305–321.