Eurographics Workshop on 3D Object Retrieval (2020)
T. Schreck, T. Theoharis, I. Pratikakis, M. Spagnuolo, and R. C. Veltkamp (Editors)

SHREC 2020 - Extended Monocular Image Based 3D Model Retrieval

Wenhui Li[1†], Dan Song[1†*], Anan Liu[1†*], Weizhi Nie[1†], Ting Zhang[1†], Xiaoqian Zhao[1†], Mingsheng Ma[1†], Yuqian Li[1†], Heyu Zhou[1†]
Beibei Zhang[2], Shengjie Le[2], Dandan Wang[2], Tongwei Ren[2], Gangshan Wu[2]
The-Anh Vu-Le[3,4], Xuan-Nhat Hoang[3,4], E-Ro Nguyen[3,4], Thang-Long Nguyen-Ho[3,4], Hai-Dang Nguyen[3,4], Trong-Le Do[3,4], Minh-Triet Tran[3,4]
[1] School of Electrical and Information Engineering, Tianjin University, China.
[2]Nanjing University, China.
[3]University of Science, Ho Chi Minh city, Vietnam.
[4]Vietnam National University, Ho Chi Minh city, Vietnam.

Abstract

Monocular image based 3D object retrieval has attracted more and more attentions in the field of 3D object retrieval. However, the research of 3D object retrieval based on 2D image is still challenging, mainly because of the gap between data from different modalities. To further support this research, we extend the previous track SHREC19'MI3DOR to organize this track, and we construct the expanded monocular image based 3D object retrieval benchmark. Compared with SHREC19'MI3DOR, this benchmark adds 19 categories for both 2D images and 3D models to the original 21 categories, taking into account the lack of categories for practical applications. Two groups participated, proposed three kinds of supervised methods and submitted 20 runs in total, and 7 commonly-used criteria are used to evaluate the retrieval performance. The results show that supervised methods still achieve satisfying retrieval results (Best NN is 96.7% for 40 categories), which are comparable to the results of SHREC19'MI3DOR. In the future, unsupervised methods are encouraged to discover in monocular image based 3D model retrieval.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Computer Graphics]: Information Systems—Information Search and Retrieval

1. Introduction

In recent years, the number of 3D models has exploded with the development of 3D related technologies, which has led to the problem of how to manage them effectively. Therefore, the task of 3D object retrieval becomes more and more important. With easier access to 2D images, retrieval of 3D models using 2D images becomes an important idea for 3D retrieval, and has attracted the attention of researchers. "Monocular image based 3D object retrieval (MI3DOR)" aims to search for relevant 3D models from a dataset when given a real-word-captured RGB image, which is novel for 3D object retrieval. As the discrepancy in both domains and modalities is a main problem in cross-modal retrieval, work of 3D object retrieval based on 2D images is challenging, which also attracts attentions from researchers.

To facilitate more innovative and interesting developments in this research, we previously constructed a benchmark of MI3DOR. The dataset has 21 categories for both 2D images and 3D models. The 2D images have 1000 samples per category, and the total number of 3D models is 7690. Based on the benchmark, we also organized the track SHREC19'MI3DOR [LLN*19]. SHREC19'MI3DOR attracted 9 groups from 4 countries and the submission of 20 runs.

In fact, objects in real life scenarios are much more diverse. Considering more practical needs, we have extended the original dataset to obtain the new benchmark. On the basis of retaining the original 21 categories, 19 categories have been added for both 2D images and 3D models, where the number of 2D images has been increased to 40,000 and the number of 3D models has been increased to 12,732. We organize this track to have the following proposals: new methods applying on the initial SHREC'19 benchmark which may bring exciting progress, or existing and new approaches to evaluate the performance of the extended SHREC'20 benchmark.

Overall, the extended benchmark contains 40 categories for both 40,000 2D images and 12,732 3D models. This track aims to encourage progress in 3D object retrieval using 2D monocular images and is divided into two tasks for 21 and 40 categories. Two teams from two countries contribute to this track, while three kinds of

† Track Organizers. * Corresponding author: Dan Song and Anan Liu. E-mail: dan.song@tju.edu.cn and anan0422@gmail.com.

DOI: 10.2312/3dor.20201163

supervised methods are proposed and 20 runs are submitted. The evaluation results show the creative contributions of each team in retrieving 3D object based on monocular images, and also reflect the prospects and potential challenges in the field.

In summary, our work has the following contributions:

- **Challenging but promising task:** Our track aims at a cross-domain 3D object retrieval task with more categories, and participants contribute different kinds of methods with multiple variants. The task is challenging while the results will encourage progress in the related research.
- **Comprehensive evaluation:** We employ 7 widely-used criteria to evaluate the proposed 3 kinds of supervised methods with 20 runs, which will give guidance for further research.
- **Dataset:** We expand the MI3DOR benchmark from 21 to 40 categories for both 2D images and 3D models (40,000 2D images and 12,732 3D models), contributing a large dataset closer to practical scenarios.

2. Extended MI3DOR Benchmark

2.1. 2D Query and 3D Gallery

The 2D images of the extended benchmark are selected from ImageNet [DDS*09] and the 3D models are selected from the popular 3D dataset NTU [CTSO03], PSB [SMKF04], ModelNet40 [WSK*15], ShapeNet [SYS*17], which share the same source with the SHREC'19 benchmark. One example per class is shown in Fig. 1 and Fig. 2.

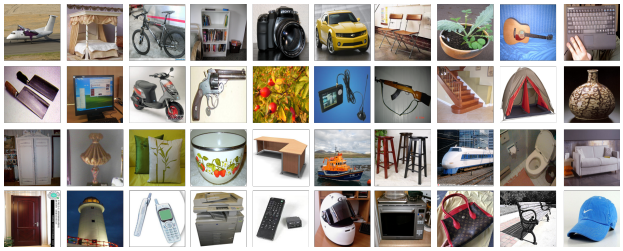

Figure 1: *2D object-centered image examples in the extended MI3DOR dataset.*

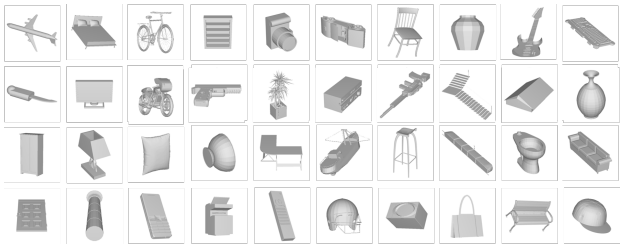

Figure 2: *3D object examples in the extended MI3DOR dataset.*

2.2. Dataset

Besides the original 21 categories of SHREC'19, 19 more categories are added in this benchmark, including lamp, pillow, bowl, desk, ship, stool, train, toilet, sofa, door, tower, telephone, printer, remote_control, helmet, microwave, bag, bench, and cap. Tab. 1 shows the data distribution for each category. The total number of 2D image samples is 40000 (.JPEG) and each class has 1000 samples. We randomly select 500 images per class for training and use the remaining samples for testing. The total number of 3D models is 12,732. We also randomly select 50% samples per class as the train set and use the remaining data for testing. As some classes are lack of samples, the train set contains 6,361 models (.OBJ), and the test set contains 6,371 models (.OBJ). We follow [SMKLM15] to render the OBJ models and get 12 views for each 3D object. The Tab. 2 shows the information of this benchmark.

Table 1: *Data distribution of the dataset.*

Category	Model	Image	Category	Model	Image
airplane	500	1000	wardrobe	500	1000
bed	500	1000	lamp	500	1000
bicycle	62	1000	pillow	76	1000
bookshelf	500	1000	bowl	232	1000
camera	90	1000	desk	500	1000
car	500	1000	ship	500	1000
chair	500	1000	stool	117	1000
flower_pot	500	1000	train	330	1000
guitar	500	1000	toilet	444	1000
keyboard	217	1000	sofa	500	1000
knife	355	1000	door	169	1000
monitor	500	1000	tower	106	1000
motorcycle	285	1000	telephone	500	1000
pistol	245	1000	printer	132	1000
remote_control	52	1000	plant	477	1000
radio	124	1000	helmet	139	1000
rifle	500	1000	microwave	121	1000
stairs	143	1000	bag	66	1000
tent	192	1000	bench	500	1000
vase	500	1000	cap	58	1000

Table 2: *Training and testing subsets of the benchmark.*

Benchmark	Image	Model	View
Train	20,000	6,361	6,361×12=76,332
Test	20,000	6,371	6,371×12=76,452
Total	40,000	12,732	12,732×12=152,784

2.3. Evaluation

For quantitative comparison, we employ the same evaluation criteria as in the SHREC'19 MI3DOR track, which are Precision-recall (PR) curve, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), F-Measure (F), Discounted Cumulative Gain (DCG), Average Normalized Modified Retrieval Rank (ANMRR) and Area Under Curve(AUC). The higher values of NN, FT, ST, F, DCG and AUC indicate better performance, while the lower value of ANMRR is better.

- **Precision-recall (PR) curve** is a curve obtained by adjusting the classification threshold with Recall as x axis and Precision as y

axis, which intuitively shows the comprehensive performance of retrieval. The area under curve (AUC) of PR-curve is a indicator for further analysis.

- **Nearest Neighbor (NN)** is an evaluation criteria for the retrieval accuracy of the first returned result.
- **First Tier (FT)** is the recall of the top k relevant results for the query.
- **Second Tier (ST)** is similar to FT but the recall of the top 2k results.
- **F-Measure** comprehensively evaluates both the Precision and Recall indicators.
- **Discounted Cumulative Gain (DCG)** considers both the relevance of the results and the position information. It assigns weights according to the ranking position information of the relevant results.
- **Average Normalized Modified Retrieval Rank (ANMRR)** is the average NMRR, where NMRR considers the ranking information of the retrieved results.The lower the ANMRR value, the better the performance.

3. Participants

1. **SORMI** submitted by MAGUS.ZLW Team (Shengjie Le, Tongwei Ren, Dandan Wang, Gangshan Wu and Beibei Zhang from Nanjing University).
2. **VSE-MI3DOR & VRQ-MI3DOR** submitted by HCMUS-Junior Team (The-Anh Vu-Le, Xuan-Nhat Hoang, E-Ro Nguyen, Thang-Long Nguyen-Ho, Hai-Dang Nguyen, Trong-Le Do, and Minh-Triet Tran from University of Science, Vietnam National University, Ho Chi Minh City, Vietnam).

4. SORMI, by MAGUS.ZLW Team

Considering the high appearance diversity within each class of both monocular images and 3D models, they propose the Semantic Similarity based 3D Object Retrieval from Monocular Image (SORMI). Fig. 3 shows the framework of the method. They first extract the semantic representation of query images and the gallery 3D models, and then measure their semantic similarities to sort the 3D models. Specifically, they utilize Inception-ResNet-v2 [SIVA17] to extract the semantic representation from monocular images, and GVCNN [FZZ*18] to extract the semantic representation from the 2D rendered views of 3D models. In semantic similarity measurement, they select the top 5 or 8 classes with maximum value from the class probability vectors of both monocular images and 3D models to measure the semantic similarity with cosine distance or vector multiplication.

As shown in Tab. 3, they finish two tasks and provide five submissions with different similarity measurement strategies for each task. In all the submissions, they augment the 2D rendered views of 3D models by capturing their new views considering the lack of views of 3D models in certain classes like bag and door. After view augmentation, each class of 3D models has 250 view groups, and each view group contains 12 views.

The main differences among their submissions lie in the similarity measurement strategies. To suppress the influence of the elements with low confidences in the class probability vectors, they only retain the top-k classes with highest values and set the scores of other classes to 0 in the semantic representations of both monocular images and 3D models. In the experiments, they find that it might obtain the best performance when the semantic representations retain the top-5 (21_s01, 21_s02, 21_s03, 40_s04, 40_s05) or top-8 (21_s04, 21_s05, 40_s01, 40_s02, 40_s03) classes. Moreover, they attempt to use different similarity measurements between the semantic representation vectors of monocular images and 3D models. Though cosine distance (21_s01, 40_s03) is more meaningful, vector multiplication (21_s02, 21_s03, 21_s04, 21_s05, 40_s01, 40_s02, 40_s04, 40_s05) performs better in the most conditions in their experiments. They also try to normalize the scores in semantic representation vector with softmax (21_s03, 21_s05, 40_s02, 40_s05).

They make attempt to capture more views for 3D models. It is observed that many of the 3D models given are tilted or upside down. Taking views from these models directly, the objects in captured views are not straight. In order to solve the problem, they straighten the 3D models by remain intact, rotating 180 degrees around the Y-axis and rotating 90 degrees around the X-axis and captured 12 views for each transformation. In this way, they can ensure that at least 12 of the 36 views are straight. However, limited by batch size, the 36 views fail to achieve equally satisfactory performance compared with the original 12 views. Hence, they do not include this method in their submissions.

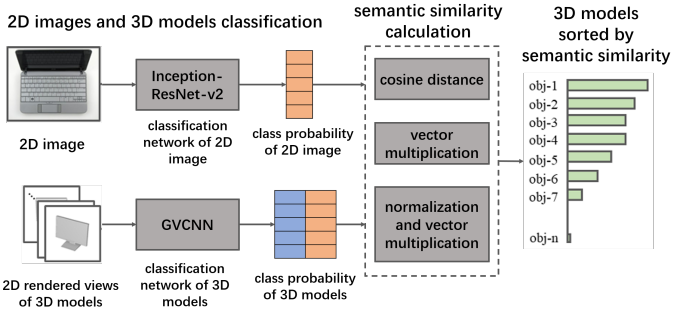

Figure 3: *The framework of SORMI.*

Table 3: *The description of the five submissions of SORMI.*

Task	Submission	Similarity Measurement
21-supervised	SORMI_s01	top5-cos
	SORMI_s02	top5-mul
	SORMI_s03	top5-mul-norm
	SORMI_s04	top8-mul
	SORMI_s05	top8-mul-norm
40-supervised	SORMI_s01	top8-mul
	SORMI_s02	top8-mul-norm
	SORMI_s03	top8-cos
	SORMI_s04	top5-mul
	SORMI_s05	top5-mul-norm

5. VSE-MI3DOR & VSQ-MI3DOR, by HCMUS-Junior Team

5.1. Solution Overview

Their main approach to this problem is to separate it into two independent tasks of classification for 2D images and 3D objects, each results in a classification module (see Fig. 4). For each input 2D image or 3D object, the corresponding classification module generates a score vector, the elements of which reflect the confidence of the module that the input belongs to each of the categories. In other words, each input x is represented as a vector $S_x \in [0,1]^C$, where C is the number of categories. They combine the results from these two modules to produce the final retrieval results.

In each 2D image or 3D model, there can be different parts related to different categories, such as *helmet* and *motocycle*, *printer* and *desk*. Therefore, instead of using a single label from the classification process, they keep classification scores for all categories to find all possible relationships between a 2D query image and 3D models. This approach is appropriate for the data in this challenge as each image or object may have more than appropriate labels.

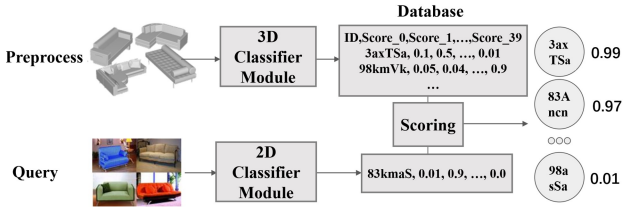

Figure 4: *Two-staged approach to 3D objects retrieval from input 2D image.*

There are two stages. In the **preprocessing** stage, they create a CSV file containing all the 3D predictions using the 3D classification module. In the **query** stage, each input 2D query image is passed into the 2D classification module to generate the scores. These scores are used to compute the matching score between the query and each of the 3D objects in the database. The final retrieval result is the sorted list of all 3D objects in descending order of the calculated scores.

5.2. 2D Image Classification with EfficientNet

To classify 2D images, they employ transfer learning using different variations of EfficientNet [TL19] models pre-trained on ImageNet. Fig. 5 describes their architecture. In their experiments, they freeze all layers, up to the global average pooling layer, and they train their final softmax layer. They also perform experiments on ResNet [HZRS16], and Xception [Cho17]. However, the EfficientNet-B1 gives the best result on their validation set.

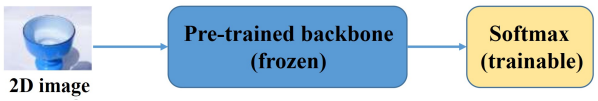

Figure 5: *Overview of their 2D classifier*

5.3. 3D Object Classification with View Set and View Sequences

Their proposed methods follow the common pipeline with 4 main steps: (1) 3D object representation, (2) 2D view embedding, (3) embedding fusion and classification, (4) ensemble of different models.

In step 1, they follow the multiple 2D-view strategy for 3D object classification (see Fig. 6). They propose two main approaches to generate the 2D views: view set and view sequences.

- In the view set approach, each 3D object is represented as a collection of 2D views and they do not exploit any spatial relationship between these images.
- For the view sequences approach, each 3D object is represented as a collection of sequences, each sequence consists of topologically ordered 2D images captured when moving a camera around the object in a specific trajectory. The view ring of [PTL*18] is an implementation of that idea, with the camera moving in a circular trajectory around an object (see View Rings of Fig. 6).

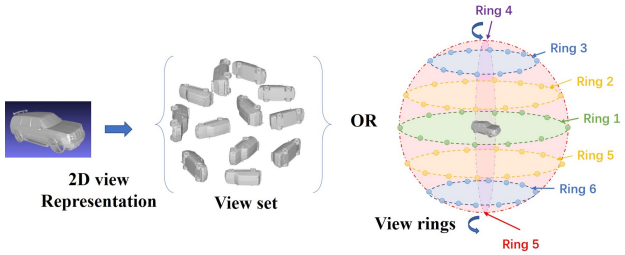

Figure 6: *3D object represented as a view set or view sequences.*

In step 2, they use a CNN with pre-trained weights on ImageNet as the feature extractor for the generated views of the 3D object.

They choose different state-of-the-art network architectures as backbone, namely different versions of ResNet and EfficientNet. They also consider freezing and unfreezing (allowing weights update) when training their proposed methods. With the unfreezing setting, they intend to further refine the pre-trained weights for image feature encoder of the chosen backbone.

Different methods for the two main approaches in step 3 are presented in Section 5.3.1 and 5.3.2.

After step 3, the initial 3D object is now embedded as a single vector, and they use a simple fully-connected network for classification into 1 of the 40 classes, gaining for each object the confidence score that the object belongs to each of the classes.

In step 4, they use either dot product or cross entropy to calculate the similarity scores between a query image and a 3D model.

5.3.1. 3D Object Classification with View Set (VSE)

Each 3D model is represented by a set of n images. They can simply use the images in their original order. Shuffling images in the view set as data augmentation can be a potential technique to enhance the result.

They propose a method for combining multiple views without

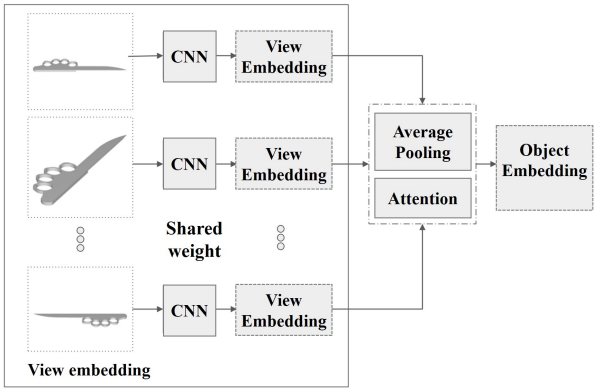

Figure 7: *View Set classification.*

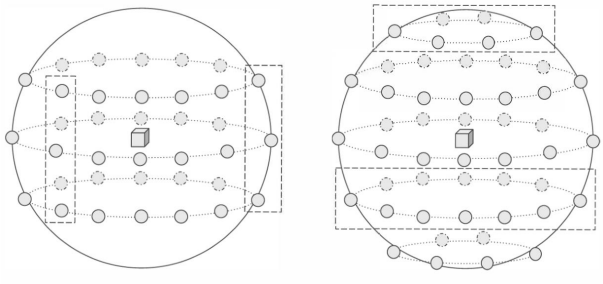

Figure 8: *Two possible settings of sequences generation (left: transposed setting, right: normal setting). Each settings come with two examples, where the views generated from camera positions bounded by each dotted box are of the same sequence.*

topological order into a single embedding(see Fig. 7): Consider all the view embeddings, that each view is now represented as a feature vector, as a collection of words and use Self Attention or LSTM to automatically attenuate the unimportant words and detect the salient keywords in the collection. The resulting vectors are then average-pooled to utilize global information.

5.3.2. 3D Object Classification with View Sequences (VSQ)

The given 3D object can be utilized by taking 2D snapshots from orbiting cameras. They consider an object by its smallest spherical hull and determine a fixed set of R latitudes of that sphere, called **rings**. The camera is positioned at V evenly divided position, facing toward the center of the object. From these cameras, a total of $R \times V$ 2D images are taken, called **views**. In this project, $V = 12$ while $R = 7$ representing the cameras on the equator and the 30/60/90 latitudes from both hemispheres. The generation method is developed from their view ring approach proposed in [PTL*18] with some modifications. First, they use 7 horizontal rings, instead of 3 horizontal and 4 vertical ones as in [PTL*18]. They intend to exploit view rings at different latitudes surrounding a 3D model. Second, each ring now includes 12 views, instead of 8 views, to get denser information on the object.

From the 2D images taken following the above steps, there are multiple ways to generate the collection of sequences. In their work, they use two different ways (see Fig. 8): (1) the normal setting considers 7 rings as 7 sequences of 12 views each, and (2) the transposed setting uses only 3 of the 7 rings (indexed 5, 1, 2 corresponding to the -30-degree latitude, the equator, and the +30-degree latitude), and instead of considering it as 3 sequences of 12 views each, this setting considers it as 12 sequences of 3 views each. Method (2) is proposed in light of the difference in quality between rings possibly affecting the classification result.

Each of the S sequences can be embedded into a vector using the following procedure: (1) each view of V views is passed through a backbone CNN which results in a D-dimensional vector, (2) these V vectors are concatenated and fed sequentially into a Bidirectional LSTM which results in a sequence of vectors of shape $2V \times D$, each 2 vectors in this sequence is the output of the Bi-LSTM at the corresponding input timestep, (3) the sequence is averaged to generate the final D-dimensional vector. In (2), independent LSTMs are used for each sequence, i.e. the weights of the LSTMs are not shared between each ring. Meanw, in (2), it is possible to follow the LSTM with an attention mechanism to potentially adjust the vectors (as proposed for some of the runs).

Given S D-dimensional vectors, using the self-attention mechanism, each vector is better calibrated based on the information of other (itself included) vectors. By taking the average of these newly calibrated vector, the global information can be utilized of every sequence. The resulting D-dimensional vector can be viewed as the vector embedding of the 3D object, which can then be passed into a simple fully connected network (classifier) to get the score for each of the C classes.

In some of their submitted runs, the parameters are $D = 1280$ (with EfficientNet-B1 as backbone), $S = 12, V = 3$ (using the transposed setting with 3 rings indexed 5, 1, 2 as described above), and $C = 40$ (total number of categories) (see Fig. 9). In some other runs, the parameters are $D = 2048$ (with ResNet50 as backbone), $S = 7, V = 12$ (using the normal setting with all 7 rings as described above), and $C = 40$.

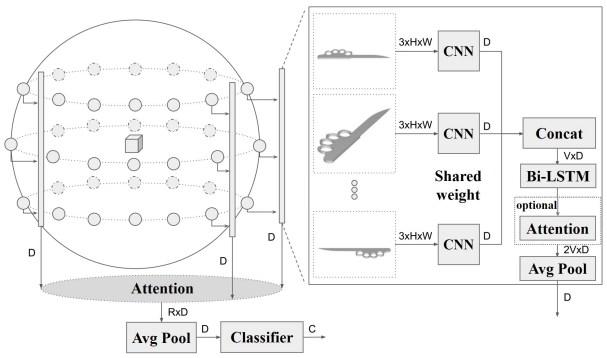

Figure 9: *Example of using the transposed setting in the proposed model. The shape associated with each arrow is the output shape of the previous step.*

Table 4: *Illustration of the all the submissions.*

Task	Submission	Feature Extractor (2D Views)	View Set (Object Embeddings)	View Sequence (Object Embeddings)	Similarity Measure
40-supervised	**VSE_s01**	Pre-trained EffcientNetB0	Attention		Cross entropy
	VSE_s02				Dot product
	VSE_s03	Finetuned EffcientNetB0	Average pooling		Cross entropy
	VSE_s04				Dot product
	VSQ_s05	EfficientNetB1		Rings 5, 1, 2 Multi-LSTM	Cross entropy
	VSQ_s06				Dot product
	VSQ_s07	ResNet50		7 View rings LSTM and Attention	Cross entropy
	VSQ_s08				Dot product
	Ensemble_s09	Ensemble 1,3,5,7 runs into one			
	Ensemble_s10	Ensemble 2,4,6,8 runs into one			

5.4. Submissions

They submit 4 runs with the View Set approaches (Runs 1-4), 4 runs with the View Sequences approaches (Runs 5-8), and 2 last runs with the ensemble technique (Runs 9-10). For the odd-number runs, they use cross entropy for final score evaluation. For the even-number runs, they use dot product to calculate the final scores. In all of the runs(Tab. 4), they use EfficientNet-B1 with frozen pre-trained weights for the 2D classification task.

- Run 1 and 2: They use EfficientNet-B0 with frozen pre-trained weights for view embedding and the attention mechanism.
- Run 3 and 4: They use EfficientNet-B0 with unfrozen pre-trained weights for view embedding, and the pooling mechanism.
- Run 5 and 6: They use EfficientNet-B1 with frozen pre-trained weights for view embedding, rings 5, 1, 2 in a transposed setting, and not following up LSTM with attention in sequence embedding.
- Run 7 and 8: They use ResNet50 with frozen pre-trained weights for view embedding, all 7 rings in a normal setting (no transposition), and following up LSTM with attention in sequence embedding.
- Run 9 and 10: They ensemble the 4 runs (1, 3, 5, and 7) or (2, 4, 6, and 8) into Run 9 and 10, respectively.

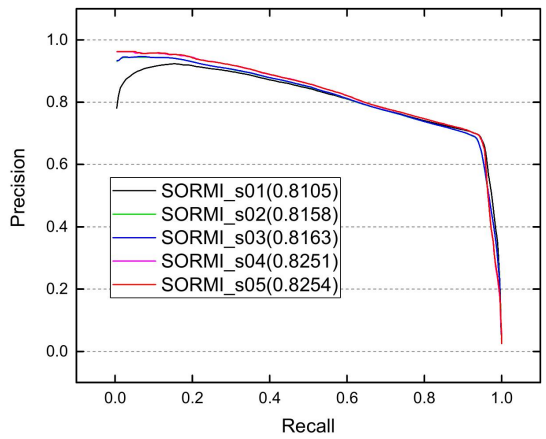

Figure 10: *PR-curve of the task for 21 categories.*

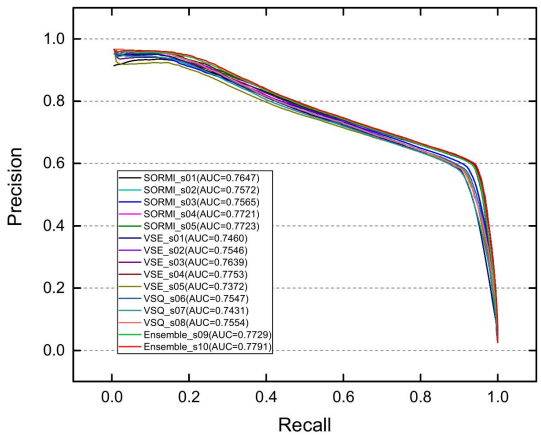

Figure 11: *PR-curve of the task for 40 categories.*

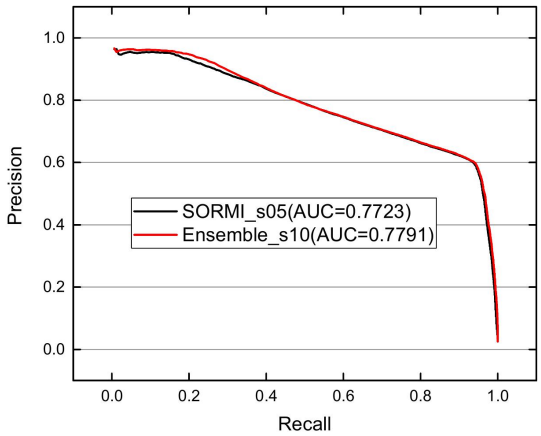

Figure 12: *Best PR-curve of the task for 40 categories.*

Table 5: *Evaluation score for 21 and 40 categories.*

Category	Method	NN	FT	ST	F-Measure	DCG	ANMRR	AUC
21	**SORMI_s01**	0.7806	0.9225	**0.9691**	0.1800	0.9224	0.0750	0.8105
	SORMI_s02	0.9311	0.9122	0.9610	0.1858	0.9228	0.0811	0.8158
	SORMI_03	0.9336	0.9126	0.9610	0.1857	0.9233	0.0807	0.8163
	SORMI_s04	0.9623	0.9255	0.9589	0.1885	0.9351	0.0685	0.8251
	SORMI_s05	**0.9628**	**0.9259**	0.9589	**0.1886**	**0.9355**	**0.0680**	**0.8254**
40	**SORMI_s01**	0.9140	0.8925	0.9608	0.2087	0.9062	0.0990	0.7647
	SORMI_s02	0.9524	0.8681	0.9475	0.2100	0.8912	0.1204	0.7572
	SORMI_s03	0.9526	0.8667	0.9459	0.2101	0.8905	0.1216	0.7565
	SORMI_s04	0.9640	0.8948	0.9550	0.2129	0.9120	0.0956	0.7721
	SORMI_s05	0.9643	0.8947	0.9548	0.2127	0.9120	0.0958	0.7723
	VSE_s01	**0.9670**	0.8551	0.9361	0.2121	0.8812	0.1310	0.7460
	VSE_s02	**0.9670**	0.8595	0.9407	0.2138	0.8859	0.1264	0.7546
	VSE_s03	**0.9670**	0.8829	0.9557	0.2099	0.8993	0.1073	0.7639
	VSE_s04	**0.9670**	0.8887	0.9585	0.2133	0.9101	0.1007	0.7753
	VSQ_s05	**0.9670**	0.8671	0.9416	0.2060	0.8833	0.1230	0.7372
	VSQ_s06	**0.9670**	0.8723	0.9455	0.2131	0.8947	0.1161	0.7547
	VSQ_s07	**0.9670**	0.8563	0.9404	0.2101	0.8804	0.1308	0.7431
	VSQ_s08	**0.9670**	0.8640	0.9455	0.2136	0.8899	0.1228	0.7554
	Ensemble_s09	**0.9670**	0.8935	0.9583	0.2134	0.9122	0.0963	0.7729
	Ensemble_s10	**0.9670**	**0.8998**	**0.9611**	**0.2144**	**0.9179**	**0.0904**	**0.7791**

6. Result

In this section, the evaluation results of all the method above are performed. The proposed methods use different network architectures to extract features for 2D images and 3D objects separately, and use the semantic feature (i.e., classification probability) for retrieval. Tab. 5 shows the evaluation scores of the supervised methods for the original 21 and extended 40 categories, which include results in terms of NN, FT, ST, F-Measure, DCG, ANMRR and AUC. The bold numbers indicate the best results among all teams in each task. From the result we can find that:

- For the supervised retrieval task on 21 categories, SORMI_s05 performs best. For the supervised retrieval task on 40 categories, Ensemble_s10 achieves the best performance. PR-curve of the supervised retrieval task on 21 and 40 categories are shown in Fig. 10 and Fig. 11(Fig. 12 shows the best results for two teams in Fig. 11), respectively. PR-curves of these submissions are close to each other, where SORMI_s05 performs best in 21-category task and Ensemble_s10 performs best in 40-category task.
- For the supervised retrieval task on 21 categories, there are 5 submissions of the SORMI method. The differences among these submissions are the similarity measurement strategies, and the number of top-k classes retained. It is obvious that the use of vector multiplication with normalized scores and remaining 8 classes bring the best performance.
- For the supervised retrieval task on 40 categories, the two teams submit 15 runs in total. Both the methods of the two teams train classifiers for two domains respectively, extract representations of the 2D images and views of 3D models, and predict classification using trained models. From the results of these submissions, specifically we have following observations: 1) For the best result of the two teams, HCMUS-Junior Team performs better than MAGUS.ZLW Team, and we assume it's due to the employment of the attention strategy in their methods. 2) For the submissions of HCMUS-Junior Team, NN values of all the runs are equal. VSE_s03 and VSE_s04 perform better in most evaluation criteria, which indicates that the finetune to the network can make progress on performance. 3) With the pre-trained base network, VSQ_05, VSQ_06, VSQ_07 and VSQ_08 have better performance than VSE_01 and VSE_02. It's mainly because VSQ exploits spatial relationship among the views of a 3D model. 4) Ensemble_s10 consistently outperforms Ensemble_s09, where the main difference is similarity measure. So it's clear that dot product measurement brings better performance. 5) Among 10 runs of HCMUS-Junior Team, Ensemble_s10 performs best. It is natural as ensemble technique combines the advantages of the strategies involved.
- For the SORMI method, which submits runs for both 21 and 40 categories, the results show that this method has good adaptability to the increase of categories.

7. Conclusion

The extended MI3DOR track of SHREC 2020 constructs a larger dataset, and two teams contribute three kinds of supervised methods and submit 20 runs of results. Both the teams contribute their creative work to monocular image based 3D object retrieval, and achieve satisfying retrieval results. The extended MI3DOR benchmark also has potentials for unsupervised learning task, which has not been discovered yet. While supervised methods usually utilize the classification task to boost the retrieval performance, unsupervised methods often put emphasis on the gap and try to narrow the gap in order to transfer the knowledge from one labeled domain to another unlabeled domain. Therefore, this direction still needs more attention from researchers. Our future work will focus more

on the unsupervised retrieval task where the 3D models are not labeled.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (61772359, 61872267, 61902277), the grant of Tianjin New Generation Artificial Intelligence Major Program (19ZXZNGX00110, 18ZXZNGX00150), the Open Project Program of the State Key Lab of CAD & CG, Zhejiang University (Grant No. A2005, A2012).

References

[Cho17] CHOLLET F.: Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1251–1258. 4

[CTSO03] CHEN D.-Y., TIAN X.-P., SHEN Y.-T., OUHYOUNG M.: On visual similarity based 3d model retrieval. In *Computer graphics forum* (2003), vol. 22, Wiley Online Library, pp. 223–232. 2

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2009), pp. 248–255. 2

[FZZ*18] FENG Y., ZHANG Z., ZHAO X., JI R., GAO Y.: Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 264–272. 3

[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. 4

[LLN*19] LI W., LIU A., NIE W., SONG D., LI Y., WANG W., XIANG S., ZHOU H., BUI N., CEN Y., CHEN Z., CHUNG-NGUYEN H., DIEP G., DO T., DOUBROVSKI E. L., DUONG A., GERAEDTS J. M. P., GUO H., HOANG T., LI Y., LIU X., LIU Z., LUU D., MA Y., NGUYEN V., NIE J., REN T., TRAN M., TRAN-NGUYEN S., TRAN M., VU-LE T., WANG C. C. L., WANG S., WU G., YANG C., YUAN M., ZHAI H., ZHANG A., ZHANG F., ZHAO S.: Monocular image based 3d model retrieval. In *12th Eurographics Workshop on 3D Object Retrieval, 3DORs* (2019), Eurographics Association, pp. 103–110. 1

[PTL*18] PHAM Q., TRAN M., LI W., XIANG S., ZHOU H., NIE W., LIU A., SU Y., TRAN M., BUI N., DO T., NINH T. V., LE T., DAO A., NGUYEN V., DO M. N., DUONG A. D., HUA B., YU L., NGUYEN D. T., YEUNG S.: RGB-D object-to-cad retrieval. In *Eurographics Workshop on 3D Object Retrieval, 3DOR 2018, 16 April 2018, Delft, The Netherlands* (2018), Telea A., Theoharis T., Veltkamp R. C., (Eds.), Eurographics Association, pp. 45–52. 4, 5

[SIVA17] SZEGEDY C., IOFFE S., VANHOUCKE V., ALEMI A. A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017). 3

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The princeton shape benchmark. In *Proceedings Shape Modeling Applications, 2004.* (2004), IEEE, pp. 167–178. 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. 2

[SYS*17] SAVVA M., YU F., SU H., KANEZAKI A., FURUYA T., OHBUCHI R., ZHOU Z., YU R., BAI S., BAI X., ET AL.: Large-scale 3d shape retrieval from shapenet core55: Shrec'17 track. In *Proceedings of the Workshop on 3D Object Retrieval* (2017), Eurographics Association, pp. 39–50. 2

[TL19] TAN M., LE Q. V.: Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946* (2019). 4

[WSK*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1912–1920. 2