Eurographics Workshop on 3D Object Retrieval (2017)
I. Pratikakis, F. Dupont, and M. Ovsjanikov (Editors)

SHREC'17: RGB-D to CAD Retrieval with ObjectNN Dataset

Binh-Son Hua [1] Quang-Trung Truong [1] Minh-Khoi Tran [1] Quang-Hieu Pham [1]
Asako Kanezaki [4] Tang Lee [5] HungYueh Chiang [5] Winston Hsu [5]
Bo Li [6] Yijuan Lu [7] Henry Johan [8] Shoki Tashiro [9] Masaki Aono [9]
Minh-Triet Tran [10] Viet-Khoi Pham [10] Hai-Dang Nguyen [10,12] Vinh-Tiep Nguyen [10,11]
Quang-Thang Tran [10] Thuyen V. Phan [10] Bao Truong [10] Minh N. Do [13] Anh-Duc Duong [11]
Lap-Fai Yu [2] Duc Thanh Nguyen [3] Sai-Kit Yeung [1]

[1] Singapore University of Technology and Design [2] University of Massachusetts Boston [3] Deakin University
[4] National Institute of Advanced Industrial Science and Technology (AIST), Japan [5] National Taiwan University
[6] University of Southern Mississippi [7] Texas State University [8] Fraunhofer IDM@NTU, Singapore [9] Toyohashi University of Technology, Japan
[10] University of Science, VNU-HCM, Vietnam [11] University of Information Technology, VNU-HCM, Vietnam
[12] John von Neumann Institute, VNU-HCM, Vietnam [13] University of Illinois at Urbana-Champaign

Abstract

*The goal of this track is to study and evaluate the performance of 3D object retrieval algorithms using RGB-D data. This is inspired from the practical need to pair an object acquired from a consumer-grade depth camera to CAD models available in public datasets on the Internet. To support the study, we propose ObjectNN, a new dataset with well segmented and annotated RGB-D objects from SceneNN [HPN*16] and CAD models from ShapeNet [CFG*15]. The evaluation results show that the RGB-D to CAD retrieval problem, while being challenging to solve due to partial and noisy 3D reconstruction, can be addressed to a good extent using deep learning techniques, particularly, convolutional neural networks trained by multi-view and 3D geometry. The best method in this track scores 82% in accuracy.*

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Computer Vision]: Scene Analysis—Object Recognition

1. Introduction

Off-the-shelf consumer depth cameras have become widely adopted in computer graphics, computer vision, gaming and medical imaging applications. In the past few years, several methods and publicly available scene and object datasets have been proposed to significantly improve the performance of scene understanding algorithms, particularly, object retrieval in the 3D domain. SHREC is an annual challenge held at the 3D Object Retrieval (3DOR) workshop that focuses on studying and benchmarking the state-of-the-art algorithms in this area.

In this SHREC track paper, we aim to study and evaluate the performance of 3D object retrieval algorithms using RGB-D data from consumer depth cameras. We focus on the problem of pairing an RGB-D object captured in a real world environment to a virtual CAD model created by 3D artists. Such pairing, when available, could be used for applications such as semantic annotation, shape completion, and scene synthesis.

To support the study in this track, we propose *ObjectNN*, a new dataset for benchmarking RGB-D Object-to-CAD Model retrieval. It contains well segmented and annotated RGB-D objects acquired in real-world setting by SceneNN [HPN*16] and richly annotated CAD models collected from the internet by ShapeNet [CFG*15]. In total, ObjectNN contains 1667 RGB-D objects and 3308 CAD models in 20 categories. Some example RGB-D objects from our dataset are shown in Figure 1.

Figure 1: *Examples of RGB-D objects (first row) and CAD models (second row) in ObjectNN. The task of this track is to retrieve a plausible CAD model given an RGB-D object. Most RGB-D objects in 3D representation are partial and might contain some noise in the geometry due to imperfect reconstruction, making the recognition and retrieval task more challenging.*

Participants are asked to run their retrieval algorithms on the proposed dataset and submit the retrieved CAD models for evaluation. Based on the evaluation results, we provided an analysis on the performance of several object retrieval algorithms.

SHREC track	Query dataset	Target dataset	Categories	Attributes
NIST [GDB*15]	60 RGB-D objects	1200 CAD models	60	Geometry only
DUTH [PSA*16]	383 range-scan models	Similar to query dataset	6	Object from cultural heritage domain
IUL [PPG*16]	200 RGB-D objects	Similar to query dataset	N.A.	Objects scanned in lab setting
Ours	1667 RGB-D objects	3308 CAD models	20	Objects from real-world environments

Table 1: *Relevant datasets in previous SHREC tracks. To serve the purpose of the contest in this track, we built a new object dataset called ObjectNN upon SceneNN [HPN*16], a public dataset with more than 100 densely annotated scene meshes acquired from the real world, and ShapeNet [CFG*15], a large-scale dataset of CAD models.*

2. Dataset

The query dataset consists of 1667 objects extracted from more than 100 3D reconstructed real-world indoor scenes in the SceneNN dataset [HPN*16]. Each object has four data entries: (1) color triangle mesh; (2) a set of RGB-D frames extracted from the scene that contain the object; (3) a mask of the object in each frame; (4) and the camera pose for each frame. All objects are carefully segmented and annotated in both 3D and 2D by utilizing the user interactive tool by Nguyen et al. [TNHYY16], which can annotate RGB-D scenes densely at per-vertex and per-pixel level.

The target dataset has 3308 CAD models of indoor objects, which are extracted from ShapeNetSem, a richly annotated subset of ShapeNet [CFG*15]. We adopt and modify the category definitions in NYU Depth v2 dataset [SHKF12] to create 20 object categories for common indoor objects such as table, chair, bookshelf, monitor, lamp, pillow, etc.

Compared to the datasets of the previous SHREC tracks, ObjectNN is designed to follow the acquisition procedure in real world settings more closely. In a general indoor scene, objects are often organized or scattered in a scene depending on their usage and functionality. The acquisition of such objects is often performed by scanning the entire scene, which often results in cluttered and partial object reconstructions. This is fundamentally different from objects acquired with lab settings whose reconstructions are much more complete. Table 1 shows a comparison of our dataset to those of some previous works. As can be seen, our dataset has a larger scale compared to the existing datasets, and thus offers a good opportunity to benchmark supervised learning techniques for object retrieval.

Ground Truth. As there is not a standard metric to measure the similarity of two shapes, particularly between an RGB-D object and a CAD model, to establish the ground truth for evaluation, we classify all objects manually using an interactive tool that can simultaneously display an RGB-D object and a CAD model. Two human subjects are asked to pair an RGB-D object in the query set to some CAD models in the target set by assigning them to a predefined category. The classification is based on shape and color similarity of the objects in each pair. After classification, for each RGB-D object, all CAD models belonging to the same category are assumed to be its ground truth targets.

Beyond categories, we also classify objects more strictly into subcategories by considering more object attributes. In total there are 20 categories and 40 subcategories. In this very first contest, we only consider 20 categories in our evaluation.

Availability. To support the training process of methods based on supervised learning, we split ObjectNN into three subsets: training, validation and test, with ratio 50%, 25%, and 25% respectively. We then release the training and validation set to participants, with ground truth categories available for both RGB-D objects and CAD models. In this contest, all CAD models are placed in the training set. The test set only contains RGB-D models and their categories are not released to ensure a fair evaluation. Note that unsupervised learning is still applicable to solve the retrieval problem, and it would become much more challenging if the ground truth categorization of CAD models is not utilized. The dataset and the ground truth annotation tool are available for download at `http://www.scenenn.net`.

3. Overview

All methods used by the participants are summarized in Table 2. For brevity, we refer to each method by a unique name of an author from each group followed by the technique they used. In total we received nine registrations for participation and five result submissions excluding a baseline submission from the organizers themselves, corresponding to a submission rate of 55.56%. Each group can propose more than one approach to solve the problem, which corresponds to one or more runs to be evaluated.

In general, the proposed approaches belong to two main classes of techniques: deep learning using convolutional neural networks (Section 4–6) and feature matching using bag-of-words or shape descriptors (Section 7–9). Interested readers could proceed to Sec-

Team	Method	Training	Domain	Color
Kanezaki	RotationNet (Sec. 4)	Y	View-based	N
Tang	3DCNN (Sec. 5.1)	Y	Full 3D	N
Tang	MVCNN (Sec. 5.1)	Y	View-based	Y
Tang	CDTNN (Sec. 5.2)	Y	View-based	Y
Truong	2D (Sec. 6.1)	Y	View-based	Y
Truong	3D (Sec. 6.2)	Y	Full 3D	N
Tran	BoW-RoPS (Sec. 7)	N	Full 3D	N
Li	SBR-VC (Sec. 8)	N	View-based	N
Tashiro	GOLD (Sec. 9)	N	View-based	Y

Table 2: *An overview of the techniques used by the participants, most of which are based on supervised learning, especially those built atop view-based and 3D volume convolutional neural networks. Unsupervised learning that performs feature matching using handcrafted descriptors can also be used to solve the problem.*

tion 4–9 for more technical description of each method. We discuss evaluation results in Section 10.

4. RotationNet

Our method is based on a convolutional neural network (CNN) named *RotationNet* [KMN16]. An overview of the training and inference process using RotationNet is illustrated in Fig. 2. Similarly to multi-view CNN [SMKLM15], it takes multi-view images of an object as input and estimates its object category. Our method treats the pose labels as latent variables, which are optimized to self align in an unsupervised manner during the training using an unaligned dataset. The code is available at https://github.com/kanezaki/rotationnet.

4.1. Training

We assume that multi-view images of each training object instance are observed from all the pre-defined viewpoints. Let M be the number of the pre-defined viewpoints and N denote the number of target object categories. A training sample consists of M images of an object $\{\mathbf{x}_i\}_{i=1}^{M}$ and its category label $y \in \{1,\ldots,N\}$. We attach a view label variable $v_i \in \{1,\ldots,M\}$ to each image $\mathbf{x}_i$ and set it to j when the image is observed from the j-th viewpoint, *i.e.*, $v_i \leftarrow j$. In our method, only the category label y is given during the training whereas the view labels $\{v_i\}$ are unknown, namely, $\{v_i\}$ are treated as latent variables that are optimized in the training process. We introduce an "incorrect view" label and append it to the target category labels. Letting $P_i = \{p_{j,k}^{(i)}\} \in \mathbb{R}_{+}^{M\times(N+1)}$ denote a matrix composed of $P(\hat{y}_i \mid \mathbf{x}_i, v_i)$ for all the M viewpoints and $N+1$ classes, the target value of P_i in the case that v_i is correctly estimated is defined as follows:

$$p_{j,k}^{(i)} = \begin{cases} 1 & (j = v_i \text{ and } k = y), \text{ or } (j \neq v_i \text{ and } k = N+1) \\ 0 & (\text{otherwise}). \end{cases} \quad (1)$$

The parameters of RotationNet are iteratively updated via standard back-propagation of M softmax losses, whereas $\{v_i\}_{i=1}^{M}$ are determined in a manner to maximize the probability that a training sample belongs to its correct category y in each iteration.

We place virtual cameras on the $M = 20$ vertices of a dodecahedron encompassing the object. There are three different patterns of rotation from a certain view, because three edges are connected to each vertex of a dodecahedron. Therefore, the number of candidates for all the view labels $\{v_i\}_{i=1}^{M}$ is 60 ($= 3M$).

4.2. Retrieval

First, we train our RotationNet model using 3308 CAD models in ShapeNetSem. Then we fine-tune the model using 830 objects in SceneNN. We did not use colors nor textures. We train two different RotationNet models, one to predict category labels and the other to predict subcategory labels. For each query, we construct a retrieval set using five different approaches described below.

Single. We predict a single category label for each query and construct a retrieval set with all the target models in the same category.

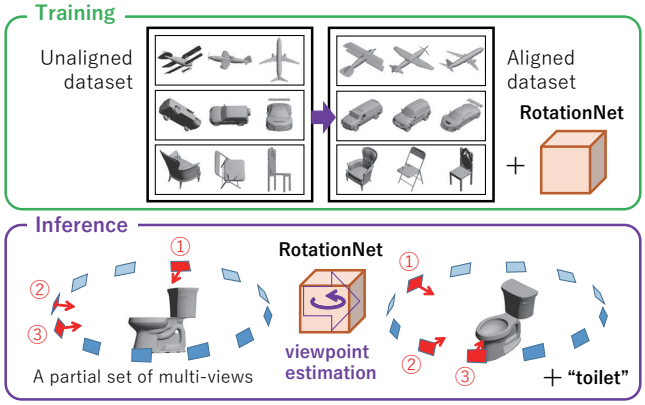


Figure 2: *Overview of training and inference using RotationNet. We jointly learn the parameters of RotationNet and estimate the pose of object instances in the unaligned 3D object dataset. In the inference phase, RotationNet takes a partial set of all the multi-view images of an object as input and outputs its object category by rotation, where the best pose is selected to maximize the object category likelihood.*

Thresh. We rank categories by the classification scores for each query. If the score of a category is higher than a certain ratio of the maximum score, we add the target models in the category to the retrieval set. We set the threshold to 0.1. In the case that the scores of all the categories (except for the predicted category) are lower than 10% of the score of the predicted category, we obtain exactly the same retrieval set as "single" case.

1000. In order of decreasing classification scores, we add the target models in the same category to the retrieval set for each query. We stop adding a target model when the number of the models in the retrieval set reaches to 1000.

In addition, we also perform the "Single" and "Thresh" retrieval but classifying the objects to subcategories for a more rigorous evaluation.

5. Multi-view and 3D Convolutional Neural Network

We propose two classes of methods to solve the problem: (1) Convolutional neural network (CNN) based techniques that include 3D CNN, multi-view 2D CNN (MVCNN), and 2D-3D fused classification, and (2) Cross-domain triplet neural network. Before applying such techniques, we preprocess SceneNN (SNN) objects and ShapeNetSem (SNS) objects as follows.

Voxelization. Voxelization is a prevailing 3D data representation method. It normalizes object size to $1 \times 1 \times 1$ space with a scaling factor and captures occupancy of the scaled object in space with $64 \times 64 \times 64$ resolution binary voxel grid, i.e., structural information and real size measurement of an object is kept. We multiply the value stored in every binary voxel with the scaling factor and augment every object by rotating it 0, 90, 180, 270 degrees horizontally, as the object may not be well aligned.

View Rendering. We render each object in both SNN and SNS from 12 views that are distributed around the object every 30 de-

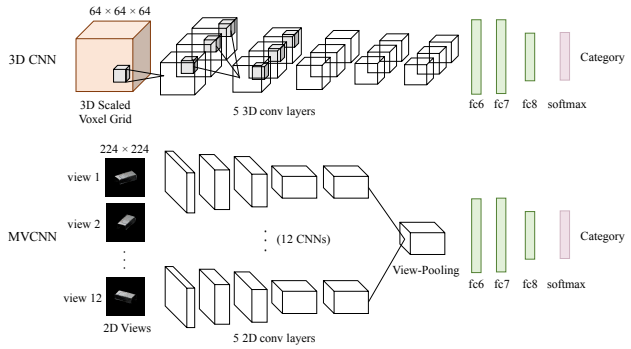


Figure 3: *The network structures of the 3DCNN and MVCNN for 3D object classification. The 3DCNN is modified based on VoxNet [MS15]. The MVCNN [SMKLM15] uses AlexNet as the base model.*

grees and elevated 30 degrees from the ground. SNN objects are rendered with color textures while SNS objects are not. All objects are rendered with Phong shading on a black background.

Since some scenes and objects in SNN are upside down, for data augmentation purpose, we additionally flip the SNN training objects vertically to render views.

5.1. CNN-based Classification

Since a retrieved object in SNS is considered correct if it is from the same category as the query SNN object, and since the categories of SNS objects are known, we can reduce the retrieval problem into a classification problem of SNN objects. We sort the predicted probabilities of the 20 categories in descending order. According to the sorted list, we first push all SNN objects from the first category into the results, then the second, and so on.

3D CNN. Since the problem is reduced to classification, we build a 3D CNN and train our network on scaled SNN voxel representations to learn the structural similarity and predict the class of an SNN object. Mirroring the success of AlexNet, we borrow the idea to model our 3D network with 8 layers including 5 convolutional layers and 3 fully-connected layers. Each convolutional layer has $3 \times 3 \times 3$ kernel and $1 \times 1 \times 1$ stride. The first, the second and the fifth convolutional layers are followed with a max pooling layer. Three fully-connected layers with output size 4096, 4096 and 20 respectively are attached after 5 convolutional layers. An illustration of our network architecture is shown in Figure 3.

MVCNN. We utilize MVCNN [SMKLM15] to learn from the views and predict the SNN objects categories. We adopt AlexNet as the base model of MVCNN, and put the View-Pooling layer (max-pooling) after pool5 layer. Each of the 12 CNNs share parameters. The parameters are pretrained with ImageNet.

2D-3D Fused Classification. 3DCNN learns objects structural and scaling information in real world, while MVCNN learns surface texture and contour of objects. We fuse the 2D and 3D methods by averaging the predicted category probabilities.

5.2. Cross-Domain Triplet Neural Network

We build a cross-domain triplet neural network (CDTNN) to learn the similarity between SNN objects and SNS objects. A triplet network [SKP15] has 3 streams of CNNs: anchor, positive, and negative. The anchor is for SNN views, and the positive and negative are for SNS views. Each stream is an MVCNN without fully connected layers, i.e. it has 12 streams of conv1 to pool5 layers of AlexNet with one view-pooling layer, generating a feature map with size 9216. The 3 streams are sharing parameters.

Since the views from SNN and SNS have different look, i.e. SNN are real object while SNS are human made and with no texture in our settings, we propose CDTNN by adding an "adaptation layer" after the anchor stream. The adaptation layer is simply a fully-connected layer with both input and output size 9216. The features generated by each stream are used as the representations, and their distances are squared Euclidean distances.

Our methods are implemented using TensorFlow. Our experiments are performed on a workstation with two NVIDIA K80 GPUs.

6. 2D-3D Retrieval using VGG and VoxNet

In this approach, the retrieval problem is solved in two steps: (1) determine the object category and (2) compare the features of the object with those of the CAD models in the same category, and determine the final similarity ranking. In Step 1, we leverage state-of-the-art object recognition techniques in both the 2D and 3D domain, i.e., convolutional neural networks. In Step 2, the features predicted by the neural networks are extracted for comparison. An overview is shown in Figure 4.

6.1. 2D Retrieval using VGG

In image classification, traditional techniques are based on hand-crafted features which are recently shown to be suboptimal compared to features learned by deep neural networks. Inspired by the impressive performance of recent pretrained networks, we employ the popular VGG model [SZ14] to solve this problem.

We first fine tune this model by using the 2D images of the RGB-D objects. As an object could appear in multiple RGB-D frames, for each object we pick and assign to it a representative image. In particular, we set a frame as representative when its object mask has

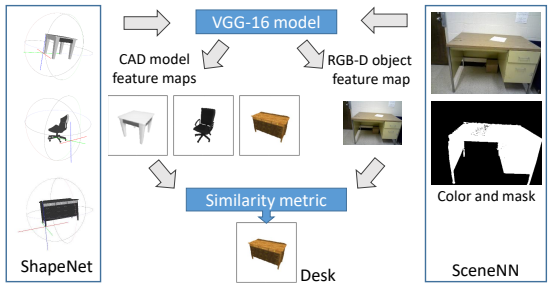


Figure 4: *An overview of object retrieval using image data with VGG. The pipeline for retrieval using 3D data is almost the same but with VGG replaced by VoxNet.*

the most number of valid pixels. We also use this mask to filter the foreground pixels in the frame, which can then be used as input to train the network. After fine tuning, we obtain an accuracy of 77.56% on the validation set, which also agrees with the 74% accuracy on the test set reported in Section 10.

Generally speaking, as we cast the current retrieval problem into object classification, obtaining the category is enough for evaluation. For each query object, we simply return all CAD models in the predicted category.

For completeness, we further compute a distance score to rank the similarity of a retrieved object to the query as follows. First, we apply a forward pass to the rendered image of all CAD models, and extract the output of the third last layer of the VGG network to obtain a feature vector with 4096 elements. During retrieval, the feature vector of the query object is extracted and compared to the feature vectors of all CAD models. We use L2-norm to compare the vectors, and then sort the distance in increasing order to obtain the ranking for the retrieved objects.

6.2. 3D Retrieval using VoxNet

For 3D, we use VoxNet [MS15], a compact convolutional neural network for object recognition that has about 1.2 million parameters. This network only contains two convolutional layers followed by two fully connected layers, and does not use dropout and pooling. We train this network from scratch using the provided object dataset.

We use a binary volume representation as the 3D CNN described in Section 5. All objects are scaled and centered at the unit cube. We achieved 50% validation accuracy by training with RGB-D objects in the training set alone. We also achieved approximately the same level of accuracy when the network is furthered trained with CAD models. To support rotation invariance, we augment the data with 128 rotation directions and retrain the work with augmented data. This results in accuracy improvement to 52.93%.

It is interesting to see that the results by 2D recognition outperforms that by 3D recognition. This could be explained by the fact that our 3D volumes do not consider object scale and color information. Also, the 2D convolutional network might be more well trained thanks to more data availability.

7. Bag-of-Words Framework with RoPS

The requirement of this challenge is to product a ranklist of target objects in ShapeNet corresponding to a test object in SceneNN. However, because query objects can be partially constructed with noise and holes, while target objects are ideally modelled, we decide not to directly evaluate the similarity between a test object with a target object. Our approach is to determine the possible categories of a query object, then retrieve all target objects (in ShapeNet) in those categories to produce the result rank list.

Our objective is to verify whether our method purely based on 3D object retrieval can be used to classify a query object with high accuracy. Thus, we do not train machine-learning-based classifiers to recognize the category of a query object. Instead, we determine the category of an object based on the the corresponding retrieved ranklist in SceneNN (c.f. Fig 5).

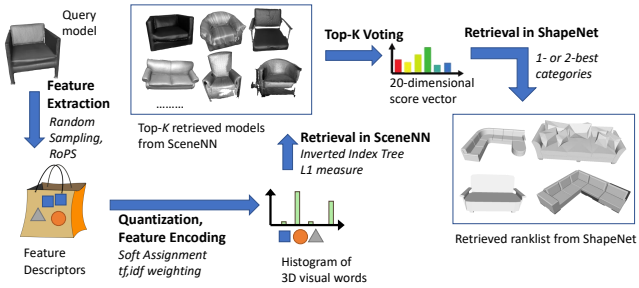


Figure 5: *Bag-of-Words framework for 3D object retrieval.*

Our method for 3D object retrieval is based on Bag-of-Words (BoW) scheme for visual object retrieval [SZ03, NNT*15]. We use our BoW framework for 3D object retrieval [PSA*16] with 3D feature RoPS [GSB*13] to retrieve $R_{SceneNN}(q)$, a rank list of top K similar labeled objects in SceneNN corresponding to a query object q. Then we use simple voting scheme to determine the top 1 and top 2 categories of q from $R_{SceneNN}(q)$. Finally, we create the result ranklist $R_{ShapeNet}(q)$ by adding target objects in ShapeNet in those categories. All selected target objects in the same category have the same similarity with a query object q.

First, all labeled and unlabeled 3D objects in SceneNN are normalized to fit in a unit cube. As shapes have irregular mesh, we subdivide the mesh to reduce significant difference in the density of vertices between different parts in a shape. We uniformly take random samples of $5\% \leq p_{Sampling} \leq 20\%$ in each object and use RoPS [GSB*13] as the feature to describe the characteristics of the mesh clipped in a sphere with supporting radius $r = 0.1$.

We train a codebook with the size relatively equal to 10% of the total number of features in the corpus, using Approximate K-Means. In our experiments, our codebook has the size of 100,000 and 150,000. We use soft-assignment [PCI*08] with 3 nearest neighbors to reduce quantization error and L1, asymmetric distance measurement [ZJS13], to evaluate the dissimilarity of each pair of objects. We experiment with four runs with the following configurations:
Run 1 The category of an object is based on Top $K = 1$ in the rank list with $p_{Sampling} = 10\%$, codebook size = 150,000.
Run 2 The two best categories of an object is based on Top $K = 4$ in the rank list with $p_{Sampling} = 20\%$, codebook size = 100,000.
Run 3 The category of an object is based on Top $K = 1$ in the rank list with $p_{Sampling} = 20\%$, codebook size = 150,000.
Run 4 The two best categories of an object is based on Top $K = 7$ in the rank list with $p_{Sampling} = 20\%$, codebook size = 100,000.

In our implementation, codebook training is performed on a 2.4 GHz Intel Xeon CPU E5-2620 v3 with 64 GB RAM. The 3D feature extraction and description are executed on a 2 GHz Intel Xeon CPU E5-2620 with 1 GB RAM. The retrieval process with feature quantization and similarity matrix output is performed on another 2.2 GHz Intel Xeon CPU E5-2660 with 12 GB RAM. The average time to calculate features of a model is 1–2 seconds and it takes on average 0.1 second to compare a test object against the training set.

Experimental results show that our method, without using 2D/3D deep learning features and color information, can retrieve with high accuracy 3D objects based on their partial 3D appearance. This is a

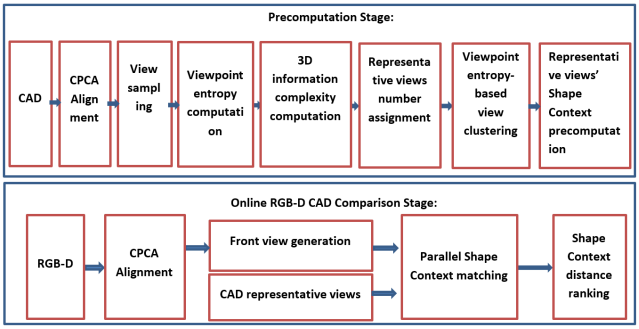


Figure 6: *Overview of the SBR-VC algorithm: the first row is for the precomputation whereas the second row is for the retrieval stage.*

promising step for our future enhanced system in which we integrate 2D views and deep learning features to boost the performance of our system.

8. View Clustering and Parallel Shape Context Matching

We adapt the sketch-based 3D model retrieval algorithm based on view clustering (SBR-VC) [LLJF16, LLJ13] to solve the problem in this track. Figure 6 illustrates an overview of the algorithm. In this approach, an RGB-D object is first aligned by Continuous Principle Component Analysis (CPCA) [Vra04]. Then, the front view is regarded as a representative view of the RGB-D object and also used to extract an outline image, similar as a sketch in the original algorithm SBR-VC, for RGB-D object and CAD model pair comparison. The remaining steps are similar to the original SBR-VC steps in [LLJF16, LLJ13].

Particularly, the SBR-VC algorithm retrieves CAD models given RGB-D queries in four steps, as below.

Step 1 Viewpoint entropy-based adaptive view clustering. It is to cluster a set of sample views of each CPCA-aligned target CAD model into an appropriate number of representative views according to its visual complexity, which is defined as the viewpoint entropy distribution of its sample views.

Step 2 Feature view generation. Outline feature views for the RGB-D queries and the 3D CAD models are generated. The RGB-D case has been explained above. In the 3D CAD model case, silhouette views are first rendered followed by outline feature extraction.

Step 3 Relative shape context computation. Rotation-invariant relative shape context features [BMP02] are extracted to represent outline feature views for both the query and target 3D objects.

Step 4 Online retrieval. The relative shape context matching is performed in parallel between the RGB-D outline feature view and several representative views of a target CAD model and the minimum matching cost is chosen as the RGB-D-to-CAD distance. The models are ranked accordingly based on the distances.

9. View-based Shape Feature

We tackle this problem by using view-based shape features. The overall steps are shown in Figure 7. First, we render each target object (CAD models) from multiple views to obtain face orientation

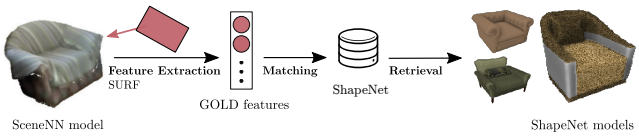


Figure 7: *Overview of the view-based shape feature method.*

images [PPTP10]. Then we extract SURF [BETVG08] local features from each image and encode them into a feature vector with Gaussian of Local Descriptors (GOLD) [SGMC15], which consists of a mean and a covariance component.

On the other hand, we render each query object (RGB-D object) from a single view under the assumption that each query object can be captured in a representative view. We calculate this view $\mathbf{v}$ with $\mathbf{v} = \left(\sum_{p\in Q} a_p \mathbf{n}_p\right) / \left\|\sum_{p\in Q} a_p \mathbf{n}_p\right\|$ where p is a triangle in the query object Q, a_p is area of the triangle, and $\mathbf{n}_p$ is the triangle surface normal. Similar to the process for CAD models, we generate a face orientation image from each query. After obtaining the face orientation image, we extract SURF local features and encode them with GOLD. For retrieval, GOLD features extracted from target objects and query objects are then power and L2 normalized. We compute dissimilarity d between query object Q and target object M as $d = \min_i \|\mathbf{f}_Q - \mathbf{f}_i\|$ where $\mathbf{f}_Q$ is GOLD feature vector of Q and $\mathbf{f}_i$ is the i-th feature vector of M.

In our experiment, we use image resolution 300×300, 66 view points based on the sub-divided octahedron, and 128 dimension SURF feature. This results in GOLD vector of 8384 dimension.

10. Evaluation

For each query model, each participant submits a ranked list where retrieved models are sorted by similarity with the query model. Each ranked list is evaluated based on the ground truth category and subcategories. We use the following measures: Precision-Recall curve, Mean Average Precision (mAP), and Normalized Discounted Cumulative Gain (NDCG). Please refer to our homepage for additional evaluation results with F-Score, Nearest Neighbor First-Tier (Tier1) and Second-Tier (Tier2).

To compute the scores, we only consider the first K retrieved objects, where K is the number of objects in the ground truth class. We found that this evaluation strategy gives consistent scores for both techniques in supervised and unsupervised categories. A precision-recall plot for all methods is shown in Figure 8, and the evaluation results with all metrics are shown in Table 3. Note that for supervised methods, a query can simply return all objects in the predicted category, resulting in extreme scores (either 0 or 1) in all metrics. After averaging, the scores across methods become mostly similar. This does not occur in the case of unsupervised learning.

In general, it can be seen that deep learning methods (Kanezaki, Tang, and Truong) outperform techniques that apply feature matching directly between an RGB-D object and a CAD model (Tashiro and Li) by a significant margin. This performance gap could be explained by the fact that the ground truth category of the CAD models are provided. This is a strong hint that supervised methods can utilize to turn the object retrieval problem into object classification, which can be solved using the state-of-the-art deep learning

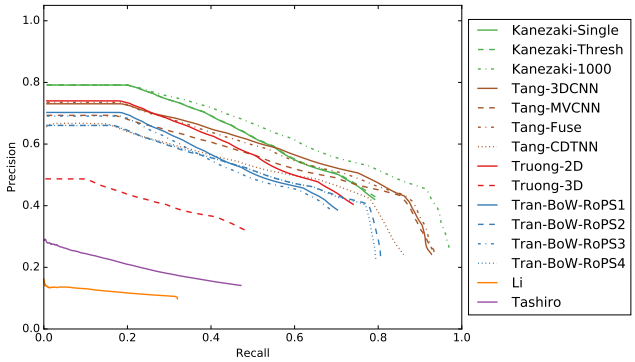


Figure 8: *Precision-recall plots of all methods submitted by participants. While not yet perfect, most methods based on convolutional neural networks perform effectively, either using multi-view or 3D shape data. Feature matching techniques have moderately low accuracy, partly due to the high variance in shape geometry and appearance in each category of our dataset. (Best view on screen with color)*

techniques, i.e., convolutional neural networks. The bag-of-words approach by Tran, while not based on deep learning, also exploits the ground truth categorization hint to achieve good accuracy. In contrast, unsupervised learning techniques not making use of the ground truth categories and simply relying on feature matching between RGB-D objects and CAD models do not work very well, because even the RGB-D objects and CAD models in a same category can vary greatly in both shape and appearance.

Among supervised methods, convolutional neural networks trained by object images are competitive, if not better, than those trained by 3D volumes. For example, Kanezaki-single, -thresh, and -1000 variant top our evaluation chart, and outperform several other 3D convolutional networks like Tang-3DCNN and Truong-3D. Despite that, it is worth noting that Tang-3DCNN is very competitive, and so as two other view-based methods like Tang-MVCNN and Truong-2D. Tang-3DCNN outperforms Truong-3D by a large gap even though both use a binary volume representation and convolutional neural network for classification. A possible reason is that Truong-3D used a more compact network and a lower resolution volume than Tang-3DCNN's, which results in lower discriminative power.

From the unsupervised learning evaluation results, it can be seen that the method by Tashiro is slightly more effective than Li's. However, there are still potential for both methods to improve. For example, a better categorization where objects in the same category has less shape and appearance variation could lead to better results.

Figure 9 shows the precision score plots for each category of all methods. It can be seen that the maximum precisions in most categories are over 75%. Some categories, such as chair, display, keyboard, cup, or bed, have precisions very close to 100%. There also exist some ambiguous categories with mean and median precisions below 50%, for example, box, machine, and printer, whose maximum precisions are just slightly above 50%. This suggests that more discriminative training data might be required to reduce ambiguity.

Team	Run	Precision	Recall	mAP	NDCG
Kanezaki	Single	0.792	0.792	0.792	0.792
Kanezaki	Thresh	0.793	0.799	0.794	0.796
Kanezaki	1000	**0.820**	**0.820**	**0.833**	**0.805**
Tang	3DCNN	0.769	0.769	0.749	0.774
Tang	MVCNN	0.727	0.727	0.710	0.735
Tang	Fuse	0.759	0.759	0.746	0.763
Tang	CDTNN	0.672	0.672	0.649	0.714
Truong	2D	0.740	0.740	0.740	0.740
Truong	3D	0.487	0.487	0.487	0.487
Tran	BoW-RoPS1	0.703	0.703	0.703	0.703
Tran	BoW-RoPS2	0.690	0.690	0.676	0.695
Tran	BoW-RoPS3	0.691	0.691	0.691	0.691
Tran	BoW-RoPS4	0.689	0.689	0.675	0.692
Li	SBR-VC	0.105	0.320	0.062	0.476
Tashiro	GOLD	0.141	0.472	0.149	0.552

Table 3: *Evaluation results on the test set. Kanezaki's methods top the scoreboard. Other methods based on convolutional neural networks also perform well.*

11. Conclusions

In this SHREC track, we proposed a new dataset to evaluate 3D object retrieval algorithms. The query is an RGB-D object and the target is a CAD model. By evaluating various techniques submitted by the participants, we found that for this problem, convolutional neural networks with multi-view representation are currently among the most effective, followed by those with 3D volume representation. Techniques based on unsupervised learning appear to be less competitive in our benchmark.

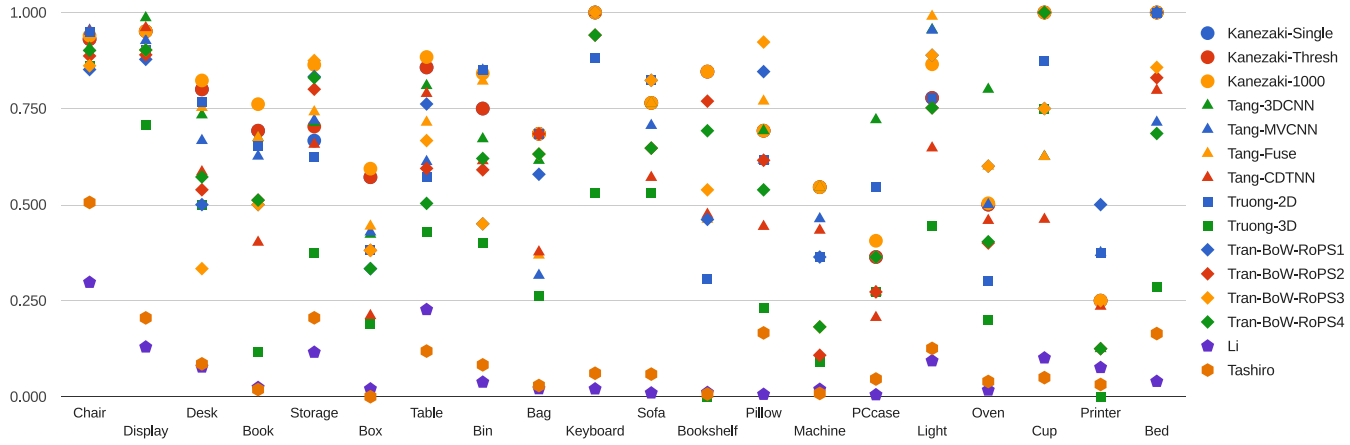
There are a few promising future research directions. First, the dataset can be extended to include more objects and categories, e.g., those from ScanNet [DCS*17] which is recently available on arXiv. In addition, stricter categorization in the ground truth would be helpful to benchmark the retrieval more rigorously. Second, object fragments could be added to pose an even more challenging track for object detection: each fragment can contain multiple objects and background structure such as floor and walls. Third, the current ground truth categorization for CAD models can be removed to challenge approaches based on supervised learning.

Acknowledgement

We acknowledge the support of the SUTD Digital Manufacturing and Design (DManD) Centre which is supported by the National Research Foundation (NRF) of Singapore. Bo Li is partially supported by the University of Southern Mississippi Faculty Startup Funds Award. The research done by Henry Johan is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative. The research conducted by Lap-Fai Yu is supported by the National Science Foundation under award number 1565978.

References

[BETVG08] BAY H., ESS A., TUYTELAARS T., VAN GOOL L.: Speeded-up robust features (surf). *Computer Vision and Image Understanding 110*, 3 (2008). 6



	Chair	Display	Desk	Book	Storage	Box	Table	Bin	Bag	Keyboard	Sofa	Bookshelf	Pillow	Machine	PCcase	Light	Oven	Cup	Printer	Bed
Mean	0.845	0.813	0.569	0.490	0.648	0.351	0.626	0.585	0.483	0.825	0.631	0.480	0.569	0.329	0.323	0.713	0.428	0.707	0.215	0.758
Median	0.907	0.902	0.585	0.512	0.714	0.381	0.667	0.620	0.615	1.000	0.765	0.468	0.615	0.364	0.364	0.778	0.500	0.750	0.250	0.857
Stdev	0.187	0.270	0.242	0.247	0.234	0.181	0.229	0.260	0.238	0.341	0.258	0.298	0.256	0.209	0.174	0.280	0.214	0.310	0.137	0.330

Figure 9: *Precision plots of all methods for each category in the dataset. Mean, median, and standard deviation of the precision score per category are provided at the bottom rows. As can be seen, most categories can be well recognized with maximum accuracy over 75%. The mean and median statistics also suggest that more training data might be necessary to improve the accuracy for a few challenging categories such as box, machine and printer. (Best view on screen with color)*

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (2002). 6

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., XIAO J., YI L., YU F.: ShapeNet: An Information-Rich 3D Model Repository. *arXiv:1512.03012* (2015). 1, 2

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T., NIESSNER M.: ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. *arXiv:1702.04405* (2017). 7

[GDB*15] GODIL A., DUTAGACI H., BUSTOS B., CHOI S., DONG S., FURUYA T., LI H., LINK N., MORIYAMA A., MERUANE R., OHBUCHI R., PAULUS D., SCHRECK T., SEIB V., SIPIRAN I., YIN H., ZHANG C.: Range scans based 3d shape retrieval. In *Eurographics Workshop on 3D Object Retrieval (3DOR)* (2015). 2

[GSB*13] GUO Y., SOHEL F. A., BENNAMOUN M., LU M., WAN J.: Rotational projection statistics for 3D local surface description and object recognition. *International Journal of Computer Vision 105*, 1 (2013). 5

[HPN*16] HUA B.-S., PHAM Q.-H., NGUYEN D. T., TRAN M.-K., YU L.-F., YEUNG S.-K.: Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)* (2016). 1, 2

[KMN16] KANEZAKI A., MATSUSHITA Y., NISHIDA Y.: Rotationnet: Joint learning of object classification and viewpoint estimation using unaligned 3d object dataset. *arXiv:1603.06208* (2016). 3

[LLJ13] LI B., LU Y., JOHAN H.: Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering. In *Eurographics Workshop on 3D Object Retrieval (3DOR)* (2013). 6

[LLJF16] LI B., LU Y., JOHAN H., FARES R.: Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information. *Multimedia Tools and Applications* (2016). 6

[MS15] MATURANA D., SCHERER S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015). 4, 5

[NNT*15] NGUYEN V., NGO T. D., TRAN M., LE D., DUONG D. A.: A combination of spatial pyramid and inverted index for large-scale image retrieval. *International Journal of Multimedia Data Engineering and Management 6*, 2 (2015). 5

[PCI*08] PHILBIN J., CHUM O., ISARD M., SIVIC J., ZISSERMAN A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR* (2008). 5

[PPG*16] PASCOAL P. B., PROENÃĞA P., GASPAR F., DIAS M. S., FERREIRA A., TATSUMA A., AONO M., LOGOGLU K. B., KALKAN S., TEMIZEL A., LI B., JOHAN H., LU Y., SEIB V., LINK N., PAULUS D.: Shape Retrieval of Low-Cost RGB-D Captures. In *Eurographics Workshop on 3D Object Retrieval (3DOR)* (2016). 2

[PPTP10] PAPADAKIS P., PRATIKAKIS I., THEOHARIS T., PERANTONIS S.: Panorama: A 3d shape descriptor based on panoramic views for unsupervised 3D object retrieval. *International Journal of Computer Vision 89*, 2 (2010). 6

[PSA*16] PRATIKAKIS I., SAVELONAS M., ARNAOUTOGLOU F., IOANNAKIS G., KOUTSOUDIS A., THEOHARIS T., TRAN M.-T., NGUYEN V.-T., PHAM V.-K., NGUYEN H.-D., LE H.-A., TRAN B.-H., TO Q., TRUONG M.-B., PHAN T., NGUYEN M.-D., THAN T.-A., MAC K.-N., DO M., DUONG A.-D., FURUYA T., OHBUCHI R., AONO M., TASHIRO S., PICKUP D., SUN X., ROSIN P., MARTIN R.: Partial Shape Queries for 3D Object Retrieval. In *Eurographics Workshop on 3D Object Retrieval (3DOR)* (2016). 2, 5

[SGMC15] SERRA G., GRANA C., MANFREDI M., CUCCHIARA R.: Gold: Gaussians of local descriptors for image representation. *Computer Vision and Image Understanding 134* (2015). 6

[SHKF12] SILBERMAN N., HOIEM D., KOHLI P., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *ECCV* (2012). 2

[SKP15] SCHROFF F., KALENICHENKO D., PHILBIN J.: Facenet: A unified embedding for face recognition and clustering. In *CVPR* (2015). 4

[SMKLM15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E. G.: Multi-view convolutional neural networks for 3D shape recognition. In *ICCV* (2015). 3, 4

[SZ03] SIVIC J., ZISSERMAN A.: Video google: A text retrieval approach to object matching in videos. In *ICCV* (2003). 5

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556* (2014). 4

[TNHYY16] THANH NGUYEN D., HUA B.-S., YU L.-F., YEUNG S.-K.: A robust 3d-2d interactive tool for scene segmentation and annotation. *arXiv:1610.05883* (2016). 2

[Vra04] VRANIC D.: *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004. 6

[ZJS13] ZHU C., JEGOU H., SATOH S.: Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV* (2013). 5