Eurographics Workshop on 3D Object Retrieval (2018), pp. 1–8
A. Telea and T. Theoharis (Editors)

SHREC'18: RGB-D Object-to-CAD Retrieval

Quang-Hieu Pham [1] Minh-Khoi Tran [1]
Wenhui Li [5] Shu Xiang [5] Heyu Zhou [5] Weizhi Nie [5] Anan Liu [5] Yuting Su [5]
Minh-Triet Tran [6] Ngoc-Minh Bui [6] Trong-Le Do [6] Tu V. Ninh [6] Tu-Khiem Le [6]
Anh-Vu Dao [6] Vinh-Tiep Nguyen [7] Minh N. Do [8] Anh-Duc Duong [7]
Binh-Son Hua [2] Lap-Fai Yu [3] Duc Thanh Nguyen [4] Sai-Kit Yeung [1]

[1] Singapore University of Technology and Design [2] The University of Tokyo
[3] University of Massachusetts Boston [4] Deakin University [5] Tianjin University
[6] University of Science, VNU-HCM [7] University of Information Technology, VNU-HCM [8] University of Illinois at Urbana-Champaign

Abstract

*Recent advances in consumer-grade depth sensors have enable the collection of massive real-world 3D objects. Together with the rise of deep learning, it brings great potential for large-scale 3D object retrieval. In this challenge, we aim to study and evaluate the performance of 3D object retrieval algorithms with RGB-D data. To support the study, we expanded the previous ObjectNN dataset [HTT*17] to include RGB-D objects from both SceneNN [HPN*16] and ScanNet [DCS*17], with the CAD models from ShapeNetSem [CFG*15]. Evaluation results show that while the RGB-D to CAD retrieval problem is indeed challenging due to incomplete RGB-D reconstructions, it can be addressed to a certain extent using deep learning techniques trained on multi-view 2D images or 3D point clouds. The best method in this track has a 82% retrieval accuracy.*

1. Introduction

With the widespread adoption of consumer-grade depth cameras in computer graphics, computer vision, and medical imaging applications, in the past few years, we have seen the emergence of several methods and datasets that aim to improve the performance of 3D scene understanding algorithms, particularly 3D object retrieval. One of such initiatives is SHREC, an annual challenge held at the 3D Object Retrieval (3DOR), where the main focuses are on studying and benchmarking state-of-the-art algorithms in this area.

In this SHREC track paper, our aims are to study and evaluate the performance of 3D object retrieval algorithms, especially on the scanned RGB-D data from consumer-grade depth cameras. Our main focus is on the problem of pairing an RGB-D object captured in a real world environment to a virtual CAD model manually designed by 3D artists. Some of the applications of such technique are semantic annotation, shape completion, and scene synthesis.

To support the study in this track, we expanded the object dataset from the same track [HTT*17], which was introduced in the previous year for benchmarking the RGB-D Object-to-CAD retrieval problem. This year we enrich the dataset by including additional RGB-D objects acquired in real-world setting from ScanNet [DCS*17]. Furthermore, we also refine the original RGB-D dataset from SceneNN [HPN*16] and CAD models from ShapeNetSem [CFG*15] with additional sub-categories information. In total, our new dataset contains 2101 RGB-D objects and 3308 CAD models in 20 categories. Figure 1 shows some example objects from our dataset.

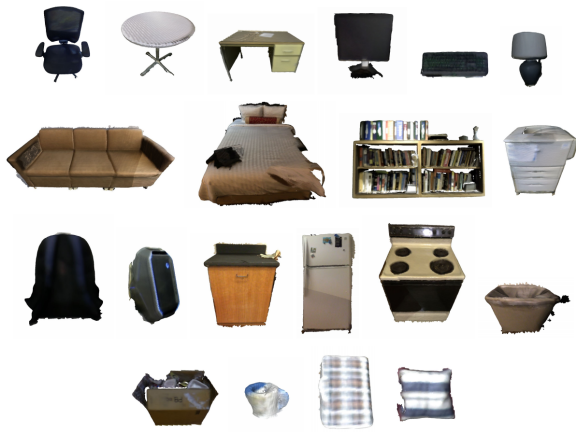

Figure 1: *Examples of RGB-D objects in the dataset. Only RGB-D objects are selected to provide rich information for learning and avoid ambiguity. However, there might still have significant difference between real objects (RGB-D) and synthetic objects (CAD models), posing great challenges to object recognition and retrieval.*

The main objective is to retrieve plausible CAD models given an RGB-D object. Participants are asked to run their retrieval al-

SHREC track	Query dataset	Target dataset	No. 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	Cultural heritage domain
IUL [PPG*16]	200 RGB-D objects	Similar to query dataset	N.A.	Lab setting
ObjectNN [HTT*17]	1667 partial RGBD objects	3308 CAD models	20	Real-world setting
Ours	2101 high-quality RGB-D objects	3308 CAD models	20	Real-world setting

Table 1: *A small comparison with relevant datasets in previous SHREC tracks. For this year, we extended and refined the ObjectNN dataset with high-quality RGB-D and CAD models from SceneNN [HPN*16], ScanNet [DCS*17], and ShapeNetSem [CFG*15]. As a result, our RGB-D objects are collected from over 600 densely annotated scene meshes in diverse real-world environments.*

gorithms on the proposed dataset and submit the retrieved CAD models for evaluation. We then perform performance analysis based on these retrieval results.

2. Dataset

In this track, the query dataset consists of 2101 objects extracted from 3D reconstructed real-world indoor scene datasets, namely SceneNN [HPN*16] and ScanNet [DCS*17]. Each object is represented as a 3D triangular mesh. For the objects from SceneNN, per-vertex segmentation and annotation are performed by experts, utilizing the user interactive tool by Nguyen et al. [TNHYY16]. For ScanNet, the annotation is provided by crowd sourcing through Amazon Mechanical Turk. The reconstruction algorithms used in SceneNN and Scannet are derivatives from the KinectFusion pipeline [NIH*11], namely ElasticReconstruction [CZK15] and BundleFusion [DNZ*17], respectively.

The target dataset is a subset of ShapeNetSem [CFG*15] that contains only models for indoor scenes, consisting of 3308 objects in total. We adopt the category definitions of both NYU Depth v2 [SHKF12] and ShapeNetSem [CFG*15] to produce a fine-grained classification of common indoor objects such as table, chair, monitor, bookshelf, etc.

Apart from designing a dataset so that it closely mirrors the conditions in real-world environments to serve the contest, we carefully select high-quality scanned models to maximize the amount of available information such as texture and geometry that could be useful for matching with the CAD models. Compared to ObjectNN [HTT*17], the objects in our dataset are less noisy and ambiguous, and the dataset also has a larger scale. This poses as a serious challenge, and thus offers a good benchmark for 3D object retrieval algorithms. A comparison of our dataset to some of the previous works can be found in Table 1.

Ground Truth. Given that there is no standard metric to measure shape similarity, especially between an RGB-D object and a CAD model, we create the ground truth pairings manually by assigning the RGB-D objects and CAD models to a predefined set of categories and sub-categories. We use an interactive tool to display the object and ask a human subject to classify them. The classification is based on the shape, color, and semantic of the objects. After classification, for each RGB-D object, we assume all CAD models of the same category to be its ground truth retrieval targets.

Improvements. Compared to the original ObjectNN dataset, we organize our dataset strictly into categories and sub-categories using object attributes. In total, there are 20 categories and 43 sub-categories. We consider both categories and sub-categories in our final evaluation. Compared to ObjectNN [HTT*17], the size of our RGB-D dataset is larger by 26%. Thus the balance between RGB-D data and its CAD counterpart is also improved. For quality control, we also removed some highly ambiguous RGB-D objects from ObjectNN. Therefore, our dataset could be regarded as an extended and more refined version of the original ObjectNN.

Availability. We split the RGB-D objects dataset into two subsets: training and test, following a 70/30 ratio. We carefully choose the split so that all of the categories and sub-categories are presented in both subsets. We release the training RGB-D objects and all of the CAD models. Being different from the RGB-D to CAD Retrieval track by Hua et al. [HTT*17], to pose a more challenging problem, we do *not* release the ground truth categories for both RGB-D objects and CAD models. This choice is to better model real-world condition where retrieval dataset usually has incompleted ground truth information. Solving this issue should be an important first step in any successful retrieval systems. Hence, participants would have to preprocess the data to establish their own categories before applying supervised learning techniques. To assist the participants, we instead provide example ranked lists for every RGB-D queries in the training set. Note that the example ranked lists are *not* exhaustive, and only cover a subset of the ground truth pairings. We also release an overall object distribution of our dataset. All of the aboved information is available at our homepage [HPN*16].

3. Overview

Table 2 summarizes all methods used by the participants. In total we received six registrations for participation and two result submissions excluding a baseline submission from the organizers themselves, yielding a submission rate of 33.33%. Each participant can propose one or more algorithms to solve the retrieval problem, which will correspond to one or more runs to be evaluated.

In general, the proposed methods can be divided into two main classes: multi-view based approach using convolutional neural networks (Section 4, 5) and full 3D based approach using point-based neural networks (Section 6). Interested readers could proceed to Section 4-6 for more technical descriptions. The final evaluation results are discussed in Section 7.

Team	Method	Domain	Color
Li	Cross-domain learning (Sec. 4)	View-based	N
Tran	Circular view-rings (Sec. 5)	View-based	N
Khoi	Pointwise-CNN (Sec. 6.1)	Full 3D	Y
Khoi	PointNet (Sec. 6.2)	Full 3D	Y

Table 2: *An overview of the techniques used by the participants, all of them are based on supervised deep learning. Here we see the two different approaches, view-based approach and point-based convolutional neural networks.*

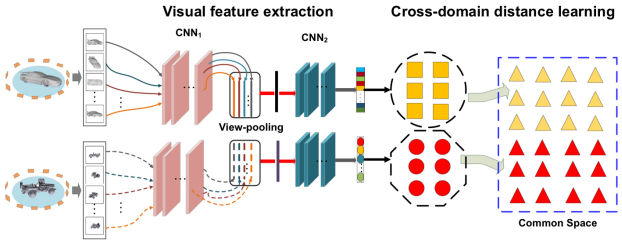

Figure 2: *Multi-view cross-domain framework for 3D object retrieval. Visual features are extracted and combined using multi-view convolutional neural networks with view-pooling. The features are then transformed into a common space for retrieval after cross-domain distance learning.*

4. Multi-view Cross Domain Retrieval

We propose a method to address cross-domain 3D model retrieval problem, where the query model and the target model come from different datasets with diverse data distribution. An overview of our method is shown in Figure 2. The proposed method consists of three successive steps: Data preprocessing, Visual feature extraction, and Cross-domain distance learning.

Data Preprocessing. Given that the categories of RGB-D objects and CAD models are not available, we leverage the partial retrieval results in the training set to divide the dataset into 20 classes. Specifically, according to the released object distribution, for each example query in the training set, we assume that the first 10 CAD models in the list belong to the same category. Then we find the intersection between these lists and divide the CAD models into 20 classes. For each CAD model, we find all the RGB-D objects whose top ten retrieval results contain the CAD model and infer the labels of the RGB-D objects. The label information is then used for supervised learning.

Visual Feature Extraction. For feature extraction, we follow the work of Su et al. [SMKLM15] and create 12 views by placing 12 virtual cameras around the model in every 30 degrees and produce a compact descriptor for individual 3D model. In our proposed method, we only utilize the geometry information from triangle meshes, discarding the color information. We use AlexNet [KSH12] as the base model to extract features from 2D views. We then place a view-pooling layer after pool5 layer to combine all the views together. For feature extraction, we first train the multi-view model with 3308 CAD models in the target dataset and then fine-tune the model with 1452 RGB-D objects in the query dataset. In particular, the output of the fc7 layer (4096-D) is used as visual features for each object.

Cross-domain Distance Learning. There is a large divergence in characteristics between the query and target datasets. RGB-D objects and CAD models can produce totally different visual images, even though they belong to the same category, which poses a great challenge for cross-domain 3D model retrieval task. Since the effect of fine-tuning on reducing the divergence between the two datasets is limited, we apply domain adaption method to solve the cross-domain retrieval problem. In our proposed approach, 3D models from different domains are matched by finding a unified transformation which transforms RGB-D and CAD features into a new common space. In more details, we leverage nonparametric Maximum Mean Discrepancy [LWD*13] to measure the difference in both marginal and conditional distributions. We then construct a new robust feature representation using Principal Component Analysis to reduce domain shifting. After the domain adaptation step, features from two domains are projected to a common space. We then measure the similarity between query and target directly by computing their Euclidean distance.

Retrieval. In this track, we submit three runs. The detailed configurations of each run is as follows:
no-cross-domain. This run computes the distance between RGB-D and CAD objects without using cross-domain learning procedure.
cross-domain-lambda-1. This run computes the distance between RGB-D and CAD objects by using cross-domain learning procedure with $\lambda = 1$.
cross-domain-lambda-10. This run computes the distance between RGB-D and CAD objects by using cross-domain learning procedure with $\lambda = 10$.

5. Circular View-Rings with Adaptive Attention

5.1. Data Preprocessing

Each RGB-D object in the training set has a corresponding partial ranked list of CAD models. First, we combine ranked lists with shared CAD objects, regardless of corresponding distances, to create 20 disjoint groups of CAD models. Next, we divide the CAD models into sub-categories based on the ranked list's retrieval scores and get a total of 348 sub-categories. In each category, we then merge sub-categories which have less than 2 items into one. After merging similar sub-categories, we have 37 major sub-categories and 14 minor sub-categories in final. To verify the consistency of our sub-categorization, we use our RVNet, a modified version of RotationNet [Kan16], with ResNet50 [HZRS16] for image encoding. The categories of RGB-D objects can be inferred from their CAD counterparts.

5.2. Circular View-Ring Classification

We inherit the ideas of using multi views in Multi-view CNN [SMKLM15] and the topological relationship between views in RotationNet [Kan16]. However, the key difference between our proposed method and RotationNet is that we do not enforce global

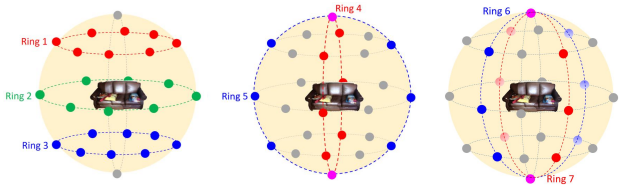

Figure 3: *Seven types of view-rings and their respective screenshot positions.*

topological relationship between all screenshots. We propose to exploit multi partial-topological relationships between different lists of screenshots.

We define a *View-Ring* as a circular list consisting of $N = 8$ screenshots of an 3D object taken uniformly around a circular orbit. As illustrated in Figure 3, we consider 2 types of view-rings: horizontal view-rings (Ring 1-3) and vertical view-rings (Ring 4-7). In each view-ring, we can preserve the topological relationship of 8 views. For each view-ring, we sequentially rotate the screenshots to get 8 *view sequences*.

Each view in a view sequence is encoded independently with ResNet50 [HZRS16] into a feature vector of 2048 elements. Then we concatenate 8 feature vectors of the 8 views in a view sequence into a feature vector of $2048 \times 8 = 16384$ elements. We use this feature to represent a view sequence.

Each RGB-D mesh is aligned using its eigenvectors and normalized to fit in a unit cube. Screenshots are taken at 26 view points distributed uniformly over the sphere centering at the mesh's center. On average It takes 2 minutes to render 26 screenshots of an RGB-D mesh object on Intel(R) Core i7 CPU @ 2.50GHZ Intel Core i7 4710HQ with 12 GB Memory.

Figure 4 shows the architecture of our view-ring classification network. The classification model for each view-ring is a fully connected neural network with 16384 input nodes; 3 hidden layers (with ReLU activation function) having 512, 256, and 256 nodes, respectively; and a softmax layer with 20 output nodes. For each view-ring i ($1 \leq i \leq 7$), we use 8 view sequences in the view-ring of all training RGB-D objects to train a classification model for the view-ring. The classification models for 7 view-rings are trained independently on Google cloud machines n1-highmem-2, each with 2 vCPUs, Intel(R) Xeon(R) CPU @ 2.50GHz Intel Xeon E5 v2, 13 GB Memory, and 1 x NVIDIA Tesla K80. The training time for a classification model is about 1-2 hours.

5.3. Adaptive Weight for Score Fusion

After classifying different view sequences in 7 view-rings of an RGB-D query object independently, we proposed attention mechanism with adaptive weighting to perform score fusion. The key idea is that we should pay more attention to view sequences with the following criteria:

- Meaningful human-oriented visual information. We use DHSNet [LH16] to generate a saliency map of a view sequence, then compute the level of visually importance of the view sequence.
- Harmony of view-ring and predicted class. From the training data, we can estimate the appropriateness of a view-ring to each category.
- High certainty for score prediction. If the score prediction vector has high information entropy, it means that such prediction might be with high uncertainty.

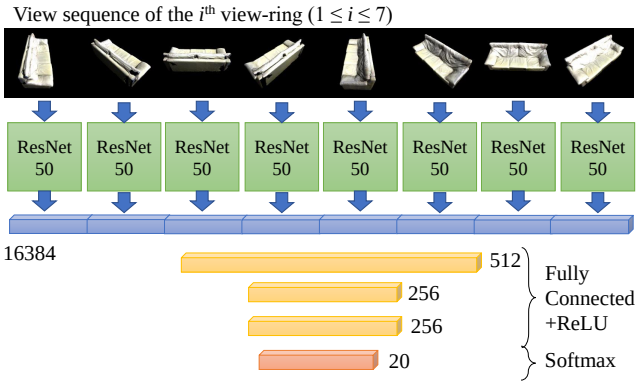

Figure 4: *The view-ring classifcation network architecture.*

The final label of an RGB-D query object is determined by the voting scheme from the prediction scores of all view sequences generated from the object using our proposed adaptive weight estimation.

5.4. Reranking with 2D BoW Retrieval

After predicting the category label for an RGB-D query object q, we insert all CAD models with the label into the output ranked list $L(q)$. To further refine this rank list, we use our Bag-of-Words (BoW) system [NNT*15, HTT*17] to retrieve a rank list $L_{RGB-D}(q)$ similar training objects in such category. From the top items in $L_{RGB-D}(q)$, we try to infer the sub-category $SC(q)$ of q among the 37 major sub-categories. The rank list $L(q)$ is sorted so that all CAD models in $SC(q)$ are in the top of the list (with distance 0) and other items are assigned with the distance 1. If we cannot determine $SC(q)$, we simply assign the distance 1 to all entries in $L(q)$. In our BoW system, we use RootSIFT without angle for keypoint descriptors, 1M codewords, soft assignment with 3 nearest neighbors, L1 asymmetric distance measurement [ZJS13].

In this track, we submit 4 runs. *view-ring-1* and *view-ring-2* only focus on the 20 main categories. We further rerank all ranked lists with sub-category information from BoW retrieval system and generate *view-ring-bow-1* and *view-ring-bow-2*, corresponding to view-ring-1 and view-ring-2, respectively.

6. Point-based Convolutional Neural Networks

We solve the retrieval problem by first performing object recognition with RGB-D point clouds as input, and then utilize the categories to return matched CAD models. To prepare for the training, the data is processed as follows.

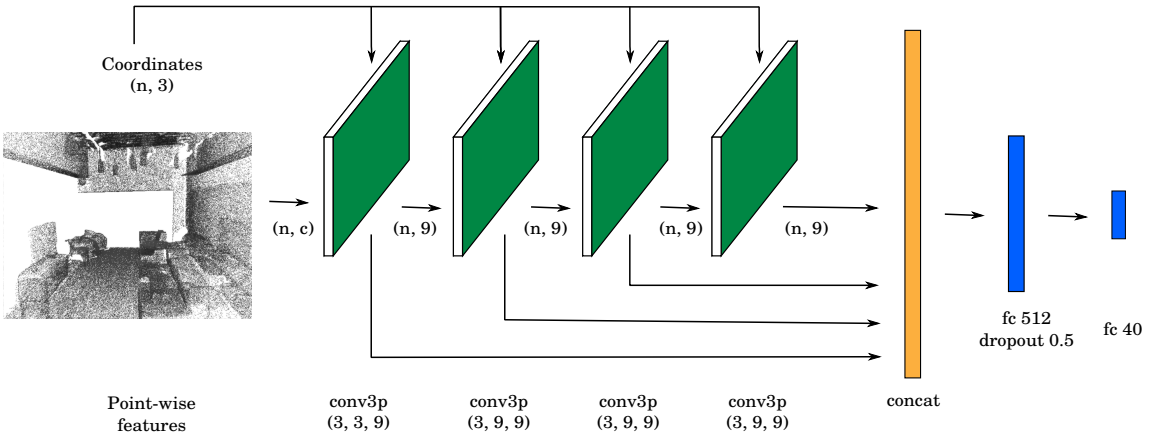

Figure 5: *The pointwise convolutional neural network architecture [HTY17].*

From the given example ranking results, we merge both RGB-D objects and CAD models to obtain 20 disjoint groups of a total of 3308 objects. For each RGB-D object, we apply uniform grid sampling to downsample the input to 2048 points, and each object is then assigned with a label indicating its category. After this step, the retrieval problem can be reformulated as an classification problem. We do this by simply return objects with the same predicted labels for each object in the test set.

The remaining task is to design neural networks to classify 3D point clouds. This is a relatively new research direction with very few existing techniques. Here, we leverage two typical techniques that can process 3D point clouds: Pointwise convolutional neural network [HTY17] and PointNet [QSMG17]. The general idea of pointwise convolution is to create a special operator that can handle convolution at each point of the point cloud, making supervised learning with point clouds similar to traditional learning with 3D volumes or 2D images. On the other hand, PointNet aims to learn point set features by handling the unordered property of point sets by a symmetric function. The details of the methods are described below.

6.1. Pointwise Convolutional Neural Network

The pointwise convolutional neural network (Pointwise-CNN) architecture is built upon a new convolution operator for point cloud, which is recently introduced by Hua et al. [HTY17]. The main idea is that the convolution kernel is placed directly at each point. Similar to an ordinary convolution, neighboring points within a chosen radius value are selected to contribute to the center point.

Formally, point-wise convolution can be written as:

$$x_i^{\ell} = \sum_k w_k \frac{1}{|\,\Omega_i(k)\,|} \sum_{p_j \in \Omega_i(k)} x_j^{\ell-1}, \qquad (1)$$

where k iterates over all sub-domains in the area within the kernel support; $\Omega_i(k)$ represents the k-th sub-domain of the kernel centered at point i; p_i is the coordinate of point i; $|\cdot|$ is the operator that counts all points within the sub-domain; w_k is the kernel weight at the k-th sub-domain, x_i and x_j the value at point i and j, and $\ell - 1$ and ℓ are the indices of the input and output layer. Interested readers could find more details about the implementation of this operator and its gradient computation in the work by Hua et al. [HTY17].

For object recognition, we keep the same architecture as in Pointwise-CNN [HTY17] (see Figure 5) and only change the last layer to support 20 categories. The network is pre-trained on the ModelNet40 dataset and then fine-tuned for 300 epochs on the training dataset. The final classification accuracy is 65%.

6.2. PointNet

To provide additional comparisons for point-based deep learning techniques, we also apply PointNet [QSMG17] for object recognition. PointNet is the seminal work for point cloud deep learning that aims to address unordered point input by using a max-pooling layer to learn order-invariant point features. Another important idea in their work is the use of transformation modules that allows rotation invariance in the input.

We also use their classification network to tackle the retrieval problem. Interestingly, we are not able to apply a pre-trained network on ModelNet40 dataset and refine with the dataset in this challenge as the fine tuning is unable to converge. We henceforth perform the training from scratch, where it smoothly converges. At test time, PointNet achieves an accuracy of about 70% in the classification task.

7. Evaluation

For each query model, each participant submits a ranked list where retrieved models are sorted according to 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, mean Average Precision (mAP), and Normalized Discounted Cumulative Gain (NDCG). Please refer to our homepage [HPN*16] for additional evaluation metrics, i.e. F-Score, Nearest Neighbor First-Tier (Tier1) and Second-Tier (Tier2).

NDCG metric will use a grade relevance to provide a more fine-grained evaluation, whereas other metrics will be evaluated on binary

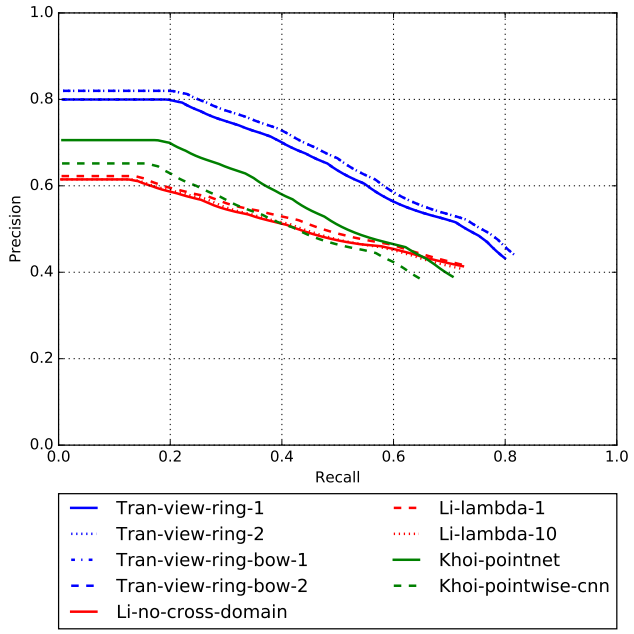

Figure 6: *Precision-recall curves of all methods submitted by participants. While not yet perfect, most proposed method perform relatively well. This show that features based on deep neural networks, either using multi-view or point based, can capture the high variance in geometry of our dataset. (Best view on screen with color)*

in-category versus out-of-category relevance. The grade relevance for NDCG is as follows: 2 for correct category and subcategory match, 1 for correct category match only, and 0 for false match.

To compute the scores, we consider the first K retrieved results, where K is the number of objects in the ground truth class. The main reason that we chose this evaluation strategy is mainly for fairer comparison; since some of the participants decided to return rank list of the entire CAD dataset for each query, and some return CAD models only in the predicted category. Note that in the second case, the scores for each query will either be 0 or 1 in all metrics, making the final scores become similar after averaging. This does not occur in the first case. Figure 6 shows precision-recall curves for all methods. The detailed evaluation results can be found in Table 3.

In general, the method proposed by Tran (Sec. 5) which is based on circular view-rings with adaptive attention outperforms both the multi-view CNN cross domain retrieval architecture proposed by Li (Sec. 4) and the point-based neural network approaches proposed by Hua (Sec. 6). This performance differences could be explained by the fact it combined multi-view features in an adaptive attention manner, which is a strong cue in 3D object recognition. All of the methods also made good use of the partial ranked lists provided for the training set to divide the dataset into disjoint categories. These categories can be utilized to turn the object retrieval problem into classification problem, which can be solved using supervised deep learning techniques.

Among view-based methods, view-rings with adaptive attention is the most competitive. For example, Tran-view-ring-2, -view-ring-bow-2 top our evaluation chart, and outperform Li's MVCNN approach by a margin. A possible reason is that Tran-view-ring methods can capture the rotation invariance of 3D models through rotating the view rings to create view sequences. Another explanation is that the feature vectors from Tran's method is much larger than Li's, i.e. 16384 elements to 4096 elements, which can correlate to better discrimination power. It is also worth noting that, when considering the NDCG scores, the Tran-view-ring-bow-2 has a small gain in performance compared to their vanilla Tran-view-ring-2 run, proving that their proposed reranking with BoW scheme is indeed effective.

For point-based methods, PointNet [QSMG17] is the most effective architecture, outperforms Pointwise-CNN [HTY17] by a small margin. Compared to Tran's method, the point-based methods give lower performance. One of the possible reasons is the size of their input point clouds is quite low, only 2048 points. This may not be enough to distinguish some ambiguous categories such as box and printer.

Figure 7 shows the precision plot for each category of all methods. The mean scores in most of the categories are over 50%. Some categories has almost perfect precision scores such as chair, display, or keyboard. Some of the ambiguous categories are printer, machine, and pc. This can be explained by the unbalance in categories of the dataset. It can be seen that Tran's method has consistent performance over different categories, never scores lower than 50% in precision, except on the printer class.

To study the effects of unbalanced data on retrieval performance, we propose an alternative evaluation. In this evaluation, each query is weighted by the inverse frequency of its ground truth category. The scores are shown in Table 3. It can be seen that Tran's method and Khoi's point-based methods are more robust to unbalanced data, dropping less than 5% in scores. Li's method shows a more than 10% loss in performance. Addressing this problem might increase the performance of future 3D object retrieval systems.

Overall, we conclude that the main technical challenges in the RGB-D object-to-CAD retrieval problem comes from the RGB-D data. Since these RGB-D objects are often acquired using a common depth reconstruction pipeline (Sec. 2), the resulting triangular meshes are generally more noisy than CAD models, causing them difficult to match in terms of geometry. In addition, color texture of real-world objects widely varies, and matching them to the set of textures of the CAD models is particularly challenging especially in 3D. While the evaluation results show that current deep learning approaches can solve this problem moderately, to improve the performance further, one should consider using additional cues such as object parts, object affordances, or scene context.

8. Conclusions

In this SHREC track, we benchmark several algorithms for 3D object retrieval, using our extended version of ObjectNN dataset. The query is an RGB-D object and the target is a CAD model. We found that for this problem, multi-view based approach with adaptive score fusion is the most effective, followed by point based convolutional neural networks. With a total domination of deep learning related approaches in the submissions, we observe an interesting trend

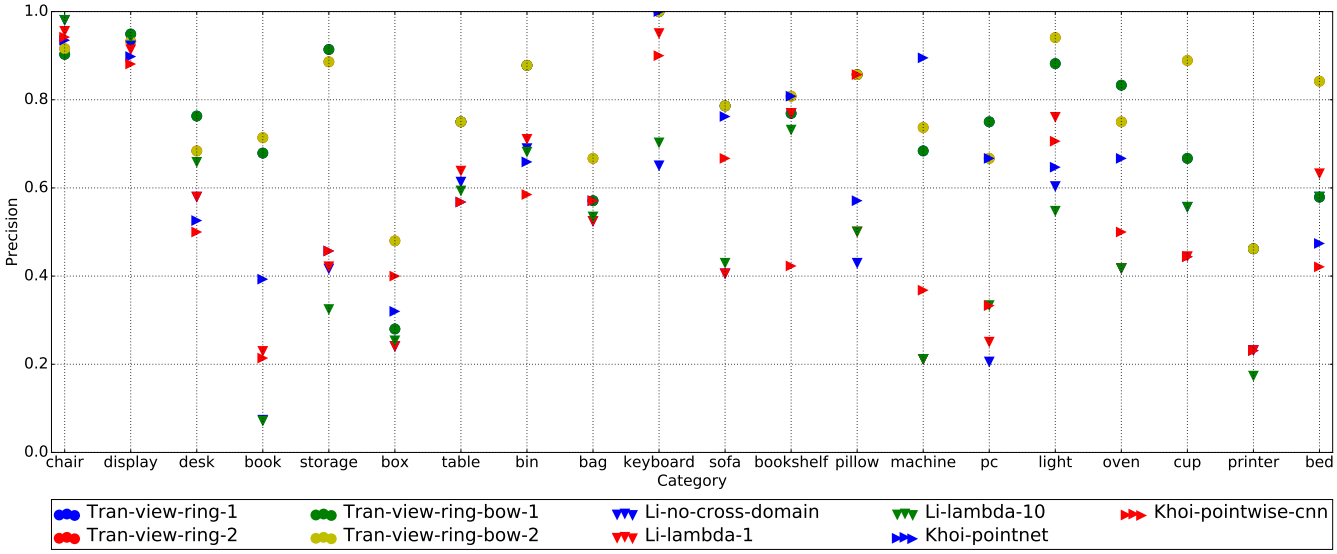

Figure 7: *Precision plot of all methods for each category in the dataset. As can be seen, most categories can be recognized well with mean precision over 50%. The plot also suggests that some categories are ambiguous such as box or printer. (Best view on screen with color)*

Dataset	Run	standard				weighted			
		Precision	Recall	mAP	NDCG	Precision	Recall	mAP	NDCG
Tran	view-ring-1	0.800	0.800	0.800	0.760	0.753	0.753	0.753	0.717
Tran	view-ring-2	0.820	0.820	0.820	0.779	0.781	0.781	0.781	0.742
Tran	view-ring-bow-1	0.800	0.800	0.800	0.781	0.753	0.753	0.753	0.717
Tran	view-ring-bow-2	**0.820**	**0.820**	**0.820**	**0.801**	**0.781**	**0.781**	**0.781**	**0.742**
Li	no-cross-domain	0.638	0.638	0.625	0.616	0.489	0.489	0.485	0.467
Li	cross-domain-lambda-1	0.657	0.657	0.638	0.631	0.538	0.538	0.530	0.514
Li	cross-domain-lambda-10	0.641	0.641	0.626	0.617	0.508	0.508	0.501	0.483
Khoi	pointwise-cnn	0.652	0.652	0.652	0.613	0.610	0.610	0.610	0.584
Khoi	pointnet	0.706	0.706	0.706	0.665	0.675	0.675	0.675	0.647

Table 3: *Evaluation results on the test set. Tran's method tops the scoreboard. Other methods based on multi-view and point based convolutional neural networks also perform well. The score for standard evaluation strategy (left) and the weighted evaluation strategy (right) are also presented.*

where traditional unsupervised methods are being unfavored by researchers compared to deep learning techniques, which appear to be more powerful in solving the 3D object retrieval problem.

There are a few promising future research directions. First, we can continue growing the dataset. For example, Matterport3D [CDF*17] is another large-scale indoor reconstruction dataset that we can look into. Second, the ground truth pairing and categorization can be further refined to create a detailed ranked lists for every objects, instead of in-category versus out-of-category relevance. Third, we want to exploit the context information in object retrieval by including the fragment of the scene where this object appears. This may pose an interesting challenge for 3D object retrieval, where only the context of the query object is known beforehand.

Acknowledgement

We thank the following teams for their kind participation and contribution to this manuscript: Li et al. for Section 4, Tran et al. for Section 5, and Minh-Khoi Tran for Section 6. We are grateful to the authors of ScanNet [DCS*17] for making their dataset publicly available, which we utilize partially for this contest.

Binh-Son Hua and Sai-Kit Yeung are supported by the SUTD Digital Manufacturing and Design Centre which is supported by the Singapore National Research Foundation (NRF). Sai-Kit Yeung is also supported by Singapore MOE Academic Research Fund MOE2016-T2-2-154, Heritage Research Grant of the National Heritage Board, Singapore, and Singapore NRF under its IDM Futures Funding Initiative and Virtual Singapore Award No. NRF2015VSGAA3DCM001-014.

References

[CDF*17] CHANG A., DAI A., FUNKHOUSER T., HALBER M., NIESSNER M., SAVVA M., SONG S., ZENG A., ZHANG Y.: Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158* (2017). 7

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., ET AL.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 1, 2

[CZK15] CHOI S., ZHOU Q.-Y., KOLTUN V.: Robust reconstruction of indoor scenes. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (2015), IEEE, pp. 5556–5565. 2

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T., NIESSNER M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* (2017). 1, 2, 7

[DNZ*17] DAI A., NIESSNER M., ZOLLHÖFER M., IZADI S., THEOBALT C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG) 36*, 3 (2017), 24. 2

[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

[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). URL: http://www.scenenn.net. 1, 2, 5

[HTT*17] HUA B.-S., TRUONG Q.-T., TRAN M.-K., PHAM Q.-H., KANEZAKI A., LEE T., CHIANG H., HSU W., LI B., LU Y., JOHAN H., TASHIRO S., AONO M., TRAN M.-T., PHAM V.-K., NGUYEN H.-D., NGUYEN V.-T., TRAN Q.-T., PHAN T. V., TRUONG B., DO M. N., DUONG A.-D., YU L.-F., NGUYEN D. T., YEUNG S.-K.: RGB-D to CAD Retrieval with ObjectNN Dataset. In *Eurographics Workshop on 3D Object Retrieval* (2017), Pratikakis I., Dupont F., Ovsjanikov M., (Eds.). 1, 2, 4

[HTY17] HUA B.-S., TRAN M.-K., YEUNG S.-K.: Point-wise convolutional neural network. *arXiv preprint arXiv:1712.05245* (2017). 5, 6

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778. 3, 4

[Kan16] KANEZAKI A.: Rotationnet: Learning object classification using unsupervised viewpoint estimation. *CoRR abs/1603.06208* (2016). 3

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105. 3

[LH16] LIU N., HAN J.: Dhsnet: Deep hierarchical saliency network for salient object detection. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on* (2016), IEEE, pp. 678–686. 4

[LWD*13] LONG M., WANG J., DING G., SUN J., YU P. S.: Transfer feature learning with joint distribution adaptation. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013* (2013), pp. 2200–2207. 3

[NIH*11] NEWCOMBE R. A., IZADI S., HILLIGES O., MOLYNEAUX D., KIM D., DAVISON A. J., KOHI P., SHOTTON J., HODGES S., FITZGIBBON A.: Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on* (2011), IEEE, pp. 127–136. 2

[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. *IJMDEM 6*, 2 (2015), 37–51. 4

[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

[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

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE 1*, 2 (2017), 4. 5, 6

[SHKF12] SILBERMAN N., HOIEM D., KOHLI P., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision* (2012), Springer, pp. 746–760. 2

[SMKLM15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E.: Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 945–953. 3

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

[ZJS13] ZHU C.-Z., JÉGOU H., SATOH S.: Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (2013), IEEE, pp. 1705–1712. 4