ANNOTATING 3D MODELS AND THEIR PARTS VIA DEEP FEATURE EMBEDDING

Kouki Omata, Takahiko Furuya, Ryutarou Ohbuchi

University of Yamanashi
cannotbe18@gmail.com, takahikof@yamanashi.ac.jp, ohbuchi@yamanashi.ac.jp

ABSTRACT

Need to organize 3D shape data has prompted studies on comparison and retrieval of 3D shape models. Being able to query 3D shape models by words, in addition to 3D model examples and 2D sketches, would be quite beneficial. This paper proposes a method to associate whole 3D models (e.g., automobile) as well as their parts (e.g., tire, body, engine) with word labels so that the 3D model can be queried by words. The associations between 3D shapes and words are learned from a dataset of 3D models whose whole model and segmented parts are labeled with words. Feature vectors of these words (distributed representation) and feature vectors of whole and partial geometries of 3D models are embedded, by Word Shape embedding Network (WSN) into a common feature embedding space. As the word feature vectors are learned by Word2Vec trained on Wikipedia corpus, the common embedding space can be queried by a wide variety of words that are not included in the labeled 3D model dataset. Experimental evaluation has shown that, with the proposed algorithm, 3D shape can be queried by labels of either whole or part shape, or labels that are semantically close but not included in the original 3D model dataset.

Index Terms— 3D model retrieval, 3D shape similarity comparison, label propagation, deep feature embedding.

1. INTRODUCTION

Prevalent modalities that have been tried for querying 3D shapes have been 3D shape examples and 2D sketches, both of which could specify geometric shape in one way or another. However, these query modalities are not the most convenient for human beings to use. A user typically does not have 3D shape close enough in shape to the desired 3D models. Sketches of 3D models are more accessible, but sketches, with their abstraction, variation in drawing style, etc. are difficult to compare with 3D shape. It is thus beneficial if 3D models can be queried by words. To do so, we must somehow associate 3D shape with words.

Previous work on labeling 3D models known to the authors associates 3D model with words [1][2] used geometric similarity of (whole) shape of 3D models to propagate word annotations from labeled 3D models to unlabeled 3D models. While these approaches achieved certain retrieval accuracy when queried using the words contained in the labeled training 3D model dataset. However, the set of words is very small and limited, and the words are associated only with the whole 3D shape. Queries based on part name, or queries based on synonyms or hypernyms can't be done.

This paper explores an approach to text-based 3D model query by associating meaningful textual labels to 3D models (e.g., airplane) and their meaningful parts (e.g., wing, body, tail, engine, …) in labeled 3D model datasets (Figure 1). Query can be made by words originally contained in the labeled 3D model dataset the method is trained with, or it can be queried by words that are close in meaning to those words. For example, a 3D model of airplane may be queried by original labels of either whole shape ("airplane") or partial shapes ("body", "wing", etc.). Or, it can be queried by related words such as "fighter", "plane", "airfoil" (synonym of a part), or "vehicle" (hypernym). To this end, the proposed algorithm embeds feature vectors of both 3D geometry and words into a common embedding space by the Word Shape embedding Network (WSN), a triplet of Deep Neural Networks (DNNs) (See Figure 2). The common feature space is necessary to compare words and 3D shapes that are inherent different in their nature. The 3D geometry feature vectors of both whole and partial shape of 3D models are extracted by using a DNN. The feature vectors of words, or distributed representation of words, are computed from words associated with labels of 3D model database as well as a text corpus (e.g., Wikipedia snapshot).

The following of this paper is structured as follows. In the next section, related work is presented. It is followed by proposed method in Section 3, experimental evaluation in Section 4, and summary and conclusion follow in Section 5.

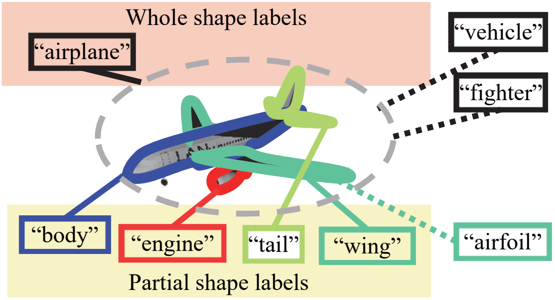


Fig. 1. Proposed method relates textual labels to both whole shape ("airplane") and its part ("body", "wing", etc.).

2. RELATED WORK

2.1. Annotating 3D CAD models with words

In their pioneering work, Goldfeder et al. [1] added word labels to 3D models without labels (unlabeled 3D models) via shape similarity comparison of 3D models. Given a set of labeled 3D models, those words are propagated to their respective *k*-nearest neighbor 3D models based on their similarity in their shape feature space. Ohbuchi et al [2] followed up on Goldfeder's work by using salient feature space induced by manifold learning of shape feature vectors for label propagation. Since the training dataset contain both labeled and unlabeled dataset, the method of [2] is a form of semi-supervised learning. Both these two methods have two limitations; only whole shape of 3D models are labeled, and that only those labels that appeared in the training set can be used to label (unlabeled) 3D models.

Using the proposed method, 3D models may be queried by labels of either their whole shape or partial shape, and that the set of words is expanded to include those semantically related words not found in the training 3D model dataset.

2.2. 3D shape classification and segmentation

Earlier work on 3D shape classification and segmentation using DNN, such as 3D ShapeNets [11] by Wu and VoxNet [12] by Maturana, employed voxel representation as their input 3D shape representation. Voxels may not be the best shape representation, however. Conversion from such 3D shape representation as polygonal surfaces into voxels inherently involve approximation and simplification, losing significant shape details in the process. This is especially true if voxel resolution is limited to about 32^3 or so.

Recently, DNNs that could directly accept such alternative shape representations as point set and triangular mesh have appeared. PointNet [5] and SO-net [13] accepts 3D point set as their input, while FeaStNet [14] accepts polygonal mesh as its input by using graph convolution. DLAN [3] and DLSF [10] accepts polygonal mesh as their input, but internally convert it into 3D oriented point representation to extract hand-crafted local geometrical features. These features are then processed by DNNs. DLSF is inherently invariant to rotation and robust against articulation thanks to its hand-crafted feature.

In this paper, we employ PointNet [5], with a modification, to extract part and whole shape features. Polygonal mesh 3D models in databases are converted into point set representation before they are given to PointNet. PointNet features are unaffected by order of input 3D points due to maximum pooling in the network. Significantly, PointNet could also perform segmentation of 3D point set into meaningful parts if trained by using a training dataset of segmented 3D models, such as ShapeNetPart [7]. We leverage this ability of segmentation for identifying and labeling parts of 3D models.

2.2. Distributed representation of word

In natural language processing, a distributed representation is a feature vector of a word in an embedding space of words. Word2Vec [4], fastText [8], and Doc2Vec [9] are examples of methods to obtain word feature vector. These methods are trained on a large text corpus, e.g., a snapshot of Wikipedia.

In the proposed method, we use Word2Vec to produce word feature vectors to be embedded into a common embedding space of 3D shape and words.

3. PROPOSED METHOD

3.1. Overall approach

Figure 2 shows the overall processing flow of the proposed algorithm using WSN. It consists of the following three parts, feature extraction, feature embedding, and labeling.

(1) **Feature extraction:** Word feature vectors are extracted from words associated with 3D models in the training set using Word2Vec [4]. To train WSN, word features of the labeled 3D models and their parts in the training dataset are extracted. In the retrieval phase, word feature of the query word is extracted.

Also, for each 3D model in the dataset, a shape feature of an entire 3D model ("whole shape feature") and shape features of the model's meaningful parts ("partial shape feature") are extracted using modified PointNet.

(2) **Feature embedding:** Word as well as shape features are embedded into a common embedding feature space by the Word Shape embedding Network (WSN) so they can be compared. The WSN consists of three neural nets, one each for word, whole shape, and partial shape. The WSN is trained by using contrastive loss [16] so that the word and shape having the same training label are close, while those having different labels are distant.

(3) **Retrieval / Labeling:** After training the WSN, features of both 3D shape models and words are embedded in the common embedding feature space, ready for comparison. For word query based retrieval, given a word as the query, its feature in the embedding space is computed. Then, shape features, either of whole or partial, closest to the word feature in the embedding space by *L2-norm* are retrieved.

To label an unlabeled 3D shape, choose its shape feature in the embedding space, and find those words that are closest to the shape feature as the shape's label.

This approach allows any word in a rich text corpus used to train Word2Vec to be used as query. A 3D model could of course be queried by using words contained as labels of whole or partial shapes of the 3D model. In addition, a rich set of words, including synonyms, hypernyms, or hyponyms of words used as labels, may also be used to query the 3D model. Note that this is not the case for the earlier algorithm

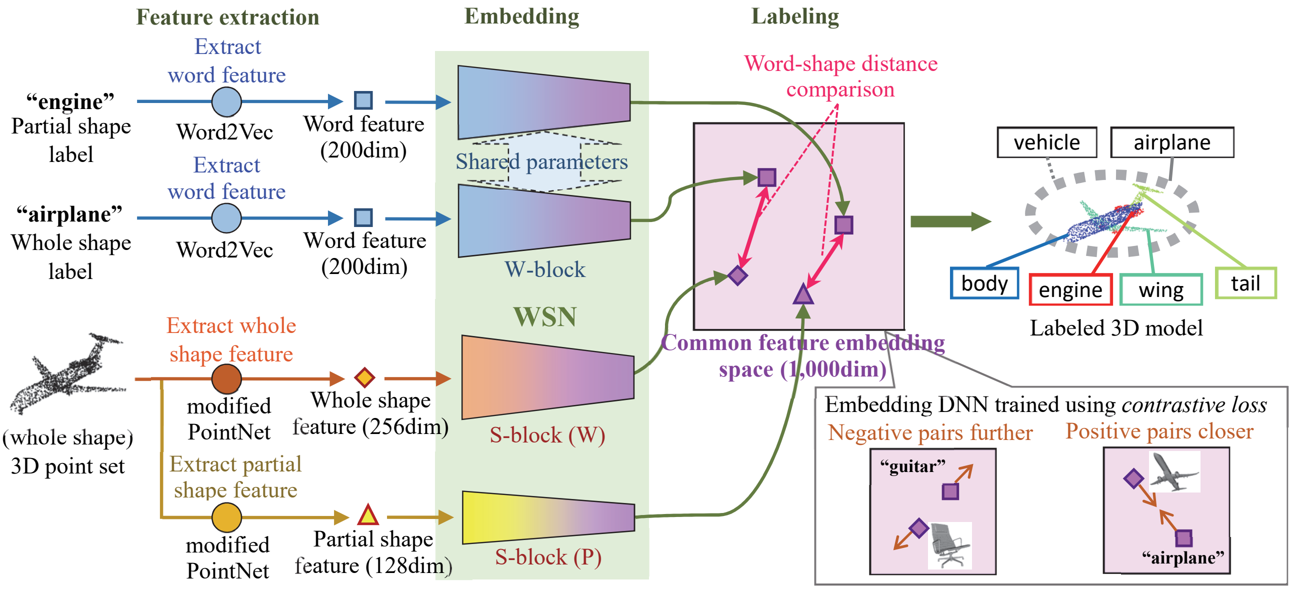


Fig. 2. Feature vectors of labels of whole shape ("airplane") and partial shapes ("body", "wing", etc.) are transformed by W-block to vectors in the common feature space. 3D shape features of the whole shape and partial shapes are transformed by S-bock (W) and S-block (P), respectively, to vectors in the common feature space.

by Goldfeder [1] or by Ohbuchi [2]. In these algorithms, only small set of words used as labels of 3D models in a database can be used for queries.

3.2. Shape and word feature embedding network WSN

WSN consists of three subparts; (1) W-block, or Word embedding block, for word embedding, (2) S-block(P), or shape embedding block for partial shape, and (3) S-Block(W), or shape embedding block for whole shape. W-block accepts a word feature vector and transforms it to a feature vector in the common feature embedding space via 4-layer fully-connected neural network. S-block(W) and S-block(P), both of which are also 4-layer fully connected neural networks, accept whole and partial shape feature vectors, respectively, and transform them to vectors in the common embedding feature space. No parameter sharing is done among W-block, S-block(P) and S-block(W).

We set the dimensionality of word feature of Word2Vec at 200. Dimensionality of whole shape feature and partial shape feature are set at 256 and 128, respectively. The dimensionality of common embedding space is chosen by experiments as 1,000 from a few alternatives. (Comprehensive search for optimal values of all the hyper parameters of WSN is too expensive for us to conduct.)

The number of neurons are, for W-block 200-1,000-4,000-1,000), for S-block(W) 256-2,000-8,000-1,000, and for S-block(P) 128-2,000-8,000-1,000. We used Rectified Linear Unit (ReLU) activation function [15] for all the neurons. Activation of the last layer's 1,000 neurons are L2-normalized to produce 1,000 dimensional vector for the embedding feature space. We note that, while far from comprehensive, we experimented with several different network architectures for the WSN to find the best performing one.

3.3. 3D shape feature extraction

Whole and partial 3D shape features are extracted by using a network based on PointNet [6]. PointNet accepts a set of 3D points, and recognizes its whole shape. Whole shape feature is simply activation of the last fully connected layer of the (standard) PointNet. We train the PointNet for (whole) shape recognition, and use the activation of 256 neurons of the last fully connected layer of the PointNet as 256 dimensional feature vector.

PointNet can segments a point set into meaningful parts by adding labels of the parts to each 3D point. This of course requires supervised training using segmented dataset such as ShapeNetPart [7]. Our algorithm regards such a segment generated by PointNet as a partial shape. If a whole 3D model is segmented into n parts, n partial shape features will be generated for the model. To obtain a partial shape feature of a segmented part, we aggregate all the per-point shape feature of points contained in the segment by using max-pooling (See Figure 3.) Each per-point shape feature vector is a set of activation of the last layer of per-point segmentation network of the PointNet. We aggregate all the per-point shape features of points having the same segmentation label by using the max-pooling layer. As per-point feature has dimensionality of 128, an aggregated per-segment feature vector also has dimensionality of 128.

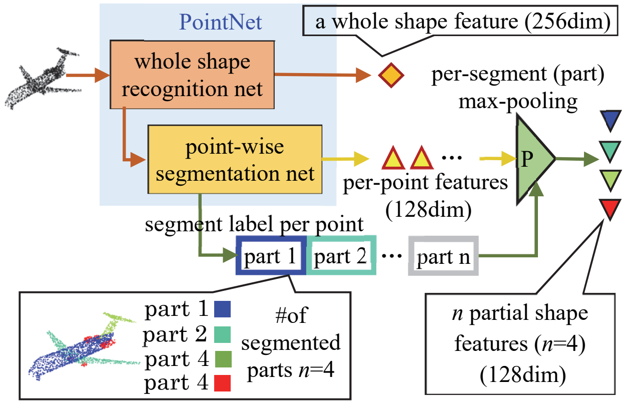

PointNet
a whole shape feature (256dim)
whole shape recognition net
per-segment (part) max-pooling
point-wise segmentation net
per-point features (128dim)
P
segment label per point
part 1
part 2
part n
part 1
part 2
part 4
part 4
#of segmented parts n=4
n partial shape features (n=4) (128dim)

Fig. 3. A partial shape feature is an aggregation of per-point shape features contained in a segmented part (e.g., wing).

3.4 Word feature extraction

Word feature, or distributed representation of word, having dimensionality 200 is computed by using Word2Vec [4]. We chose Word2Vec over fastText, for the former showed better accuracy in our preliminary experiment (63.1% over 60.4% in f-measure for whole shape labeling accuracy).

Some of the words that appeared in the labels of 3D models are not found in the training set of the pre-trained Word2Vec model. We located sentences in the Word2Vec training set that contain synonyms of those unfound words, and replaced these synonyms with the unfound words of the 3D model labels. We then retrained the Word2Vec model. For the training, we used CBOW having window size 4.

3.5. Training WSN

WSN is trained to embed heterogeneous features of shape and word by using contrastive loss [16]. Training data for contrastive loss are pairs of data, that are either positive or negative. A positive pair consists of a shape and its correct label, e.g., a 3D model of human and the word label "human". A negative pair consists of a shape and an unrelated word, e.g., a human model paired with the word "automobile".

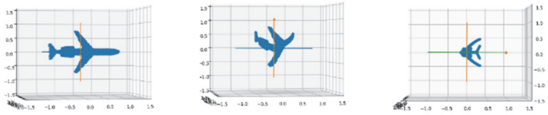
An effective training requires (1) equal number of pairs across classes, and (2) equal number of negative and positive pairs. ShapeNetPart [7], with its segmentation and labeling, has significant variation in sizes of classes. Furthermore, using a straightforward pairing strategy, about 200k positive pairs and 11,000k negative pairs are generated.

To smooth out class imbalance, we try to match sizes of classes to that of the class having the largest number of positive pairs. To do so, we increase number of positive pairs in classes having smaller number of positive pairs by duplication. After the number of positive pairs are evened out, negative pairs for the class are generated to match the number (after inflation) of the positive pairs. In the end, we generated about 2 million each of positive as well as negative pairs for training. These balancing of the number of training pairs improve labeling accuracy quite significantly (in f-measure, from 54.3% to 63.1%).

The number of 3D models in ShapeNetPart, is not large enough for training DNNs. We thus perform data augmentation of 3D models via three methods below.

(1) Scale: Scale the 3D model along one of 6 axes, that are, *x*, *y*, and *z* coordinate axes and three principal axes computed via principal component analysis (PCA) of point distribution of each 3D point set model.

(2) Bending: Bend the 3D model off from a plane determined by the principal axes. Control points for the bending, and the direction and amount of displacement are determined randomly. The following pictures show original (left) and exaggerated examples of bending.



(3) Region growing: Segmentation produced by PointNet is somewhat arbitrary. In an attempt to randomize the segmentation, we grow the size of each segmented region randomly using one of several predetermined methods.

We implemented the WSN using TensorFlow. A hyper parameter of the contrastive loss, the margin size *m*, is set at 0.07 via preliminary experiments. Optimization of the neural network loss function is done by using Adam [17] with initial learning coefficient of 0.0003. All the parameters of the neural network are randomly initialized. The training is done until decrease in contrastive loss converges. We iterated until about 200 epochs, which took about 3 days using Intel Core i7-6700 paired with NVIDIA GeoForce GTX Titan.

4. EXPERIMENTS AND RESULTS

We perform quantitative evaluation of (1) label accuracies of whole shape labeling and partial shape labeling, and (2) impact data augmentation has on labeling accuracy.

We also wish to do quantitative analysis of retrieval accuracy using words semantically related to but different from word labels attached to the training 3D models. However, we know of no dataset of 3D models with ground truth of related semantically words. Instead, we show examples of query by words including some semantically related words.

4.1. 3D shape model dataset

For quantitative evaluation of accuracy of whole shape labeling, we used ShapeNetCore55 [6] dataset. To qualitatively evaluate whole and partial shape labeling, we used ShapeNetPart [7] dataset (Figure 4 for examples).

The ShapeNetPart is a dataset in which all the 3D models are segmented into parts, and all the parts of all the models are semantically labeled. It contains about 15,000 3D models divided into 16 classes. Polygon mesh models in the database are converted into 3D point set representation having 2,048 points by randomly sampling the mesh surfaces to be sent to PointNet. Every point of the converted models are labeled by respective part label (e.g., "wing").

ShapeNetCore55 is one of the largest 3D model dataset labeled by whole shape. Models in the dataset are classified into 55 classes ("airplane", "sofa", "flowerpot", etc.). The dataset contains 35,764 model training set and 10,265 model test set. We use the training set to train WSN, and test set for evaluation. Models in this dataset are also converted to 3D point sets for processing.

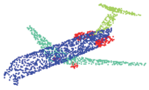
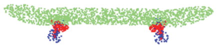
Labels	Segmentation
airplane 🟦 body 🟩 wing 🟩 tail 🟥 engine	
skateboard 🟦 wheel 🟩 skateboard 🟥 deck ⬛ other	 (axsles are labeled "other")

Fig. 4. Examples of segmented 3D point set model data generated from ShapeNetPart [7].

4.2. Whole shape label accuracy

We computed accuracy indices, in F-measure, for when whole shape label and partial shape labels are inferenced together given a whole shape 3D point set model. ShapeNetCore55 dataset is used for this experiment. Proposed method is compared against Goldfeder-PointNet, which is the method of Goldfeder [1], except that its shape feature is "modernized" to the whole shape feature obtained from the same PointNet as used in the proposed method.

Whole shape label accuracy among the two methods are nearly identical, as depicted in Table 2. This is understandable as the *k*-nearest neighbor is known to perform quite well. The difference is that, while the Goldfeder-PointNet can't handle partial shape label, our proposed method can, with the accuracy of 77.4%. We suspect that the accuracy of 77%, when queried by using partial shape label, is useful as is for some of the real world applications.

Table 2. Accuracy of whole shape labeling and partial shape labeling (f-measure [%]).

Methods	Whole shape label	Partial shape label
Goldfeder-PointNet	95.8	Not possible
Proposed	**96.0**	**77.4**

4.3. Impact of data augmentation

Data augmentation impacted labeling accuracy significantly, as the Table 3 shows for whole shape label. This experiment uses ShapeNetCore55 dataset and evaluated using retrieval accuracy queried by whole shape labels. Of the three augmentation methods, bending appears the most effective. Interestingly, when the three augmentation methods are combined, accuracy improved significantly by nearly 9%.

Table 2. 3D shape data augmentation and accuracy of whole shape retrieval.

3D shape augmentation methods	f-measure (%)
None	68.6
(1) Scaling	64.4
(2) Bending	75.0
(3) Region expansion	69.4
(1)+(2)+(3)	**77.4**

4.4. Examples of partial shape labeling and word-based retrieval

Figure 5 shows example of success as well as failure cases for labeling unlabeled 3D shapes. Upper two rows are success cases, while lower two rows are failure cases. It appears that most failures are due to mis-segmentation by PointNet. Some other failures are due to "incorrect ground truth" in [8] that are semi-automatically generated.

Table 3 shows examples of retrieval, in which query word airplane (whole shape label), vehicle (hypernym of whole shape label), wing (partial shape label), and airfoil (synonym of partial shape label) are used.

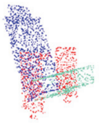
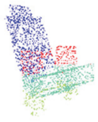
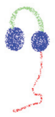
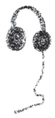
Ground truth label	Gournd truth segmentation	Segmentation	Added labels
car 🟦 roof 🟩 hood 🟩 wheel			car 🟦 roof 🟩 hood 🟩 wheel 🟥 other
guitar 🟦 head 🟥 body 🟩 nec			guitar 🟦 head 🟥 body 🟩 nec
chair 🟦 back 🟥 arm 🟩 seat			chair 🟦 back 🟩 leg 🟥 arm 🟩 seat
earphone 🟦 earphone 🟥 other 🟩 headband			earphone ⬛ mug ⬛ peak

Fig. 5. Examples of partial shape labeling. Success case (upper two), and failure case (lower two).

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to associate whole 3D models (e.g., automobile) as well as their parts (e.g., tire, body, engine) with word labels so that the 3D model can be queried by words. The words and shape are related in a common embedding space induced by both Word2Vec word features trained on Wikipedia corpus and whole as well as part shape features of segmented 3D model corpus. Experimental evaluation showed that the proposed algorithm achieved query by word retrieval accuracy, in f-measure, of 96% for whole shape and 77% for partial shape.

We would also like to quantitatively evaluate the accuracy of query by using semantically related words, e.g., synonyms or hypernyms, that are not used in the labels of training 3D models. This is currently difficult for there is no database of 3D models having ground truth labels of these semantically related words. We also would like to use large collection of 3D models for both training and testing.

REFERENCES

[1] C. Goldfeder and P. Allen. "Autotagging to Improve Text Search for 3D Models", *JCDL* 2008, 2008.

[2] R. Ohbuchi, and S. Kawamura. "Shape-Based Autotagging og 3D Models for Retrieval" *Proc. SAMT* 2009.

[3] T. Furuya, and R. Ohbuchi. "Deep Aggregation of Local 3D Geometric Features for 3D Model Retrival." *Proc. BMVC* 2016, 2016.

[4] T. Mikolov, et al. "Distributed Representations of Words and Phrases and their Compositionality", *Proc. NIPS* 2013, 2013.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3d classification and segmentation," *Proc. CVPR* 2017, pp. 77-85, 2017.

[6] A. X. Chang, et al. "ShapeNet: An Information-Rich 3D Model Repository", arXiv:1512.03012, 2015.

[7] L. Yi, et al. "A scalable active framework for region annotation in 3D shape collections." *SIGGRAPH Asia*, 2016, 2016.

[8] Piotr Bojanowski, et al. "Enriching Word Vectors with Subword Information." arXiv:1607.04606v1. 2016.

[9] Q. V. Le and T. Mikolov. "Distributed Representations of Sentences and Documents", *1st Workshop on Representation Learning for NLP*. 2015.

[10] E. Rodolà, et al., "Deformable Shape Retrieval with Missing Parts", *Proc. EG 3DOR* 2017. 2017.

[11] Z. Wu, et al. 3D ShapeNets: A deep representation for volumetric shape modeling. *Proc. CVPR 2015*, 2015.

[12] D. Maturana and S. Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. *Proc. IROS 2015*, pp.922–928, 2015.

[13] J. Li, B. M. Chen, G. H. Lee. "SO-Net: Self-Organizing Network for Point Cloud Analysis" *Proc. CVPR 2018*.

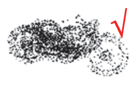
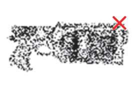
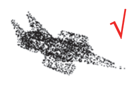
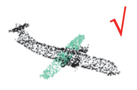
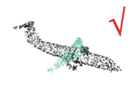
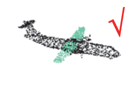
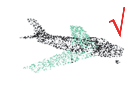
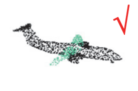
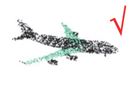
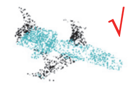
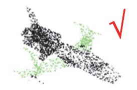
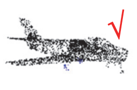
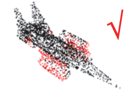
[14] N. Verma et al. "FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis" *Proc. CVPR 2018*.

[15] V. Nair and E. G. Hinton. "Rectified linear units improve restricted Boltzmann machines", *Proc ICML 2010*, 2010.

[16] R. Hadsell, S. Chopra, and Y. LeCun. "Dimensionality Reduction by Learning an Invariant Mapping", *Proc. CVPR 2006*, 2006.

[17] P. D. Kingma and B. J. Lei. "Adam: A Method for Stochastic Optimization", *Proc. ICLR 2015*, 2015.

Table 3. Examples of query by whole shape label, synonym of whole shape label, partial shape label, and synonym of partial shape label.

Query/rank	1	2	3	4	5	6	7	8
airplane (whole shape label)	√	√	√	√	√	√	√	√
	airplane	airplane	airplane	airplane	airplane	airplane	airplane	airplane
vehicle (synonym of whole shape label)	√	 √	 ×	√	 √	√	?	√
	airplane	motorbike	pistol	airplane	airplane	airplane	rocket	car
wing (partial shape label)	√	 √	 √	 √	 √	 √	 √	√
	airplane	airplane	airplane	airplane	airplane	airplane	airplane	airplane
airfoil (synonym of partial shape label)	 √	 √	√	 √	?	?	√	 √
	airplane	airplane	airplane	airplane	rocket	rocket	airplane	airplane