Eurographics Workshop on 3D Object Retrieval (2019)
S. Biasotti and G. Lavoué (Editors)

SHREC 2019 - Monocular Image Based 3D Model Retrieval

Wenhui Li[1†], Anan Liu[*1†], Weizhi Nie[*1†], Dan Song[1†], Yuqian Li[1†], Weijie Wang[1†], Shu Xiang[1†], Heyu Zhou[1†]
Ngoc-Minh Bui[2], Yunchi Cen[3], Zenian Chen[3], Huy-Hoang Chung-Nguyen[2], Gia-Han Diep[2], Trong-Le Do[2], Eugeni L. Doubrovski[4], Anh-Duc Duong[5], Jo M.P. Geraedts[4], Haobin Guo[6], Trung-Hieu Hoang[2], Yichen Li[7], Xing Liu[9], Zishun Liu[4], Duc-Tuan Luu[2], Yunsheng Ma[10], Vinh-Tiep Nguyen[5], Jie Nie[11], Tongwei Ren[6], Mai-Khiem Tran[2], Son-Thanh Tran-Nguyen[2], Minh-Triet Tran[2], The-Anh Vu-Le[2], Charlie C.L. Wang[8], Shijie Wang[9], Gangshan Wu[6], Caifei Yang[9], Meng Yuan[11], Hao Zhai[7], Ao Zhang[6], Fan Zhang[3], Sicheng Zhao[10]

[1] Shool of Electrical and Information Engineering, Tianjin University, China.
[2] University of Science, VNU-HCM, Vietnam.
[3] State Key Laboratory of Virtual Reality Technology and System, Beihang University, China.
[4] Delft University of Technology, Netherlands.
[5] University of Information Technology, VNU-HCM, Vietnam.
[6] Nanjing University, China.
[7] SuoAo Technology Center, SAEE, University of Science and Technology Beijing, China.
[8] Chinese University of Hong Kong, China.
[9] School of Software, Dalian University of Technology, China.
[10] Department of Electrical Engineering and Computer Sciences, University of California Berkeley, USA
[11] Ocean University of China, China.

Abstract

Monocular image based 3D object retrieval is a novel and challenging research topic in the field of 3D object retrieval. Given a RGB image captured in real world, it aims to search for relevant 3D objects from a dataset. To advance this promising research, we organize this SHREC track and build the first monocular image based 3D object retrieval benchmark by collecting 2D images from ImageNet and 3D objects from popular 3D datasets such as NTU, PSB, ModelNet40 and ShapeNet. The benchmark contains classified 21,000 2D images and 7,690 3D objects of 21 categories. This track attracted 9 groups from 4 countries and the submission of 20 runs. To have a comprehensive comparison, 7 commonly-used retrieval performance metrics have been used to evaluate their retrieval performance. We wish this publicly available benchmark, comparative evaluation results and corresponding evaluation code, will further enrich and boost the research of monocular image based 3D object retrieval and its applications.

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

1. Introduction

As the rapid development of 3D technologies for modeling, reconstruction, printing an so on have produced increasing number of 3D models, 3D model retrieval becomes more and more important. Monocular image based 3D object retrieval (MI3DOR) aims to retrieve 3D object using a RGB image captured in real world. It helps users to get access to valuable 3D models by easily available 2D images, which is significant and promising.

However, few work focuses on MI3DOR with the following two reasons: (1) lack of related retrieval benchmarks, and (2) the gap between two modalities makes the problem extremely challenging.

The fundamental challenge in cross-modal retrieval lies in the heterogeneity of different modalities of data. In recent years, some efforts have been made to bridge the gap between different modalities and different domains, such as text-to-image retrieval and image-to-image domain adaption. SHREC18'IBR [ARYL*18] aims to search for relevant 3D scenes with 2D scene image, which is also a cross-modal retrieval task. Compared with SHREC18'IBR, this track has the following different aspects: (1) Different from collecting the scene images and models, we focus on individual object, which is useful for many applications related to 3D objects. (2) We contribute a dataset with more data and more categories, which makes the retrieval task based on this dataset more convincing.

In summary, the objective of this track is to retrieve 3D objects using 2D monocular image. Our collection is composed of 21,000

† Track organizer. * Corresponding Author Email: anan0422@gmail.com and weizhinie@tju.edu.cn.

2D images and 7,690 3D objects of 21 categories. 9 groups participated in this track and 20 runs were submitted. The evaluation results show a promising scenario about monocular image based 3D model retrieval methods, and reveal interesting insights in dealing with cross-modal data. The dataset will be made publicly available so as to enable rapid progress based on this promising technology.

2. MI3DOR Benchmark

2.1. Dataset and Queries

Our MI3DOR benchmark includes 21 classes for both 2D images and 3D objects. Table 1 shows the number of samples in training dataset and testing dataset. The exact number of images and objects for each class is shown in Fig. 1. We randomly selected 50% samples per class as the training set and used the remaining data for testing. We follow [SMKLM15] to render the 3D object (.OBJ) and get 12 views for each 3D object.

Table 1: *Training and testing datasets of our MI3DOR benchmark.*

Benchmark	Image	Model	View
Train	10500	3842	3842×12=46104
Test	10500	3848	3848×12=46176
Total	21000	7690	7690×12=92280

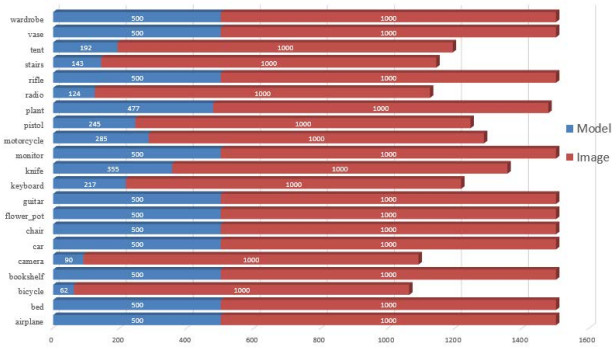

Figure 1: *Data distribution*

The 2D object-centered image dataset contains uniformly classified 21,000 images of 21 categories, which are all collected from ImageNet [DDS*09]. Figure 2 shows one example for each class in monocular image dataset.

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

The objects in 3D object dataset are selected from the popular 3D dataset NTU [CTSO03], PSB [SMKF04], ModelNet40 [WSK*15] and ShapeNet [SYS*17]. Figure 3 shows one example for each class in 3D object dataset. Besides 3D object (.OBJ) file, we render 3D model and get 12 views for each model.

Figure 3: *3D object examples in MI3DOR dataset.*

2.2. Evaluation

We adopt the evaluation criteria that have been widely employed in existing 3D object retrieval work, which are Precision-Recall (PR) diagram, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), F-Measure (F), Discounted Cumulated Gain (DCG) and Average Normalized Modified Retrieval Rank (ANMRR). A lower ANMRR value indicates a better performance.

3. Participants

9 groups participated in this track and 20 runs were submitted. The participant details and the corresponding contributors are shows as follows.

1. **RNF-MVCVR** submitted by HCMUS Team (Ngoc-Minh Bui, Trong-Le Do, Mai-Khiem Tran, Trung-Hieu Hoang, Minh-Triet Tran from University of Science, VNU-HCM and Vinh-Tiep Nguyen, Anh-Duc Duong from University of Information Technology, VNU-HCM).
2. **SORMI** submitted by MAGUS.ZinG Team (Ao Zhang, Haobin Guo, Tongwei Ren and Gangshan Wu from Nanjing University).
3. **RNFETL** submitted by Z. Liu, E.L. Doubrovski, J.M.P. Geraedts from Delft University of Technology and C.C.L.Wang from Chinese University of Hong Kong.
4. **CLA** submitted by HCMUS-Juniors Team. The-Anh Vu-Le, Huy-Hoang Chung-Nguyen, Gia-Han Diep, Duc-Tuan Luu, Son-Thanh Tran-Nguyen, Minh-Triet Tran from University of Science, VNU-HCM and Vinh-Tiep Nguyen from University of Information Technology, VNU-HCM.
5. **MLIS** submitted by Caifei Yang, Xing Liu, Shijie Wang from school of software, Dalian University of Technology.
6. **ADDA-MVCNN** and **SRN** submitted by Yunchi Cen, Fan Zhang and Zenian Chen from Beihang University.
7. **ALIGN** submitted by Hao Zhai and Yichen Li from University of Science and Technology Beijing.
8. **JGSA** submitted by Yunsheng Ma and Sicheng Zhao from UC Berkeley.
9. **JAN** submitted by Jie Nie and Meng Yuan from Ocean University of China.

4. Supervised Methods

4.1. RNF-MVCR, by HCMUS Team

4.1.1. 2D Query Image Classification with ResNet-based Fusion

For each query image, they use multiple classification models as follows: First, they use pretrained ResNet-50 [HZRS16] to get the feature vector. Then they use the feature vectors as inputs for multiple fully-connected neural networks that have one or two hidden layers. The number of hidden nodes of each layer is selected between 64, 100, 128 and 192, respectively. Then they use the majority voting scheme to select the best result from these models.

4.1.2. 3D Target Model Classification with Multi-Views and Circular ViewRings

They inherite the idea of Multi-view CNN to classify 3D objects based on the projection views. Different approaches have been tried to classify 3D objects based on 12 views that are provided by organizers and 26 views that participants render.

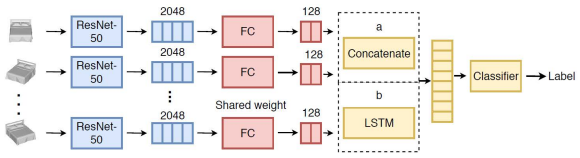

Figure 4: *View-based 3D classification.*

Figure 4 illustrates the architecture of this classification model. For each view, they use pre-trained ResNet-50 model to extract 2048-dimensional feature vector representing that view. Then, they use these vectors as inputs to a neural network.

Each view feature will be fed to a fully connected layer, whose weight is shared among views, to produce a 128-dimensional vector. Then these vectors are fused to form a unique vector to represent the 3D object. They adopt two different strategies to fuse the vectors. One approach is to simply concatenate these vectors to form a single 1536-dimensional vector. The second approach is to treat the input vectors as a sequence that has topological order and feed each view feature as an input at a time step to a recurrent neural network with LSTM cell, then they use the cell state of the last time step to represent the 3D object.

The output vector of one of these two strategies will be fed to a fully connected layer to produces a 21-dimensional output vector, where each element represent a class probability.

They also applied the method [PTL*18] in which each 3D target object is represented by multiple Circular View-rings, and tried different numbers of rings, ranging from 1 to 7 rings.

4.2. SORMI, by MAGUS.ZinG Team

Considering the high appearance diversity within each class of both monocular images and 3D models, this method proposes a **Semantic similarity based 3d Object Retrieval from Monocular Image (SORMI)** method. Figure 5 shows the framework of the method. They first extract the semantic representation of the query image and the retrieved 3D models respectively, and measure their semantic similarities to sort the 3D models. Specifically, they utilize Resnet-50 [HZRS16] or InceptionNet-v4 [SIVA17] to extract the semantic representation from monocular images, and MVCNN [SMKLM15] or 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 6 classes from the classeme vectors of both monocular images and 3D models to generate their semantic representations, and measure the similarity with cosine distance or vector multiplication.

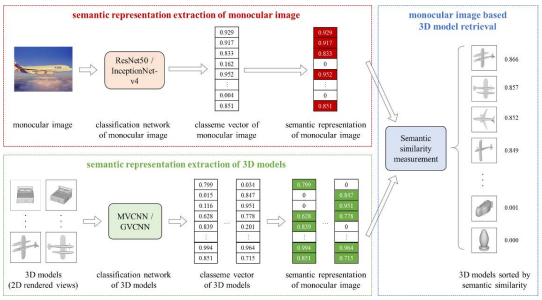

Figure 5: *The framework of the proposed method.*

As shown in Table 2, they provide five submissions with different image classification networks, 3D model classification networks and similarity measurement strategies. In all the submissions, they augmented the 2D rendered views of 3D models by capturing their new views, because of the lack of views in some specific classes such as bicycle and tent. After view augmentation, each class of 3D models has 250 view groups, and each view group contains 12 views.

Table 2: *Illustration of the five submssions*

Submission	Image Classification Network	3D Model Classification Network	Similarity Measurement
SORMI-1	Resnet-50	MVCNN	top5-cos
SORMI-2			top5-mul
SORMI-3			top6-mul
SORMI-4			top5-mul-norm
SORMI-5	InceptionNet-v4	GVCNN	top5-mul-norm

They utilize Resnet-50 and MVCNN as the networks for monocular image classification and 3D model classification, respectively. They also attempt to apply some recently proposed methods, such as InceptionNet-v4 and GVCNN for semantic representation extraction. These methods brought in slight improvements in 3D model classification, but might cause sorting failure sometimes in their experiments. Hence, they use Resnet-50 and MVCNN in four submissions and InceptionNet-v4 and GVCNN in one submission.

Additionally, they made attempt on representing monocular images and 3D models with the outputs of FC layer, and measuring the feature distances with the assistance of adversarial transfer method. However, the performance of all the attempts was significantly worse than the methods based on semantic similarity. Hence, they do not include such methods in the submissions.

4.3. RNFETL, by Z. Liu, E.L. Doubrovski, J.M.P. Geraedts, and C.C.L.Wang

They propose **ResNet-50 Feature Embedding with Triplet Loss (RNFETL)** for MI3DOR task. In this attempt, query images and rendered views of target 3D objects are all embedded into the same Euclidean space. The distance between a query image and a 3D object is measured according to the L_2 distances between the image and the object's views in the embedded feature space. Multi-layer neural networks are trained with triplet loss to learn the embedding.

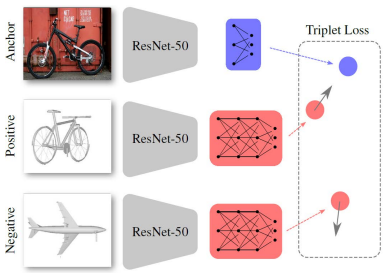

Figure 6: *Illustration of RNFETL.*

The adopted framework is illustrated in Fig. 6. Each training triplet contains an anchor monocular image u, a positive view v_p, and a negative view v_n. The positive view is rendered with a 3D model from the same category as the anchor image, while the negative one is rendered with a model from a different category. After sampling triplets, images and views are feed into the image network f_u and view network f_v respectively. Both networks map an input data instance to a k-dimensional vector. The networks are trained to minimize the loss

$$\sum_{(u,v_p,v_n)} [\| f_u(u) - f_v(v_p) \|_2^2 - \| f_u(u) - f_v(v_n) \|_2^2 + \alpha]_+ \qquad (1)$$

in which $\alpha > 0$ enforces a margin between the distance of feature vectors of different categories.

Instead of training from scratch, the networks f_u and f_v are both built on top of a pre-trained network. The ResNet-50 [HZRS16] trained on ImageNet dataset is used here. After removing the last softmax layer, ResNet-50 becomes a function f_{RN} that maps data to 2048-dimensional vectors. Taking advantage of its powerful data representation ability, $f_{RN}(x), x = u, v_p, v_n$ are used to train shallow embedding networks $\tilde{f}_u$ and $\tilde{f}_v$. That is to say, $f_x(\cdot) = \tilde{f}_x(f_{RN}(\cdot)), x = u, v$ is trained to minimize the loss in Eq. 1.

As there are 21 categories of query images, a classification network is trained to serve as $\tilde{f}_u$, of which the softmax output will be the embedeed feature vector. Thus the value of k will be 21. The image embedding network $\tilde{f}_u$ has only one fully-connected layer. It is trained with cross entropy loss, after which its parameters are fixed. Only $\tilde{f}_u$ is trained in the final training stage to minimize Eq. 1. The view embedding network $\tilde{f}_u$ has two fully-connected layers, each of which has 512 nodes. Activation functions in both layers are rectifiers. Dropout with 0.5 probability is used in the second layer. In Eqn. 1, the margin parameter a is set as 0.2.

In the retrieval stage, each query image is represented as a 21-d feature vector $f_u(u)$ and each target 3D model is represented as a collection of $f_v(v_i), i = 1, \cdots, N_v$, in which $N_v = 12$ is the number of rendered views for each shape, all of which are released by the organizers of this track. To measure the distance between the query image and the target shape, all N_v distances $\| f_u(u) - f_v(v_i) \|_2$ are collected as a set. The average of the smallest m values is used as the estimation of the image-to-model distance. The value of m is picked as 8 by a 5-fold cross-validation on the training set.

4.4. CLA, by HCMUS-Juniors

This approach mainly consists of two classifiers, one is for the 2D images and the other is for the 3D models. Each classifier can output the probabilities of the input image (or model) belonging to each of the 21 classes. The ResNet-18 (or ResNet-50) model, pre-trained on the ImageNet dataset, has been proven to achieve high results in classification of common objects [HZRS16]. Thus we choose this model to be our main feature extractor, which yields a 512-dimension (or 2048-dimension) representation vector for each input image.

4.4.1. 2D image classification

The classification model is a neural network with one hidden layer comprising of K nodes, with K varies from 100 to 1024. Stochastic gradient descent is used to minimize the categorical cross entropy loss function, together with techniques like learning rate scheduling or early stopping. The output score will be processed by a Softmax function, resulting in a 21-dimensional probability vector. It should be noted that some images can be classified into multiple classes (commonly up to two), with the probability as the deciding factor. Multiple networks are trained with varying values for K. The concluding probabilities of the image belonging to each of the classes are the weighted average of all networks' prediction.

4.4.2. 3D model classification

For 3D model classification, they analyze several approaches (described below) to obtain the final reliable labels for each 3D model.

Majority voting. For each 3D model, the 12 3D images captured from different angles of that model are fed into a Neural Network with 2 hidden layers. The output prediction of each image is calculated by a Softmax function which results in a vector with 21 elements that respectively represents probability of predicted labels. The output label of each image is taken from the element with the highest probability value. For labeling each object, the label which has the highest occurrence in the set of twelve view images of that object is the final label of that object.

Long Short Term Memory. With 12 images taken from different angles orbiting the object, it is natural to think of this as a time series with each angle as a time step. To classify the 3D model, they build a stacked Long Short Term Memory (LSTM) neural network with 2 LSTM layers. The gates of the first layer consists of 1024 nodes while the gates of the second layer consisting of half of that number. There is also a DropOut layer for regularization. The output of the last timestep will be fed into a densely connected neural network and processed by a Softmax function to produce a 21-dimension vector as probabilities for the models belong to each class. This neural network is trained with 3842 training data, each

with 12 time steps, each timestep is a 512-dimension representation vector (outputted by the ResNet model as stated before).

Data Augmentation using Rotation. As the images are captured from different but continuously changing views which orbit in a full circle around the 3D model, the order of the 12 images can be rotated to create different inputs, increasing the training data size by 144. They perform the rotation on the input vector with shape (3848, 1000) to create a vector of (46176, 12, 1000) including twelve elements with size (12, 12, 1000) each, for every 3D model. Then, each input will be trained with an LSTM model, examining varies parameters and layers to get the optimal result; Though, it is still significantly lower than our non-rotated results.

Commonly misidentified classes. Observing the training data as well as the validation result, it is clear that the objects of the three classes vase, flowerpot, and plant are easily mistaken as another. Our approach on this problem is to separate the objects of these three classes and label them as one large class. we then build two classifiers, one to classify the models into 1 of the 18 categories (the 17 classes except vase, flowerpot, and plant and 1 class for those three) and the other to classify between the three separated classes.

4.5. MLIS, by Caifei Yang, Xing Liu and Shijie Wang

This approach uses **Metric Learning for cross-domain 3D object retrieval in the Identical Subspace (MLIS)**, which try to bridge the gap between 3D objects and 2D images and learn the feature representation.

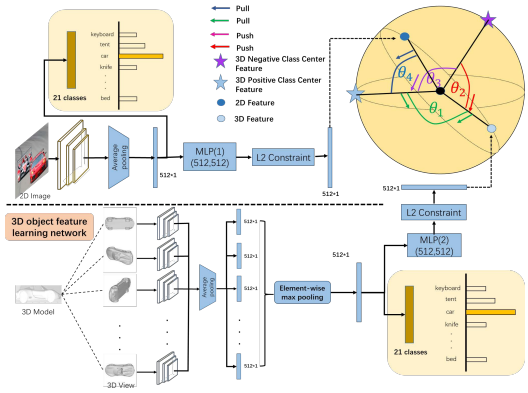

Figure 7: *Method Framework.*

The method framework is illustrated in Figure 7. This framework is divided into two parts. The first part (bottom part in the figure) is the feature learning for 3D objects, which is built upon the framework of MVCNN [SMKLM15]. They use the $L2$ norm to constrain the features of 3D objects on a unit hypersphere centered at the origin and propose a new metric learning loss, named **Triple Center Angular (TCA)** loss function. This loss function learns a center for each class and requires that the angles between features of 3D objects and centers from the same class relative to the origin are smaller than those from different classes, such that the features of 3D objects from same class are pulled closer to the corresponding center and pushed away from the other centers of different classes. In addition, this loss function requires that the angles of different class centers are larger relative to the origin, which makes the centers of different classes far away and increases the discriminability between different classes of features. They combine the TCA loss and cross entropy loss to train 3D object feature learning network and fix the center features of all classes of 3D objects after the joint loss function convergence.

The second part (top part in the figure) is to construct the relationship between the features of 2D images and 3D objects. In this part, they also use the $L2$ norm to constrain the features of 2D images to the same hypersphere as the features of 3D objects, which allows to learn the relationship between the features of 2D images and 3D objects. Since the distribution between the features of 2D images and 3D objects is extremely inconsistent, which makes the loss function difficult to optimize. In order to solve the addressed problem, they propose the **Center Angular (CA)** loss function. CA loss function requires that the angles between 2D images' features and fixed center features from the same class of 3D objects relative to the origin are as small as possible. This loss function only considers the relationship between the features of 2D images and 3D objects of the same classes, which is convenient for the optimization of CA loss function. They combine the CA loss and weakened cross entropy loss to train the relationship network.

4.6. ADDA-MVCNN, by Yunchi Cen, Fan Zhang and Zenian Chen

This method starts with learning the shape representations using two different CNN models. These two independent CNN streams are used to handle the two kinds of samples respectively, which is more powerful to extract features from two different domains. More importantly, they use an adversarial discriminative domain adaption approach to help solving the cross-domain problem. This approach couples the two input sources into the same target space, which allows to compare the similarities of cross-domain features direcly using a simple distance function.

4.6.1. Learning feature representations for image-based 3D shape retrieval

Recent deep learning has achieved great success on many computer vision tasks. With a deep structure, CNN can effectively learn complicated mappings from raw images to the target, which requires less domain knowledge compared with handcrafted features learning framework. Since the two input sources have distinctive intrinsic properties, we utilize ResNet-50 [HZRS16] pretrained on the ImageNet [DDS*09] as the initial network parameters. And MVCNN [SMKLM15] framework is adapted as another CNN, which takes 12 view images of a 3D model as input. The last fully connected layer of both networks are replaced by a 20 dimension embedding vector, and we fine-tune the two CNNs respectively.

4.6.2. Cross-domain matching using ADDA

If the correct mapping in each domain and cross-domain relations are learned, the two different feature domains may be correctly aligned in the feature space. After the cross domain mapping learning, matching can be performed cross domain. Fig.8 illustrates their

method for adversarial discriminative domain adaption. They treat the 2D images in the real world as the source domain and the 2D view images of the 3D models as the target domain. The basic idea of the adversarial learning is to carry on fine-turning the encoder parts that map the source and target data to the same space through a domain-adversarial loss. For detail, in the training stage, the encoder parts of the ResNet and MVCNN are connected to the same classifier and discriminator, where the discriminator is responsible for distinguishing source images from target images, while the encoder parts of the ResNet and MVCNN make efforts in fooling the discriminator and doing the correct classification. The parameters of the ResNet, MVCNN, discriminator are updated alternately. As a result, the feature distribution from the source domain and target domain are close to each other step by step.

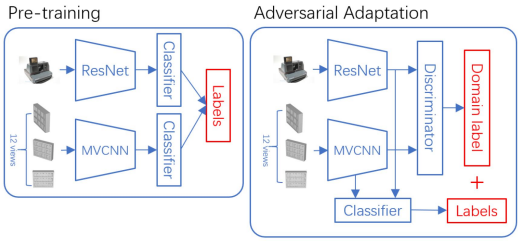

Figure 8: *Adversarial adaption of the method.*

4.7. SRN, by Yunchi Cen, Fan Zhang and Zenian Chen

This method proposes to use a weakly supervised metric learning method named **Siamese Ranking Network (SRN)** [BJ05] which consists of two identical sub-convolutional networks. The goal of the network is to make the output vectors similar if input pairs are labled similar, and dissimilar for the input pairs that are labeled as dissimilar. The output score will directly measure the similarities between 2D images and 2D view images.

4.7.1. Network Architecture

They adopt ResNet-50 [HZRS16] and MVCNN [SMKLM15] as the backbone of the Siamese network, and Fig.9 shows the architecture. The features extracted from ResNet and MVCNN are stitched together and further used to output a score indicating whether the two inputs belong to the same category or not. The loss function consists of three parts, two classification losses for each branch separately and a discriminating loss for the category consistency of the two inputs. All parameters in the network will be updated simultaneously for each back-propagation.

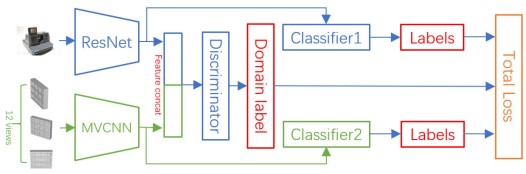

Figure 9: *Network architecture of the method.*

4.8. ALIGN, by Hao Zhai and Yichen Li

Considering the differences between two domains, this method uses different neural networks to conduct the basic information extraction individually and uses their fully connect layer output as a new training set for a teacher network. The teacher network is designed to minimize the output differences when the inputs of two domains belong to the same class, while maximizing the output differences when they belong to different classes.

4.8.1. Pre-train

They use DenseNet-201 to deal with 2D images, while multiple ResNet-18(s) is used to process 2D views of 3D model. In the training step, they split the train set as (train: verify = 3: 1). They use the pretrained parameters† for training.

4.8.2. Cross-domain Train

The design of alignment for cross-domain training is inspired by [AVT17]. They extract the feature layer (the layer just before the full-connected layers) as the data of cross-domain training. Before the teacher net, they introduce two networks for dimension reduction to 1-dimensional vector. Then the two streams are input to a teacher net with the same structure. Besides the conventional labels' cross-entropy loss, they additionally employ a ranking loss function to obtain both aligned and discriminative representations:$\sum_i^N \sum_{j \neq i} max\{0, \Delta - \psi(x_i, y_i) + \psi(x_i, y_j)\}$. Δ is a hyper-parameter, which is set to 0.02 with several trials. ψ is a cosine similarity function, which is $\psi(x,y) = \frac{x \cdot y}{\sqrt{x^2}\sqrt{y^2}}$.

5. Unsupervised Methods

There are two teams train the model without using the train set labels of target domains.

5.1. JGSA, by Yuesheng Ma and Sicheng Zhao

Traditional domain adaption methods consists of two successive steps: multi-view visual representation and cross-domain distance learning.

5.1.1. Multi-view visual representation

For the view-based methods, a 3D model is usually represented by a set of views captured from different directions. To transform each 3D model into a set of images, Phong reflection model is used to capture and render multiple views of 3D models. The 12 views of 3D model are inputted into AlexNet where they share identical architecture and the same parameters for feature extraction. Then they take element-wise maximum operation over 12 features into one, which acts as the final 3D model representation.

† https://download.pytorch.org/models/resnet18-5c106cde.pth and https://download.pytorch.org/models/densenet201-c1103571.pth

5.1.2. Cross-domain distance learning

Following [ZLO17], they learn two coupled projections to map the source and target data into respective subspaces. After the projections, 1) the variance of target domain data is maximized to preserve the target domain data properties, 2) the discriminative information of source data is preserved to effectively transfer the class information, 3) both the marginal and conditional distribution divergences between source and target domains are minimized to reduce the domain shift statistically, and 4) the divergence of two projections is constrained to be small to reduce domain shift geometrically.

This method does not require the strong assumption that a unified transformation can reduce the distribution shift while preserving the data properties. Different from subspace centric based methods, they not only reduce the shift of subspace geometries but also reduce the distribution shifts of two domains.

5.2. JAN, by Jie Nie and Meng Yuan

They well adapt the work [LZWJ17] by aligning the multiple layers distribution to address monocular image-based 3D model retrieval problem, where the query is a 2D monocular real image and the target is 3D object models. Those data come from different datasets with diverse data distribution and have different modalities. They address this task by rendering the 3D object information with multiple views.

5.2.1. Methodology

Given the source domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ of n_s labeled samples and a target domain $D_t = \{x_j^t\}_{j=1}^{n_t}$ of n_t unlabeled samples. The source domain and target domain are from different distributions $S(X^s, Y^s) \neq T(X^t, Y^t)$. They design a new network to reduce the shift in the joint distributions across domains by minimizing the source risk and domain discrepancy. The classic CNN classifier error is defined as follow:

$$\min_f \frac{1}{n} \sum_{i=1}^{n} J(f(x_i), y_i)) \tag{2}$$

where $J(\cdot,\cdot)$ is the cross-entropy loss function.

Based on the quantification study of [YCBL14], the convolutioinal layers can learn transferable generic features across domains. Considering the joint maximum mean discrepancy, they can integrate it over the domain-specific layers ζ into the CNN classifier error, the joint distributions are matched end-to-end with network training:

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(x_i), y_i)) + \lambda \hat{D_\zeta}(S, T,)) \tag{3}$$

where $\lambda > 0$ is a tradeoff parameter of the JMMD penalty and the $\hat{D_\zeta}(S,T)$ is computed as the squared distance between the empirical kernel mean embeddings.

5.2.2. Experiment setting

They use the Alexnet [SZ14] architecture finetuned on the Imagenet as the basic models and they utilize the two fully connection layers fc6 and fc7 as the layer set L to formally reduces the shifts in the joint distributions across domains. They use mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and the learning rate is 0.001.

6. Results

Table 3: *Evaluation Score of Supervised Methods.(Red indicates best performance and cyan blue indicates second-best performance.)*

Method	NN	FT	ST	F	DCG	ANM	AUC
RNF-MVCVR-1	0.974	0.921	0.928	0.202	0.935	0.069	0.855
RNF-MVCVR-2	0.974	0.921	0.937	0.200	0.935	0.069	0.850
RNF-MVCVR-3	0.974	0.922	0.937	0.200	0.936	0.069	0.850
RNF-MVCVR-4	0.974	0.918	0.937	0.200	0.933	0.072	0.846
SORMI-1	0.929	0.918	0.959	0.184	0.924	0.078	0.809
SORMI-2	0.945	0.917	0.959	0.186	0.925	0.078	0.812
SORMI-3	0.945	0.917	0.959	0.186	0.925	0.078	0.812
SORMI-4	0.945	0.907	0.936	0.180	0.913	0.091	0.782
SORMI-5	0.947	0.922	0.964	0.186	0.929	0.074	0.813
RNFETL	0.970	0.911	0.974	0.189	0.924	0.079	0.832
CLA-1	0.952	0.887	0.893	0.203	0.903	0.103	0.827
CLA-2	0.952	0.887	0.893	0.203	0.904	0.103	0.827
CLA-3	0.952	0.887	0.895	0.202	0.903	0.103	0.826
CLA-4	0.952	0.887	0.896	0.202	0.904	0.103	0.826
MLIS	0.942	0.910	0.963	0.186	0.919	0.084	0.815
ADDA-MVCNN	0.875	0.863	0.878	0.178	0.876	0.130	0.727
SRN	0.894	0.867	0.878	0.182	0.882	0.124	0.739
ALIGN	0.642	0.695	0.801	0.138	0.695	0.300	0.556

Table 4: *Evaluation Score of Unsupervised Methods*

Method	NN	FT	ST	F	DCG	ANM	AUC
JGSA	0.681	0.611	0.751	0.135	0.631	0.377	0.515
JAN	0.446	0.343	0.495	0.085	0.364	0.647	0.241

In this section, we perform a comparative evaluation of the proposed supervised methods and unsupervised methods in terms of PR-Curve, NN, FT, ST, F-Measure, DCG, ANMRR and AUC (the area under PR-curve). The evaluation scores of supervised methods and unsupervised methods are shown in Table 3 and 4 respectively. PR-curve of supervised and unsupervised methods is shown in Fig. 3 and 4 respectively.

The results have shown monocular image based 3D object retrieval performance using multiple views of 3D model from all the participants. From the results, we can have the following observations.

- All the methods use the multiple views to represent 3D models, which leverage the huge gap between 2D images and 3D point object and translate the cross modality retrieval to cross domain learning.
- Most of the supervised methods are classifier-based adaptation and get the excellent performance by joining the retrieval task with classification. By training the classifiers of two domains, they apply the trained model to predict the pseudo labels of test monocular images and 3D models and use the label to design the similarity measure.

- The unsupervised cross domain learning (without target labels) get lower performance comparing to the supervised methods. It is still a big challenge for real application.

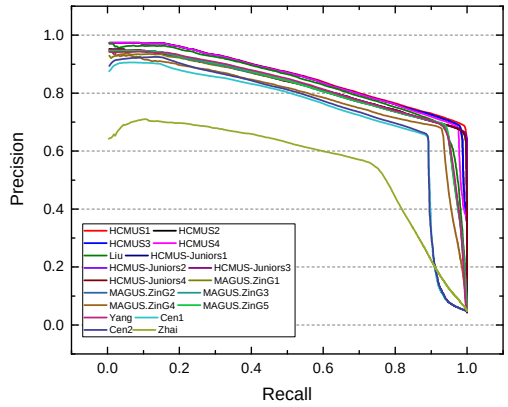

Figure 10: *PR-Curve of supervised methods*

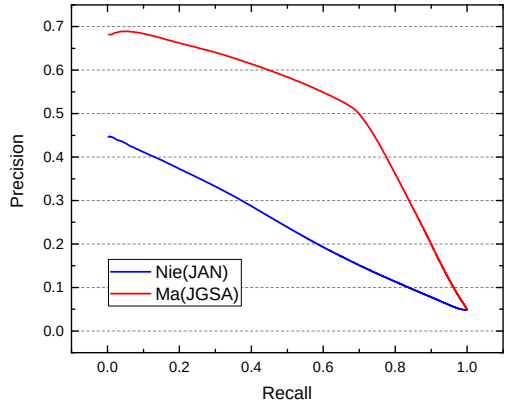

Figure 11: *PR-Curve of unsupervised methods*

7. Conclusion

In conclusion, this track has attracted research attention on 3D object retrieval using multimodal views. We have 9 groups who have successfully participated in the track and contributed 20 runs. This track serves as a platform to solicit existing monocular based 3D object retrieval methods. Also all the participated methods have achieved improved performance, the task is still challenging and the results are far from satisfactory and practical applications. There are still a long way for monocular image based 3D object retrieval.

8. Acknowledgements

The organizers are supported in part by the National Natural Science Foundation of China (61772359,61572356,61872267), the grant of Tianjin New Generation Artificial Intelligence Major Program (18ZXZNGX00150), the Open Project Program of the State Key Lab of CAD&CG (A1907), Zhejiang University, and the grant of Elite Scholar Program of Tianjin University (2019XRX-0035).

References

[ARYL*18] ABDUL-RASHID H., YUAN J., LI B., LU Y., BAI S., BAI X., BUI N.-M., DO M. N., ZHOU H., ZHOU Y., ET AL.: ShrecâĂŹ18 track: 2d image-based 3d scene retrieval. In *Proceedings of the Workshop on 3D Object Retrieval* (2018), Eurographics Association. 1

[AVT17] AYTAR Y., VONDRICK C., TORRALBA A.: See, hear, and read: Deep aligned representations. *arXiv preprint arXiv:1706.00932* (2017). 6

[BJ05] BOVENBERG A. L., JACOBS B.: Redistribution and education subsidies are siamese twins. *Journal of Public Economics 89*, 11-12 (2005). 6

[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, 5

[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. 3, 4, 5, 6

[LZWJ17] LONG M., ZHU H., WANG J., JORDAN M. I.: Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017* (2017), pp. 2208–2217. 7

[PTL*18] PHAM Q.-H., TRAN M.-K., LI W., XIANG S., ZHOU H., NIE W., LIU A., SU Y., TRAN M.-T., BUI N.-M., ET AL.: Rgb-d object-to-cad retrieval. In *Proceedings of the 11th Eurographics Workshop on 3D Object Retrieval* (2018), Eurographics Association, pp. 45–52. 3

[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, 3, 5, 6

[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

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014). 7

[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

[YCBL14] YOSINSKI J., CLUNE J., BENGIO Y., LIPSON H.: How transferable are features in deep neural networks? *CoRR abs/1411.1792* (2014). URL: http://arxiv.org/abs/1411.1792. 7

[ZLO17] ZHANG J., LI W., OGUNBONA P.: Joint geometrical and statistical alignment for visual domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* (2017), pp. 5150–5158. 7