Computer Vision and Image Understanding 201 (2020) 103070

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

A comparison of methods for 3D scene shape retrieval

Juefei Yuan [a], Hameed Abdul-Rashid [a], Bo Li [a,*], Yijuan Lu [b], Tobias Schreck [c], Song Bai [d], Xiang Bai [d], Ngoc-Minh Bui [e,j], Minh N. Do [f], Trong-Le Do [e], Anh-Duc Duong [e], Kai He [a], Xinwei He [d], Mike Holenderski [g], Dmitri Jarnikov [g,k], Tu-Khiem Le [e,l], Wenhui Li [h], Anan Liu [h], Xiaolong Liu [d], Vlado Menkovski [g], Khac-Tuan Nguyen [e], Thanh-An Nguyen [e], Vinh-Tiep Nguyen [e], Weizhi Nie [h], Van-Tu Ninh [e,l], Perez Rey [g,k], Yuting Su [h], Vinh Ton-That [e], Minh-Triet Tran [e], Tianyang Wang [i], Shu Xiang [h], Shandian Zhe [m], Heyu Zhou [h], Yang Zhou [d], Zhichao Zhou [d]

[a] School of Computing Sciences and Computer Engineering, University of Southern Mississippi, Long Beach, USA
[b] Department of Computer Science, Texas State University, San Marcos, USA
[c] Institute of Computer Graphics and Knowledge Visualization, Graz University of Technology, Graz, Austria
[d] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China
[e] Faculty of Information Technology, Vietnam National University - Ho Chi Minh City, Ho Chi Minh City, Vietnam
[f] Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, USA
[g] Department of Mathematics and Computer Science, Eindhoven, Eindhoven University of Technology, Netherlands
[h] School of Electrical and Information Engineering, Tianjin University, Tianjin, China
[i] Department of Computer Science & Information Technology, Austin Peay State University, Clarksville, USA
[j] Department of Computer Science, John Hopkins University, Baltimore, USA
[k] Prosus AI, Prosus, Amsterdam, The Netherlands
[l] School of Computing, Dublin City University, Dublin, Ireland
[m] School of Computing, University of Utah, Salt Lake City, USA

ARTICLE INFO

Communicated by Nicu Sebe

Keywords:
3D scenes
3D shape retrieval
Scene benchmark
Performance evaluation
Query-by-Sketch
Query-by-Image
Scene understanding
Scene semantics
SHREC

ABSTRACT

3D scene shape retrieval is a brand new but important research direction in content-based 3D shape retrieval. To promote this research area, two Shape Retrieval Contest (SHREC) tracks on 2D scene sketch-based and image-based 3D scene model retrieval have been organized by us in 2018 and 2019, respectively. In 2018, we built the first benchmark for each track which contains 2D and 3D scene data for ten (10) categories, while they share the same 3D scene target dataset. Four and five distinct 3D scene shape retrieval methods have competed with each other in these two contests, respectively. In 2019, to measure and compare the scalability performance of the participating and other promising Query-by-Sketch or Query-by-Image 3D scene shape retrieval methods, we built a much larger extended benchmark for each type of retrieval which has thirty (30) classes and organized two extended tracks. Again, two and three different 3D scene shape retrieval methods have contended in these two tracks, separately. To solicit state-of-the-art approaches, we perform a comprehensive comparison of all the above methods and an additional new retrieval methods by evaluating them on the two benchmarks. The benchmarks, evaluation results and tools are publicly available at our track websites (Yuan et al., 2019 [1]; Abdul-Rashid et al., 2019 [2]; Yuan et al., 2019 [3]; Abdul-Rashid et al., 2019 [4]), while code for the evaluated methods are also available: http://github.com/3DSceneRetrieval.

1. Introduction

Currently, there is a lot of research in 3D model retrieval, which usually targets the problem of retrieving a list of candidate 3D models using a single sketch, image, or model as input. 3D scene shape retrieval is a brand new research topic in the field of 3D object retrieval. Traditional 3D model retrieval ideally assumes that each query contains only a single object. However, 3D scene retrieval is a different and new type of 3D model retrieval which involves 2D/3D scenes comprising multiple objects that may overlap each other and also having spatial context configuration information. It is more challenging, but also has vast applications such as 3D scene reconstruction, autonomous driving cars,

* Correspondence to: 730 East Beach Blvd, Long Beach, MS 39560, United States of America.
E-mail address: bo.li@usm.edu (B. Li).

https://doi.org/10.1016/j.cviu.2020.103070
Received 3 November 2019; Received in revised form 19 June 2020; Accepted 18 August 2020
Available online 25 August 2020

3D geometry video retrieval, and 3D AR/VR entertainment. Therefore, this research topic deserves our further exploration.

Depending on the queries, 3D scene shape retrieval can be divided into three schemes: Query-by-Sketch, Query-by-Image, Query-by-Model. In this paper, we only cover the first two types of retrieval schemes.

Query-by-Sketch (Sketch-based) 3D scene shape retrieval is to retrieve relevant 3D scenes using a 2D scene sketch as input. It has the intuitiveness advantage over other two schemes and is also convenient for users to learn and retrieve 3D scenes. This retrieval scheme is also very promising and has great potential in many applications such as 3D scene reconstruction, 3D geometry video retrieval, virtual reality (VR) and augmented reality (AR) in 3D Entertainment like Disney World's Avatar Flight of Passage Ride (Wikipedia, 2019; Attractions, 2019; the Magic, 2019). However, although there are many existing 2D sketch-based 3D shape retrieval systems, there is little existing research work on 2D scene sketch-based 3D scene retrieval due to two major reasons: (1) It is challenging to collect a large-scale 3D scene dataset and there exists a very limited number of available 3D scene shape benchmarks; (2) Like 2D sketch-based 3D shape retrieval, there is a big semantic gap between the iconic representation of 2D scene sketches and the accurate 3D coordinate representations of 3D scenes. All of the above reasons make the task of retrieving 3D scene models using 2D scene sketch queries a challenging, although interesting and promising, research direction.

Query-by-Image (Image-based) 3D scene shape retrieval is an intuitive and convenient framework which allows users to learn, search, and utilize the retrieved results for related applications. For example, it can be applied in automatic 3D content generation based on one or a sequence of captured images for AR/VR applications. Other application scenarios include: autonomous driving cars, 3D movie, game and animation production, and robotic vision (i.e. path finding). In addition, we can also utilize it in developing consumer electronics apps, which facilitate users to efficiently generate a 3D scene after taking an image of a real scene. Last but not least, it is also very promising and has great potential in other related applications such as 3D geometry video retrieval, and highly capable autonomous vehicles like the Renault SYMBIOZ (Renault, 2019; Tips, 2019). However, there is little research in 2D scene image-based 3D scene shape retrieval (Merrell et al., 2011; Xu et al., 2016) due to at least two reasons: (1) the problem itself is challenging to cope with; (2) lack of related retrieval benchmarks. Seeing the benefit of advances in retrieving 3D scene models using 2D scene image queries makes the research direction meaningful, interesting and promising.

To promote the research on 3D scene shape retrieval, during the past two years (2018 and 2019), we have successfully organized four Shape Retrieval Contest (SHREC) tracks (Yuan et al., 2018; Abdul-Rashid et al., 2018; Yuan et al., 2019c; Abdul-Rashid et al., 2019) on the research topic of 3D scene retrieval: one for Query-by-Sketch and one for Query-by-Image during each year. In 2018, starting from a 2D scene sketch dataset named Scene250 (Ye et al., 2016) which consists of 250 2D scene sketches that are equally classified into 10 classes, we built the first 2D scene sketch-based 3D scene retrieval benchmark SceneSBR2018 by collecting 100 3D scene models for each class from 3D Warehouse (Trimble, 2018). Based on this benchmark, we organized the SHREC'18 2D scene sketch-based 3D scene retrieval track (Yuan et al., 2018). Considering the popularity of 2D scene images that also can be used as queries, we further collected 1000 2D scene images for each class as the new query dataset, and then still used the same 3D scene model target dataset that we already had in the SceneSBR2018 benchmark to curate the first 2D scene image-based 3D scene retrieval benchmark SceneIBR2018. Similarly, we organized another SHREC'18 track on 2D scene image-based 3D scene retrieval (Abdul-Rashid et al., 2018). We combine these two benchmarks SceneSBR2018 and SceneIBR2018 to form our *basic* 2D scene sketch/image-based 3D scene retrieval benchmark **Scene_SBR_IBR_2018**.

However, as can be seen, **Scene_SBR_IBR_2018** contains only 10 distinct scene classes, and this is also one of the reasons that all the three deep learning-based participating methods have achieved excellent performance on it. Considering this, after the track we have tripled (Yuan et al., 2019b) the size of **Scene_SBR_IBR_2018**, resulting in an *extended* benchmark **Scene_SBR_IBR_2019**, which has 750 2D scene sketches, 30,000 2D scene images, and 3000 3D scene models. Similarly, all the 2D scene sketches and images, as well as 3D scene models are equally classified into 30 classes. We have kept the same set of 2D scene sketches and images, and 3D scene models belonging to the initial 10 classes of **Scene_SBR_IBR_2018**. Based on the extended benchmark **Scene_SBR_IBR_2019**, in 2019 in a similar way we organized the SHREC'19 extended 2D scene sketch-based 3D scene retrieval (SceneSBR2019) track (Yuan et al., 2019c) and the SHREC'19 extended 2D scene image-based 3D scene retrieval (SceneIBR2019) track (Abdul-Rashid et al., 2019). Our main purpose for organizing these two tracks is to further advance this important but also challenging research area by soliciting the state-of-the-art retrieval methods for comparison, especially in terms of their scalability to a bigger and more challenging 3D scene retrieval dataset.

In the rest of the paper, we first review the related work (w.r.t. techniques and benchmarks) in Section 2. In Section 3, we introduce the motivation, building process, contents, and evaluation metrics of the two 3D scene retrieval benchmarks we built. A short and concise description of each contributed method (including an additional new method) is presented in Section 4. Section 5 describes the evaluation results of the six (6) Query-by-Sketch and eight (8) Query-by-Image 3D scene retrieval algorithms on the benchmarks. Section 6 concludes the paper and lists several future research directions.

2. Related work

2.1. Scene understanding: classification, object detection, semantic segmentation, and recognition

Naseer et al. (2018) conducted a survey on diverse indoor scene understanding tasks such as scene classification and reconstruction, semantic segmentation, object detection and pose estimation. They also reviewed related evaluation performance metrics for the above tasks, and proposed current challenges and open research problems that require further investigation.

Zhou et al. (2014) proposed a new large-scale scene image dataset which is 60 times bigger than the standard SUN (Xiao et al., 2016) dataset. They show that deep networks learned on object-centric datasets like ImageNet are not optimal for scene recognition, whereas training similar networks with a large amount of scene images substantially improves their performance.

Armeni et al. (2017) presented a large-scale indoor spaces dataset that provides a variety of mutually registered modalities from 2D, 2.5D and 3D domains, with instance-level semantic and geometric annotations, enabling the development of joint and cross-modal learning models and potentially unsupervised approaches utilizing the regularities existing in indoor spaces.

Chen et al. (2018) highlighted convolution with up-sampled filters, or "atrous convolution", as a powerful tool in dense prediction tasks, proposed Atrous Spatial Pyramid Pooling (ASPP) to robustly segment objects at multiple scales, and improved the localization of object boundaries by combining methods from DCNNs and probabilistic graphical models. Their DeepLab model combining the three innovations sets the new state-of-art performance at the PASCAL VOC-2012 semantic image segmentation task (Everingham et al., 2012).

2.2. Scene semantics

Semantics-based scene recognition and reconstruction. Oliva and Torralba (2001) first put forward the term of Spatial Envelope, defined by five features (naturalness, openness, roughness, expansion, and ruggedness) which are estimated via energy spectrum and coarsely localized information. Instead of relying on specific details about objects and segmentation information of regions, their computational model uses Spatial Envelope to recognize real world scenes. Xiang et al. (2016) built a large-scale dataset for 3D object recognition, named ObjectNet3D, which contains both images and 3D shapes and aligns objects in the 2D images with the 3D shapes. The dataset is therefore useful for recognizing the 3D location, pose, and shape of objects from 2D images, joint 2D detection and 3D object pose estimation, and image-based 3D shape retrieval. Zeng et al. (2017) proposed a data-driven RGB–D reconstruction model named 3DMatch which learns a local volumetric patch descriptor for establishing correspondences between partial 3D data. They also designed a self-supervised feature learning method that leverages corresponding labels found in existing RGB–D reconstructions. Mikolov et al. (2013) studied the quality of vector representations of words derived by various models on a collection of syntactic and semantic language tasks. Focusing on the distributed representations of words learned by neural networks, a new model architecture named *Word2Vec* was proposed for computing high-quality word vectors from huge data sets with billions of words. It is often used in 3D scene recognition (Chen et al., 2019) and scene parsing (Zhao et al., 2017).

Scene Parsing. Zhou et al. (2017, 2016) introduced the ADE20K dataset, covering a wide range of scenes and object categories with dense and detailed annotations for scene parsing. Moreover, a generic network design called Cascade Segmentation Module is proposed to enable the evaluated segmentation networks to parse a scene into stuffs, objects, and object parts in a cascade. Mohan (2014) proposed an end-to-end deep convolutional neural network architecture that employs multi-patch training to learn highly-hierarchical image structure and features for scene parsing. Armeni et al. (2016) proposed a semantic parsing method for the 3D point cloud of an indoor building using a hierarchical approach. It first parses the raw data into semantically meaningful spaces, and then breaks down the spaces into structural and building elements. Huang et al. (2018) proposed Holistic Scene Grammar (HSG) to represent the 3D scene structure based on a joint functional and geometric distribution of indoor scenes. They also devised a joint inference algorithm to estimate a holistic 3D configuration of a RGB indoor scene image based on a set of existing CAD models using a stochastic grammar model. Zhao et al. (2017) first presented a new task which parses scenes with a large and open vocabulary, and then came up with an approach to solve the task by jointly embedding image pixels and word concepts with the help of several evaluation metrics. Hung et al. (2017) presented a scene parsing method that utilizes global context information based on both parametric and non-parametric models. Their global context network is based on scene similarities and it performs favorably compared with previous methods that exploit only local relationship between objects.

Semantics-based 3D scene retrieval techniques. Hoàng et al. (2010) presented an image content description of the Triangular Spatial Relationships (Δ-TSR) between visual entities, which improves scene retrieval performance as well as execution time when evaluated on several datasets of city landmarks. Fisher et al. (2011) represented scenes as graphs that encode models and their semantic relationships, then defined kernels between the graphs that compare common virtual substructures and capture the similarity between corresponding scenes. It is shown that by incorporating structural relationships they have achieved better results in several scene modeling problems such as finding similar scenes, relevance feedback, and 3D model retrieval.

2.3. 2D/3D scene benchmarks

2.3.1. Xiao et al.'s SUN and SUN3D datasets (2010, 2016)

Xiao et al. (2010) built the Scene UNderstanding (SUN) image dataset for the purpose of fostering improvements in large scale scene recognition. SUN was initially comprised of 899 scene categories and 130,519 images. Later, SUN was extended to include 908 distinct classes (Xiao et al., 2016). Xiao et al. (2013) further created SUN3D, a RGB–D video dataset with camera pose information and object labels, to capture full-extend of 3D places. They used the videos for partial 3D reconstruction, propagated labels from one frame to another, and then used the labels to refine the partial reconstruction.

2.3.2. Silberman et al.'s NYU depth dataset V2 (2012)

Silberman et al. (2012) built a RGB–D indoor scene video dataset captured by the Microsoft Kinect. It comprises 1449 densely-annotated RGB–D images for 464 different scenes of three cities over 26 scene classes and 407,024 unlabeled frames.

2.3.3. Patterson et al.'s SUN attribute dataset (2012, 2014)

Patterson and Hays (2012) and Patterson et al. (2014) built the first large-scale scene attribute dataset, which contains 102 distinctive attributes for 14,340 images belonging to 707 scene categories. They found that scene attributes are helpful for many scene understanding tasks including classification, zero short learning, captioning, search, and parsing, while even the attribute features alone can achieve the state-of-the-art performance.

2.3.4. Lin et al.'s COCO dataset (2014) and Caesar et al.'s COCO-Stuff dataset (2018)

Lin et al. (2014) created a large-scale object detection, segmentation, and captioning dataset, named Common Objects in Context (COCO) dataset. It annotates the 80 object classes and 91 stuff classes existing in a collection of 328K images, containing 2.5M objects in total.

Based on COCO (Lin et al., 2014), Caesar et al. (2018) further annotated the stuff (background regions) in the images and built the COCO-Stuff dataset, which contains annotations of 91 stuff classes (e.g. grass, sky) based on superpixels.

2.3.5. Hua et al.'s SceneNN dataset (2016)

Hua et al. (2016) released SceneNN, a richly annotated RGB–D indoor scene dataset which contains 100 scenes annotated at the vertex, mesh and pixel level. This level of detail in annotation is in hopes of promoting research in various computer vision and scene understanding applications.

2.3.6. Xiang et al.'s ObjectNet3D dataset (2016)

Xiang et al. (2016) curated ObjectNet3D, a large 3D scene dataset across 100 categories. The dataset is comprised of 90,127 scene images, 201,888 objects within the scene images and 44,147 3D objects. ObjectNet3D aligns 2D images with 3D shapes, and provides 3D pose annotations and approximate 3D shape annotations. ObjectNet3D's goal is to provide annotations at a large scale comparable to that of recent 2D datasets.

2.3.7. Handa et al.'s SceneNet network and dataset (2016)

Handa et al. (2016) designed SceneNet, a framework that automatically generates much needed labeled training data for 3D scene understanding, such as synthetic 3D scenes as well as RGB–D videos with semantic annotations. SceneNet utilizes 57 hand created scenes across 5 indoor scene categories and leverages existing indoor scene annotations to find correlation and semantics between objects. Once relationships of the objects are extracted, SceneNet samples CAD repositories and constructs a new synthetic scene with annotations. Finally, they generated around 10k synthetic views for the five types of 3D scenes for different scene understanding experiments.

2.3.8. Song et al.'s SUNCG dataset (2017)

Song et al. (2017) constructed Scene UNderstanding Computer Graphics (SUNCG), a dataset of synthetic 3D scenes with manually labeled voxel occupancy and semantic labels. SUNCG has 45,622 different scenes and 2644 objects across 84 categories. They also developed the Semantic Scene Completion Network (SSCNet), an end-to-end 3D convolutional neural network, which uses a single depth image as input and produces semantic labels as well as a voxel occupancy grid. They trained SSCNet with SUNCG, and achieved state-of-the-art performance in both scene completion and semantic labeling.

2.3.9. Zhou et al.'s places dataset (2018)

Zhou et al. (2018) compiled Places, a dataset of 10,624,928 scene images across 434 scene categories. While Places is not annotated at the object level, it provides the most diverse scene composition as well as insights into solutions to scene understanding problems.

2.3.10. Zou et al.'s SketchyScene dataset (2018)

Zou et al. (2018) curated SketchyScene, a dataset with 29,000 scene-sketches, over 7000 pairs of scene templates and photos, and over 11,000 object sketches. Each scene is comprised of object-based semantic ground truth and instance mask. They also provided insights into the use of SketchyScene to explore potential methods trained to perform semantic segmentation as well as image retrieval, captioning and sketch coloring.

2.3.11. Gao et al.'s SketchyCOCO dataset (2020)

Gao et al. (2020) proposed to generate a full-scene image from a hand-drawn scene sketch. To evaluate their approach, they built SketchyCOCO which contains 14K+ pairs of scene images and sketches based on the COCO-Stuff dataset (Caesar et al., 2018). Their two-staged approach generates the foreground and background of an image separately. Therefore, SktechCOCO also includes 20K+ sets of foreground sketches, images and their edge maps, which span 14 classes; as well as 27K+ pairs of background sketches and images falling into 3 categories.

3. Benchmarks

In the SHREC'18 and SHREC'19 scene retrieval tracks (Yuan et al., 2018; Abdul-Rashid et al., 2018; Yuan et al., 2019c; Abdul-Rashid et al., 2019), we have built two sketch/image-based 3D scene retrieval benchmarks, featuring a basic and an extended benchmark, respectively. To make our presentation self-contained, we also define seven commonly-used performance evaluation metrics to evaluate retrieval algorithms.

3.1. Basic benchmark: SHREC'18 sketch/image-based 3D scene retrieval track benchmark **Scene_SBR_IBR_2018**

3.1.1. Overview

Our basic 2D Scene Sketch/Image-Based 3D scene Retrieval benchmark **Scene_SBR_IBR_2018** is publicly available (Yuan et al., 2019d; Abdul-Rashid et al., 2019a). It utilizes: (1) the 250 2D scene sketches in Scene250 (Ye et al., 2016) as its 2D scene sketch query dataset; (2) 10,000 2D scene images selected from ImageNet (Deng et al., 2009) as its 2D scene image query dataset; (3) 1000 SketchUp 3D scene models ("OBJ" and "SKP" format) as its 3D scene target dataset. All of the above three datasets have the same ten classes, and each of them contains the same number of 2D scene images (1000 per class), 2D scene images (25 per class), and 3D scene models (100 per class).

To facilitate learning-based retrieval, we randomly select 18 sketches, 700 images, and 70 models from each class for training and use the remaining 7 sketches, 300 images, and 30 models for testing, as indicated in Table 1. The SHREC'18 scene sketch/image track participants are required to submit results on the testing dataset if they use a learning-based approach. Otherwise, the retrieval results on the complete (250 sketches/10,000 images, and 1000 models) dataset are needed. To provide a complete reference for future users of our **Scene_SBR_IBR_2018** benchmark, we evaluate the participating algorithms on both the testing dataset (7 sketches/300 images, and 30 models per query) for learning-based approaches and the complete **Scene_SBR_IBR_2018** benchmark (25 sketches/1000 images and 100 models per class) for non-learning based approaches.

Table 1
Training and testing dataset information of our **Scene_SBR_IBR_2018** benchmark.

Datasets	Sketches	Images	Models
Training (per class)	18	700	70
Testing (per class)	7	300	30
Total (per class)	25	1000	100
Total (all 10 classes)	250	10,000	1000

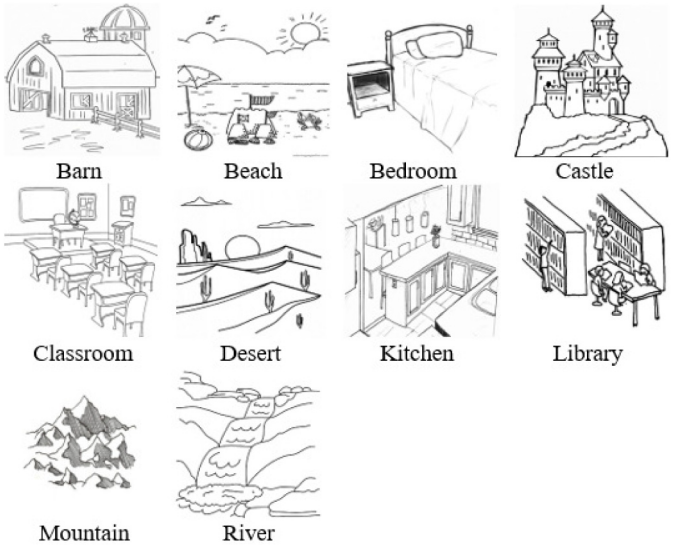


Fig. 1. 2D scene sketch query examples (one example per class) (Ye et al., 2016) in our **Scene_SBR_IBR_2018** benchmark.

3.1.2. 2D scene sketch query dataset

To facilitate Query-by-Sketch 3D scene retrieval, we built the 2D scene sketch query dataset comprising 250 2D scene sketches (10 classes, each with 25 sketches), while all the classes have relevant models in the 3D scene target dataset which are downloaded from 3D Warehouse (Trimble, 2018). One example per class is demonstrated in Fig. 1.

3.1.3. 2D scene image query dataset

Similarly, to facilitate Query-by-Image 3D scene retrieval, we created the 2D scene image query dataset which is composed of 10,000 scene images (10 classes, each with 1000 images) that are all from ImageNet (Deng et al., 2009). One example per class is demonstrated in Fig. 2.

3.1.4. 3D scene model target dataset

The 3D scene target dataset is built on the selected 1000 3D scene models downloaded from 3D Warehouse. Each class has 100 3D scene models. One example per class is shown in Fig. 3.

3.2. Extended benchmark: SHREC'19 sketch/image -based 3D scene retrieval track benchmark **Scene_SBR_IBR_2019**

3.2.1. Overview

To further promote the research of 3D scene retrieval, in 2019 we built a unified 3D scene benchmark supporting both sketch and image

Fig. 2. 2D scene image query examples (one example per class) in our **Scene_SBR_IBR_2018** benchmark.

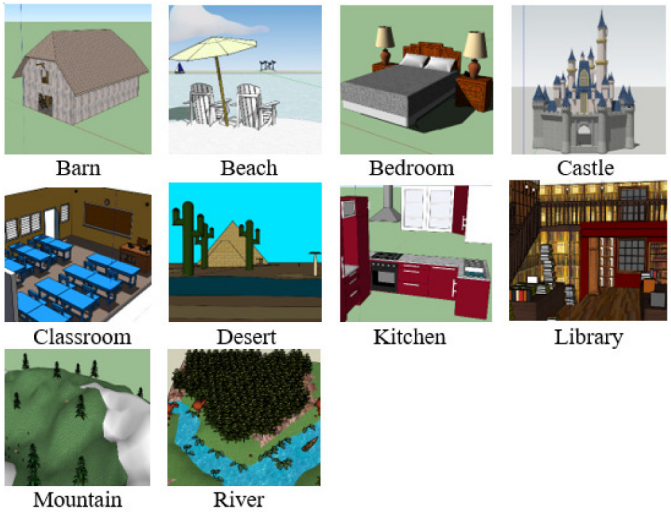


Fig. 3. 3D scene model target examples (one example per class) in our **Scene_SBR_IBR_2018** benchmark.

queries by substantially extending **Scene_SBR_IBR_2018** by means of identifying and consolidating the same number of sketches/images/models for another additional 20 classes from the most popular 2D/3D data resources. This work is the first to form a new and larger benchmark corpus for both sketch-based and image-based 3D scene retrieval. This benchmark provides an important resource for the community of 3D scene retrieval and will likely foster the development of practical sketch-based and image-based 3D scene retrieval applications.

3.2.2. Motivation

As mentioned in Section 3.1, to foster the research direction of sketch-based and image-based 3D scene retrieval, we built the first benchmark **Scene_SBR_IBR_2018** respectively and organized two related Shape Retrieval Contest (SHREC) tracks (Yuan et al., 2018; Abdul-Rashid et al., 2018). During the competitions, we found that both of these two benchmarks were not challenging and comprehensive enough since they cover only 10 distinctive categories. Considering this, we decided to further increase the comprehensiveness of the benchmarks by building a significantly larger and unified benchmark which supports both types of retrieval.

Table 2
Training and testing dataset information of our **Scene_SBR_IBR_2019** benchmark.

Datasets	Sketches	Images	Models
Training (per class)	18	700	70
Testing (per class)	7	300	30
Total (per class)	25	1000	100
Total (all 30 classes)	750	30,000	3000

3.2.3. Building process

By referring to several of the most popular 2D/3D scene datasets, such as Places (Zhou et al., 2018) and SUN (Xiao et al., 2010), we finally selected 30 scene classes (including the initial 10 classes in **Scene_SBR_IBR_2018**) based on the criteria of *popularity*, in terms of the degree to which they are commonly seen. Based on a voting mechanism among three people (two graduate student voters and a faculty moderator), the most popular 30 scene classes were selected from the 88 common scene labels in the Places88 dataset (Zhou et al., 2018). It is worth noting that the 88 scene categories are already shared by ImageNet (Deng et al., 2009), SUN (Xiao et al., 2016), and Places (Zhou et al., 2018). For the additional 20 classes' (sketches, images and models) data collection, we gathered their sketches and images from Flickr (2018) as well as Google Images (Google, 2018), and downloaded their SketchUp 3D scene models (in both the original ".SKP" format and our transformed ".OBJ" format) from 3D Warehouse (Trimble, 2018).

All of the above mentioned datasets (Places, SUN, ImageNet, Flickr, Google Images, and 3D Warehouse) are among the most popular sketch/image/model online repositories, whose data come from practical scenarios (i.e., captured by consumer cameras) or created by professionals who build 3D models for practical applications (i.e., people upload and share 3D models via 3D Warehouse). These design considerations are to make our datasets generalize to real applications.

3.2.4. Benchmark details

Our extended 3D scene retrieval benchmark **Scene_SBR_IBR_2019** is publicly available (Yuan et al., 2019a; Abdul-Rashid et al., 2019b). It expands the initial 10 classes of **Scene_SBR_IBR_2018** by adding 20 new classes to form a more comprehensive dataset of 30 classes. 500 more 2D scene sketches and 20,000 more images have been added to its 2D scene sketch and image query datasets respectively, and 2000 more SketchUp 3D scene models (".SKP" and ".OBJ" formats) to its 3D scene dataset. Each of the additional 20 classes has the same number of 2D scene sketches (25), 2D scene images (1000), and 3D scene models (100), as well. Therefore, **Scene_SBR_IBR_2019** contains a complete dataset of 750 2D scene sketches (25 per class), 30,000 2D scene images (1000 per class), and 3000 3D scene models (100 per class) across 30 scene categories. Examples for each class are demonstrated in Figs. 4, 5, and 6.

Similar to the **Scene_SBR_IBR_2018**, we randomly select 18 sketches, 700 images, and 70 models from each class for training and the remaining 7 sketches, 300 images, and 30 models are used for testing, as shown in Table 2. The participants are asked to submit results on the training and testing datasets, respectively, if they use a learning-based approach. Otherwise, the retrieval results based on the complete (750 sketch queries or 30,000 image queries, and 3000 scene model targets) dataset are needed.

3.3. Evaluation metrics

To conduct a solid evaluation of the sketch/image-based 3D scene retrieval algorithms based on our two scene retrieval benchmarks, we adopt seven performance evaluation metrics that are commonly used in information retrieval: Precision–Recall plot (PR), Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-Measures (E), Discounted Cumulated Gain (DCG) (Shilane et al., 2004) and Average Precision (AP) (Li and Johan, 2013). For users' convenience, we also have

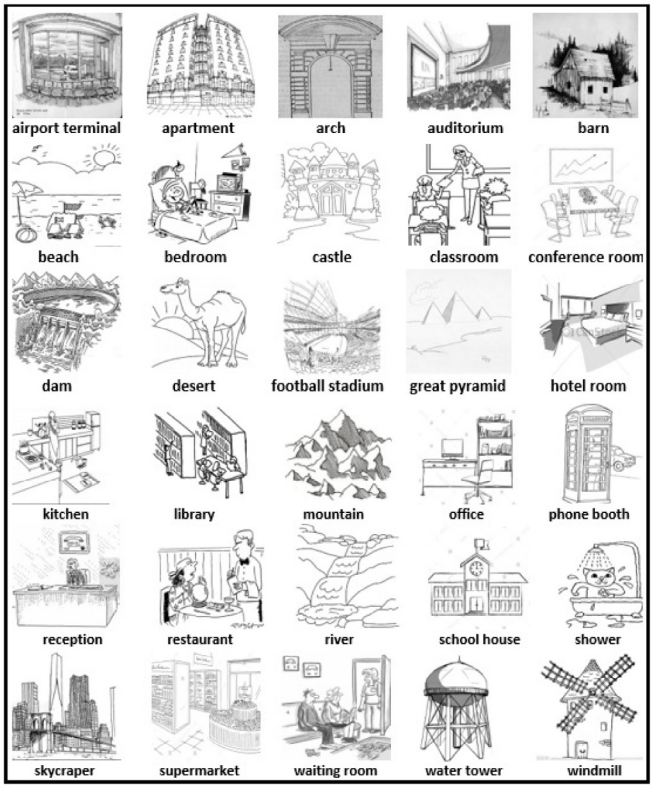


Fig. 4. 2D scene sketch query examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown.

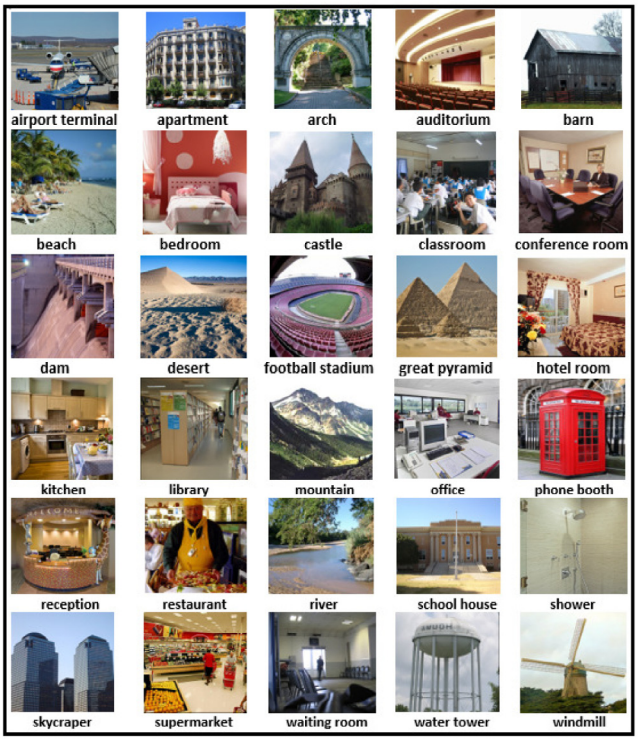


Fig. 5. 2D scene image query examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown.

developed an evaluation toolkit to compute them for each of the two benchmarks, and made them publicly available via the corresponding

Fig. 6. 3D scene model target examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown.

four tracks (Yuan et al., 2019d; Abdul-Rashid et al., 2019a; Yuan et al., 2019a; Abdul-Rashid et al., 2019b). For convenience and completeness, we explain the meaning and definition for each of the seven metrics below.

Here, we look at how to calculate the performance metrics for a sketch/image query q. We need to average over all the queries' performance to generate the performance of a 3D scene retrieval algorithm. Let us assume that in the 3D scene model target dataset of the benchmark, there are n models in total, where C models are relevant, that is, they have the same categorical label as the query q.

- **Precision–Recall plot (PR):** This curve plot (Recall is the horizontal axis, while Precision is the vertical one) measures the overall retrieval performance, thus it is one of the most important metrics to compare the general performance of different retrieval algorithms. Each point on the curve corresponds to a rank list R_K, while $1 \leq K \leq n$. The precision P value of the point is to measure the hitting accuracy of the retrieval list. For example, if there are H relevant models (hits) in the rank list, then the precision $P = \frac{H}{K}$. While, the recall R value of that point is to find out how much percentage of the relevant models in the whole dataset have been retrieved so far in that top K rank list, that is, $R = \frac{H}{C}$.
- **Nearest Neighbor (NN):** NN measures the precision (hitting accuracy) of the top 1 rank list.
- **First Tier (FT):** FT is the recall of the top C rank list.
- **Second Tier (ST):** ST is the recall of the top $2C$ rank list.
- **E-Measure (E):** Considering the importance of the first page of results, we use E-Measure to measure the overall performance of the top 32 returned models that can fit in that page: $E = \frac{2}{\frac{1}{P}+\frac{1}{R}}$.
- **Discounted Cumulated Gain (DCG):** Relevant models appear in different locations have different weights, thus DCG is created to measure the overall performance by accumulating the contributions of all the relevant models weighted by their ranking

positions. We first create a label vector G, where G_i=1 for a relevant model and G_i=0 for a irrelevant model. Then, DCG is defined as follows based on a logarithmic decay weighting factor,

$$DCG_i = \begin{cases} G_1 & i = 1 \\ DCG_{i-1} + \frac{G_i}{\lg_2 i} & \text{otherwise} \end{cases} \quad (1)$$

Finally, we normalize it by its optimum,

$$DCG = \frac{DCG_n}{1 + \sum_{j=2}^{C} \frac{1}{\lg_2 j}} \quad (2)$$

- **Average Precision (AP):** AP measures the overall performance as well since it combines both precision and recall. It averages all the precision values along the Precision–Recall plot. Therefore, it is equal to the total area under the Precision–Recall curve plot.

4. Methods

The first five authors of this paper built the above two benchmarks and organized the four SHREC'18/SHREC'19 tracks on the topics of sketch-based and image-based 3D scene model retrieval as well as this follow-up study. In total, the four tracks' participants contributed *twelve (12)* runs of *five (5)* different Query-by-Sketch and *eighteen (18)* runs of *seven (7)* distinctive Query-by-Image 3D scene retrieval algorithms. In addition, one run for each of the four tracks based on a newly introduced additional method named *DRF* (Section 4.1.6) has been incorporated in this paper; while one and two new runs of the TCL method (Section 4.1.2) are also provided here for the first time on the SHREC'19 sketch and image track respectively to evaluate its scalability performance. In this section, we introduce each Query-by-Sketch and Query-by-Image participating method in detail. However, except *BoW* (Section 4.2.1), other six Query-by-Image algorithms (i.e., *VGG* (Section 4.1.1), *MMD-VGG* (Section 4.1.1), *TCL* (Section 4.1.2), *VMV-VGG* (Section 4.1.5), *RNIRAP* (Sections 4.1.3~4.1.4), and *DRF* (Section 4.1.6)) are almost identical to their counterparts in the Query-by-Sketch category (*RNSRAP* for *RNIRAP*). Therefore, we merge their presentations only in Section 4.1 when we present the Query-by-Sketch methods. We also need to mention that each method has some parameter settings, which can be found in each method's description below.

To provide an even better overview of the *fourteen (14)* evaluated 3D model retrieval algorithms, we classify them in Table 3 based on the following taxonomy: type of feature (e.g., local or global), feature coding/matching methods (e.g., Direct Feature Matching (DFM), Bag-of-Words (BoW) or Bag-of-Features (BoF) framework, or Classification-Based Retrieval (CBR) framework), learning scheme (e.g., Domain Adaption (DA), Convolutional Neural Network (CNN), or Variational Autoencoder (VAE)), CNN model used for learning-based approaches, and semantic information (e.g., usage of classification or label information).

4.1. Query-by-sketch retrieval

4.1.1. MMD-VGG: Maximum mean discrepancy domain adaption on the VGG-Net, by W. Li, S. Xiang, H. Zhou, W. Nie, A. Liu, and Y. Su

Overview. They proposed the Maximum Mean Discrepancy domain adaption based on the VGG model (MMD-VGG) to address the scene sketch/image-based 3D scene retrieval problem, where the query is a 2D scene sketch/image and the targets are 3D scene models. Those two types of data come from different datasets with diverse data distribution. They address this task from two settings, learning-based setting and non-learning based setting. This method mainly contains two successive steps: data preprocessing and feature representation.

Data preprocessing. For 3D scene data, they use SketchUp, which is a very popular and easy-to-use 3D design software, to capture the representative views of all the 3D models automatically. The format of the input model is ".SKP" and the output of the model in SketchUp is a 480 ∗ 480 image. Several example representative views are shown in Fig. 7.

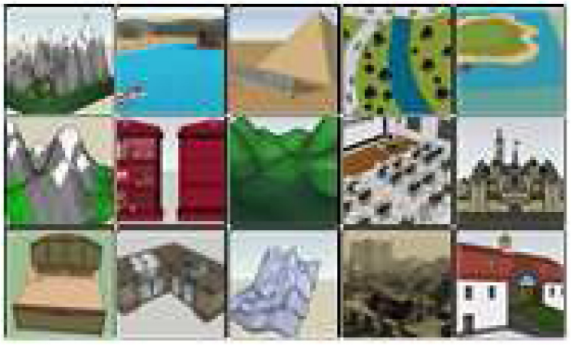


Fig. 7. Several example representative views.

Feature representation. After obtaining the representative views of all the 3D models, the 2D-to-3D retrieval task can be transformed into a 2D-to-2D retrieval task. For the feature representation, they use two settings: learning-based setting and non-learning based setting.

Learning-based setting. Inspired by the impressive performance of deep networks, they employ the VGG (Simonyan and Zisserman, 2014) model pretrained on the Places (Zhou et al., 2018) dataset as the initial network parameters. Then, they fine-tune the network on all the training sketches/images and all the representative views of training 3D models. Finally, they use the output of last but one fully connected layer (fc7) as the feature representation of each image.

It is obvious that the domain divergence between the targets and the query is quite huge. A scene sketch/image dataset and a 3D scene dataset can own different visual features even though when they depict the same category, which makes it difficult for cross-domain 3D model retrieval. Since the fine-tuning operation can only moderately reduce the divergence between these two datasets, they apply a domain adaption method to help to solve the cross-domain problem. In this algorithm, they aim to find a unified transformation which learns a new common space for features from two different domains. In detail, the nonparametric Maximum Mean Discrepancy (Long et al., 2013) is leveraged to measure the difference in both marginal and conditional distributions. Then, they unify it by Principal Component Analysis (PCA) to construct a feature representation which is robust and efficient for the domain shift reduction. After the domain adaptation, the features of two domains are projected into a common space. They measure the similarity between the query and target directly by computing their Euclidean distance.

Non-learning based setting. For non-learning based setting, they directly use the VGG (Simonyan and Zisserman, 2014) model pretrained on the Places dataset to extract the features of sketches/images/views. Then, they directly compute the Euclidean distances between the scene sketches/images and the representative views of the 3D scene models to measure their similarities.

4.1.2. TCL: Triplet center loss, by X. Liu, X. He, Z. Zhou, Y. Zhou, S. Bai , and X. Bai

Their method is based on a two-stream CNN which processes samples from either domain with a corresponding CNN stream. Based on triplet center loss (He et al., 2018) and softmax loss supervision, the network is trained to learn a unified feature embedding for each sample, which is then used for similarity measurement in the following retrieval procedure. Below is the detailed description of the method.

View rendering. Their approach exploits the view-based representations of 3D scene models. For each 3D scene model (with color

Table 3
Classification of the fourteen evaluated methods. Terms involved in the evaluated methods: (1) MMD: Maximum Mean Discrepancy; (2) TCL: Triplet Center Loss; (3) RNSRAP: ResNet50/ResNet18 based Sketch Recognition and Adapting Place classification; (4) VMV: View and Majority Vote; (5) BoW: Bag-of-Words; (6) RNIRAP: ResNet50/ResNet18 based Image Recognition and Adapting Place classification; (7) CVAE: Conditional Variational AutoEncoders; (8) DRF: Deep Random Field. When classifying Query-by-Sketch/Image methods, we refer to Li et al. (2014a) for "Feature type": local or global 2D feature. Two different retrieval frameworks: (1) DFM: Direct Feature Matching; (2) CBR: Classification-Based 3D model Retrieval framework. Learning schemes: (1) DA: Domain Adaption; (2) CNN: Convolutional Neural Network; (3) VAE: Variational Autoencoder. CNN model(s) used if it adopts a CNN-based learning scheme. "–" means not applicable.

Index	Evaluated method	Feature type	Feature coding/matching	Learning scheme	CNN model	Semantic information	Section	Reference(s)
Query-by-Sketch								
1	VGG	Local	DFM	No	VGG	No	4.1.1	Simonyan and Zisserman (2014)
2	MMD-VGG	Local	DFM	DA	VGG	No	4.1.1	Long et al. (2013) and Simonyan and Zisserman (2014)
3	TCL	Local	DFM	CNN	VGG, ResNet	No	4.1.2	He et al. (2018)
4	RNSRAP	Local	CBR	CNN	ResNet	Yes	4.1.3, 4.1.4	Zhou et al. (2018), Tzeng et al. (2017) and Ren et al. (2015)
5	VMV-AlexNet	Local	CBR	CNN	AlexNet	No	4.1.5	Yuan et al. (2019b)
6	VMV-VGG	Local	CBR	CNN	VGG	No	4.1.5	Yuan et al. (2019b)
7	DRF	Local	CBR	CNN	VGG	Yes	4.1.6	Yuan et al. (2020)
Query-by-Image								
8	VGG	Local	DFM	No	VGG	No	4.1.1	Simonyan and Zisserman (2014)
9	MMD-VGG	Local	DFM	DA	VGG		4.1.1	Long et al. (2013) and Simonyan and Zisserman (2014)
10	TCL	Local	DFM	CNN	VGG, ResNet	No	4.1.2	He et al. (2018)
11	VMV-VGG	Local	CBR	CNN	VGG	No	4.1.5	Yuan et al. (2019b)
12	BoW	Local	BoW	No	–	No	4.2.1	Nguyen et al. (2015) and Limberger et al. (2017)
13	RNIRAP	Local	CBR	CNN	ResNet	No	4.1.3, 4.1.4	Zhou et al. (2018), Tzeng et al. (2017) and Ren et al. (2015)
14	CVAE	Local	DFM	VAE	–	No	4.2.2	Kingma et al. (2014)
15	CVAE-VGG	Local	DFM	VAE	VGG	No	4.2.2	Kingma et al. (2014)
16	DRF	Local	CBR	CNN	VGG	Yes	4.1.6	Yuan et al. (2020)

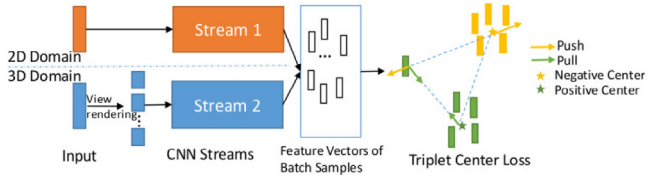


Fig. 8. Illustration of the network architecture. Two separate CNN streams are used to extract features for the two domains. Triplet center loss along with softmax loss (not depicted here) is used to optimize the whole network.

texture), they render it into multiple color images from N_v (N_v = 12 in their experiments) view directions. Each view image is of size 256 × 256. To fit the pre-defined CNNs during training, images of size 224 × 224 are randomly cropped as input from these rendered view images. While for testing, they only take the center crop of the same size from each view image.

Network architectures. An overview of the feature learning network is depicted in Fig. 8. Considering the huge semantic gap between images and 3D scene models, they adopt two separate CNN streams for samples from the two different domains. A normal CNN (Stream 1) is used to extract the features of sketches/images, while the MVCNN (Su et al., 2015) framework (Stream 2) is adopted to obtain features from the rendered view images. In their experiments, these two streams are based on the same backbone (e.g. VGG11-bn Simonyan and Zisserman, 2014). But note that their parameters are not shared. The last fully connected layer of each stream outputs a N_c-dimension embedding vector, where N_c is the number of categories.

Triplet Center Loss. In order to increase the discrimination of the features, they adopt triplet center loss (TCL) (He et al., 2018) for feature learning. Given a batch of training data with M samples, they define TCL as,

$$L_{tc} = \sum_{i=1}^{M} \max\left(D(f_i, c_{y_i}) + m - \min_{j \in C \setminus \{y_i\}} D(f_i, c_j), 0 \right) \tag{3}$$

where $D(\cdot)$ represents the squared Euclidean distance function. y_i and f_i are the ground-truth label and the embedding for sample i respectively. C is the label set. c_{y_i} (or c_j) is the center of embedding vectors for class c_{y_i} (or j). Intuitively, TCL is to enforce the distances between the samples and their corresponding center c_{y_i} (called *positive center*) smaller than the distances between the samples and their nearest *negative center* (i.e. centers of other classes $C \setminus \{y_i\}$) by a margin m. For a better performance, softmax loss is also employed.

Retrieval. In the testing stage, the two CNN streams are employed to extract the feature embeddings of both the 2D scene sketches/images and the 3D scene models, respectively. Euclidean distance is adopted as the distance metric to calculate the similarity matrix between the sketches/images and 3D scene models. To further improve the retrieval performance, an efficient re-ranking algorithm utilized in GIFT (Bai et al., 2016) is taken as a post-processing step. Three runs with different experimental settings are provided, they are, *Run1* with a single VGG11-bn model as the backbone network, *Run2* and *Run3* which are the ensemble results computed using different backbone models including VGG11-bn (Simonyan and Zisserman, 2014), ResNet50 (He et al., 2016) and ResNet101 (He et al., 2016) and different re-ranking parameter settings. Originally, only the results on the two SHREC'18 tracks are available. To evaluate TCL's scalability with respect to a larger dataset, the track organizers have implemented the TCL1's running on both SHREC'19 tracks as well as the TCL2's running on the SHREC'19 image track, with the help from this paper's co-author Tianyang Wang, first author Juefei Yuan, and the TCL method's authors. Therefore, we name it as a new group "Wang & Yuan & Liu", in short "WYL". Due to the unavailability of related code and limited time, the aforementioned re-ranking step is not included in the running.

4.1.3. RNSRAP/RNIRAP (SHREC'18 basic version): ResNet50/ResNet18 based sketch/image recognition and adapting place classification for 3D models using adversarial training, by M. Tran, T. Le, V. Ninh, K. Nguyen, N. Bui, V. Ton-That, T. Do, V. Nguyen, M. Do, and A. Duong

Except for the first step, the two methods RNSRAP and RNIRAP share other steps. Therefore, we only present their first steps separately.

Sketch recognition with ResNet50 encoding. In sketch classification task, the output of ResNet50 (He et al., 2016) is employed to encode a sketch into a feature vector of 2048 elements. Due to the extremely small-scale data in sketch data, it is difficult to use only the extracted features to train their neural network model directly, so they create variant samples by data augmentation. From the original training dataset, different variations of a sketch image can be generated. Regular transformations can be applied, including flipping, rotation, translation, and cropping. From the saliency map of an image, they extract different patches with their natural boundaries corresponding

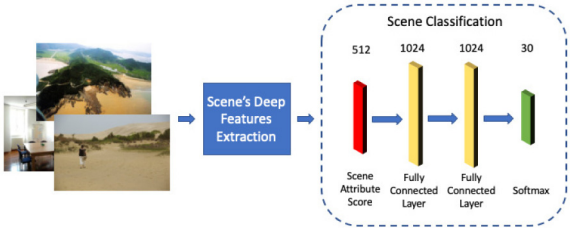


Fig. 9. 2D scene classification with scene attributes.

to different entities in the image and synthesize other sketch images by matting these patches. By this way, they enrich the training dataset with 2000 images.

Two types of fully connected neural networks are constructed. The first network type (Type 1) contains two hidden layers to train extracted feature vectors. The number of nodes in the first and second hidden layers are 256 and 128, respectively. The second network type (Type 2) uses only one hidden layer with 200 nodes. Extracted features from ResNet50 of all training sketch images, including the original and synthesized extra samples, are used to train different classification models conforming the two proposed neural network structures.

Owing to the small-scale training data, Batch Gradient Descent with Adam optimizer (Kingma and Ba, 2014) is used to minimize the cross entropy loss function in the training process. The output scores are processed through softmax function to provide proper predicted probability for each class.

They improve the performance and accuracy of their system by training multiple classification networks with different initializations for random variables for the two types of neural networks. They fuse the results of those models by using the majority-vote scheme to determine the label of a sketch image query.

They also improve the performance and accuracy of the retrieval system by training multiple classification networks with different numbers of nodes K in the hidden layer and different initializations for random variables. Finally, they obtain five classification models with the same structure and fuse the results of those models with the voting scheme to determine the label of a 2D scene image query.

An ASUS-Notebook SKU X541UV computer with Intel(R) Core(TM) i5-6198DU CPU @ 2.30 GHz, 8 GB Memory, and 1 x NVIDIA GeForce 920MX was used. The training time for a classification model is about 30 min. It takes less than 1 s to predict the category of a sketch image.

Scene image classification with ResNet18 encoding. A 2D scene image can be classified into one of the ten categories by using the scene attributes of that image, such as open area, indoor lighting, natural light, wood, etc. Thus, they employ the output of Places365-CNNs (Zhou et al., 2018) as the input feature vector for their neural network. They choose the ResNet18 model in the core of Place365 network and extract the scores of its scene attributes which yield a vector of 102 elements. By feeding the model with 7000 training 2D scene images, they obtain a training data with a dimension of 7000×102 used as the input vector for the 2D scene classification task.

The classification model is a fully connected neural network having one hidden layer with K nodes, $100 \leq K \leq 200$ (see Fig. 9). A training algorithm called Batch Gradient Descent with Adam optimizer (Kingma and Ba, 2014) is used to minimize the cross entropy loss function in training process. The output scores are processed through softmax function to provide the predicted probability for each class. It should be noticed that some query images may be classified into more than one categories. For example, some images contain a river but also has a mountain in the background. Thus, they assign up to two best predicted classes to each 2D scene image query.

To improve the performance and accuracy of the retrieval system, they train multiple classification networks with different numbers of nodes K in the hidden layer and different initializations for random variables. Finally, they obtain five classification models with the same structure and fuse the results of those models with the voting scheme to determine the label of a 2D scene image query. Using the same computer, it takes about one hour to train each classification model.

Saliency-based selection of 2D screenshots. For a 3D model, there exist multiple viewpoints to capture screenshots, some capture the general views of the model while others focus on a specific set of entities in the scene. They randomly generate multiple screenshots from different viewpoints at 3 different scales: general views, views on a set of entities, and views on a specific entity. Screenshots with many occlusions are removed. Then, they estimate the saliency map of a screenshot with DHSNet (Liu and Han, 2016) to evaluate if this view has sufficient human-oriented visually attracted details. By this way, they generate a set of visually information-rich screenshots for each 3D model. In this task, experimental results show that using no more than 5 appropriate views can be sufficient to classify the place of a 3D model with high accuracy.

Place classification adaptation for 3D models. Adversarial training is a promising approach for training robust deep neural network. Adversarial approaches are also possible to unsupervised domain adaptation (Tzeng et al., 2017; Sohn et al., 2017). They apply the adversarial adaptive method to minimize the distance between the source and target mapping distributions. This approach aims to create an efficient target mapping model due to substantial variance between the two domains.

In this approach, the source domain is a set of natural images that are used to train Places365-CNN models, while the target domain is a set of 3D place screenshots that are captured from given 3D models. Inspired by the idea of adversarial discriminative domain adaptation for face recognition (Tzeng et al., 2017), they propose their method to train the target mapping model so as to match the source distribution for place classification. Fig. 10 illustrates the overview of their proposed method to adapt a place classification system from natural images to screenshots of 3D models. They first train a target representation M_t to encode a screenshot of a 3D model into a feature vector that cannot be distinguished with the feature from a natural image by the domain discriminator. Then they train a classifier C that can correctly classify target images.

In the **Adversarial Adaptation** step, a natural image is encoded by a source representation M_s and a screenshot of a 3D model is encoded by a target representation M_t. The goal of this step is to learn M_t so that the discriminator cannot distinguish the domain of a feature vector encoded by either M_s or M_t. They keep the source representation M_s fixed and train the target representation M_t using a basic adversarial loss until the feature maps of the two domains are indistinguishable by the discriminator. By this way, they obtain a transformation to match the target distribution (screenshots from 3D models) with the source distribution (natural images).

In the **Classification for Target Domain** step, they use M_t to encode screenshots of 3D models and train a classifier with data from the training dataset. The label for a 3D model is determined by voting from the results of its selected screenshots with the coefficient weights corresponding to the prediction score of each view. To further boost the overall accuracy for place classification of 3D models from 2D screenshots, they train multiple classifiers with the same network structure and assemble the output results with voting scheme. They use Google cloud machines n1-highmem-2, each with 2 vCPUs, Intel(R) Xeon(R) CPU @ 2.50 GHz Intel Xeon E5 v2, 13 GB Memory, and 1 x NVIDIA Tesla K80.

Ranking generation. Because of the wide variation of sketch images, for each sketch image in the test set, they consider up to the two best labels of the sketch image, then retrieve all related 3D models (via their common labels), and finally sort all retrieved items (3D models) in ascending order of dissimilarity.

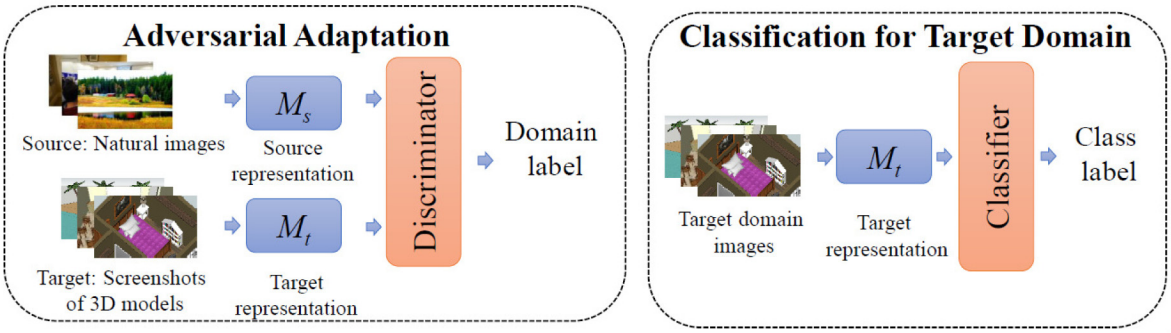


Fig. 10. Place classification for screenshots of 3D models with adversarial discriminative domain adaptation.

- Single-labeled sketch image: they select all the 3D models corresponding to the label of a sketch image and insert them into the rank list in a descending order of confidence scores measuring the possibility that a 3D model belongs to that category.
- Multi-labeled sketch image: the similarity score between a sketch image and a 3D model is determined by the product of the confidence score of the sketch image and that of the 3D model. All 3D models in the categories related to a sketch image are inserted into the rank list and sorted in descending order of similarity, i.e. ascending order of distance.

They submit 3 runs to the SHREC'18 sketch/image track.

- Run 1: they use the single label of a sketch/image from one network in Type 1 and the single label of a 3D model from one place classification model.
- Run 2: they use the single label of a sketch/image from the fusion of 3 networks (one Type 1 and two Type 2 networks) and the single label of a 3D model from the fusion of 5 place classification models.
- Run 3: they use the two best labels of a sketch/image from one network in Type 1 and the single label of a 3D model from the fusion of 5 place classification models.

4.1.4. RNSRAP/RNIRAP (SHREC'19 extended version): ResNet50-based sketch/image recognition with scene attributes and adapting place classification for 3D models using adversarial training, by N. Bui, T. Do, K. Nguyen, T. Nguyen, V. Nguyen, and M. Tran

Similarly, the two methods RNSRAP'19 and RNIRAP'19 share all the steps, except the first one. Therefore, we only present their first steps respectively.

Sketch image classification with data augmentation. They use data augmentation to enrich the training data for sketch recognition. They first collect a dataset of natural scene images from Google. They do not only crawl images with exactly 30 concepts in this track but also extend the list of concepts with semantically related concepts. For example, instead of searching only "desert" images, they expand the query terms into "desert", "camel", "cactus", etc. By using this query expansion strategy, they expect that their natural scene corpus can be used to link the gap of visual differences in the sketch image dataset.

The natural scene images are transformed into sketch-like images. For this track, they simply use automated tools for image transformation. However, they intend to use image translation to adapt images from the natural domain into the sketch-like domain.

For each image in the enriched dataset, they use ResNet-50 (He et al., 2016) to extract features and train a simple image classification network with 30 concepts.

2D scene image classification with scenes' deep features. To classify an image into one of the 30 scene categories in this track, they apply their method (used in SceneIBR2018, Section 4.1.3) to extract scenes' deep features using MIT Places API (Zhou et al., 2018). They train a simple network with the extracted features from Places API and use this network to classify an input image with 30 labels.

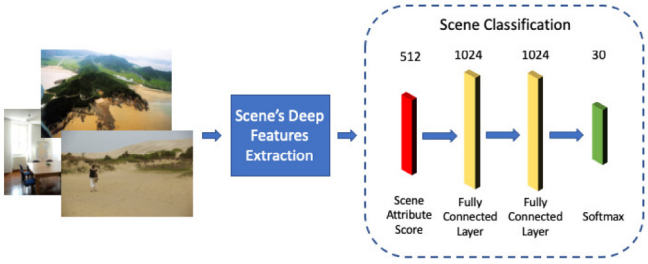


Fig. 11. 2D scene classification with scenes' deep features.

In their first step, an input image is represented as a feature vector in Places API domain vector space using a pre-trained ResNet-50 (He et al., 2016) model on the MIT Places API scene recognition network. Instead of using 102 scene attributes as in their previous SceneIBR2018 competition, they use a 512-dimensional deep feature representation which is generated before being processed through dense layers for classification.

Next, they utilize the extracted features to train a neural network classifier on the provided 30 scene categories. Different from their method used in the SceneIBR2018 track, the input feature is processed through two dense hidden layers with a dimension of 1024 for each layer, instead of a small network of $100 \leq K \leq 200$ dimensions as stated in their previous method. The visualization of their network configuration is demonstrated in Fig. 11. The network is trained on a server with 1 × NVIDIA Tesla K80 GPU. An Adam optimizer with learning rate at 0.0001 being hyperparameters. Three models were trained using this network configuration. The final label prediction of an image is outputted by using a majority voting scheme from these three models.

3D scene classification with multiple screenshots, domain adaptation, and concept augmentation. They perform a two-step process for 3D scene classification with multiple screenshots. The first step of their method is to use a number of classification models and domain adaptation to classify the 3D scene. The second step is to take advantage of visual concepts to refine the final result. The overview of the method is illustrated in Fig. 12.

In the first step, they train multiple classification models and use the voting scheme to ensemble the classification results. Because there are fair resemblances between 3D scene models and scenery images, they perform transfer learning from models pretrained on two datasets: ImageNet (Deng et al., 2009) and Places365 (Zhou et al., 2018).

The first model is to extract feature vectors for each image using ResNet-50 (He et al., 2016) pretrained on the ImageNet and Places365 datasets, respectively, then feed these feature vectors to a fully-connected neural network that has one to two hidden layers. The

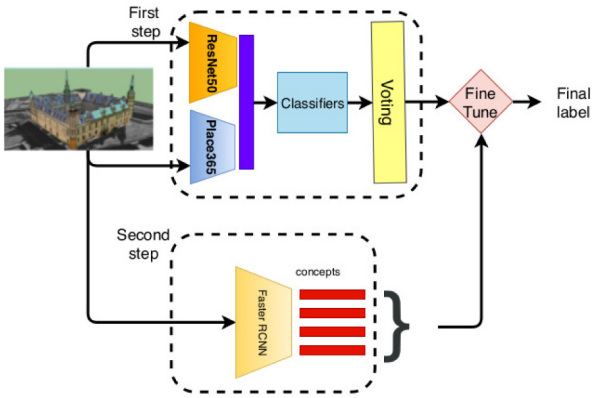


Fig. 12. Two-step process of the 3D scene classification method.

number of nodes in each hidden layers is set to 128, 192, 256, or 320 nodes and they choose the architecture that yields the highest classification accuracy to be the final result of this model.

They also extract 365-D scene attribute features for each image using Places365 and then concatenate them with the 2048-D feature vector of that image to form a 2413-D feature, which is later reduced to 512-D by PCA to train a third classification network. The extracted scene attributes may provide useful information, such as "outdoor", "natural light" , "trees" for a screenshot from a model in the "mountain" category. Concatenating two vectors' results in a higher dimensional input may make the model prone to overfitting. Therefore, each feature is normalized to have zero mean and unit variance and then they use PCA to reduce the size of the input vectors to 512-D.

Their second model is to collect real images of the 30 different categories from Places365 dataset and the Internet (for the "great_pyramid" class). They collect 1000 images per category. Then they use the weights of the last fully connected layer trained by this small-scale dataset to initialize the weights of the model when trained on the screenshot dataset.

Next, they apply their proposal of domain adaptation (used in SHREC 2018) (Yuan et al., 2019d, 2018) to classify a 2D screenshot of a 3D scene. Concretely, they first train an adversarial network to learn the representation of a 3D model to be close to the representation of a corresponding natural image. They treat the natural image domain as the source domain and the screenshots of the 3D model as the target domain. A discriminator is used to distinguish between the representations of the two domains. They train the target representation via an adversarial loss so that the two representations are indistinguishable to the discriminator. Then, using the adaptive representation of a 3D model, they train a number of simple networks. The predicted labels from the networks are assembled via voting to select the final label for the 3D model.

Because of the wide variation in the design of a 3D scene, it is not enough to classify the category of a scene simply by extracting the feature (from ResNet-50) or from the features of scene attributes (from Places365, even after domain adaptation). This motivates their proposal to employ object/entity detectors to identify entities related to certain concepts existing in a screenshot.

In the second step of the proposed method, they first collect a dataset of natural images from the Internet corresponding to the concepts that are related to the 30 scene categories. For example, they use the query terms such as "cactus" and "camel" to serve the scene classification for "desert". They train their set of object detectors from this dataset of natural images with Faster RCNN (Ren et al., 2015). Then they apply their detectors to identify entities that might appear in a scene, such as "book" (in a library), and "umbrella" (in a beach). By this way, they further refine their retrieval results.

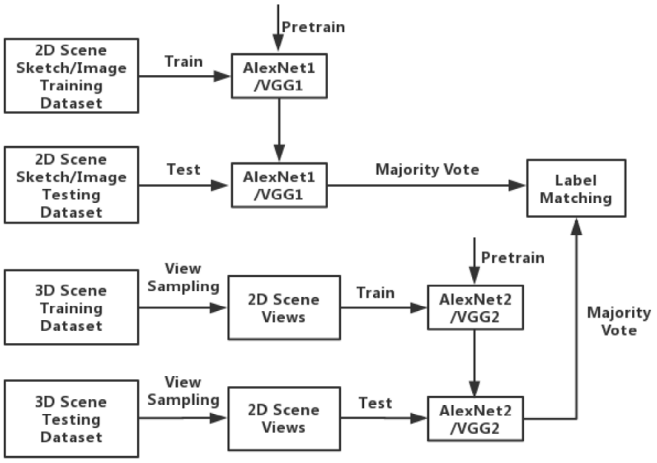


Fig. 13. VMV architecture (Yuan et al., 2019b).

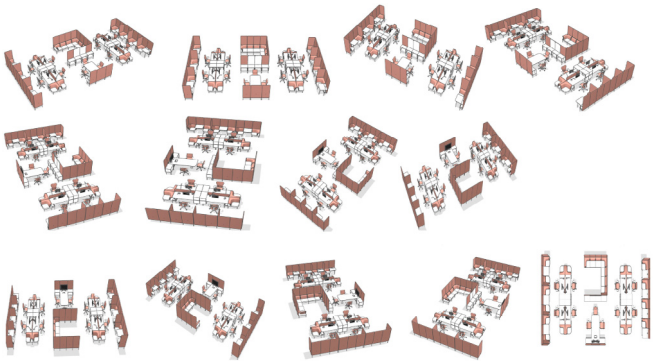


Fig. 14. A 13 sampled scene view images example of an apartment scene model (Yuan et al., 2019b).

4.1.5. VMV-AlexNet, VMV-VGG: View and majority vote based 3D scene retrieval algorithm, by J. Yuan, H. Abdul-Rashid, B. Li, T. Wang , and Y. Lu

They proposed a View and Majority Vote based 3D scene retrieval algorithm (VMV) (Yuan et al., 2019b) by either employing the AlexNet (for Query-by-Sketch only) or the VGG-16 model. Its architecture is illustrated in Fig. 13.

3D scene view sampling. For each 3D scene model, they center each 3D scene model in a 3D sphere. They develop a QMacro script program to automate the operations of the SketchUp software to perform the view sampling, and sample 13 scene view images automatically. They uniformly arrange 12 cameras on the equator of the bounding sphere of a 3D scene model, and one on the top of the sphere. One example is shown in Fig. 14.

Data augmentation. To avoid overfitting issues, before each pre-training or training, they employ data augmentation technique (rotations, shifts and flips) (Ye et al., 2016) to enlarge the related dataset's size by 500 times.

Pre-training and fine-tuning. They pre-train the AlexNet1/VGG1 model on the TU-Berlin sketch dataset (Eitz et al., 2012) for 500 epochs, and pre-train AlexNet2/VGG2 on the Places scene image dataset (Zhou et al., 2018) for 100 epochs. After pre-training, they fine-tune the AlexNet1/VGG1 on the 2D scene sketch/image training dataset, and fine-tune the AlexNet2/VGG2 on the 2D scene views training dataset, respectively.

Sketch/image/view classification and majority vote-based label matching. They obtain classification vectors by feeding well-trained AlexNet1/VGG1 with a 2D scene sketch/image query, or AlexNet2/VGG2 with the 2D scene views testing target dataset. Finally,

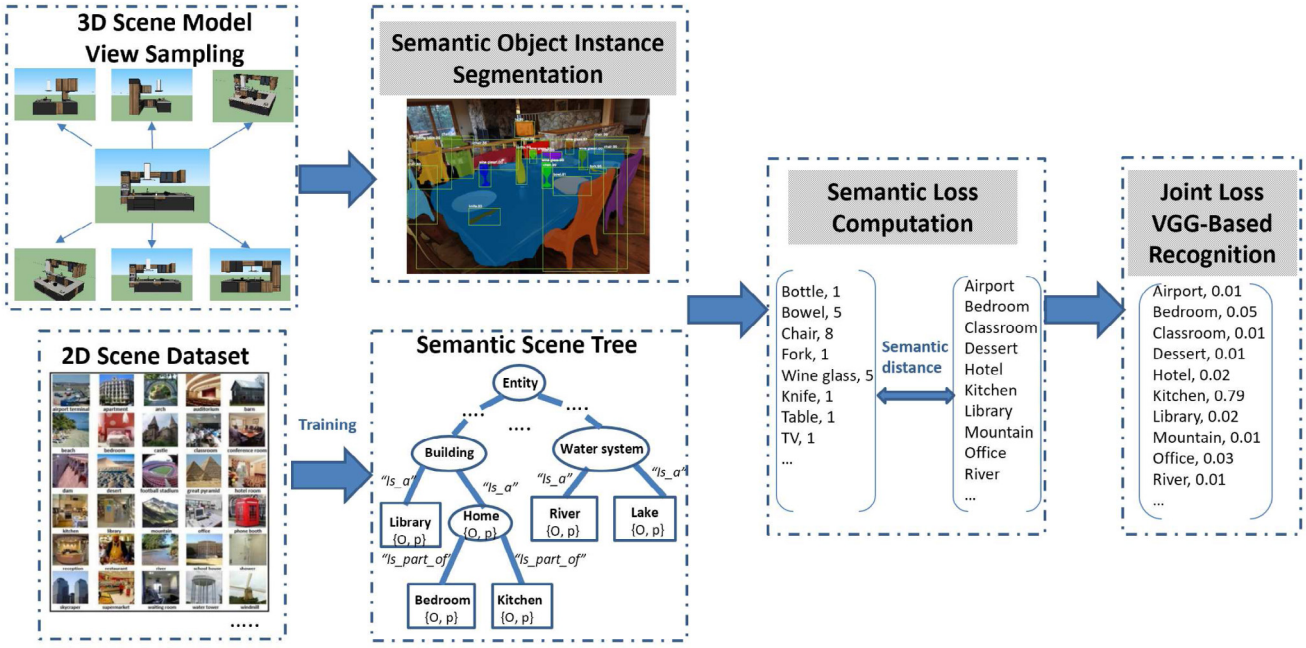


Fig. 15. Overview of the Deep Random Field (DRF) model for semantic tree-based 3D scene model recognition. Object occurrence distribution $\{O, p\}$ for a scene category consists of a list of pairs of an object class O and its occurrence probability p in that scene (Yuan et al., 2020).

based on the query's classification vector and a 3D scene target's 13 classification vectors, they generate the rank list for each sketch/image query by using a majority vote-based label matching method.

For more details, please refer to Yuan et al. (2019b), while the code is also publicly accessible.[1]

4.1.6. *DRF: Deep random field based semantic 3D scene retrieval algorithm, by J. Yuan, T. Wang, S. Zhe, Y. Lu, and B. Li*

This retrieval algorithm extends the semantic tree-based 3D scene model recognition model named Deep Random Field (DRF) proposed by Yuan et al. (2020), as illustrated in Fig. 15. The motivation of this retrieval algorithm is to utilize the semantic information existing in 2D scene images/sketches and 3D scene models to improve the retrieval performance. To organize such semantic information, we first build a Scene Semantic Tree (SST) based on the semantic ontology of WordNet (Miller, 1995) and its available hierarchical tree of semantically-related concepts. Then, an individual DRF model is trained respectively on the training query/target dataset of the basic and extended benchmark. Finally, a classification and majority vote-based matching which is similar to that of VMV (Section 4.1.5, last step) is applied to generate a rank list for a query.

DRF adopts the same multi-view convolutional neural network (MVCNN) based recognition framework as Su et al. (2015). However, besides the standard CNN-related loss, its loss function also includes a semantic information-based loss during the learning process, by utilizing the pre-constructed semantic scene tree. The DRF-based retrieval algorithm contains the following four steps.

(1) **Sampling 3D scene views**: a set of 13 color sample scene views are rendered for each 3D scene model by uniformly setting 12 cameras on the bounding sphere of the model with an elevation angle of 20 degrees, and 1 camera on the north pole.

(2) **Building a Scene Semantic Tree (SST)**: based on all the 2D/3D scene sketches/images/models available in the training query and target datasets, a Scene Semantic Tree (SST) is constructed to encodes the semantic class and attribute (i.e., scene object) information existing in the 2D/3D scene data. To build the tree, firstly, the YOLOv3 (Redmon and Farhadi, 2018) model is employed to detect the objects available in each scene sketch/image/view. One example for a kitchen view image and the related definition of object occurrence distribution can be found in Fig. 15.

(3) **Training a DRF query/target classification model respectively**: The VGG16 model is used, while its joint loss function of the DRF model is defined as follows,

$$\mathcal{L} = \lambda \mathcal{L}_{\text{DNN}} + (1-\lambda)\mathcal{L}_{\text{SST}}(\{P(O_i|S)\}, \{R_i * c_i\}), \tag{4}$$

where, $\mathcal{L}_{\text{DNN}}$ is the standard loss of a Deep Neural Network (DNN) classifier; $\mathcal{L}_{\text{SST}}$ is the Scene Semantics Tree-related semantic loss. λ is a hyperparameter, where $\lambda \in [0, 1]$. The object occurrence probability $\{P(O_i|S)\}$ is learned based on the corresponding training query/target dataset. It is the conditional probability that an object class O_i appears in a candidate scene S, and serves as the scene semantics information of S. R_i is the Lesk (Lesk, 1986)-based semantic relatedness between O_i and S. c_i is the number of occurrences of O_i detected by YOLOv3 in a training scene sketch/image/view. Both losses are scaled to [0, 1] before combination.

(4) **Sketch/image/view classification and majority vote-based label matching**: it is almost the same as the last step of Section 4.1.5. Please refer to it for more details.

For the original DRF related code, data, and experimental results, please refer to Yuan et al. (2020).

4.2. *Query-by-image retrieval*

4.2.1. *BoW: Bag-of-words framework based retrieval, M. Tran, V. Ninh, T. Le, K. Nguyen, V. Ton-that, N. Bui, T. Do, V. Nguyen, M. N. Do, and A. Duong*

The same participating group as that of Section 4.1.3 contributed another two runs based on the Bag-of-Words framework. In this approach, they do not train a model to classify a 2D scene image or a 3D model. Instead, their non-learning based method takes advantage of their framework on Bag-of-Word retrieval (Nguyen et al., 2015; Limberger et al., 2017) to determine the category of a 2D scene (query) and a 3D model (target). They also employ the same method to generate a set of useful views for each 3D model (see Section 4.1.3).

For both 2D scene images and 3D model views, they follow the same retrieval process. First, they apply the Hessian Affine detector to detect

[1] URL: http://orca.st.usm.edu/~bli/Scene_SBR_IBR/index.html.

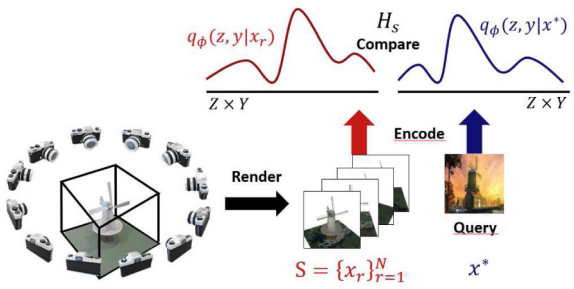


Fig. 16. Overview of scene sampling and CVAE distribution learning.

the interest points N in each image, either a 2D scene image or a 2D view of a 3D model. They use RootSIFT (Arandjelovic and Zisserman, 2012) without angle for keypoint descriptors and train the codebook using an approximate K-Means algorithm with 1 million codewords. They perform the quantization on all the training images with k-d tree data-structure to calculate the BoW representation of each image. They also perform soft assignment with 3 nearest neighbors, L2 asymmetric distance measurement (Zhu et al., 2013), TF–IDF weighting, and average pooling for each representation.

For each unlabeled 2D scene image, they retrieve a rank list of relevant images. Then they determine the top-M most voted labels from those of the retrieved images and assign these candidate labels to the input image. In this task, they choose $M = 1$ or 2. Similarly, they also determine the top-M most voted labels for each 2D view, then assign the most reliable label to the corresponding 3D model.

The codebook training module using Python 2.7 is deployed on a computer with a Ubuntu 14.04 OS and 2.4 GHz Intel Xeon E5-2620 v3 CPU, and 64 GB RAM. It takes 2 h to create a codebook with 1 million visual words from 15 million features. The retrieval process in Matlab R2012b with feature quantization and dissimilarity matrix calculation is performed on a computer with a Windows Server 2008 R2 OS, a 2.2 GHz Intel Xeon E5-2660 CPU, and 12 GB RAM. It takes less than 1 s to perform the retrieval for each image.

There are two runs in this method. In this first run, they determine only one label for each scene image and only one label for each 3D model. In the second one, they determine up to two labels for each scene image and up to two labels for each 3D model.

4.2.2. CVAE: Conditional variational autoencoders for image based scene retrieval, by P. Rey, M. Holenderski, D. Jarnikov, and V. Menkovski

Overview. Their proposed approach consists of image-to-image comparison with Conditional Variational AutoEncoders (CVAE) (Kingma et al., 2014), as shown in Fig. 16. The CVAE is a semi-supervised method for approximating the underlying generative model that produced a set of images and their corresponding class labels in terms of the so-called unobserved latent variables. Each of the input images is described in terms of a probability distribution over the latent variables and the classes.

Their approach consists of using the probability distributions calculated by the CVAE for each image as a descriptor. They compare an image query and the 3D scene renderings by using the distributions obtained from the CVAE. Their method consists of several steps: data pre-processing, training and retrieval described in the following subsections.

Data preprocessing. They obtain thirteen renderings for each of the 3D scenes. Twelve views are rendered at different angles around the scene as in Su et al. (2015) and one view is obtained from the 3D scene's predefined view once it is loaded into the SketchUp software. Their training dataset consists of these renderings together with the training images provided. All images are resized to a resolution of 64×64 with three color channels and all pixel values are normalized to the interval $[0, 1]$. Any image x is a part of the data space set $X = [0, 1]^{64\times64\times3}$. They have performed image data augmentation during training using a horizontal flip to all images.

Training. The CVAE consists of an encoder and a decoder neural network. The encoder network calculates from an input image $x \in X$ the parameters of a probability density $q_\phi(z|x)$ over the latent space $Z = \mathbb{R}^d$ and a density $q_\phi(y|x)$ over the thirty class values in $Y = \{1, 2, 3, \ldots, 30\}$ where ϕ represents the network parameters. On the other hand, the decoder network receives as an input a sampled latent variable $z \sim q_\phi(z|x)$ and a sampled class label $y \sim q_\phi(y|x)$ and returns a reconstruction of the original image x which is interpreted as the location parameter of a normal distribution over the data space X.

The distribution $q_\phi(z|x)$ is chosen to be a normal distribution over Z and $q_\phi(y|x)$ a categorical distribution over Y. The probabilistic model used corresponds to the M2 model described in the article (Kingma et al., 2014). Both the encoding and decoding neural networks are convolutional.

The CVAE is fed with batches of labeled images during training. The loss function is the sum of the negative Evidence Lower Bound (ELBO) and a classification loss. The ELBO is approximated by means of the parametrization trick described in Kingma et al. (2014) and Kingma and Welling (2013) and represents the variational inference objective. The classification loss for their encoding distributions over Y corresponds to the cross entropy between the probability distribution over Y with respect to the input label.

Retrieval. Each image $x \in X$ can be described as a conditional joint distribution over $Z \times Y$. Assuming that the latent variable z and the categorical value y for an image x are independent, this joint probability density corresponds to $q_\phi(z, y|x) = q_\phi(z|x)q_\phi(y|x)$.

The similarity D between an input image query $x^* \in X$ and a 3D scene represented by its N rendered images $S = \{x_r\}_{r=1}^N$ is given by the minimum symmetrized cross entropy H_s between the query and the rendered images' probability distributions (see Fig. 16).

$$D(x^*, S) = \min_{r\in\{1,2,\ldots,13\}} H_s(q_\phi(z|x^*), q_\phi(z|x_r)) + \alpha H_s(q_\phi(y|x^*), q_\phi(y|x_r)) \quad (5)$$

The parameter α corresponds to a weighting factor taking into account the importance of label matching. They have used a value of $\alpha = 64 \times 64 \times 3$. A ranking of 3D scenes is obtained for each query according to this similarity.

Five runs. They have sent five submissions corresponding to methods who differ only on the architecture of the encoding and decoding neural networks. These are described as follows:

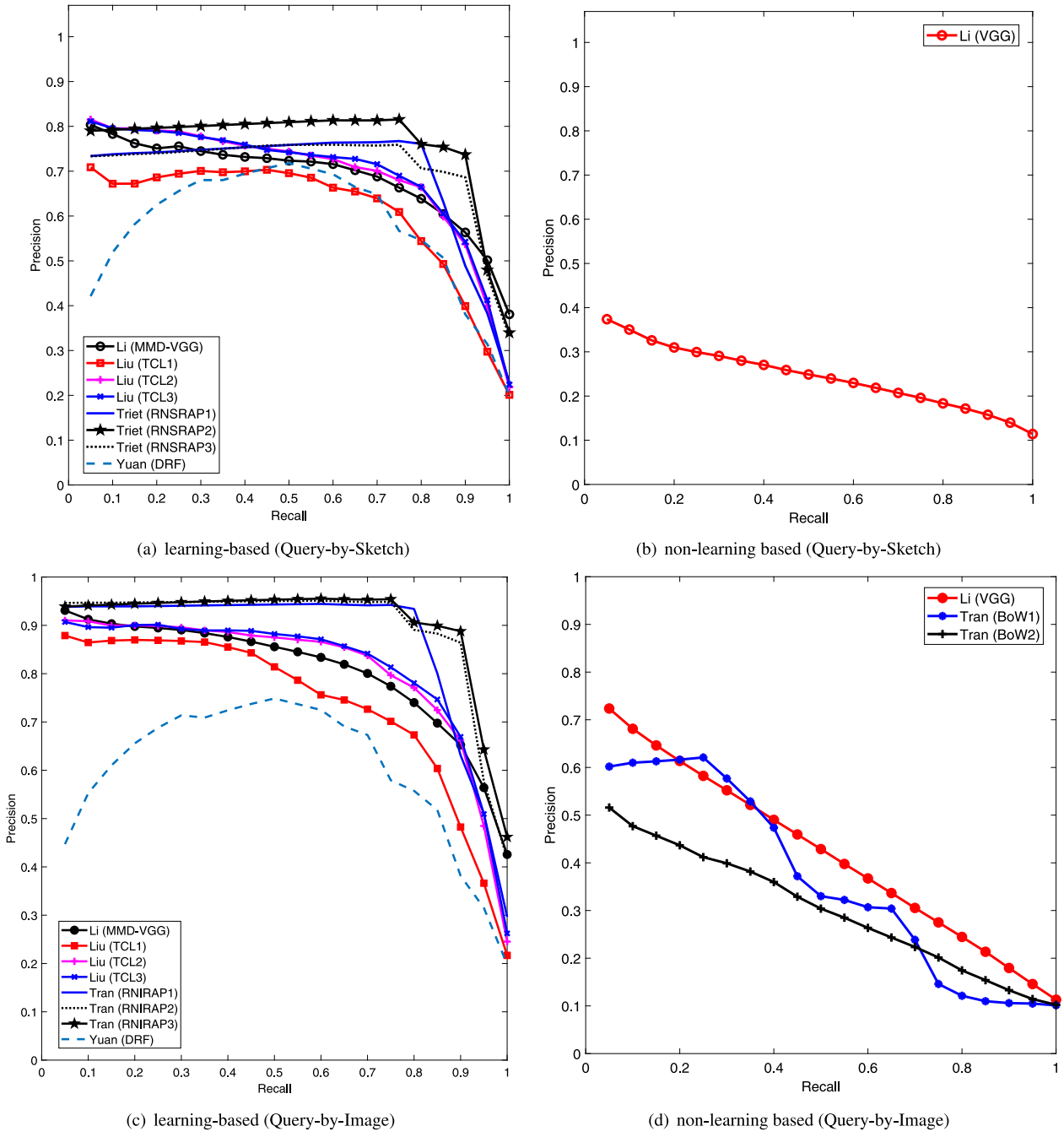
1. **CVAE-(1,2,3,4)**: CVAE with different CNN architectures for the encoder and decoder.
2. **CVAE-VGG**: CVAE with features from pre-trained VGG (Kalliatakis, 2017) on the Places data set (Zhou et al., 2018) as part of the encoder.

5. Results

For clarity, we conduct comparative evaluations with respect to the two different sketch/image-based 3D scene retrieval benchmarks that we have built. We measure retrieval performance based on the seven metrics described in Section 3.3: PR, NN, FT, ST, E, DCG and AP.

*5.1. **Scene_SBR_IBR_2018** benchmark*

Based on the our **Scene_SBR_IBR_2018** benchmark described in Section 3.1, we organized two SHREC'18 tracks on the topics of either 2D scene sketch or 2D scene image-based 3D scene retrieval, for which we refer to as SceneSBR2018 and SceneIBR2018. Fig. 17 and Table 4 compare the three learning-based and one non-learning based Query-by-Sketch retrieval methods submitted to SceneSBR2018, as well as the three learning-based and two non-learning based Query-by-Image retrieval methods submitted to SceneIBR2018, based on the corresponding testing and complete datasets of our **Scene_SBR_IBR_2018** benchmark. We also evaluate the newly contributed learning-based semantic approach DRF together with them.



(a) learning-based (Query-by-Sketch)

(b) non-learning based (Query-by-Sketch)

(c) learning-based (Query-by-Image)

(d) non-learning based (Query-by-Image)

Fig. 17. Query-by-Sketch and Query-by-Image Precision–Recall diagram performance comparisons on our **Scene_SBR_IBR_2018** benchmark.

5.1.1. Peer performance evaluation

Query-by-Sketch retrieval. We first perform a comparative evaluation of the eight runs of the four methods submitted to the SceneSBR2018 track by the three groups. As shown in the aforementioned figure and table, in the learning based category, Tran's RNSRAP algorithm (run 2) performs the best, followed by Liu's TCL method (run 3), while the overall performance of all the track participating learning-based methods are close to each other. We find that the performance of Yuan's DRF method is relatively lower, which should be due to the fact that it is an ongoing research approach and not optimized yet. In the non-learning based category, there is only one participating method, whose performance is much inferior if compared with learning-based ones. More details about the retrieval performance of each individual query of every participating method can be found on the SceneSBR2018 track homepage (Yuan et al., 2019d).

Though we cannot directly compare non-learning based approaches and learning-based approaches together, we have found much more promising results in learning-based approaches. The CNNs contribute a lot to the top performance of those three learning-based approaches. Considering many latest sketch-based 3D model retrieval methods utilize deep learning techniques, we regard it as the currently most popular and promising machine learning technique for 2D/3D feature learning and related retrieval. In fact, the three methods that adopt certain deep learning models also perform well when adapted to this challenging benchmark.

Finally, we classify all the SceneSBR2018 track participating methods with respect to the techniques employed: all the four participating groups (Li, Liu, Tran, Yuan) utilize local features. All of the four groups (Li, Liu, Tran, Yuan) employ deep learning framework to automatically learn the features. But Tran further applies regular transformations and

Table 4
Query-by-Sketch and Query-by-Image performance metrics comparison on our **Scene_SBR_IBR_2018** benchmark.

Participant	Method	NN	FT	ST	E	DCG	AP
Query-by-Sketch							
Learning-based methods							
Li	MMD-VGG	0.771	0.630	**0.835**	0.633	0.856	0.685
Liu	TCL1	0.643	0.582	0.753	0.579	0.810	0.606
	TCL2	**0.814**	0.630	0.794	0.626	0.860	0.688
	TCL3	0.800	0.640	0.801	0.633	0.861	0.691
Tran	RNSRAP1	0.729	0.658	0.659	0.637	0.826	0.689
	RNSRAP2	0.786	**0.729**	0.734	**0.707**	**0.864**	**0.757**
	RNSRAP3	0.729	0.652	0.766	0.637	0.834	0.707
Yuan	DRF	0.200	0.621	0.740	0.618	0.745	0.576
Non-learning based methods							
Li	VGG	0.336	0.262	0.428	0.151	0.684	0.243
Query-by-Image							
Learning-based methods							
Li	MMD-VGG	0.910	0.750	0.899	0.750	0.929	0.803
Liu	TCL1	0.823	0.689	0.856	0.687	0.900	0.733
	TCL2	0.871	0.751	0.888	0.759	0.927	0.803
	TCL3	0.864	0.760	0.893	0.762	0.927	0.809
Tran	RNIRAP1	0.864	0.760	0.893	0.762	0.927	0.809
	RNIRAP2	**0.944**	**0.882**	0.890	**0.854**	0.954	0.893
	RNIRAP3	0.936	0.875	**0.941**	0.850	**0.958**	**0.902**
Yuan	DRF	0.203	0.547	0.767	0.645	0.762	0.598
Non-learning based methods							
Li	VGG	**0.719**	**0.416**	**0.585**	**0.291**	**0.803**	**0.414**
Tran	BoW1	0.575	0.316	0.396	0.272	0.735	0.360
	BoW2	0.501	0.311	0.469	0.196	0.719	0.298

adversarial training, while Yuan utilizes available semantic information as well. On the other hand, Li and Liu directly compute the 2D–3D distances based on the distributions of sketches and models by using the Euclidean distance metric, while Tran and Yuan conduct the retrieval based on 2D/3D classification.

Query-by-Image retrieval. Similarly, we perform a comparative evaluation of the ten runs of the five methods submitted to SceneIBR2018 track by the three groups, together with one run from the new method DRF. As shown in the aforementioned figure and table, in the learning-based category, Tran's RNIRAP algorithm (run 3) performs the best, closely followed by Li's MMD-VGG and Liu's TCL method (run 3), which are close to each other as well. That is, the performance of all the three learning-based methods are similar to each other. DRF's performance is still relatively lower than those three SHREC'18 participating methods. In the non-learning based category, Li's VGG algorithm outperforms Tran's BoW method. For each participating method, more details about the rank list and evaluated retrieval performance of each query can be found on the SceneIBR2018 track website (Abdul-Rashid et al., 2019a).

Although it is not fair to compare non-learning based approaches with learning-based approaches, it is easy to find that the learning-based approaches have produced much more appealing accuracies. In Tran's top-performing learning based approach RNIRAP, in terms of automatically learning the features, the deep learning approach Place365-CNN (Zhou et al., 2018) contributes a lot to its better accuracy among the learning based approaches.

Finally, all the five SceneIBR2018 track participating methods are categorized according to the techniques they employed. All the three learning-based methods (MMD-VGG, TCL, RNIRAP) from three participating groups (Li, Liu, Tran) utilize deep learning techniques to automatically learn local features. Therefore, all of the three groups have considered the deep learning framework for feature learning. DRF also adopts a deep learning-based approach to learn local features. In the non-learning based category, Tran's BoW method employs the Bag-of-Words, while Li's VGG method uses a pre-trained model VGG to directly extract local features. Only Tran's RNIRAP and Yuan's DRF utilize a classification-based 3D model retrieval framework.

5.1.2. *Cross-track performance evaluation*

As can be seen from Fig. 17 and Table 4, both the SceneSBR2018 and SceneIBR2018 tracks have almost the same four participating methods. However, for the same method each performance metric achieved on the SceneIBR2018 track is significantly better than that on the SceneSBR2018 track, while its Precision–Recall curve is also often higher on the image track. We believe at least the following three differences of SceneIBR2018 contribute to its better performance: (1) it has a 40 times larger query dataset which is very helpful for the training of the deep neural networks; (2) compared with the sketch queries of SceneSBR2018, SceneIBR2018's image queries contain much more accurate 3D shape information; and (3) each of SceneIBR2018's image queries has additional color information to correlate to the texture information existing in the 3D scene models. Therefore, there is a much smaller semantic gap to bridge between the query and target datasets for the SceneIBR2018 track, while the SceneSBR2018 track is much more challenging due to a big semantic gap there. It is also interesting to find that DRF does not follow this trend since it achieves similar performance on both tracks, in terms of all the evaluation metrics including Precision–Recall plot. We believe this is due to the semantic retrieval approach targets bridging the semantic gap between 2D scene sketches/images and 3D scenes by incorporating the WordNet-based Scene Semantic Tree into the retrieval process, which helps it to achieve consistency in its retrieval performance on either track.

5.2. ***Scene_SBR_IBR_2019*** *benchmark*

Similarly, based on the our **Scene_SBR_IBR_2019** benchmark described in Section 3.2, we organized two SHREC'19 tracks on 2D scene sketch/image-based 3D scene retrieval, for which we refer to as SceneSBR2019 and SceneIBR2019. Fig. 18 and Table 5 compare the two learning-based Query-by-Sketch retrieval methods submitted

to SceneSBR2019, as well as the three learning-based Query-by-Image retrieval methods submitted to SceneIBR2019, based on the corresponding testing and complete datasets of our **Scene_SBR_IBR_2019** benchmark. Likewise, five new runs coming from the newly introduced approach DRF and SHREC'18 participating method TCL are also evaluated together with the 12 runs of SHREC'19 participating methods.

5.2.1. Peer performance evaluation

Query-by-Sketch retrieval. In this subsection, we comparatively evaluate the six runs of the four methods submitted by the four groups. All the four methods are learning-based methods. As shown in the Fig. 18 and Table 5, Bui's RNSRAP algorithm (run 2) performs the best, followed by their RNSRAP (run 1), a close pair of Yuan's DRF and WYL's TCL1, and VMV-VGG. More details about the retrieval performance of each individual query of every evaluated method are available on the SceneSBR2019 track homepage (Yuan et al., 2019a). An interesting finding is about DRF and VMV-VGG: they use the same CNN model (VGG) and both adopt a classification-based framework, while the main difference is that DRF integrates a semantic loss during its model training process. It is evident to find that there is a very significant improvement in the performance after utilizing semantic information. For example, there is a 78.6% and 10.3% increase in terms of AP on the sketch and image track, respectively. In terms of Precision–Recall plot performance, DRF also outperforms VMV-VGG by a non-trivial margin.

All the four evaluated methods utilized CNN models, which contribute a lot to the achieved performance of those two learning-based approaches. Since deep learning techniques are widely utilized in many latest sketch-based 3D model retrieval methods, it can be regarded as the currently most popular and promising machine learning technique for 2D/3D feature learning and related retrieval. In fact, we can see that the deep learning models which are adopted in these four methods, especially Bui's method, perform well in dealing with this challenging retrieval task. They improved their method used in the SceneIBR2018 track by utilizing object-level semantic information for data augmentation and refining retrieval results, which helps to advance the retrieval performance further. The significant impact on the retrieval performance by utilizing semantic information has also been reflected by the above comparative evaluation of DRF and VMV-VGG. Considering there is still much room for further improvement in the retrieval accuracy as well as the scalability issue, we believe it is very promising to further propose a practical retrieval algorithm for large-scale 2D sketch-based 3D scene retrieval by utilizing both deep learning and scene semantic information.

Finally, we classify all the four evaluated methods based on the techniques adopted: all of them utilize local features, employ a deep learning framework to automatically learn the features, and apply regular transformations (e.g., flipping, translation, rotation). While, Bui further applies adversarial training as well. On the other hand, Liu's TCL adopts a direct feature matching approach, while Yuan's two approaches (VMV and DRF) mainly adopt an image/sketch classification framework and then uses majority vote-based label matching to generate the retrieved result. However, Bui conducts the retrieval based on both 2D sketch recognition and 3D model classification, as well as both object detection and recognition.

Query-by-Image retrieval. As can be seen in the aforementioned figure and table, Bui's RNIRAP algorithm (run 2) performs the best, followed by TCL2, DRF, the baseline method VMV-VGG, TCL1, and the CVAE method (CVAE2). More details about the retrieval performance of each individual query of every evaluated method are available on the SceneIBR2019 track website (Abdul-Rashid et al., 2019b). Here, we want to have a closer study on TCL and DRF. Among all the evaluated approaches, only TCL proposes a so-called triplet-center loss to improve extracted features' discriminative power, while all other five methods completely (i.e., RNIRAP (ResNet), VMV (VGG), and DRF (VGG)) or partially (i.e., CVAE) utilizes a traditional classification loss. The triplet-center loss optimizes each class' center such that relevant samples are closer to it than the centers of other classes. It is obvious to find out the more discriminative power of such approach for retrieval purpose based on its superior performance than the classification loss-based models (i.e., pure VGG/ResNet-based ones). Again, DRF achieves a significant jump (i.e. 10.3% increase in AP) in its performance after integrating a semantic loss with a traditional classification loss during its VGG-based model training. Therefore, it can be anticipated that an integration of a more powerful model, like the triplet-center loss based TCL, and a semantic retrieval framework, such as the semantic tree-based DRF approach, will push the limit of such retrieval framework's performance even further.

Firstly, all the three methods submitted to the SceneIBR2019 track by the three participating groups and all the currently evaluated six methods are leaning-based methods, while there is no submission involving a non-learning based approach during the SceneIBR2019 track time. In addition, all of the six methods have employed a deep neural networks based learning approach.

Secondly, we could further classify the submitted approaches at a finer granular level. RNIRAP, VMV-VGG, and DRF utilize CNN models and a classification-based approach, which contribute a lot to their better accuracies. While, TCL utilizes a trained DNN model to extract feature vectors to perform direct feature matching for retrieval; and the CVAE-based method uses a conditional VAE generative model and resulted latent features to measure the 2D–3D similarities.

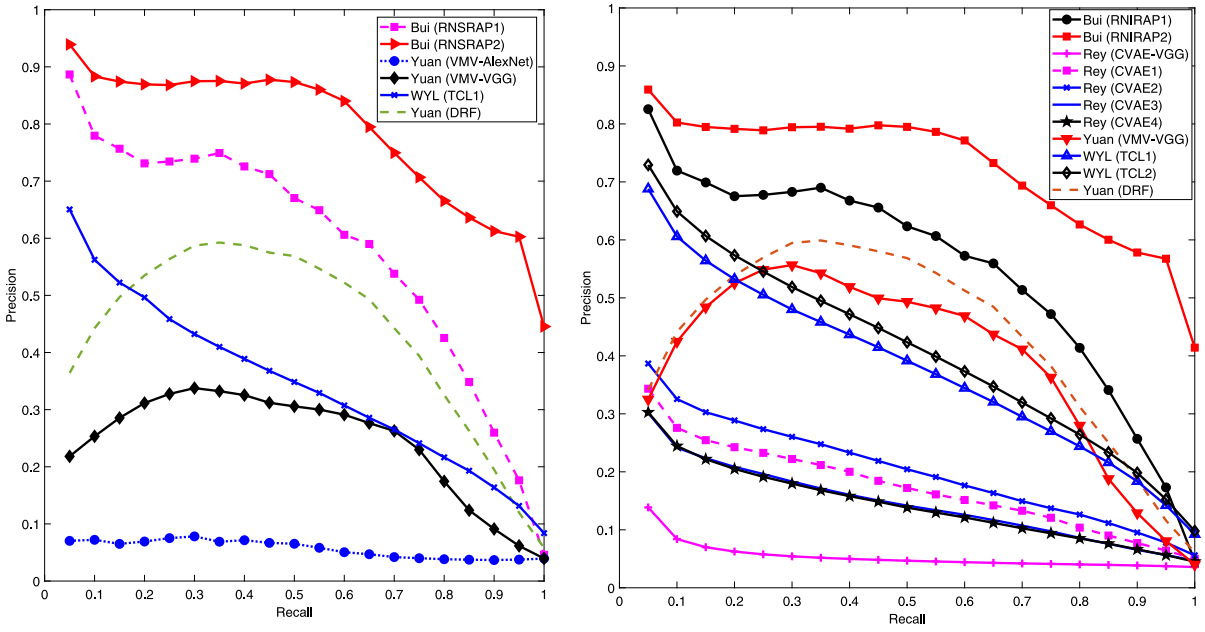
Therefore, according to these two years' SHREC tracks (SHREC'19 and SHREC'18) on this topic, deep learning-based techniques are still the most promising and popular approach in tackling this new and challenging research direction. To further improve the retrieval performance, Bui used scene object semantic information during the stages of data augmentation and retrieval results refinement.

5.2.2. Cross-track performance comparison

Except CVAE, these two tracks share other two participating methods (with minor differences). It is the second time that we have found that generally the performance achieved in the "Image-Based 3D Scene Retrieval (IBR)" track is significantly better, compared with that achieved on the back to back "Sketch-Based 3D Scene Retrieval (SBR)" track. This should be attributed to the same reasons as we have concluded in Section 5.1.2: IBR has a much larger training query dataset which contains images, instead of sketches, that have much more details and color information as well, which makes the semantic gap between the 2D image query and 3D scene targets much smaller. It is also the second time to find that DRF performs differently from the SHREC'19 participating methods. It achieves very similar cross-track performance on all the seven evaluation metrics (NN, FT, ST, E, DCG, AP, and Precision–Recall plot) on the SHREC'19 tracks, which should be attributed to the same reason as mentioned in Section 5.1.2.

5.3. Timing performance evaluation

Table 6 lists the running time information in terms of average response time per query for all the 15 evaluated sketch/image-based 3D scene retrieval algorithm. We define response time as the time difference between the start of a retrieval after submitting the query and the end of the retrieval when a rank list is generated for it. It can be found that most algorithms are very fast and can meet the requirement for real-time retrieval. Typically, it takes from several hours (i.e. approximately 6 h for CVAE on the SHREC'19 image track) to several days (i.e. around 3 days for TCL1 on the same SHREC'19 image track) for the training on the SHREC'18/SHREC'19 track benchmarks. However, since they are offline and all the times are still within a reasonable range, we do not directly compare them in Table 6. In a word, we think most evaluated algorithms have excellent scalability performance in terms of efficiency for large-scale 3D scene retrieval scenarios.



(a) learning-based (Query-by-Sketch) (b) learning-based (Query-by-Image)

Fig. 18. Query-by-Sketch and Query-by-Image Precision–Recall diagram performance comparisons on our **Scene_SBR_IBR_2019** benchmark.

Table 5
Query-by-Sketch and Query-by-Image performance metrics comparison on our **Scene_SBR_IBR_2019** benchmark.

Participant	Method	NN	FT	ST	E	DCG	AP
Query-by-Sketch							
Learning-based methods							
Bui	RNSRAP1	0.914	0.668	0.728	0.665	0.825	0.581
	RNSRAP2	**0.943**	**0.818**	**0.870**	**0.814**	**0.913**	**0.786**
Wang &Yuan & Liu (WYL)	TCL1	0.610	0.345	0.486	0.350	0.680	0.343
Yuan	VMV-AlexNet	0.024	0.046	0.084	0.047	0.386	0.057
	VMV-VGG	0.081	0.281	0.369	0.280	0.533	0.243
Yuan	DRF	0.148	0.500	0.588	0.494	0.670	0.434
Query-by-Image							
Learning-based methods							
Bui	RNIRAP1	0.845	0.620	0.674	0.618	0.791	0.544
	RNIRAP2	**0.865**	**0.749**	**0.792**	**0.745**	**0.863**	**0.722**
Rey	CVAE-VGG	0.071	0.054	0.099	0.055	0.405	0.054
	CVAE1	0.235	0.187	0.295	0.189	0.532	0.172
	CVAE2	0.272	0.217	0.331	0.219	0.560	0.201
	CVAE3	0.199	0.154	0.251	0.157	0.507	0.145
	CVAE4	0.211	0.149	0.246	0.152	0.505	0.142
Wang &Yuan & Liu (WYL)	TCL1	0.632	0.375	0.521	0.376	0.706	0.378
	TCL2	0.677	0.403	0.551	0.403	0.728	0.407
Yuan	VMV-VGG	0.122	0.458	0.573	0.452	0.644	0.390
Yuan	DRF	0.094	0.505	0.595	0.500	0.667	0.430

5.4. Scalability performance evaluation

To evaluate an algorithm's scalability to a larger benchmark, we plan to compare its performance on our two benchmarks **Scene_SBR_IBR_2018** and **Scene_SBR_IBR_2019**. From Tables 4 and 5, we can find that the top-performing algorithms RNSRAP and RNIRAP, as well as TCL (run1) and the new method DRF have available results on both benchmarks.

For the best-performing approaches RNSRAP and RNIRAP, we need to mention that there are some further improvement in their 2019 version if compared with their 2018 version, which can be found in Sections 4.1.3 and 4.1.4. For example, some changes in RNIRAP are listed below. (1) use ResNet50 in SceneIBR2019, in comparison to the ResNet18 model used in SceneIBR2018; (2) to represent the deep learning feature vector, they elevated its dimension from 102 which was used in SceneSBR2018 to 512 in SceneSBR2019; (3) they also increased the dimension of the two hidden layers of the classifier from less than 200 to 1024. For RNSRAP, there is a significant change in their sketch classification in ScceneSBR2019: a query expansion technique was added by searching semantically related natural images and then added their transformed sketch-like images into the sketch training dataset for the training of ResNet50 for feature extraction.

Now, we consider all the four methods (RNSRAP, RNIRAP, TCL and DRF) together. In a direct comparison to the results from SceneIBR2018, SceneIBR2019 results do not preform as well for each of them, including the top methods RNSRAP and RNIRAP even though after several improvements mentioned above. If we compare their Precision–Recall (PR) plots, we can find that it is common the Precision (P)

Table 6
Available timing information comparison of the five Query-by-Sketch and seven Query-by-Image retrieval algorithms: T_S / T_I is the average response time (in seconds) per query for a Query-by-Sketch / Query-by-Image retrieval method. "R" denotes the ranking order of all the runs within their respective type of retrieval (Query-by-Sketch, or Query-by-Image). "-" means not applicable.

Contributor (with computer configuration)	Method	Language	**Scene_SBR_IBR_2018**		**Scene_SBR_IBR_2019**	
			T_S	T_I	T_S	T_I
Li (CPU: Intel(R) @3.3 GHz (single core); Memory: 8 GB; OS: Windows 7)	VGG	C++, Matlab	2.29	2.41	–	–
	MMD-VGG	C++, Matlab	10.14	33.93	–	–
Liu (CPU: Intel(R) Core i3-2350M @2.3 GHz; GPU: 1 x NVIDIA Titan Xp; Memory: 6 GB; OS: Windows 2003 32-bit)	TCL1	Python	0.06	0.09	0.04	0.04
	TCL2	Python	0.09	0.08	–	0.04
	TCL3	Python	0.09	0.07	–	–
Tran & Bui (CPU: Intel(R) Core i5-6198DU @2.30 GHz; GPU: 1 x NVIDIA GeForce 920MX; Memory: 8 GB; OS: Ubuntu) (for BoW only): CPU: Intel(R) Xeon E5-2660 @2.2 GHz; Memory: 12 GB; OS: Windows Server 2008 R2	RNSRAP	Python	0.01	–	0.01	–
	RNIRAP	Python	–	0.01	–	0.01
	BoW1	Python	–	0.01	–	–
	BoW2	Python	–	0.01	–	–
Yuan (CPU: Intel(R) Core i7 6850K @3.6 GHz (6 cores); GPU: 1 x NVIDIA Titan Xp; Memory: 32 GB; OS: Windows 10)	VMV-AlexNet	C++, Matlab	–	–	0.02	–
	VMV-VGG	C++, Matlab	–	–	0.06	0.04
	DRF	C++, Python	0.02	0.03	0.05	0.03
Rey (CPU: Intel(R) Xeon(R) E5-2698v4 @2.2 GHz (4 processors, 20 cores); Memory: 256 GB; OS: Ubuntu 18.04)	CVAE	Ruby, Python	–	–	–	0.09
	CVAE-VGG	Ruby, Python	–	–	–	0.22

values will drop much more quickly from the start of the PR plots on the SHREC'19 tracks than those on the SHREC'18 tracks for all the evaluated methods. These are to be expected since the 10 scene categories in the SceneIBR2018 benchmark are distinct and have few correlations. In fact, this trend is consistent in the SceneSBR2019 as well, which can be found the generally lower performance achieved on the more challenging **Scene_SBR_IBR_2019** benchmark. This has also been explored by us in our prior work (Yuan et al., 2019b): the significant drop in performance can be attributed to the introduction of many correlating scene categories.

Therefore, this raise our interest in developing more robust 3D scene retrieval algorithms which are scalable in a large-scale retrieval scenario.

6. Conclusions and future work

6.1. Conclusions

2D sketch/image 3D scene retrieval is a new, challenging yet important research direction in 3D object retrieval. It has a large amount of related applications. To promote the research in 3D scene retrieval, we built the first 2D scene sketch/image-based 3D scene retrieval benchmark **Scene_SBR_IBR_2018** and organized two SHREC'18 tracks (Yuan et al., 2018; Abdul-Rashid et al., 2018). In 2019, we have further extended the number of categories from 10 to 30 and built the most diverse and comprehensive 2D/3D scene dataset to date **Scene_SBR_IBR_2019**, and further extended the line of our SHREC related research work on sketch/image-based 3D shape retrieval (i.e., SHREC'12 (Li et al., 2012, 2014a), SHREC'13 (Li et al., 2013, 2014a), SHREC'14 (Li et al., 2014b, 2015), SHREC'16 (Li et al., 2016), SHREC'18 (Yuan et al., 2018; Abdul-Rashid et al., 2018)) by running another two related tracks (Yuan et al., 2019c; Abdul-Rashid et al., 2019) in SHREC'19.

Participating groups of these four tracks have explored many different approaches to solve the intractable task of 2D to 3D scene understanding. Currently, six Query-by-Sketch and eight Query-by-Image 3D scene retrieval algorithms have been evaluated on our two benchmarks, including a newly incorporated semantic retrieval method DRF for each track. We have conducted a comprehensive comparison of all these 14 retrieval methods by evaluating them on the two benchmarks. We also made the benchmarks, evaluation results and evaluation toolkits publicly available at our websites (Yuan et al., 2019d; Abdul-Rashid et al., 2019a; Yuan et al., 2019a; Abdul-Rashid et al., 2019b). We also review the related techniques and datasets, and provide a method description for each retrieval algorithm in the paper. We believe all of these will become an important and useful resource for the researchers that are interested in this topic as well as many related applications.

6.2. Future work

The four tracks not only help us identify state-of-the-art methods, but also existing problems, current challenges and future research directions for this important, new and interesting research topic.

- **Building a large-scale and/or multimodal 2D scene-based 3D scene retrieval benchmarks.** Our proposed **Scene_SBR_IBR_2018** contains only ten scene classes, which is one of the reasons that all the three deep learning-based participating methods have achieved excellent performance. Our **Scene_SBR_IBR_2019**, even as the largest benchmark for 2D scene image-based 3D scene retrieval, has only thirty scene categories. This again can partially explain the still relatively good performance that has been achieved by the top deep learning-based participating methods. However, we did see an apparent drop in the overall performance. Therefore, testing the scalability of a retrieval algorithm with respect to a large-scale retrieval scenario and various 2D/3D data formats is very important for many practical applications. Therefore, our next target is to build a large-scale benchmark which supports multiple modalities of 2D queries (i.e. images and sketches) and/or 3D model targets (i.e. meshes, RGB–D, LIDAR, and range scans). Then, we will invite people to adapt and run their algorithms on the new benchmark again to evaluate their scalability in a large-scale and/or multimodal 3D scene retrieval scenario.
- **More realistic 3D scenes models.** Some of the SketchUp 3D scene models that we downloaded from 3D Warehouse (Trimble, 2018) are not as realistic as relevant 2D scene images. For example, in the "mountain" category, the ratio between trees and mountains is not real, which could reduce the 3D scene retrieval accuracy. Due to this reason, a more realistic 3D scene dataset is also necessary.
- **Semantics-driven 2D scene sketch/image-based 3D scene retrieval.** Since a scene is composed of one or more objects, the semantic information existing in 2D scene sketches and 3D scene models and the relationships between objects or between objects and related scenes are very useful for 3D scene retrieval. For instance, Bui's team utilized the known semantic information for data augmentation, e.g., they manually collected and added "camel" and "cactus" images to the "desert" category during training. They also employed object detection and recognition to refine their retrieval results. There is a lot of semantic information in both the 2D sketch/image queries and the 3D scene model targets in our two scene retrieval benchmarks. To improve either the accuracy or efficiency of a 2D scene sketch/image-based 3D

scene retrieval algorithm, we need to consider utilizing the semantic information. However, we find that only one participating group has considered this, probably due to limited time for the competition. Therefore, we can expect even better performance if they also incorporate the semantic information into their methods. We also believe that related applications (i.e. online 3D scene retrieval, 3D Entertainment contents development, and autonomous driving cars) will benefit a lot from the retrieval based on extracted semantic information in both the queries and targets. These have been partially proved by the DRF approach (Yuan et al., 2020).

- **Extending the feature vectors by incorporating the geolocation estimation features.** Photo geolocation estimation is to predict the GPS coordinates for a photo image. This information is helpful in classifying certain scene images. By classifying the earth's geographical cells based on deep learning, a recent work conducted by Müller-Budack et al. (2018) has shown that photo geolocalization without any limitations can work to some extend reliably, even though with a small training dataset. Therefore, it is promising to achieve even better results by taking the scene's geographical information into account when forming a feature representation for the retrieval.
- **Classification-based retrieval.** It can be found that class-based or classification-based 3D model retrieval (i.e. RNIRAP, VMV-VGG, and DRF) is potential to achieve even better performance compared to other algorithms which utilize a more traditional 3D model retrieval pipeline. This also coincides with our prior findings related to class-based 3D model retrieval (Li and Johan, 2013) or semantic information-based 3D model retrieval (Li et al., 2017). This relatively new framework contributes to better NN, FT and the overall performance metrics such as DCG and AP.
- **Application-oriented 2D scene-based 3D scene retrieval.** It targets developing a 2D scene-based 3D scene retrieval dedicated for a related application, such as creating 3D scene contents for a new 4D immersive program, like Disney World's Avatar Flight of Passage Ride (Wikipedia, 2019; Attractions, 2019; the Magic, 2019). Other example applications include but not limited to retrieving domain-specific 3D scenes such as indoor/outdoor scenes, sand table models for real estate applications, rainforest scenes for cartoon or movie production. For instance, automatically retrieving scenes from movies, computer games, and educational content by utilizing text and speech recognition to extract semantic scene information. This will help us build much larger benchmarks as well.
- **Deep learning models specifically designed for 3D scene retrieval.** From the method evaluations, we can find that deep learning techniques have great potential in achieving promising retrieved performance. However, we can find that all the related algorithms adapt the existing neural network models designed for other purposes (e.g., objects classification), thus lacking considerations of the characteristics of this scene retrieval problem. Therefore, it is promising to achieve even better retrieval result if we develop new deep learning models which fit this scenario well.

CRediT authorship contribution statement

Juefei Yuan: Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Hameed Abdul-Rashid:** Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing - original draft. **Bo Li:** Conceptualization, Resources, Writing - original draft, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Yijuan Lu:** Conceptualization, Resources. **Tobias Schreck:** Conceptualization, Visualization, Writing - review & editing. **Song Bai:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Xiang Bai:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Ngoc-Minh Bui:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Minh N. Do:** Methodology, Software, Visualization, Writing original draft, Writing - review & editing. **Trong-Le Do:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Anh-Duc Duong:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Kai He:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Xinwei He:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Mike Holenderski:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Dmitri Jarnikov:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Tu-Khiem Le:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Wenhui Li:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Anan Liu:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Xiaolong Liu:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Vlado Menkovski:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Khac-Tuan Nguyen:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Thanh-An Nguyen:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Vinh-Tiep Nguyen:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Weizhi Nie:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Van-Tu Ninh:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Perez Rey:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Yuting Su:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Vinh Ton-That:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Minh-Triet Tran:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Tianyang Wang:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Shu Xiang:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Shandian Zhe:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Heyu Zhou:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Yang Zhou:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Zhichao Zhou:** Methodology, Software, Visualization, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project is supported by the University of Southern Mississippi, USA Faculty Startup Funds Award to Dr. Bo Li, and the Texas State Research Enhancement Program, USA and NSF, USA CR1-1305302 Awards to Dr. Yijuan Lu. We gratefully acknowledge the support from NVIDIA Corporation, USA for the donation of the Titan X/Xp GPUs used in this research and anonymous content creators from the Internet.

References

Abdul-Rashid, H., Yuan, J., Li, B., Lu, Y., 2019a. SHREC'18 2D scene image-based 3D scene retrieval track website. http://orca.st.usm.edu/~bli/SceneIBR2018/.

Abdul-Rashid, H., Yuan, J., Li, B., Lu, Y., 2019b. SHREC'19 extended 2D scene image-based 3D scene retrieval track website. http://orca.st.usm.edu/~bli/SceneIBR2019/.

Abdul-Rashid, H., Yuan, J., Li, B., Lu, Y., Bai, S., Bai, X., Bui, N., Do, M.N., Do, T., Duong, A.D., He, X., Le, T., Li, W., Liu, A., Liu, X., Nguyen, K., Nguyen, V., Nie, W., Ninh, V., Su, Y., Ton-That, V., Tran, M., Xiang, S., Zhou, H., Zhou, Y., Zhou, Z., 2018. SHREC'18: 2D image-based 3D scene retrieval. In: Eurographics Workshop on 3D Object Retrieval, 3DOR 2018, 16 April 2018. Delft, the Netherlands, pp. 37–44.

Abdul-Rashid, H., Yuan, J., Li, B., Lu, Y., Schreck, T., Bui, N., Do, T., Holenderski, M., Jarnikov, D., Le, T., Menkovski, V., Nguyen, K., Nguyen, T., Nguyen, V., Ninh, V., Rey, L.A.P., Tran, M., Wang, T., 2019. SHREC'19: Extended 2D scene image-based 3D scene retrieval. In: 12th Eurographics Workshop on 3D Object Retrieval, 3DOR 2019, Genoa, Italy, May 5-6, 2019. pp. 41–48.

Arandjelovic, R., Zisserman, A., 2012. Three things everyone should know to improve object retrieval. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012. pp. 2911–2918.

Armeni, I., Sax, S., Zamir, A.R., Savarese, S., 2017. Joint 2D-3D-semantic data for indoor scene understanding. CoRR abs/1702.01105.

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I.K., Fischer, M., Savarese, S., 2016. 3D semantic parsing of large-scale indoor spaces. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 1534–1543.

Attractions, W., 2019. New ride!!!! disney world animal kingdom: Avatar flight of passage ride video 4k hd video (pov). http://www.youtube.com/watch?v=f-cw7iCUY3c.

Bai, S., Bai, X., Zhou, Z., Zhang, Z., Latecki, L.J., 2016. GIFT: A real-time and scalable 3D shape search engine. In: CVPR. IEEE, pp. 5023–5032.

Caesar, H., Uijlings, J.R.R., Ferrari, V., 2018. COCO-Stuff: Thing and stuff classes in context. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 1209–1218.

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Chen, B.X., Sahdev, R., Wu, D., Zhao, X., Papagelis, M., Tsotsos, J.K., 2019. Scene classification in indoor environments for robots using context based word embeddings. CoRR abs/1908.06422.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F., 2009. Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255.

Eitz, M., Hays, J., Alexa, M., 2012. How do humans sketch objects?. ACM Trans. Graph. (Proc. SIGGRAPH) 31 (4), 44:1–44:10.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2012. The PASCAL visual object classes challenge 2012 (VOC2012) results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Fisher, M., Savva, M., Hanrahan, P., 2011. Characterizing structural relationships in scenes using graph kernels. ACM Trans. Graph. (TOG) 30 (4), 34.

Flickr, 2018. Flickr website. https://www.flickr.com/.

Gao, C., Liu, Q., Xu, Q., Liu, J., Wang, L., Zou, C., 2020. SketchyCOCO: Image generation from freehand scene sketches. CoRR abs/2003.02683.

Google, 2018. Google images. https://www.google.com/imghp?hl=EN.

Handa, A., Patraucean, V., Stent, S., Cipolla, R., 2016. SceneNet: An annotated model generator for indoor scene understanding. In: ICRA. IEEE, pp. 5737–5743.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: CVPR. pp. 770–778.

He, X., Zhou, Y., Zhou, Z., Bai, S., Bai, X., 2018. Triplet center loss for multi-view 3D object retrieval. In: CVPR.

Hoàng, N.V., Gouet-Brunet, V., Rukoz, M., Manouvrier, M., 2010. Embedding spatial information into image content description for scene retrieval. Pattern Recognit. 43 (9), 3013–3024.

Hua, B., Pham, Q., Nguyen, D.T., Tran, M., Yu, L., Yeung, S., 2016. SceneNN: A scene meshes dataset with annotations. In: 3DV. IEEE Computer Society, pp. 92–101.

Huang, S., Qi, S., Zhu, Y., Xiao, Y., Xu, Y., Zhu, S., 2018. Holistic 3D scene parsing and reconstruction from a single RGB image. CoRR abs/1808.02201.

Hung, W.-C., Tsai, Y.-H., Shen, X., Lin, Z.L., Sunkavalli, K., Lu, X., Yang, M.-H., 2017. Scene parsing with global context embedding. In: ICCV. pp. 2650–2658.

Kalliatakis, G., 2017. Keras-VGG16-Places365. GitHub, https://github.com/GKalliatakis/Keras-VGG16-places365.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980.

Kingma, D.P., Mohamed, S., Rezende, D.J., Welling, M., 2014. Semi-supervised learning with deep generative models. In: Advances in Neural Information Processing Systems. pp. 3581–3589.

Kingma, D.P., Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Lesk, M., 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC 1986, Toronto, Ontario, Canada, 1986, pp. 24–26.

Li, B., Johan, H., 2013. 3D model retrieval using hybrid features and class information. Multimedia Tools Appl. 62 (3), 821–846.

Li, B., Lu, Y., Duan, F., Dong, S., Fan, Y., Qian, L., Laga, H., Li, H., Li, Y., Liu, P., Ovsjanikov, M., Tabia, H., Ye, Y., Yin, H., Xue, Z., 2016. SHREC'16: 3D sketch-based 3D shape retrieval. In: 3DOR 2016.

Li, B., Lu, Y., Godil, A., Schreck, T., Aono, M., Johan, H., Saavedra, J.M., Tashiro, S., 2013. SHREC'13 track: Large scale sketch-based 3D shape retrieval. In: 3DOR. pp. 89–96.

Li, B., Lu, Y., Godil, A., Schreck, T., Bustos, B., Ferreira, A., Furuya, T., Fonseca, M.J., Johan, H., Matsuda, T., Ohbuchi, R., Pascoal, P.B., Saavedra, J.M., 2014a. A comparison of methods for sketch-based 3D shape retrieval. CVIU 119, 57–80.

Li, B., Lu, Y., Johan, H., Fares, R., 2017. Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information. Multimedia Tools Appl. 76 (24), 26603–26631.

Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burtscher, M., Chen, Q., Chowdhury, N.K., Fang, B., Fu, H., Furuya, T., Li, H., Liu, J., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A., Wan, Y., Zhang, C., Zou, C., 2015. A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. CVIU 131, 1–27.

Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burtscher, M., Fu, H., Furuya, T., Johan, H., Liu, J., Ohbuchi, R., Tatsuma, A., Zou, C., 2014b. SHREC'14 track: extended large scale sketch-based 3D shape retrieval. In: 3DOR. pp. 121–130.

Li, B., Schreck, T., Godil, A., Alexa, M., Boubekeur, T., Bustos, B., Chen, J., Eitz, M., Furuya, T., Hildebrand, K., Huang, S., Johan, H., Kuijper, A., Ohbuchi, R., Richter, R., Saavedra, J.M., Scherer, M., Yanagimachi, T., Yoon, G.-J., Yoon, S.M., 2012. SHREC'12 track: Sketch-based 3D shape retrieval. In: 3DOR. pp. 109–118.

Limberger, F.A., Wilson, R.C., Aono, M., Audebert, N., Boulch, A., Bustos, B., Giachetti, A., Godil, A., Saux, B.L., Li, B., Lu, Y., Nguyen, H.D., Nguyen, V., Pham, V., Sipiran, I., Tatsuma, A., Tran, M., Velasco-Forero, S., 2017. SHREC'17: Point-cloud shape retrieval of non-rigid toys. In: 3DOR.

Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: common objects in context. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V. pp. 740–755.

Liu, N., Han, J., 2016. DHSNet: Deep hierarchical saliency network for salient object detection. In: CVPR. pp. 678–686.

Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S., 2013. Transfer feature learning with joint distribution adaptation. In: ICCV. pp. 2200–2207.

the Magic, I., 2019. New flight of passage ride queue, pre-show in pandora - the world of avatar at walt disney world. http://www.youtube.com/watch?v=eM8f47Igtu8.

Merrell, P., Schkufza, E., Li, Z., Agrawala, M., Koltun, V., 2011. Interactive furniture layout using interior design guidelines. ACM Trans. Graph. 30 (4), 87.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. CoRR abs/1301.3781.

Miller, G.A., 1995. WordNet: A lexical database for english. Commun. ACM 38 (11), 39–41.

Mohan, R., 2014. Deep deconvolutional networks for scene parsing. CoRR abs/1411.4101.

Müller-Budack, E., Pustu-Iren, K., Ewerth, R., 2018. Geolocation estimation of photos using a hierarchical model and scene classification. In: Computer Vision – ECCV 2018. Springer International Publishing, Cham, pp. 575–592.

Naseer, M., Khan, S.H., Porikli, F., 2018. Indoor scene understanding in 2.5/3D: A survey. CoRR abs/1803.03352.

Nguyen, V., Ngo, T.D., Tran, M., Le, D., Duong, D.A., 2015. A combination of spatial pyramid and inverted index for large-scale image retrieval. IJMDEM 6 (2), 37–51.

Oliva, A., Torralba, A., 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vis. 42 (3), 145–175.

Patterson, G., Hays, J., 2012. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012. pp. 2751–2758.

Patterson, G., Xu, C., Su, H., Hays, J., 2014. The SUN attribute database: Beyond categories for deeper scene understanding. Int. J. Comput. Vis. 108 (1–2), 59–81.

Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement. CoRR abs/1804.02767.

Ren, S., He, K., Girshick, R.B., Sun, J., 2015. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497.

Renault, 2019. Renault SYMBOIZ concept. http://www.renault.co.uk/vehicles/concept-cars/symbioz-concept.html.

Shilane, P., Min, P., Kazhdan, M.M., Funkhouser, T.A., 2004. The Princeton shape benchmark. In: SMI. pp. 167–178.

Silberman, N., Hoiem, D., Kohli, P., Fergus, R., 2012. Indoor segmentation and support inference from RGBD images. In: Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V. pp. 746–760.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556.

Sohn, K., Liu, S., Zhong, G., Yu, X., Yang, M., Chandraker, M., 2017. Unsupervised domain adaptation for face recognition in unlabeled videos. CoRR abs/1708.02191.

Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.A., 2017. Semantic scene completion from a single depth image. In: CVPR. IEEE Computer Society, pp. 190–198.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G., 2015. Multi-view convolutional neural networks for 3D shape recognition. In: ICCV. pp. 945–953.

Tips, L.T., 2019. Driving a multi-million dollar autonomous car. http://www.youtube.com/watch?v=vlIJfV1u2hM&feature=youtu.be.

Trimble, 2018. 3D warehouse. http://3dwarehouse.sketchup.com/?hl=en.

Tzeng, E., Hoffman, J., Saenko, K., Darrell, T., 2017. Adversarial discriminative domain adaptation. In: CVPR. pp. 2962–2971.

Wikipedia, 2019. Avatar flight of passage. http://en.wikipedia.org/wiki/Avatar_Flight_of_Passage.

Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C.B., Su, H., Mottaghi, R., Guibas, L.J., Savarese, S., 2016. ObjectNet3D: A large scale database for 3D object recognition. In: ECCV (8). In: Lecture Notes in Computer Science, vol. 9912, Springer, pp. 160–176.

Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A., 2016. SUN database: Exploring a large collection of scene categories. Int. J. Comput. Vis. 119 (1), 3–22.

Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A., 2010. SUN database: Large-scale scene recognition from abbey to zoo. In: CVPR. IEEE Computer Society, pp. 3485–3492.

Xiao, J., Owens, A., Torralba, A., 2013. SUN3D: A database of big spaces reconstructed using SfM and object labels. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013. pp. 1625–1632.

Xu, K., Kim, V.G., Huang, Q., Mitra, N., Kalogerakis, E., 2016. Data-driven shape analysis and processing. In: SIGGRAPH ASIA 2016 Courses. ACM, p. 4.

Ye, Y., Li, B., Lu, Y., 2016. 3D sketch-based 3D model retrieval with convolutional neural network. In: 2016 23rd International Conference on Pattern Recognition (ICPR). pp. 2936–2941.

Ye, Y., Lu, Y., Jiang, H., 2016. Human's scene sketch understanding. In: ICMR '16. pp. 355–358.

Yuan, J., Abdul-Rashid, H., Li, B., Lu, Y., 2019a. SHREC'19 extended 2D scene sketch-based 3D scene retrieval track website. http://orca.st.usm.edu/~bli/SceneSBR2019/.

Yuan, J., Abdul-Rashid, H., Li, B., Lu, Y., 2019b. Sketch/image-based 3D scene retrieval: Benchmark, algorithm, evaluation. In: 2nd IEEE Conference on Multimedia Information Processing and Retrieval, MIPR 2019, San Jose, CA, USA, March 28-30, 2019. pp. 264–269.

Yuan, J., Abdul-Rashid, H., Li, B., Lu, Y., Schreck, T., Bui, N., Do, T., Nguyen, K., Nguyen, T., Nguyen, V., Tran, M., Wang, T., 2019c. SHREC'19: Extended 2d scene sketch-based 3D scene retrieval. In: 12th Eurographics Workshop on 3D Object Retrieval, 3DOR 2019, Genoa, Italy, May 5-6, 2019. pp. 33–39.

Yuan, J., Li, B., Lu, Y., 2019d. SHREC'18 2D scene sketch-based 3D scene retrieval track website. http://orca.st.usm.edu/~bli/SceneSBR2018/.

Yuan, J., Li, B., Lu, Y., Bai, S., Bai, X., Bui, N., Do, M.N., Do, T., Duong, A.D., He, X., Le, T., Li, W., Liu, A., Liu, X., Nguyen, K., Nguyen, V., Nie, W., Ninh, V., Su, Y., Ton-That, V., Tran, M., Xiang, S., Zhou, H., Zhou, Y., Zhou, Z., 2018. SHREC'18: 2D scene sketch-based 3D scene retrieval. In: Eurographics Workshop on 3D Object Retrieval, 3DOR 2018, 16 April 2018. Delft, the Netherlands, pp. 29–36.

Yuan, J., Wang, T., Zhe, S., Lu, Y., Li, B., 2020. Semantic tree based 3D scene model recognition. In: The IEEE 3rd International Conference on Multimedia Information Processing and Retrieval, MIPR 2020, Shenzhen, China, August 6-9, 2020.

Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.A., 2017. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In: CVPR. IEEE Computer Society, pp. 199–208.

Zhao, H., Puig, X., Zhou, B., Fidler, S., Torralba, A., 2017. Open vocabulary scene parsing. In: ICCV. IEEE Computer Society, pp. 2021–2029.

Zhou, B., Lapedriza, À., Khosla, A., Oliva, A., Torralba, A., 2018. Places: A 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. 40 (6), 1452–1464.

Zhou, B., Lapedriza, À., Xiao, J., Torralba, A., Oliva, A., 2014. Learning deep features for scene recognition using places database. In: NIPS. pp. 487–495.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., 2016. Semantic understanding of scenes through the ADE20K dataset. CoRR abs/1608.05442.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A., 2017. Scene parsing through ADE20K dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 5122–5130.

Zhu, C., Jegou, H., Satoh, S., 2013. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In: ICCV. pp. 1705–1712.

Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.-Z., Xiang, T., Gao, C., Chen, B., Zhang, H., 2018. SketchyScene: Richly-Annotated Scene Sketches. In: Proc. of ECCV.