Summer 8-1-2021

Semantics-Driven Large-Scale 3D Scene Retrieval

Juefei Yuan

SEMANTICS-DRIVEN LARGE-SCALE 3D SCENE RETRIEVAL

by

Juefei Yuan

A Dissertation
Submitted to the Graduate School,
the College of Arts and Sciences,
and the School of Computing Sciences and Computer Engineering
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved by:

Dr. Bo Li, Committee Chair

Dr. Chaoyang Zhang

Dr. Bikramjit Banerjee

Dr. Zhaoxian Zhou

Dr. Ras Pandey

Dr. Karen S. Coats, Dean of the Graduate School

August 2021

ABSTRACT

2D scene sketch/image-based 3D scene retrieval is to retrieve man-made 3D scene models given a user's hand-drawn 2D scene sketch or a 2D scene image usually captured by a camera. Due to the intuitiveness in sketching and ubiquitous availability in image capturing, this research topic has many applications such as 3D scene reconstruction, autonomous driving cars, 3D geometry video retrieval, and 3D AR/VR Entertainment. It is a brand new but also very challenging research topic in the field of 3D object retrieval due to the semantic gap in their representations: 3D scene models or their views differ from either non-realistic 2D scene sketches or realistic 2D scene images.

For 2D sketch-based 3D model retrieval, it has the intuitiveness advantage over other types of retrieval schemes and there is a lot of research in sketch-based 3D model retrieval, which usually targets the problem of retrieving a list of candidate 3D models using a single sketch as input. 2D scene sketch-based 3D scene retrieval is a brand new research topic in the field of 3D object retrieval. Unlike traditional sketch-based 3D model retrieval which ideally assumes that a query sketch contains only a single object, this is a new 3D model retrieval topic within the context of a 2D scene sketch which contains several objects that may overlap each other and thus be occluded and also have relative location configurations. It is challenging due to the semantic gap existing between the iconic 2D representation of sketches and more accurate 3D representation of 3D models. For 2D scene image-based 3D scene retrieval, which has the same semantic gap problem, it has an intuitive and convenient framework which allows users to learn, search, and utilize the retrieved results for vast related applications.

To boost this interesting and important research, we have built the currently largest and most comprehensive 2D scene sketch/image-based 3D scene retrieval benchmark, developed a convolutional neural network (CNN) based 3D scene retrieval algorithm, further organized four related Eurographics Shape Retrieval Contest (SHREC) tracks based on

the benchmarks we curated, and finally conducted a comprehensive evaluation of all the participating methods on the benchmarks.

We developed a semantics-driven large-scale 3D scene retrieval framework in this project. It comprises the following three main components.

Firstly, we built a **richly-annotated hierarchical scene database**, named SceneNet, to support large-scale 3D scene retrieval that will generate good retrieval performance in terms of accuracy, efficiency and extensibility, for a variety of applications including 3D printing, animation production, and virtual reality (VR)-based applications, such as online touring.

Secondly, to bridge the semantic gap between scene sketches/images and models, based on SceneNet, we proposed both **semantic tree and gradient descend-based 3D scene retrieval framework**. The proposed approaches can effectively capture semantic information of 2D scene sketches/images and 3D scene models, accurately measure their similarities, and therefore greatly enhance the retrieval performance.

Finally, we notice the fact that there are many publicly available 2D scene images online, but there are much fewer 2D scene sketches to train a deep model. What's more, in the existing 2D scen sketch datasets, there are not as many categories for 2D scene sketches as there are for 2D scene images. If we can utilize 2D scene images to directly convert them into 2D scene sketches, this problem will be solved. However, the problem of image-to-sketch (I2S) synthesis still remains open and challenging. To further explore this research direction, we propose a framework for generating full-scene sketch representations from natural scene images, aiming to generate outputs that approximate hand-drawn scene sketches. Specifically, we exploit generative adversarial models to produce full-scene sketches given arbitrary input images that are actually conditions which are incorporated to guide the distribution mapping in the context of adversarial learning. We conduct extensive experiments to validate the proposed framework and provide detailed quantitative and qualitative evaluations to demonstrate its effectiveness.

ACKNOWLEDGMENTS

Without the help and support of so many people who are gratefully acknowledged here, this dissertation cannot be completed smoothly.

I would like to thank my supervisor, Dr. Bo Li, for his time, effort and constant support for guiding me of doing research during my Ph.D. career. His professional academic level and standards, enthusiasm for research and serious attitude will bring me guidance and benefit me in the future. I am also grateful to Prof. Chaoyang Zhang for providing me a lot of helpful advices both in research and courses learning, and Dr. Bikramjit Banerjee, Dr. Ras Pandey and Dr. Zhaoxian Zhou for their invaluable lectures and advices.

I also want to acknowledge Mr. Hameed Abdul-Rashid who collaborated with me in 3D scene retrieval projects and helped me in expanding my knowledge in the deep learning field.

I want to thank Mr. Tom Rishel for his teaching assistant support as well.

I should appreciate the financial support to this research provided by the University of Southern Mississippi, and the equipment provided by the School.

Finally, I want to thank my family for their unconditional support, encouragement and blessings. Because of them, I have the courage to face and overcome all difficulties.

TABLE OF CONTENTS

ABSTRACT iv

ACKNOWLEDGMENTS v

LIST OF ILLUSTRATIONS viii

LIST OF TABLES x

LIST OF ABBREVIATIONS xii

1 BACKGROUND **1**
1.1 Background and Motivation 1
1.2 Project Overview 3
1.3 Overview of the Approach 6
1.4 Thesis Organization 10
1.5 Summary 10

2 LITERATURE REVIEW **11**
2.1 Introduction 11
2.2 Terminologies 12
2.3 3D Scene Classification 14
2.4 3D Scene Recognition 17
2.5 3D Scene Retrieval 21
2.6 3D Scene Reconstruction 22
2.7 3D Scene Generation 27
2.8 3D Scene Datasets 29
2.9 Other Related Deep Learning Based 3D Scene Understanding 34
2.10 Adversarial Networks Related Research Directions, Techniques and Benchmarks 36
2.11 Summary 39

3 BENCHMARKS BUILDING AND METHODS EVALUATION **41**
3.1 Introduction 41
3.2 Benchmarks 44
3.3 Methods 53
3.4 Results 72
3.5 Summary 82

4 SEMANTICS-BASED 3D SCENE RETRIEVAL 84
4.1 Introduction 84
4.2 Semantics-driven 3D scene model retrieval 86
4.3 Experiments and discussions 90
4.4 Summary 94

5 IMAGE TO SCENE SKETCH GENERATION 95
5.1 Introduction 95
5.2 Methodology 97
5.3 Experiments and discussions 103
5.4 Summary 108

6 CONCLUSIONS AND FUTURE WORK 110
6.1 Conclusions 110
6.2 Future Work 111

LIST OF ILLUSTRATIONS

Figure

1.1 Semantic sketch/image-based 3D scene retrieval framework 4
1.2 WordNet ontology example. 9

2.1 A VDRAE-based 3D object layout prediction example. Segmented 3D point cloud as input (top left), processed by VDRAE system (top right, make the objects in the same category have the same color), and make fully objects contained in 3D bounding boxes (bottom) [170]. 18
2.2 Embed image pixel features and word concepts jointly [231]. 20
2.3 VBV-VGG architecture [223]. 23
2.4 Performance on both synthetic (row $2 \sim 3$)and real scenes (row $4 \sim 9$). Each row contains the scene area, # of robots (#R), # of planning intervals (#I), planning time for each planning interval (PT), time of each planning interval (IT), total scanning time (TT), and all robots' total movement distance (TD) [55]. 26
2.5 A training set bedroom example with the corresponding scene hierarchy. The root node have five children which are one floor node and four wall nodes. Then each wall has its own subtree with more detailed object-object relations [116]. 30

3.1 2D scene sketch query examples [219] in our **Scene_SBR_IBR_2018** benchmark. 45
3.2 2D scene image query examples in our **Scene_SBR_IBR_2018** benchmark. . . 46
3.3 3D scene model target examples in our **Scene_SBR_IBR_2018** benchmark. . . 47
3.4 2D scene sketch query examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown. 50
3.5 2D scene image query examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown. 51
3.6 3D scene model target examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown. 52
3.7 Several example representative views. 56
3.8 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. 57
3.9 2D scene classification with scene attributes. 60
3.10 Place classification for screenshots of 3D models with adversarial discriminative domain adaptation. 62
3.11 2D scene classification with scenes' deep features. 64
3.12 Two-step process of the 3D scene classification method. 65
3.13 VMV architecture [223]. 67
3.14 A 13 sampled scene view images example of an apartment scene model [223]. 68

3.15 Overview of scene sampling and CVAE distribution learning. 70
3.16 Query-by-Sketch and Query-by-Image Precision-Recall diagram performance comparisons on our **Scene_SBR_IBR_2018** benchmark. 73
3.17 Query-by-Sketch and Query-by-Image Precision-Recall diagram performance comparisons on our **Scene_SBR_IBR_2019** benchmark. 77

4.1 3D target scene model and 2D scene image examples in our **Scene_SBR_IBR** benchmark. One example per class is shown. 91
4.2 Object occurrence probabilities for the airport terminal scene category. 92

5.1 Example sketches rendered by our method based on given images. Row 1: given images. Row 2: rendered sketches. 96
5.2 Sketches generated by the CycleGAN [15] using the given images as direct inputs. Row 1: given images. Row 2: generated sketches. 98
5.3 The proposed framework I2S^2 for full-scene Image-to-Scene Sketch translation. A natural images goes through two stages: HED edge detection-based feature selection and CycleGAN-based distribution mapping. 100
5.4 Sketch generation example with our model. (A) represents a given color image, (B) is the corresponding conditional input, and (C) is a generated full-scene sketch. 102
5.5 Qualitative evaluations of different methods. Row 1: given images. Row 2: results of HED [214]. Row 3: results of Photo-Sketching [117]. Row 4: our results. 103
5.6 Generated sketches when different loss functions are employed to train the generative model. Row 1: given images. Row 2: results of the WGAN-loss [17] (**WGAN+**). Row 3: results of the CycleGAN-loss [15] (Our approach). . 104
5.7 Generated sketches when different conditional inputs are used. Row 1: given images. Row 2: when the conditional input is provided by the Canny edge detector (**Canny+**). Row 3: when the conditional input is provided by the HED method [214] (Our method). 105
5.8 Full-scene sketches generated by our method. Row 1: given images. Row 2: generated full-scene sketches. 106

LIST OF TABLES

Table

2.1 Overview of the thirty-five (35) 3D scene analysis and processing papers reviewed in this dissertation w.r.t different research directions, inputs, and approaches. 14

3.1 Training and testing dataset information of our **Scene_SBR_IBR_2018** benchmark. 45

3.2 Training and testing dataset information of our **Scene_SBR_IBR_2019** benchmark. 49

3.3 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 [109] 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. 55

3.4 Query-by-Sketch and Query-by-Image performance metrics comparison on our **Scene_SBR_IBR_2018** benchmark. 74

3.5 Query-by-Sketch and Query-by-Image performance metrics comparison on our **Scene_SBR_IBR_2019** benchmark. 78

3.6 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. . . 81

4.1 Scene recognition accuracy comparison on the testing dataset of **Scene_SBR_IBR**. 93

4.2 3D scene retrieval performance comparison on the 3D scene testing dataset of **Scene_SBR_IBR**. 94

5.1 Quantitative evaluations of generated sketches by different methods. For FID, the lower the better, and the higher the better for the other metrics. In **Canny+**, we adopt the Canny detector to detect the edge map from an image, and use this edge map as the conditional input. In **CycleGAN**, we directly use regular images as conditional inputs. In **WGAN+**, Wasserstein loss is used within our framework. 107

Chapter 1

BACKGROUND

1.1 Background and Motivation

With ubiquitous cameras and popular 3D scanning and capturing devices to help us capture 2D/3D scene data, there are many scene understanding related applications, as well as quite a few important and interesting research problems in processing, analyzing, and understanding the available scene data. During the recent several years, there is a significant advancement in different research directions in this field and quite a few novel 3D scene analysis and processing methods that have been proposed correspondingly in each direction.

Compared to 3D objects, 3D scenes are much closer to our daily life. There are a large amount of real life relevant applications, such as autonomous driving cars, 3D geometry video retrieval, 3D AR/VR Entertainment, etc. Therefore, recently researchers have proposed many 3D scene analysis and processing methods and contributed significantly to this research area. The research directions within this area include: a) **3D Scene Classification**, which is to classify the 3D scene models into different certain categories; b) **3D Scene Generation**, which is to generate 3D scene models from 2D images or nature languages (e.g., "a person besides a table"); c) **3D Scene Recognition**, which is to recognize the category of a given 3D scene; d) **3D Scene Reconstruction**, which is to reconstruct three-dimensional scene models from multiple 2D projected scene images, whose depth information may be missing; e) **3D Scene Retrieval**, which is to retrieve 3D scene models given an input query (2D scene sketches/images) provided by the users.

The research focuses on semantics-based 3D scene retrieval. Although there are many existing 3D shape retrieval systems, there is little existing research work on 2D scene sketch/image-based 3D scene retrieval due to at least four 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) The problem itself is challenging to cope with (e.g., 3D scene data handling). 3) If the input queries are sketches or images, there is a big semantic gap between the 2D scene sketches/images 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/image queries a challenging, although interesting and promising, research direction.

To promote this research direction, and based on the **Scene250** benchmark [219] for sketches, and ImageNet [50], we built the **SceneSBR2018** and **SceneIBR2018** benchmarks [4, 225] and organized two SHREC'18 tracks on 3D scene retrieval in 2018. However, either **SceneSBR2018** or **SceneIBR2018** contains only 10 distinct scene classes. In order to make the benchmarks more comprehensive, after these two tracks, we have tripled [5, 222] the size of **SceneSBR2018** and **SceneIBR2018** in 2019, resulting in two extended benchmarks **SceneSBR2019** and **SceneIBR2019**, which have 750 2D scene sketches, 30,000 2D scene images and 3,000 3D scene models. Similarly, all the 2D scene sketches, 2D scene images and 3D scene models are equally classified into 30 classes. We have kept the same set of 2D scene sketches, 2D scene images and 3D scene models belonging to the initial 10 classes of **SceneSBR2018** and **SceneIBR2018**.

However, all the deep learning-based participating methods from the tracks in 2018 and 2019 are data-driven based techniques, which consider only the feature vectors (shape descriptor, position relationship of pixels). These techniques work well on 3D object retrieval, because objects belonging to the same category have similar shapes. But in 3D scene retrieval, scenes belonging to the same category may have quite different shapes, thus, the accuracy to retrieve 3D scene models is usually much lower without using other techniques.

In order to improve the 3D scene retrieval accuracy, we introduce the concept of semantics into the retrieval process. The semantic information of a 3D scene is that the objects appear in a certain scene often have object-object and object-scene relatedness information. For example, in a classroom scene, chair and desk objects have a high probability of appearance, because they are highly correlated in the context of a classroom scene. In addition, by this means, we can also improve the object detection performance for scene sketches, images, and models. For instance, if an elephant object is detected in a classroom scene, it will raise an alarm for mistakes. Therefore, by incorporating semantics information, we can increase the 3D scene retrieval accuracy.

In addition, due to lack of available high-quality 2D scene sketch data, there is an urgent need to curate a large-scale scene sketch dataset for 2D scene sketch-based 3D scene retrieval. Currently available and related scene sketch/contour datasets are either too small in terms of size or limited in within-class variations in terms of quality. Even by using Amazon Mechanical Turk, collecting/generating a large number of scene sketches

for training deep learning models for 2D scene sketch-based 3D scene retrieval is still a challenging task. Therefore, we proposed a Generative Adversarial Network (GAN)-based scene sketch generation approach to automatically generate 2D scene sketches by utilizing the existing large amount of 2D natural scene images.

1.2 Project Overview

The project aims at building a richly-annotated hierarchical scene database, named SceneNet, to support related large-scale scene understanding applications, especially large-scale 3D scene retrieval. Based on SceneNet, we propose a large-scale 3D scene retrieval framework which generates good retrieval performance in terms of accuracy, efficiency and extensibility, for a variety of applications including 3D printing, animation production, and virtual reality (VR)-based applications, such as online touring. 3D scene retrieval is to retrieve man-made 3D scene models given a user's hand-drawn 2D scene sketch or a 2D scene image usually captured by a camera. There is a gap in their domains: 3D scene models or views differ from either non-realistic 2D scene sketches or realistic 2D scene images. In fact, due to the even bigger representation gap between rough 2D scene sketch representation and accurate 3D scene model coordinates, 2D scene sketch-based 3D model retrieval (SceneSBR) is one of the most challenging research topics in the field of 3D scene retrieval. To bridge the gap in semantics (i.e. categories) due to their diverse representations for even the same 3D real scene, a novel **semantic tree-based 3D scene retrieval framework** is proposed in this project. The proposed approach can effectively capture semantic information of 2D scene sketches/images and 3D scene models, accurately measure their similarities, and therefore greatly enhance the retrieval performance. We also adapt the retrieval framework for related applications research in semantics-driven 2D scene sketch/image-based 3D scene retrieval. The project will accelerate the speed of applying 3D scene retrieval techniques in related large-scale applications by providing an infrastructure to perform information search at semantic level, like Google.

Content-Based 3D Model Retrieval. 3D models consist of 3D data (typically a list of vertices and faces) to represent 3D objects. 3D models are widely used in a lot of fields, such as industry product design, visualization and entertainment, 3D modeling, rendering, and animation. In recent years, the number of 3D models keeps increasing drastically, which triggers urgent research tasks and a lot of research interests in developing effective and

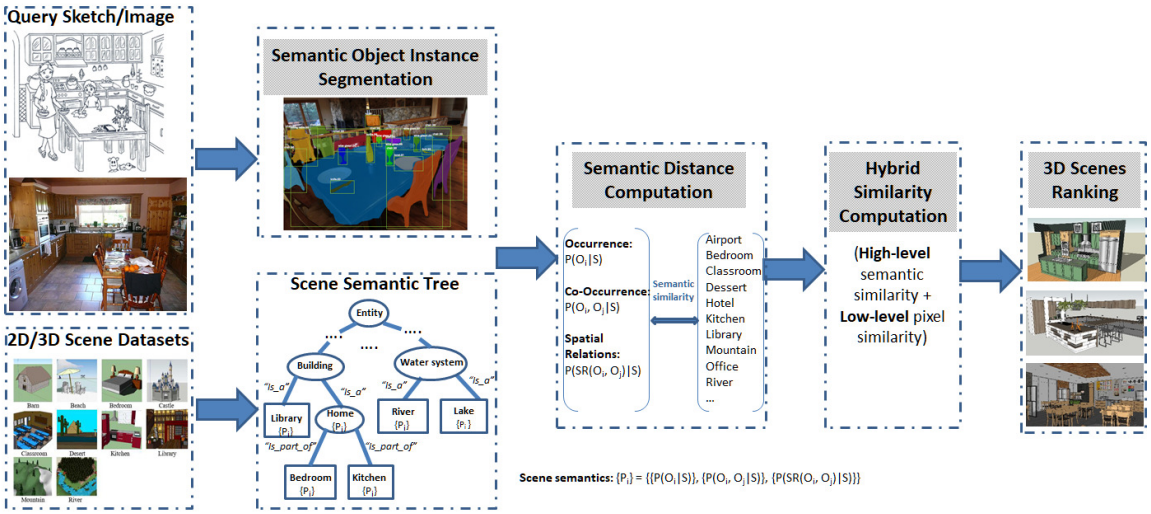


Figure 1.1: Semantic sketch/image-based 3D scene retrieval framework. Scene semantics of a particular scene category S contain the following three probability distributions: (1) **Object occurrence probability** $P(O_i|S)$ is the conditional probability that an object class O_i appears in S; (2) **Object co-occurrence probability** $P(O_i, O_j|S)$ is the conditional probability that both of two object classes O_i and O_j appear simultaneously in S; (3) **Spatial relation probability** $P(SR(O_i, O_j)|S)$ is the conditional probability that two object classes O_i and O_j have a certain spatial relation (SR, a spatial preposition) in S , e.g., $SR(O_i, O_j) =$ support / surround / near, that is, O_i supports / surrounds / is near to O_j.

efficient 3D shape retrieval algorithms for related applications. Given a query which is often a 2D sketch/image or a 3D model, content-based 3D model retrieval is to retrieve relevant 3D models (typically only single object models) coming from the same category as the query, and then rank them in the front part of the rank list as much as possible, while at the same time pushing irrelevant 3D models to the back of the rank list. Effectiveness, efficiency, and scalability are three most important performance metrics, which can be measured by a set of performance metrics commonly used in the field of information retrieval.

Most existing content-based 3D model retrieval algorithms target on single 3D object model retrieval, we are the **first** who built the first 3D scene retrieval benchmark [7, 227] based on either 2D scene image or sketch queries. We are also the first group who **pioneered** this research direction by organizing two 2018 Eurographics Shape Retrieval Contest (SHREC) tracks [4, 225].

2D Scene Sketch-Based 3D Scene Retrieval is to retrieve relevant 3D scenes using a 2D scene sketch as input. This scheme is intuitive and convenient for users to learn and

search for 3D scenes. It is also very promising and has great potentials in many applications such as autonomous driving cars, 3D scene reconstruction, 3D geometry video retrieval, and virtual reality (VR) in 3D Entertainment like Disney World's Avatar Flight of Passage Ride [21, 186, 201]. While, existing 3D model retrieval algorithms have mainly focused on single object retrieval and have not handled retrieving such 3D scene content, which involves a lot of new research questions and challenges. Considering its vast application scenarios, we believe that this research topic does deserve our further explorations and will raise more and more interests and attentions from both inside and outside of the 3D object retrieval research community.

In addition, there are many existing 2D sketch-based 3D shape retrieval algorithms that usually targets the problem of retrieving a list of candidate 3D models using a single sketch as input, there is little existing research work on 2D scene sketch-based 3D scene retrieval. 2D scene sketch-based 3D scene retrieval is a brand new research topic in the field of 3D object retrieval. Unlike traditional sketch-based 3D model retrieval which ideally assumes that a query sketch contains only a single object, this is a new 3D model retrieval topic within the context of a 2D scene sketch which contains several objects that may overlap with each other and thus be occluded and also have relative location configurations. It is challenging due to the semantic gap existing between the iconic 2D representation of scene sketches and more accurate 3D representation of 3D models.

2D Scene Image-Based 3D Scene Retrieval is also a new research topic in the field of 3D object retrieval. Given a 2D scene image, it is to search for relevant 3D scenes from a dataset. It has an intuitive and convenient framework which allows users to learn, search, and utilize the retrieved results for vast related applications. For example, automatic 3D content generation based on one or a sequence of captured images for AR applications, or 3D cartoon animation production, robotic vision (i.e. path finding), and consumer electronics apps development, which facilitate users to efficiently generate a 3D scene after taking an image of a real scene. It is also very promising and has great potentials in other related applications such as 3D geometry video retrieval, and highly capable autonomous vehicles like the Renault SYMBIOZ [164] [187]. However, similarly there is little research in 2D scene image-based 3D scene shape retrieval [136] [216] 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.

Project Goal & **Approach.** Therefore, now we can find that there is either a big **semantic gap** between 2D scene sketches and 3D scene models for 2D scene sketch-based 3D scene shape retrieval, or a scarce of substantial research in 2D scene image-based 3D scene shape retrieval due to its **challenges and difficulties**. In this project, we propose a semantic-tree based large scale 3D scene retrieval strategy to both bridge the semantic gap and overcome the challenges. Assisted by a semantic tree for 2D/3D scenes, we will develop a new sketch/image-based 3D scene search approach, which will 1) be able to accurately retrieve similar 3D scenes given users' sketches/images; 2) have low computational costs, suitable for real-time online/mobile applications; 3) be scalable to large-scale 3D scene retrieval.

Although the representation of 2D sketch/images and 3D scene differ a lot, there exists a common thing shared by them, which is semantic information. Semantic information describes high-level representation of 2D sketches/images and 3D scenes. It provides a possible bridge to reduce the representation gap among them. Motivated by this, an interesting question is raised: "Can we use semantic information to bridge the semantic gap?" Proposed work concentrates on semantics-based 3D scene retrieval framework.

Semantics-Based 3D Scene Retrieval Framework. We propose a comprehensive and novel semantic tree-based 3D scene retrieval framework, as illustrated in **Fig. 1.1**. Given a 2D query sketch/image and a dataset of 3D scene models, we will first build up a scene semantic tree in two steps. **Step 1:** Build a scene semantic tree based on the semantic ontology in WordNet [140]. Then, classify collected 2D and 3D scene models into certain nodes in the tree according to their semantic classification/label information (i.e. semantic concepts or names). **Step 2:** Next, we will identify the semantic attributes (i.e. object labels contained in the scenes) that the 2D query sketch/image contains via a deep learning-based classification approach. **Step 3:** Finally, by measuring the semantic similarity between the 2D sketch's or 2D image's semantic attributes and the nodes in the semantic tree, we will compute the similarities between the 2D sketches/images and 3D scenes to find out the most relevant 3D scenes.

1.3 Overview of the Approach

1.3.1 Challenges in Existing 3D Model Retrieval Techniques.

Previous 3D model search approaches mainly target a Query-by-Model framework (using existing 3D models as queries) due to its simplicity. However, this search strategy is not a human-natural way to search 3D models and therefore cannot satisfy the need of many

applications. For example, when human design a product, an architecture, or a cartoon animation role/scene, human usually sketch it on a paper or on a touch device (i.e. cellphone, pad, or laptop) to search for its similar 3D models. In virtual reality and 3D games, people would like to involve their hands/control console in creating 3D scenes based on existing 3D models. Hence, retrieving 3D models based on human sketches becomes a desired and necessary searching strategy. It will provide a valuable tool for a lot of applications including human computer interaction, 3D cartoon animation, game design, and virtual reality etc.

In recent years, more than a dozen of sketch-based 3D model retrieval algorithms [103, 105, 108–113] have been proposed. Most methods render a large number of 2D views from 3D models and match a 2D query sketch with 2D views. That is, given a 2D sketch query and a 3D model dataset, a set of views will be sampled first for each 3D model, then extract a 2D shape descriptor for each view and the query sketch. Next, the minimum shape descriptor distance between the query sketch and all the sample views is computed and it is regarded as the sketch-model distance. Finally, all the target 3D models are ranked by sorting the sketch-model distances.

However, such kinds of methods usually suffer from *high computational cost* and *low retrieval accuracy* [109, 112]. This is due to a big **semantic gap** between 2D sketch and 3D model. Human sketches always have arbitrary styles, iconic representations in 2D space, high-level abstraction, and drastic simplification, which bring a lot of difficulties in sketch description and representation. A 3D model of an object is generally an accurate representation of its geometry information. Such difference between the representation of 2D sketches and 3D models produces a big semantic gap, which make the search based on a direct 2D-3D comparison extremely difficult even if we sample views densely. Therefore, it is still a very challenging task for existing algorithms to achieve outstanding performance [109, 112], in terms of both effectiveness and efficiency, especially when applied on a large-scale sketch/image-based 3D scene retrieval scenario.

1.3.2 WordNet and WordNet-Based Semantic Multimedia Retrieval.

WordNet [140] is a lexical database of concepts/synsets, represented by a set of synonyms. Each node in the tree represents one word, which has one or more senses (meanings). Each sense has its synset and a set of words are related through the following three relationships: hypernyms/hyponyms (IS_A relation), holonyms (MEMBER_OF relation) and meronyms (PART_OF relation), as demonstrated in **Fig. 1.2**. As a lexical dictionary of semantic

concepts, WordNet has been vastly applied in semantic multimedia retrieval of either text or image objects. Aslandogan et al. [20] utilized WordNet for query expansion in image retrieval. They considered synonyms of nouns and verbs, different number of senses of a word, and other three relationships (IS_A, MEMBER_OF, and PART_OF) mentioned before. They found that for query expansion the optimal setting is using synonyms of all senses, or considering the synonyms and the IS_A and MEMBER_OF relations of the first sense of a word. Marszalek and Schmid [133] proposed to utilize WordNet to build a semantic and hierarchical graph for the objects involved in recognition. Based on labeled training data, they learned a binary classifier for each node in the graph. Wang et al. [198] proposed to build an ontology based on WordNet for a 3D model benchmark, infer 3D semantic properties by rule engine based on Semantic Web Rule Language (SWRL), and perform semantic retrieval using the ontology. A survey on three typical semantics processing (relevance feedback, machine learning, and ontology) has been performed in [65], while Tousch et al. [188] presented a survey on semantics-based image labeling. According to our previous related research experience [104, 108, 111], class-based or semantic information-based 3D model retrieval is a promising approach to improve retrieval accuracy, especially for certain performance metrics like NN and FT since we can push forward more 3D models that have been correctly classified into one class, to the front part of a retrieval rank list. There are some other important related applications employing 2D/3D semantics, such as scene recognition [237], 3D reconstruction and segmentation [220], and part-based 3D retrieval [64].

1.3.3 Scene Semantic Tree Definition.

WordNet [140] provides a broad and deep taxonomy with over 80K distinct synsets representing distinct noun concepts arranged as a directed acyclic graph (DAG) network of hyponym relationships (e.g., "table" is a hyponym of "furniture"). As shown in **Fig. 1.1**, a scene semantic tree is a hierarchy of classes with corresponding 2D sketches, images and 3D scene models organized based on the semantic hierarchy in WordNet synsets. Each class (synset) of the scene semantic tree has several attributes (i.e. is-a, has-part, or is-made-of relations) according to its gloss defined in WordNet. Each leaf node of the scene semantic tree has a number of 2D sketches, images and 3D scene models belonging to the leaf node class. It also contains scene semantics information for the scene class. For a particular scene category S, we form its scene semantics based on the following three probability distributions.

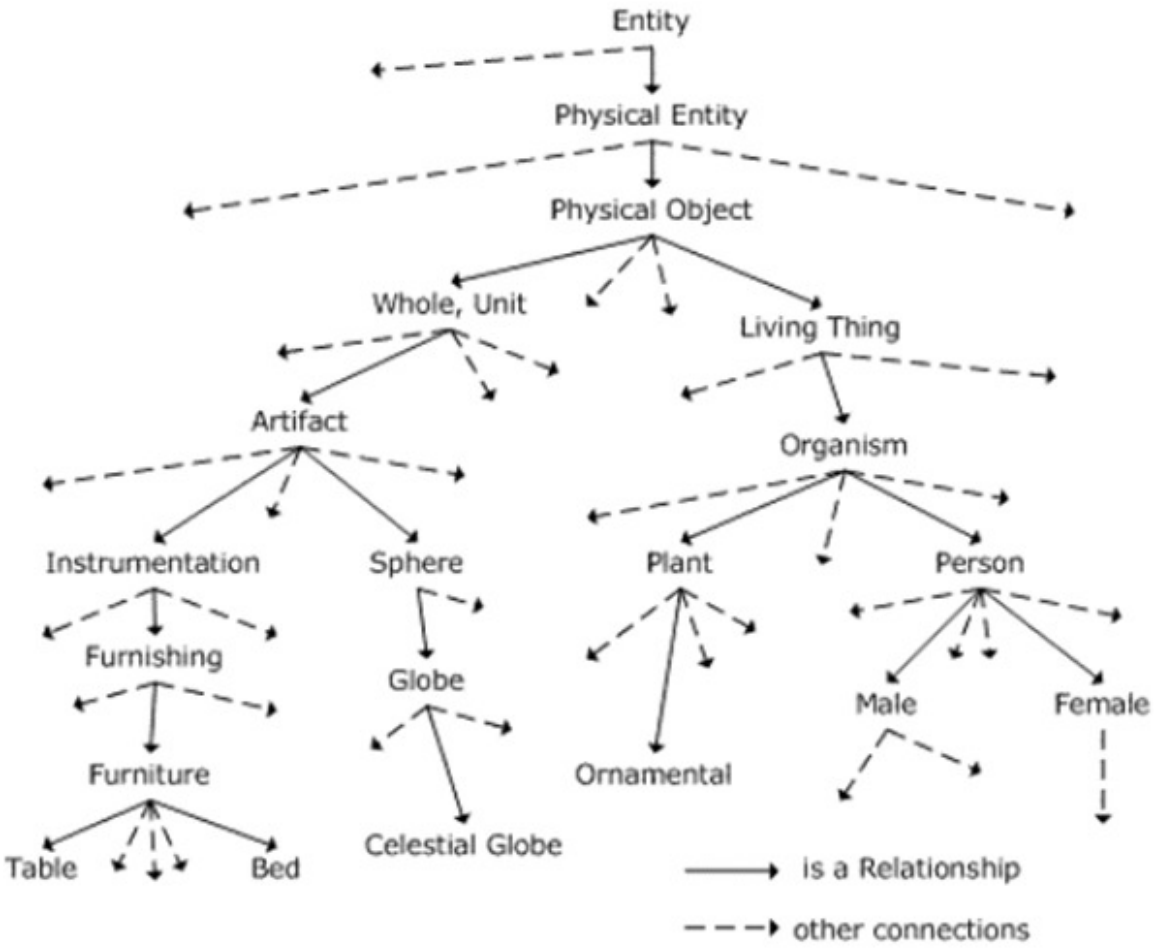


Figure 1.2: WordNet ontology example.

1. **Object occurrence probability** $P(O_i|S)$: the conditional probability that an object class O_i appears in S;

2. **Object co-occurrence probability** $P(O_i, O_j|S)$: the conditional probability that both of two object classes O_i and O_j appear simultaneously in S;

3. **Spatial relation probability** $P(SR(O_i, O_j)|S)$: the conditional probability that two object classes O_i and O_j have a certain spatial relation (SR, a spatial preposition) in S , e.g., $SR(O_i, O_j) =$ support / surround / near, that is, O_i supports / surrounds / is near to O_j.

Therefore, the scene semantic tree forms a network of classes, attributes (i.e. scene object categorical names and their statistics) and related scene files (images, sketches and models).

1.4 Thesis Organization

I first conduct a a literature review in Chapter 2. Chapter 3 presents a comparison of methods for 3D scene shape retrieval from the related four SHREC'18 and SHREC'19 tracks based on the benchmarks built by us. Chapter 4 presents a probabilistic deep learning scene semantics generation method, a semantics-based 3D scene retrieval approach, as well as the results and our findings. Chapter 5 demonstrates how to utilize adversarial networks to automatically generate 2D scene sketches from 2D scene images. Finally, I draw a conclusion and propose our initial algorithm designs for the future work in Chapter 6.

1.5 Summary

Sketch/Image-based 3D scene retrieval are brand new, interesting, and challenging research topics with a lot of application potentials. There is extremely limited preliminary work in this field, which allows us to explore many promising ideas and interesting results. In this project, we mainly propose a semantics-driven large-scale 3D scene retrieval framework which builds on a richly-annotated hierarchical scene database that has been curated by us.

Chapter 2

LITERATURE REVIEW

With ubiquitous cameras and popular 3D scanning and capturing devices to help us capture 2D/3D scene data, there are many scene understanding related applications, as well as quite a few important and interesting research problems in processing, analyzing, and understanding the available scene data. During the recent several years, there is significant advancement in different research directions in this field and quite a few novel 3D scene analysis and processing methods have been proposed correspondingly in each direction. This dissertation provides a review and critical evaluation on the most recent (i.e., within five recent years) and novel data-driven or semantics-driven 3D scene analysis and processing methods, as well as several involved 3D scene datasets. For each method, its advantage(s) and disadvantage(s) are discussed, after an overview and/or analysis of the approach. Finally, based on the review, we propose several promising future research directions in this field.

2.1 Introduction

Nowadays, more and more different types of 3D sensing devices could help us capture 3D scene data, such as Acuity Laser [9], Light Detection and Ranging (LIDAR) [204], Leap Motion [203], etc. Those captured 3D scenes include not only indoor scenes, but also outdoor scenes. In addition, in order to deal with different situations or meet different research requirements, researchers have built different benchmarks [46], [234], [183], [191], and [72]. [46], [183] and [191] are built for 3D indoor and/or outdoor scene research, while [234] and [72] are more accurate and more comprehensive than previous benchmarks.

We organized 4 3D scene retrieval SHREC tracks [225], [4], [222], and [5] in 2019 and 2020. In those tracks, we built a **SceneSBRIBR** benchmark which has 3,000 3D scene models for all the participants train and test their models, as organizers, we proposed a method as well.

From all the tracks, we realized that in the field of artificial intelligence, the demand for 3D scene analysis and processing area are getting higher and higher. Compared to 3D

objects, 3D scenes are more directly related to our daily life. There are a large amount of real life relevant applications, such as autonomous driving cars, 3D geometry video retrieval, 3D AR/VR Entertainment, etc. Therefore, recently researchers have proposed many 3D scene analysis and processing methods and contributed significantly to this research area. 3D scene retrieval is one part of this area. The research directions within this area include: a) 3D scene classification, which is to classify the 3D scene models into different certain categories; b) 3D scene recognition, which is to recognize the category of a given 3D scene; c) 3D scene retrieval, which is to retrieve 3D scene models given an input query (i.e., a 2D scene sketch/image) provided by the users; d) 3D scene reconstruction, which is to reconstruct three-dimensional scene models from multiple 2D projected scene images, whose depth information may be missing; e) 3D scene generation, which is to generate 3D scene models from 2D images or nature languages (e.g., "a person besides a table").

These research directions may involve either data-driven or semantics-driven based techniques: a) **Data-driven methods** are the methods that are based on the original raw data or the data preprocessed by some techniques like redundant data points reduction, error data removal [71], GPU parallel calculating, etc. b) Correspondingly, **Semantics-driven methods** are the techniques that are not only based on the data, but also incorporate semantic information of the objects or context in the 3D scenes. For examples, [157] proposed a 3D scene classification method based on semantic labels extracted from 3D scenes. [14] presented a web-based 3D room layout generation system, which utilizes the semantic information of each furniture in a 3D room and each furniture's related objects. In addition, to improve reconstruction accuracy, [192] reconstructed a 3D scene by fusing the 3D map with the semantic information of each objects in the scene, etc.

Section 2.2 provides an overview by defining several typical related terminologies, and summarizing the papers to be reviewed in the survey. Sections 2.3$\sim$2.8 introduce and review each direction individually. Finally, after a conclusion, several promising future work directions are proposed in Section 2.11.

2.2 Terminologies

In this section, we first provide a definition for the most commonly used terminologies in 3D scene analysis and processing techniques.

3D Scene. In computer world, we define a 3D scene as an arrangement of scenery objects and properties to represent a recognizable place, where the objects that appear, and

their shapes, sizes, and spatial relationships, as well as the background (i.e., ground, and sky) are important features to characterize the place.

3D Scene Shape Representations. 3D scene contains a list of objects, which are entitled independent representations to represent their shapes and textures. To represent and easily maintain the semantic relationship between the objects in a scene, a scene graph data structure is often used. People have developed quite a few 3D object representations to meet the needs of practical applications, for example, (1) meshes; (2) point sets; (3) Spline surfaces; (4) Volumetric representations (i.e., voxels, particle systems, and finite element method (FEM); (4) Subdivision surfaces (i.e., Loop subdivision surface [127]); (5) Constructive solid geometry (CSG) (define a shape based on boolean operations on simple objects); and (6) Implicit surfaces (a surface defined by a mathematical equation).

Besides the above representations, RGB-D is a popular 3D scene representation to represent 3D scenes captured by various 3D capturing and sensing devices.

3D Scene Features. We can divide 3D scene features into low-level 3D scene features and high-level 3D scene features. **Low-level 3D Scene Features:** characterize a 3D scene at a lower level, e.g., pixel level, by focusing one details like color, texture, shape (e.g., lines, dots), spatial location, etc. **High-level 3D Scene Features:** represent a scene at a higher level, e.g., object-level or object-group level, by examining the spatial and semantic relationships between the objects in the scene.

3D Scene Semantics Information. Semantics information is used to interpret a special entity. There are a lot of semantic information (i.e. objects, object parts and object groups) existing in 3D scene models. To improve 3D scene analysis and processing accuracy, we could incorporate such semantic information into the learning process.

3D Scene Datasets. A 3D scene dataset is a collection of 3D scene data spanning over different categories, and often contains both training and testing subsets. Different 3D scene datasets are built for different purpose, e.g., Cordts et al. [46] released a Cityscapes dataset for urban street 3D scene analysis, while Vasiljevic et al. [191] curated a Dense Indoor and Outdoor DEpth (DIODE) dataset for both indoor and outdoor 3D scene analysis.

In this dissertation, we review very recently (i.e., within five recent years) published thirty-five (35) papers related to the five research directions (3D scene classification/ recognition/ retrieval/ reconstruction/ generation). We further group them into two different inputs (2D and 3D), as well as two types of approaches (data-driven and semantics-driven). Table 2.1 gives the overview of the above information. In the following five sections, we will review each of the five research directions individually.

Table 2.1: Overview of the thirty-five (35) 3D scene analysis and processing papers reviewed in this dissertation w.r.t different research directions, inputs, and approaches.

Tasks	**Input (2D)**	**Input (3D)**	**Data-driven**	**Semantics-driven**
classification	[3],[38],[40],[50],[58]	[12],[32]	[3],[32],[38],[50],[58]	[12],[40]
recognition	[6],[67]	[37],[45],[65],[69]	[6],[45],[69]	[37],[65],[67]
retrieval	[44]	[63](methods 1∼3), [64]	[44],[63](methods 1∼2), [64]	[63](method 3)
reconstruction		[8],[15],[18],[22] [23],[35],[39],[41],[42] [43],[47],[54],[62]	[8],[15],[18] [22],[23],[39],[41] [42],[47],[62]	[35],[43],[54]
generation	[5],[31],[34]	[21],[57],[66]	[66]	[5],[21],[31],[34],[57]

2.3 3D Scene Classification

Given a 3D scene model, 3D scene classification is to classify this scene model into one of the candidate categories.

2.3.1 Data-driven 3D Scene Classification

A variety of data-driven based methods have been proposed and many of them work well under certain circumstances.

Steinhauser et al. [182] proposed a scene classification method based on the data collected from a LIDAR laser scanner. It can be used to collect and classify the raw data of the real time surrounding environment of the vehicle into safe condition for driving road and unsafe obstacles (static obstacles or moving obstacles). They tested the method on the university campus and forest tracks, and the approach generates good results in estimating safe road (e.g., untarred road). However, it still has some places to be improved: (a) need to reduce the time cost of the method so as to run in real time, (b) make the method be able to deal with the LIDAR failure issues (e.g., scanner may fail in a small degree range, like 10 degrees), (c) the algorithm may not work well if there are quite a few moving cars around the vehicle, or when only a small number of landmarks are visible or the trees beside the road are dense and hard to distinguish them from each other.

Naseer et al. [146] 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.

Lin et al. [121] proposed a method for indoor scene understanding based on RGB-D data.

They utilized the Constrained Parametric Min-Cuts (CPMC) [35] framework to generate candidate cuboids for the 3D objects in a 3D scene, and then classify these cuboids. With 2D segmentation information, 3D geometry properties, and the contexture relationship between objects and scenes integrated in this method, the 3D object and 3D scene classification can be solved together. Compared to the Part Based Model DPM [59], their method made a good performance improvement on the NYU v2 dataset [172]: the F1-score accuracy, which is the harmonic mean value of the precision and recall [52], has been increased considerably.

Wang et al. [196] proposed two contributions to solve the two issues existing in scene recognition/classification: (a) large intra-class variations; (b) label ambiguity. Firstly, they proposed a multi-resolution CNN architecture, which consists two parts: (a) coarse-resolution CNNs, which deal with global features and large objects in the scene; (b) fine-resolution CNNs, which deal with local features and small objects in the scene. They are complementary to each other. Secondly, for the label ambiguity issue, they adopted two ways to deal with it: (i) utilizing a confusion matrix technique (by computing the similarity between any two categories), which can put those ambiguous scene categories into one super category (e.g., outdoor athletic, and outdoor track scenes); (ii) using other networks to predict the label of each scene, which is called soft label. Then train the model with the guidance of super category labels and soft labels. However, there still exist some failure examples: some scene categories still cannot be distinguished with each other easily, e.g., supermarkets and top shops are similar if looked from outside.

Aiger et al. [12] proposed a multi-view based CNN model, which has a good accuracy in classifying water and trees. Compared with the state-of-the-art model Inception-V3 [185], the related accuracy has been increased from 79% to 96%. The method requires neither fully-segmented labels, nor marked object class boundaries in the scene image, while it only requires sparsely labeled pixels.

Muller-Budack et al. [144] treated the geolocalization (subdividing the earth into multiple geographical cells) of a photo as a scene classification problem. To incorporate the hierarchical knowledge of different spatial resolutions, they adopted a multi-partition CNN model, which can be used to compute geolocalization loss. Moreover, they extracted the scene labels information from different scene types (indoor, nature, urban, etc) by using the ResNet model [77], and incorporated the information into the multi-partition CNN model as well. They ran their method on two benchmarks Im2GPS [75] and Im2GPS3k [194], and compared with the PlaNet [200] approach and demonstrated that their method has improved the classification accuracy. This CNN model requires a small number of training images and

does not rely on the retrieval results from any datasets for verification. To further improve the geolocalization, they could also incorporate other contextual information into the CNN model, such as specific landmarks, image styles, etc.

Mohammad and Hossein [156] presented a geometric features-based system for 3D model categorization. The system extracts the geometric features from faces and vertexes. And a histogram of geometrical features are used for the 3D models classification. The histogram includes two variables: (1) deviation angle of vertex normal vector from center-to-vertex vector [156]; (2) distance of the center of the object to vertex. They also adopt mutual Euclidean distance histogram to improve the categorization accuracy, utilize the Probabilistic Neural Network (PNN) and Support Vector Machine (SVM) classifiers to compare their categorization result and speed.

2.3.2 Semantics-driven 3D Scene Classification

Unlike data-driven 3D scene classification that only focuses on the scene data itself, semantics-driven 3D scene classification also considers the semantic relatedness between objects, or between objects and scenes.

Since it is challenging for robotics to achieve a high accuracy in 3D indoor scene classification due to a large number of scene categories in related datasets, Chen et al. [40] proposed a word vector (a.k.a word embedding) based algorithm for the 3D indoor scene classification task. This algorithm first uses GPS to locate a robot's rough area, e.g., school, shopping mall, etc. Then it just needs to search the objects belonging to this area instead of searching all the object categories. They employed different CNN models for different purposes in their approach, which consists of four modules. The first is a typical CNN-based scene classification module to obtain the top-5 prediction labels. The second is a CNN-based scene parsing module which is to detect the objects, background and foreground in a scene. Next, the third module word embedding is to compute the vector for the objects in a scene image and the vector for the top-5 prediction labels. Finally, the fourth module refines the rank list of the top-5 labels based on the comparison of the above two vectors. They adopted ResNet50 as the CNN model. After incorporating the word vector information into the CNN model, they further increased its classification accuracies on both the Places365 dataset and their selected indoor scenes dataset, that is, the school, home and shopping mall scenes selected from the original Places365 dataset.

Rangel et al. [157] proposed a scene classifier based on the semantic labels recognized by the Clarifai [180] descriptor. This paper compares the Clarifai-based approach with other

descriptors (i.e., GIST, ESF), and shows that the Clarifai-based descriptor is competitive if compared with those state-of-the-art ones. Moreover, the Clarifai-based approach performs the best when dealing with general scenes. For example, after this approach is trained on the semantic sequences of one type of building scenes, it can obtain good classification results on the semantic sequences of another type of building scenes.

2.4 3D Scene Recognition

3D scene recognition is to recognize the category of a given 3D scene, unlike 3D scene classification, the candidate categories are not provided. Similar to 3D scene classification, 3D scene recognition can be categorized into data-driven and semantics-driven approaches.

2.4.1 Data-driven 3D Scene Recognition

Behl et al. [25] proposed a new system for estimating 3D scene traffic flow for autonomous driving. This system addresses the large displacement or local ambiguity (due to lack of texture or surface reflection) problems which can fail the estimation in existing methods. It is a recognition-based approach instead of like existing ones relying on local features. They conducted experiments on 2D bounding boxes calculation, 2D instance segmentations, and 3D object part predictions. The results demonstrated that the approach improves the performance by a lot when dealing with large displacement or local ambiguities.

Zhong et al. [235] proposed a method for 3D text recognition in 3D scenes. It helps in shadow detection and removal. This method segments shadow pixels from background and text pixels by utilizing the Gabor kernel, then removes their depth information, and finally converts the 3D texts into a 2D text image. Since it is the first attempt in 3D text recognition, there are still some room for improvement. For example, the thresholds determined by the Gabor kernel for shadow detection cannot achieve good performance where there are low contrast, small fonts, non-uniform illumination effects, and so on.

Shi et al. [170] proposed a variational denoising recursive autoencoder (VDRAE) system to predict the 3D scene layout of a 3D point cloud indoor scene, as demonstrated in **Fig.** 2.1. This system generates and denoises the predicted 3D object proposals by incorporating the hierarchical context information of 3D objects. The denoised indoor scenes can improve the 3D scene recognition accuracy. However, this system is not an end-to-end system. For example, the hierarchical proposals prediction and denoising steps are done separately.

Caglayan et al. [32] proposed a two-stage object and scene recognition framework, which

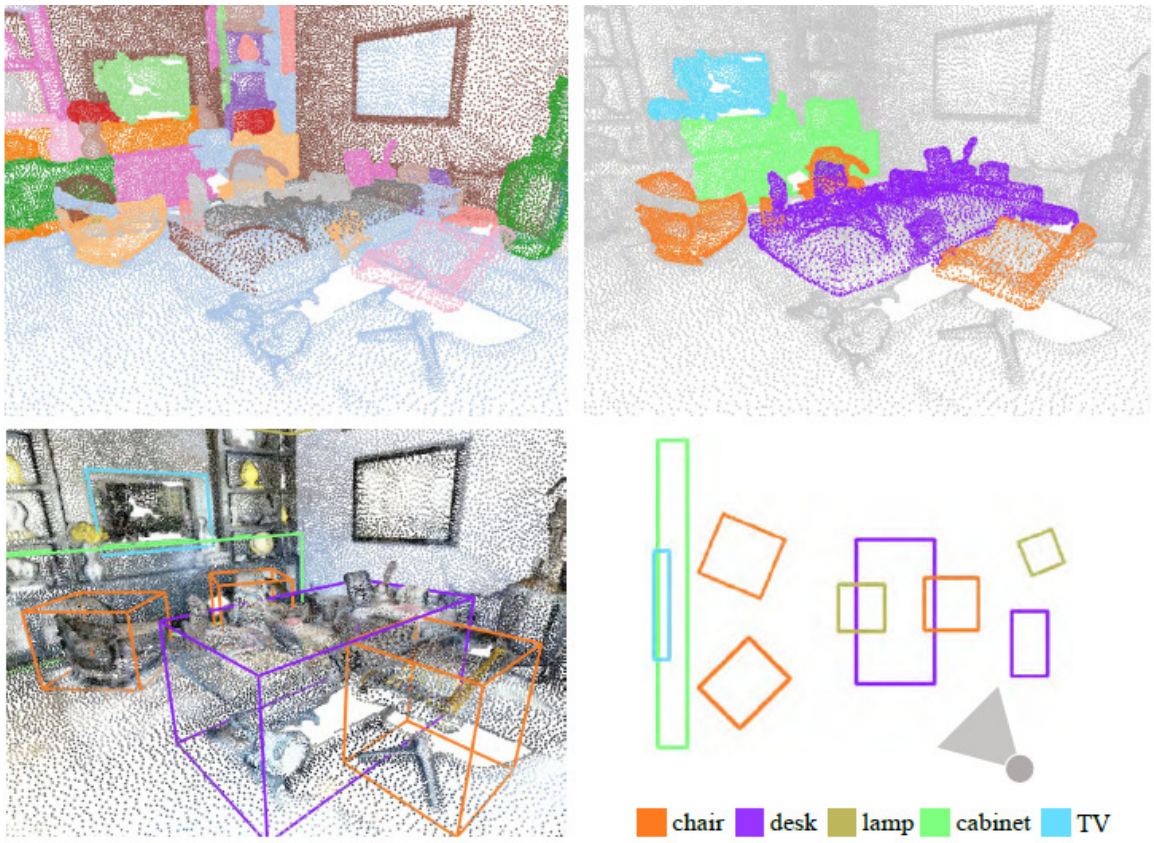


Figure 2.1: A VDRAE-based 3D object layout prediction example. Segmented 3D point cloud as input (top left), processed by VDRAE system (top right, make the objects in the same category have the same color), and make fully objects contained in 3D bounding boxes (bottom) [170].

can recognize scenes from RGB-D images captured by RGB-D sensors. The first stage is a pretrained CNN model that can extract different level visual features. The second stage utilizes multiple random recursive neural networks (RNNs) to map the extracted features into high-level representations. In addition, they extended the idea of randomness in RCNNs and proposed a randomized pooling schema to improve the recognition accuracy.

2.4.2 Semantics-driven 3D Scene Recognition

Zhao et al. [231] proposed a framework that can parse scene images at both pixel level and word concept level. They jointly embedded them into a high-dimensional positive vector space, as demonstrated in **Fig.** 2.2. At the word concept level, their framework incorporates

the semantic word-word relations, i.e., using a hypernym/hyponym based on WordNet [58]. They made rules for the space construction process: making the pixel level features close to their annotated labels and keeping the semantic relations unchanged. In general, their framework includes two streams: (a) **Concept stream**, which is to incorporate the semantic relationship information into the embedding space; (b) **Image stream**, which is to segment the image by using a fully convolutional network. Then, their framework combines the two streams by a joint loss function to measure the similarity in their image features and word concept hierarchies, while the weights of the two streams in the loss function are predefined. They selected 150 object categories from the ADE20K dataset [240] to train and test their framework based on certain evaluation measures, e.g., using weighted intersection-over-union (IoU) [202] as a baseline flat metric. They also compared their jointly embedding framework with other models, such as Word2Vec [138]. The results show that their framework has achieved better performance and demonstrated two main advantages: (a) It has more freedom for the user to label an object at different grained levels (e.g., Husky and dog categories) without sacrificing the training accuracy. (b) The system is end-to-end, thus the semantic relationship information can be extended easily in the system. Nevertheless, it also has some limitations that may affect its performance, such as: (a) the training data and target data are very different from each other; (b) compared to the label set, the size of the image dataset is too small.

Huang et al. [82] proposed a multitask learning-based [36] 3D indoor scene recognition method. This method classify the 3D indoor scenes based on 3D point cloud or voxels data instead of 2D images. They also incorporate the semantic object label information during the 3D indoor scene recognition process. By combing the geometry and the object label information of the scene, the multitask method can reach 90.3% accuracy on ScanNet dataset.

Miksik et al. [139] presented an augmented reality system for 3D outdoor scene recognition and reconstruction. This system simulates the 3D outdoor scene map in real-time and allows users to segment objects manually. With a machine learning model learned from the existing 3D object/scene datasets and objects drawn by the users, the system can recognize the scene in a more accurate way. The limitations of this system are in three-fold: (a) the computational load is heavy; (b) it needs powerful GPUs, which limits the laptop usage for outdoor scenes; and (c) the learning and prediction processes require users' voice commands to switch, and these two functions cannot be used at the same time, while in the mean time the feedback of the two processes could amplify the errors and decrease the

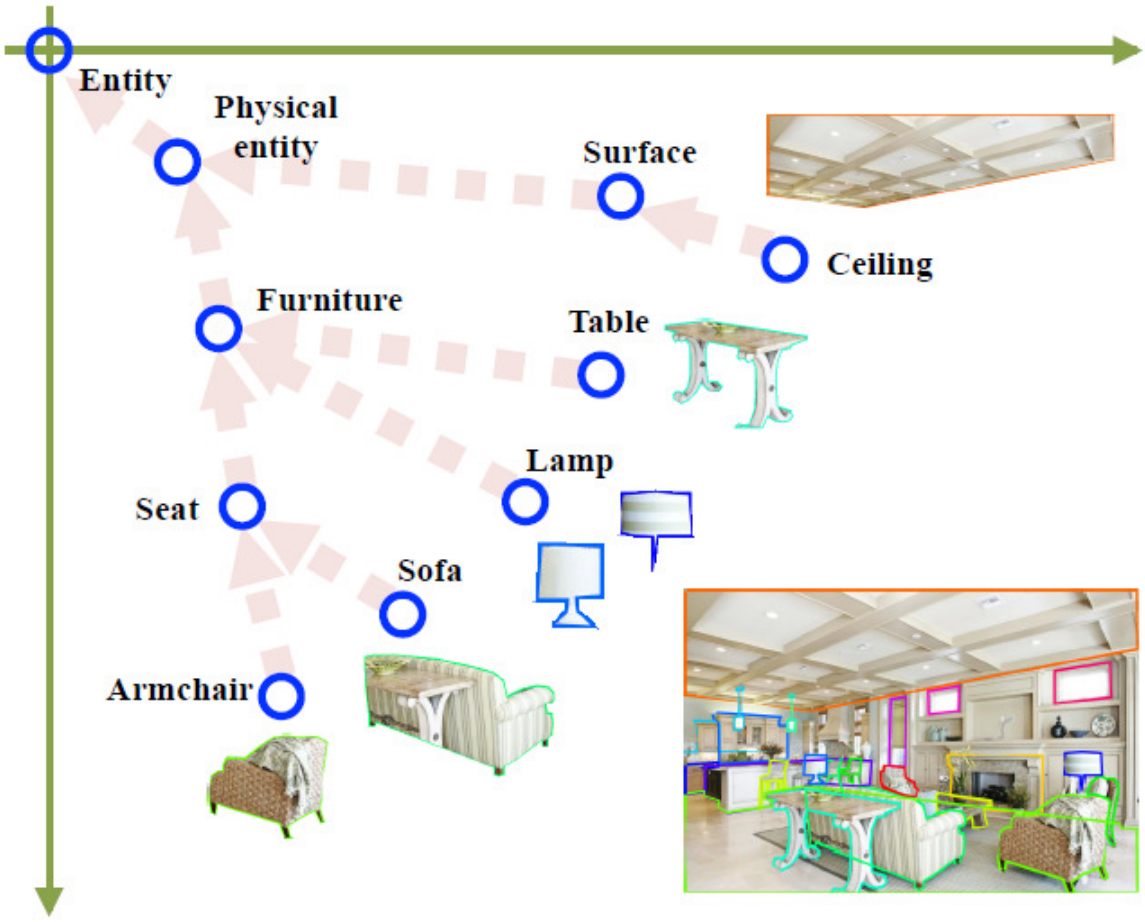


Figure 2.2: Embed image pixel features and word concepts jointly [231].

accuracy.

Yuan et al. [228] proposed a semantic tree-based framework for 3D scene model recognition. Firstly, this framework builds a scene semantic tree based on the semantic ontology in WordNet [58]. Secondly, the framework can identify the semantic attributes (e.g., object labels contained in the scenes) that the 2D query image contains via a deep learning-based recognition approach. And Finally, by measuring the semantic similarity between the 2D image's semantic attributes and the nodes in the semantic tree, the framework could recognize the target 3D scene categories. In this framework, the scene semantics of a particular scene category contain three probability distributions: (a) object occurrence probability, it is the conditional probability that an object class appears in the scene category, (b) object co-occurrence probability, it is the conditional probability that both of two object

classes appear simultaneously in the scene category, and (c) spatial relation probability, it is the conditional probability that two object classes have a certain spatial relation in the scene category.

2.5 3D Scene Retrieval

3D scene retrieval is to retrieve 3D scene models given an input query provided by the users. This research topic has vast applications such as 3D scene reconstruction, 3D geometry video retrieval, and 3D AR/VR entertainment.

2.5.1 Data-driven 3D Scene Retrieval

Savva et al. [169] advised a system to design and help retrieve 3D indoor scenes. This system is based on a large-scale learned 3D priors set which is extracted from existing 3D scenes. These priors are related to static support, position, and orientation. Moreover, by using those priors, this system provides suggestions for 3D object placement and assembles 3D objects with regard to desired scene category. However, this system does not consider collision detection between two objects, which may lead to incorrect placement.

Hoàng et al. [80] 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.

Yuan et al. [223] proposed a sketch/image-based 3D scene retrieval algorithm. The input query of the approach is a user's hand-drawn 2D scene sketch or a 2D scene image. This method represents a 3D scene model by multiple 2D view images sampled from different viewpoints. Then, they train two CNN models separately on the 2D scene sketches/images, and the scene view images, as shown in **Fig.** 2.3. Finally, the ranking is based on the two CNN classification results on the corresponding testing datasets.

Li et al, one of the participant groups in two Shape Retrieval Contest 2019 (SHREC'19) tracks on 3D scene retrieval tracks [5, 222], presented the Maximum Mean Discrepancy domain adaption method based on the VGG model (MMD-VGG) to tackle 3D scene retrieval task. The query is a 2D scene sketch/image and the target is 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.

Liu et al, one more participant group in the two SHREC tracks, proposed a two-stream CNN-based method. In their method, the 2D scene sketch/images dataset is regarded as

the source domain, and the 3D scene models dataset is regarded as the target domain. It processes samples from either domain with a corresponding CNN stream. They adopted triplet center loss [78] and softmax loss for training supervision, the network is trained to learn a unified feature embedding for each sample, which is then used for similarity measurement for the retrieval process.

2.5.2 Semantics-driven 3D Scene Retrieval

Minh-Triet Tran et al, another participant group in the two SHREC'19 tracks on 3D scene retrieval, proposed a domain adaptation based method named ResNet50-Based Sketch Recognition and Adapting Place Classification for 3D Models Using Adversarial Training (RNSRAP). In addition to the training dataset provided, they performed data augmentation by adding semantic related sketches/images (e.g., add camel, cactus sketches/images to the desert category). Due to the substantial variance exists in the two domains (source domain and target domain), the adversarial adaptive method they utilized is to minimize the variance between the source and target domains. As a result, the trained domain adaptation model can be used for classification in both the source and target domains.

Fisher et al. [60] 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.

2.6 3D Scene Reconstruction

Similar to 3D object reconstruction, 3D scene reconstruction is to reconstruct a three-dimensional scene model from multiple 2D projected scene view images, whose depth information needs to be recovered.

2.6.1 Data-driven 3D Scene Reconstruction

Hossein and Hassan [56] proposed a space curve-based method that can reconstruct a moving three-dimensional (3D) object from stereo rigs captured image sequences. Space curves are extracted from the stereo images, this method ensures accurate geometry, minimize the number of outliers, and photometric information is not required by adopting the new space

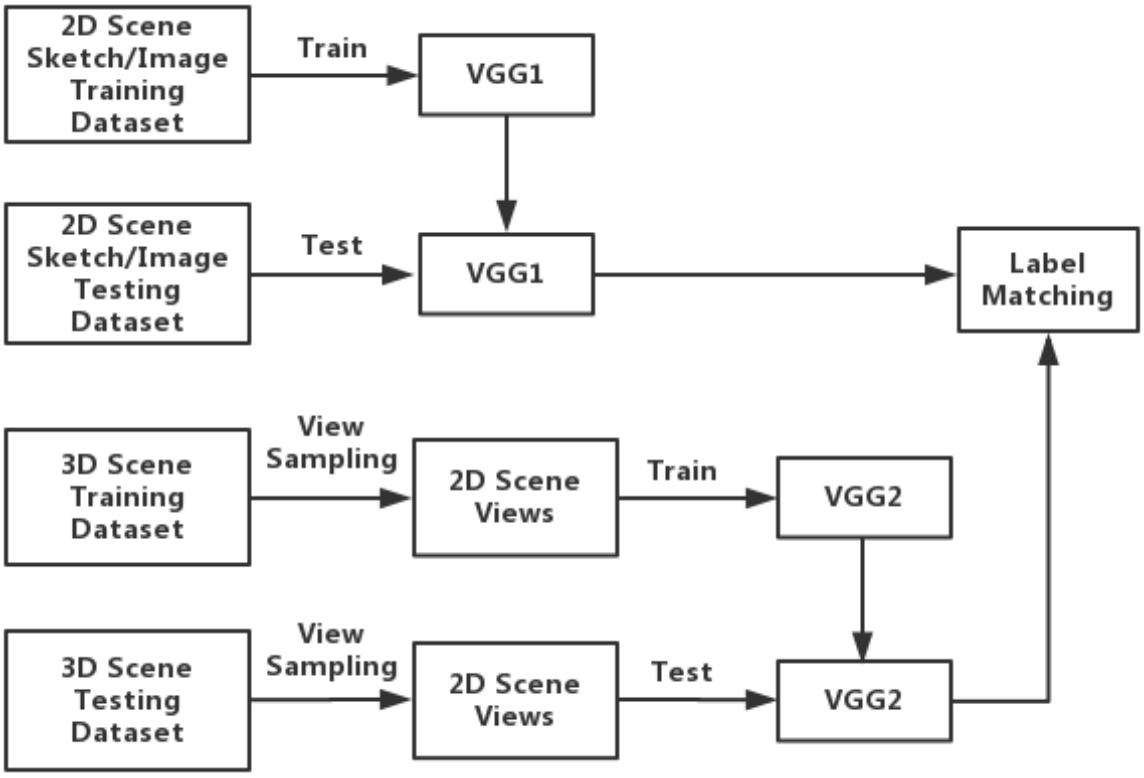


Figure 2.3: VBV-VGG architecture [223].

curve extraction method. In addition, by utilizing perpendicular double stereo, the method can estimate the motion of the 3D object more accurately. And based on the estimated motion, they construct multiple virtual cameras to obtain multiple views and extract the finest visual hull of the 3D object, which is useful for reconstructing poorly textured objects.

Song et al. [178] presented an end-to-end system named semantic scene completion network (SSCNet), which is based on convolutional neural network techniques. It is able to reconstruct a 3D indoor scene by using 3D voxel representations and predict semantic segmentation labels with a 2D depth image as input. This system takes both 3D scene reconstruction and semantic labels into consideration simultaneously, which were handled individually in previous work. This system solves two issues: a) extend the receptive field of the network to effectively capture 3D volume data context information; b) manually build a 3D scene dataset named SUNCG, which provides complete labeled 3D objects information.

Bobenrieth et al. [28] proposed a 3D indoor scene reconstruction method. Due to the reason that some applications require a complete scanning data captured by some scanning

devices like Kinect, which is a time-consuming process, their method only requires a few shots of the 3D scene, and also no overlapping requirement is required to generate a seamless scene. This method aligns these shots by looking for a group of transformations, and constructs an alignment graph which is used to find a global solution for all the transformations. However, since their method searches all the possible solutions, the time cost is highly dependent on the provided number of shots, e.g., it only takes a few seconds for simple cases, but the time may increase rapidly if the provided number of shots increases sharply.

Penner and Zhang [151] proposed a method to perform soft (keeping depth uncertainty) 3D scene reconstruction and view synthesis. During each stage, their method keeps the depth uncertainty, which can help to refine the depth estimates of object boundaries during the 3D reconstruction step. It also helps to adjust view rays and texture mapping rays during the view synthesis step. Their approach accepts a variety of inputs, which include not only structured images and wide-baseline captures, but also unstructured images and narrow-baseline captures.

Dai et al. [48] presented a data-driven based system named ScanComplete, which can reconstruct a high-resolution 3D scene from an incomplete RGB-D 3D scene scan. This system utilizes fully-convolutional neural network techniques to train on small subvolumes of the 3D scene and test on either small or large 3D scenes. In addition, in order to obtain high-resolution outputs with regard to the 3D scene size, the system adopts a coarse-to-fine strategy to predict small details and global structure simultaneously. The results show that it improves the quality of the 3D scene reconstruction with incomplete RGB-D 3D scan input as well as the semantic segmentation performance when compared with other methods.

Xu et al. [217] presented a system of reconstructing unknown 3D indoor scenes automatically with a single robot. This system enables the robot to scan and reconstruct the scene simultaneously, while taking care of both exploration efficiency and high quality scans. The system utilizes a time-varing 2D tensor field, a 2D image computed over the partial scanned scene, to guide the movement and camera control of the robot along its movement path. The system flexibly guides the camera's movement instead of using a fixed camera.

In 2018, Guo and Guo [71] presented a method to improve the reconstruction of urban scenes with buildings based on multi-view images. This method fuses the reconstructed dense points and line segments. According to the fusion process, it helps remove error line segments, sample the correct line segments with points, and finally determine and fuse the corrected line segments and points for the 3D scene. The results show that the approach

provides more accurate edge information in some parts with rare point features to represent the 3D urban scenes, such as windows and walls, and the time cost is acceptable.

Ritchie et al. [165] presented a 3D indoor scene synthesis model that can solve the following three limitations existing in previous work: (1) cannot place reasonable objects in the scene; (2) fail to take the size of an object into consideration; and (3) time-consuming. This model is a deep convolutional generative model, which can generate data distributions. It utilizes a top-down scene image, extracted from a 3D scene and fed into the model to iteratively synthesize new objects into the 3D scene. The synthesis process involves the decisions of objects' categories, positions, orientations, and sizes.

Rematas et al. [161] developed an end-to-end system to reconstruct a 3D soccer field with moving players from a soccer game video. It can detect the players in the video and estimate the depth map for each player. Compared to other methods that need to set up many synchronized cameras in a real soccer field, this system can reduce the cost. However, this system also has some limitations, e.g., if the system fails to detect the player(s), the player(s) will not be presented in the reconstruction result, and the overlap between the players may cause incorrect depth estimation, etc.

In 2019, Dong et al. [55] presented an end-to-end system that allows multiple robots to collaboratively scan unknown 3D indoor scenes for 3D scene reconstruction. This system utilizes an approach, named Optimal Mass Transport (OMT), to solve the resource distribution problem for the robots scanning the 3D indoor scenes. It adopts a divide-and-conquer scheme to assign tasks to the robots and optimize their paths. The timing and statistics performance information can be found in **Fig.** 2.4. However, this system is greedy-based, thus, may fall into local minimum.

Flynn et al. [63] proposed a 3D scene view synthesis method, which first generates a multiplane image (MPI), a kind of representation that can model the exterior effects of light fields such as transparency, and then uses it for view synthesis, based on a sparse set of views of the 3D scene. This method utilizes and improves the learned gradient descent-based method (LGD) [10], which is to update the prediction model parameters, by replacing its update rule with a deep neural network parameters update. This method can deal with depth complexity, object boundaries, light reflections, and thin structures as well, and the results demonstrate the state-of-the-art performance.

Zak et al. [145] presented an end-to-end method for 3D scene reconstruction. Unlike traditional 3D scene construction approach which utilizes the intermediate representation of depth maps for predicting the truncated signed distance function (TSDF) values [199].

Scene	Area	#R	#I	PT	IT	TT	TD
SunCG#1	110 m^2	3	16	0.9 sec	22 sec	6 min	48 m
Matterport3D#1	125 m^2	6	17	2.2 sec	26 sec	8 min	55 m
Office	60 m^2	3	10	0.8 sec	17 sec	3 min	32 m
Sitting_room	85 m^2	4	9	1.1 sec	25 sec	4 min	21 m
Classroom	120 m^2	5	5	1.2 sec	40 sec	5 min	41 m
Meeting_room	80 m^2	3	4	0.8 sec	46 sec	4 min	18 m
Dorm	35 m^2	2	4	0.5 sec	40 sec	3 min	17 m
Lab	300 m^2	6	40	2.3 sec	6 sec	5 min	156 m

Figure 2.4: Performance on both synthetic (row 2 $\sim$ 3)and real scenes (row 4 $\sim$ 9). Each row contains the scene area, # of robots (#R), # of planning intervals (#I), planning time for each planning interval (PT), time of each planning interval (IT), total scanning time (TT), and all robots' total movement distance (TD) [55].

Their reconstruction approach can predict the TSDF directly without requiring depth maps inputs, one CNN is used first to extracts features from each RGB image, and then, another CNN is used to refine the features and predict the TSDF values. In addition, their model obtains semantic segmentation label information with low computation. After the evaluation on Scannet dataset, their method performs better than state-of-the-art baselines.

2.6.2 Semantics-driven 3D Scene Reconstruction

Vineet et al. [192] proposed an end-to-end 3D scene reconstruction system. This system can efficiently perform dense, large-scale semantic 3D scene reconstruction. This system can also deal with moving objects in the 3D scenes by fusing the semantic information of the objects with the 3D map. The core of the system is that they adopt a hash-based fusion approach and a volumetric mean-field (a technique that can gradually refine the edges of each voxel in iterations [193]) based optimization approach for 3D scene reconstruction and object labeling separately.

To improve the 3D scene reconstruction accuracy, Blaha et al. [27] presented a 3D scene reconstruction method that takes both 3D scene reconstruction and semantic labeling into consideration at the same time, because these two themes can affect each other. This method is adaptive, which means it only reconstructs the necessary regions (near to the

predicted surfaces) of the 3D scenes. This can save much memory and time, and as a result, it can reconstruct large-scale scenes.

Savinov et al. [168] proposed an approach for dense semantic 3D reconstruction. It utilizes two schemes: one is continuous regularization and the other one is ray potentials. While ray potentials means that a ray is composed of voxels, and its information is contained in the pixels of the observed images. Therefore, by using correct ray potentials, it can achieve more accurate reconstructions. While, continuous regularization is performed to handle the noise in the input data. Particularly, this approach can also reconstruct thin objects due to the accurate representation of the input data, which is optimized by continuous regularization on the surfaces.

Ma et al. [131] presented a hybrid framework to reconstruct semantic 3D dense models from monocular images. This framework utilizes the conditional random fields (CRFs)-based [100] method as the baseline method. It considers the correlation between 3D space points and image pixels, which helps to obtain consistent object segmentation from multi-view images. With those semantic information from images, it can remove the noisy points in the 3D space, correct wrongly-labeled voxels, and fill the space where points are difficult to recover during the reconstruction process.

Armeni et al. [18] presented a semi-automatic framework to construct a 3D scene graph, which can carry various types of semantics (e.g., objects labels, scene categories, material types, etc) in a 3D scene. This 3D scene graph contains 4 layers which can represent the semantics information, camera position and 3D spacial relations (e.g., occlusion, relative volume, etc.). The construction of such kind of 3D scene graph is usually done manually and the labor is heavy. This framework can mitigate this problem by utilizing existing learning methods (e.g., Mask R-CNN) and improves them based on two ways: 1. Framing, which is a frame to sample the query images on panoramas so as to enhance the performance of 2D detectors. 2. Multi-view consistency, which is to deal with the issues originating from 2D detection of different camera positions.

2.7 3D Scene Generation

3D scene generation is to generate 3D scene models guided by the purpose of the generation method. Since many professionals such as autonomous vehicle designers, game developers, VR/AR engineers, and architects are increasingly using virtual 3D scenes for prototyping as well as end products design, the demand for related 3D scene data is high, which triggers

the need of 3D scene generation.

2.7.1 Data-driven 3D Scene Generation

Zhang et al. [230] proposed a 3D indoor scene modification framework to help users to enrich 3D indoor scenes with many small objects with regard to three scheduling rules related to: (a) object category, (b) object placement, and (c) object arrangement. The modification process could make the scenes more realistic based on the users' preferences. It adopts a cost function that integrates both the constraints proposed by the framework and the user-specified scheduling rules. However, it fails to involve the occurrence information of small objects. For example, the laptop and mouse objects normally appear in the same place, while the framework may separate them far away from each other.

2.7.2 Semantics-driven 3D Scene Generation

Akase and Okada [14] proposed a web-based system that deals with 3D room layout according to users' preferences. Their method is based on the Interactive Evolutionary Computation (IEC) method [13], and they used a predictive approach to narrow down the search space and adopted a multi-screen interface to reduce the fatigue of each user by using IEC. They created a semantic database which hosts the information of each single furniture object. This helps us to know each furniture's related objects. In addition, by computing the feature elements' importance, they performed a conjoint analysis on user preferences to generate satisfactory 3D scenes, and it achieved high user satisfaction.

Fisher et al. [61] proposed an activity-centric scene generation technique. It first anchors an observed 3D scene, then scans the activities supported by the 3D scene environment. Based on those activities, they finally determined semantically reasonable arrangements of the retrieved objects from an object database. The limitation of this technique is that it is expected to support a more general class of activities.

Walczak and Flotynski [195] presented a scene generation method by first creating semantically described 3D meta-scenes (3D content representations), and then generating customized 3D scenes based on those 3D meta-scenes. The advantages of using the semantics include: (1) making 3D scene customization simple; (2) supporting high-level abstraction operation and complex content customization.

Ma et al. [130] proposed a sub-scene level framework that can generate 3D indoor scenes by using natural language commands. It contains two steps: (1) retrieve related sub-scene(s) from a 3D scene database; (2) synthesize a new 3D scene by using the sub-scenes

and the current 3D environment. To bridge the gap between user language commands and scene modeling operations, they adopted a representation named Semantic Scene Graph (SSG), which contains objects' information, attributes and relationships to encode geometric and semantic scene information. To demonstrate the scalability of their framework, they need to train their model on a larger set of group relations and natural language commands.

Li et al. [116] presented a non-convolutional generative recursive neural network (RvNN) which also focuses on indoor 3D scenes. This network can learn hierarchical scene structures by utilizing a variational autoencoder (VAE) [34]. **Fig.** 2.5 shows an example scene hierarchy. Besides the semantic object-object relations, they also proposed three grouping operations (support, surround, and co-occurrence), and utilized object co-occurrences during the generation process. However, global scene hierarchies have some limitations due to certain reasons like imperfect training data, and unsatisfactory performance on complicated scenes (i.e. messy offices), so a network that can learn sub-scene level structures by itself may address this issue.

Li et al. [115] proposed an anisotropic convolutional network (AIC-Net) for 3 dimensional semantic scene completion. This network can overcome the limitation that exists in standard 3D convolution methods, which utilize a fixed 3D receptive field. The AIC-Net utilizes an anisotropic 3D receptive field, it decomposes the 3D convolution into three 1D convolutions. These stacked 1D convolutions can improve the voxel-wise modeling performance by determining the kernel size for each 1D convolution adaptively. Therefore, it allows the network to control the receptive field of each voxel more flexibly. And the core module of AIC-Net can be used as a plug-in for other existing networks.

2.8 3D Scene Datasets

2.8.1 Silberman et al.'s NYU Depth dataset V2 (2012)

Silberman et al. [173] built a RGB-D indoor scene video dataset captured by the Microsoft Kinect. It comprises 1,449 densely-annotated RGB-D images for 464 different scenes of three cities over 26 scene classes and 407,024 unlabeled frames.

2.8.2 Patterson et al.'s SUN Attribute dataset (2012, 2014)

Patterson et al. [149, 150] 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,

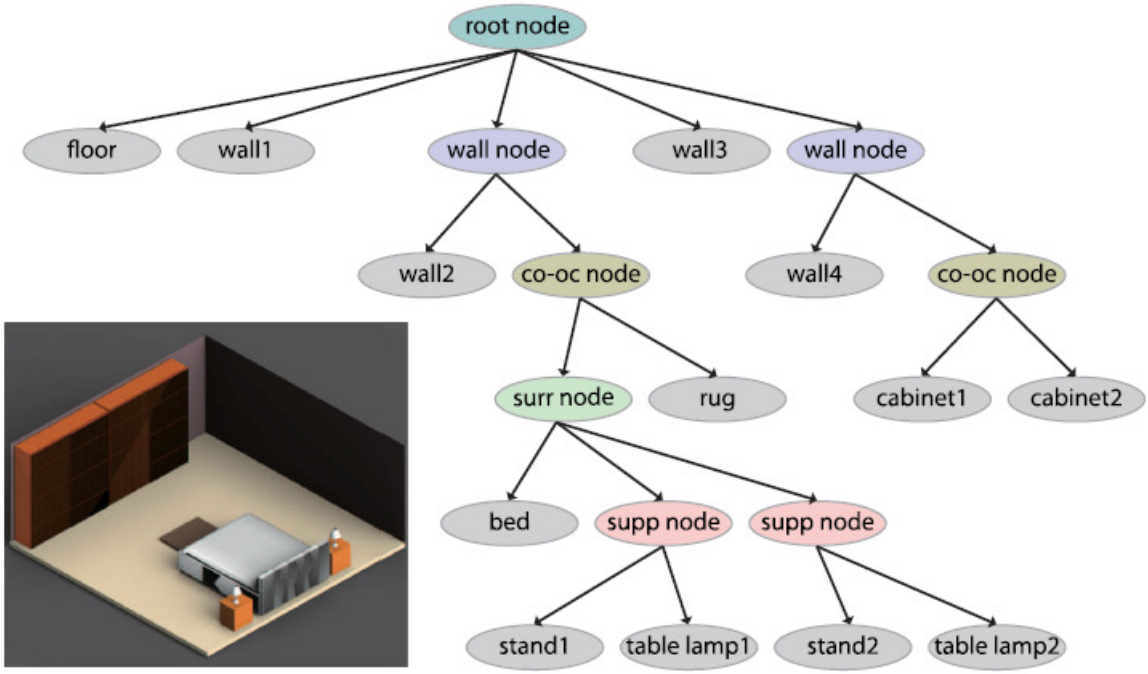


Figure 2.5: A training set bedroom example with the corresponding scene hierarchy. The root node have five children which are one floor node and four wall nodes. Then each wall has its own subtree with more detailed object-object relations [116].

zero short learning, captioning, search, and parsing, while even the attribute features alone can achieve the state-of-the-art performance.

2.8.3 Zhou et al.'s Places dataset (2014)

Zhou et al. [239] proposed a new large-scale scene image dataset which is 60 times bigger than the standard SUN [210] 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.

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

Xiao et al. [212] 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 [210]. Xiao et al. [213] 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.

2.8.5 Cordts et al.'s Cityscapes dataset (2016)

Cordts et al. [46] built a 2D scene image dataset, named Cityscapes, which contains urban street scenes recorded by stereo video for 50 cities. This dataset is composed of finely-annotated and coarsely-annotated images. 5,000 finely-annotated images are manually selected from 27 cities, which contain highly diverse objects, background and layouts. 20,000 coarsely-annotated images are automatically selected from the videos. In order to increase the annotation speed, the object boundaries are not as accurate as finely-annotated images, but they still have a 97% segmentation accuracy.

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

Hua et al. [81] created a richly annotated RGB-D indoor scene dataset named SceneNN. It contains 100 scenes categories annotated at vertex, mesh and pixel level, respectively. This multi-level annotation was designed to promote its usage in diverse related applications.

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

Xiang et al. [209] released a large scale dataset called ObjectNet3D containing 100 categories of scene data. There are 90,127 scene images comprising 201,888 objects, and 44,147 3D objects in the dataset. It has performed 2D images-3D shapes alignment, and also provides pose and shape annotations for the 3D shapes.

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

Handa et al. [74] designed an automatic 3D scene data synthesis framework to generate synthetic 3D scenes by utilizing existing CAD repositories, and generated about 10,000 synthetic views for five different types of 3D indoor scenes.

2.8.9 Armeni et al.'s Joint 2D-3D-Semantic dataset (2017)

Armeni et al. [19] 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.

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

Song et al. [179] constructed a SUNCG dataset, a synthetic 3D scene database with manually labeled voxel occupancy and semantic labels. This dataset has 84 categories, and 45,622 different scenes and 2,644 objects across those categories.

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

Lin et al. [123] 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 [123], Caesar et al. [31] 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.

2.8.12 Zhou et al.'s Places dataset (2018)

Zhou et al. [238] 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.

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

Zou et al. [243] curated SketchyScene, a dataset with 29,000 scene-sketches, over 7,000 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.

2.8.14 Yuan et al.'s SceneSBR2019 and SceneIBR2019 dataset (2019)

Yuan et al. [222] and Abdul-Rashid et al. [5] compiled two 3D scene retrieval benchmarks, named SceneSBR2019 and SceneIBR2019. SceneSBR2019 is using 2D scene sketches as the input query while SceneIBR2019 is using 2D scene images as the input query. Both

benchmarks contain 30 categories, which were selected from the Places88 dataset [236] scene labels. The 88 categories of the Place88 dataset are also shared by the ImageNet [50] and SUN datasets [211]. SceneSBR2019 contains 25 scene sketches for each category, while SceneSBR2019 contains 1,000 scene images for each category. Both SceneSBR2019 and SceneIBR2019 share the same 3,000 3D scene models, which is the target dataset. It is currently the first and largest benchmark for 2D scene sketch/image-based 3D scene retrieval.

2.8.15 Zheng et al.'s Structured3D dataset (2019)

Zheng et al. [234] built a synthetic dataset, named Structured3D, to meet the increasing demand of symmetries (e.g., lines, cuboids, surfaces) for 3D indoor scene reconstruction and recognition. They first collected a lot of 3D indoor scenes designed by professional specialists. Then, they extracted 3D structures (ceiling, floor, wall, etc) annotations as ground truth from those 3D scenes. Finally, based on the extracted 3D structures, they synthesized and generated high-quality (photo-realistic) 2D scene images.

2.8.16 Straub et al.'s Replica dataset (2019)

Straub et al. [183] created a dataset, named Replica, which contains 18 different indoor scenes. Compared to other 3D scene datasets such as [47] or [37], the Replica dataset is more realistic because it captures the full indoor scenes and has no missing surfaces. In addition, for each mesh primitive, Replica introduces high dynamic range (HDR) textures by changing the settings of the RGB texture camera. Moreover, Replica also contains glass and mirror reflectors surface information, which also can be rendered and make the 3D scenes appear more realistic.

2.8.17 Vasiljevic et al.'s DEpth Dataset (DIODE) dataset (2019)

Vasiljevic et al. [191] curated a RGB-D 2D scene image dataset, named Dense Indoor and Outdoor DEpth Dataset (DIODE), which contains both indoor and outdoor scene categories. Most existing datasets only contain one domain (either indoor or outdoor) since due to different scene types of the two domains, indoor and outdoor scene images are obtained with different types of sensor suites. As a result, it is difficult to obtain a good accuracy for related cross-domain problems. This dataset adopts one sensor type, thus making the indoor and outdoor scenes have the same scene type.

2.8.18 Gupta et al.'s Large Vocabulary Instance Segmentation dataset (LVIS) dataset (2019)

Gupta et al. [72] constructed a 2D scene image dataset, named Large Vocabulary Instance Segmentation dataset (LVIS), which contains about 2 million object segmentation masks for more than 1,000 object categories, and about 164K 2D scene images in total. Compared to some related datasets, e.g., COCO [123], LVIS provides a more accurate mask for each segmented object instance, thus will be more beneficial in improving the accuracy of a learning method for scene image object detection or segmentation.

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

Gao et al. [66] 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 [31]. Their two-staged approach generates the foreground and background of an image separately. Therefore, SketchyCOCO 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.

2.9 Other Related Deep Learning Based 3D Scene Understanding

According to Goodfellow et al. [67], the human visual system does much more than just recognizing objects. It is able to understand entire scenes including many objects and relationships between objects, and process rich 3D geometric information needed for our bodies to interface with the world.

Zhao et al. [232] proposed a framework to parse scene images at both pixel feature and word concept levels by jointly embedding the two levels of information into a high-dimensional vector space. At the word concept level, they incorporated the semantic word-word relations (hypernym/hyponym) based on WordNet [140] and compared their jointly embedding framework with other models, such as Word2Vec [137] and demonstrated better performance.

Choi et al. [42] proposed a hierarchical visual scene understanding model named 3D Geometric Phrase Model, which captures both semantic and geometric relationships of the objects in a scene, as well as their grouping information. Su et al. [184] devised a multi-view convolutional neural network (MVCNN) framework for 3D shape recognition by first learning features from multiple rendered views of a 3D model via a CNN model,

and then fusing all the extracted features via a max-pooling like view pooling approach, and finally using another CNN as a classifier for the 3D shape recognition. We also utilized the MVCNN framework in our approach. PointNet [153] is a deep neural network designed on top of point clouds, and it directly consumes point clouds. Such an interaction better preserves the permutation invariance of points in the input, and thus mitigates the issues caused by transforming point cloud to regular 3D voxel grids or collections of images. The proposed unified architecture is applicable for a wide range of applications including object classification, part segmentation and scene parsing, and has demonstrated promising results, as well.

Ku et al. [97] organized a semantic objects segmentation contest based on provided 3D point cloud for street scenes, which are annotated with five classes (building, ground, car, vegetation and pole), in SHREC 2020. This contest is challenging due to the fact that by utilizing LiDAR scanners, the captured raw large-scale 3D point clouds data contains tremendous points and are usually non-uniformly distributed.Ku also provided a state-of-the-art deep learning based baseline method, which is PointNet++ for the semantic segmentation task. There are 4 (3 learning-based, 1 non-learning based) methods are provided by the participants and all of those 4 methods outperformed the baseline method. Specifically, the non-learning based approach obtained a good result proves that well-designed feature descriptors could plays a more important role in segmentation than learned features, especially for the unbalanced data.

Joseph et al. proposed a YOLO (v1 [158], v2 [159], v3 [160]) system which can be used for image or video object detection. It is a state-of-the-art, real-time, one-stage, end-to-end object detection system. It has an advantages with comparing with other object detection: Image processing speed is fast while maintaining a high accuracy at the same time. Compared to YOLOv1 and YOLOv2, YOLOv3 has been greatly enhanced in performance: 1. Multi labels predection: They adopted binary cross-entropy loss (logistic classifier) instead of softmax to solve the multi labels problem (e.g, man and person); 2. Small objects detection: solve the small objects detection problem by using short cut connections; 3. Feature extractor network: the backbone net structure has been improved from darknet-19 in v2 to darknet-53 with deeper layers. In the dissertation, we adopted the latest YOLOv3 for the task of object occurrence prediction in Section 4.

Xinwei et al. [78] presented a Triplet-Center Loss (TCL)-based 3D object retrieval method. In this paper, the loss function contains two parts, which are triplet loss and center loss. Compared to traditional softmax loss which is usually utilized for 3D shape

classification, triplet-center loss could learn more discriminative features for 3D shape retrieval task. The triplet-center loss can learn a center for each class, and guarantees that the samples from the same class as the center has closer distance to the center than the samples from different classes. In addition, the triplet-center loss is also able to find an embedding space where the distance between samples of the same class is less than the distance between samples of different classes. In the dissertation, we adopted the TCL loss function as one part of our network loss function.

Due to the reason that some acquired 3D scene data may be incomplete during the process of 3D capturing, e.g., object overlap each other, views are insufficient while capturing, etc. Wang et al. [197] proposed a U-Net like structures octree-based CNNs (O-CNN) network to complete and clean such defective 3D scene data. This network supports very deep network layers (e.g., up to 70 layers) and contains an output-guided skip-connection, which can preserve current input geometry and learn new geometry from input data. Results demonstrate that the prediction accuracy has improved and outperforms the state-of-the-art.

2.10 Adversarial Networks Related Research Directions, Techniques and Benchmarks

3D/2D line drawing generation. We can generate 3D line drawings [43, 44] from a 3D model directly based on different 3D features, such as ridges and valleys [83], suggestive contours [49], apparent ridges [86], photic extremum lines (PELs) [215], demarcating curves [96], abstraction feature lines [135], programmable line drawings [70], and stylized stereoscopic 3D line drawings [85]. Based on a 3D line drawing of a 3D model which is still a 3D representation of the object, we can generate its different 2D line drawing images from different viewpoints, i.e., simply by projection transformation. A recurrent neural network (RNN)-based neural representation and related generative model were proposed in [73] to represent and generate single-object sketch drawings under both conditioned and unconditioned situations.

Sketch-to-image and image-to-sketch synthesis Based on sketches, we can generate corresponding images of different styles, which can be found in recent image-to-image translation algorithms based on different types of GANs: deep convolutional GAN (DCGAN) [155], conditional GAN (cGAN) [84], CycleGAN [242], SketchyGAN [41], and contextual GAN [128].

However, there is much less research work on the other direction: image-to-sketch synthesis. Berger et al. [26] proposed to generate portraits coming from the same artistic style at different levels of abstraction based on an image. They first learned a model based on a portrait sketch dataset collected from different artists to encode both their styles and abstraction process. Then, they directly synthesized a portrait sketch based on the available strokes in the dataset by modifying their shape, curvature, and length based on the learned model. Li et al. [118] proposed an algorithm to perform perceptual grouping of the semantic parts of a sketch, and then by utilizing a human stroke dataset they generated sketches from an image by exploiting a deformable stroke model-based optimization approach. To handle recognition of an incomplete sketch, Liu et al. [124] advised to first complete then sketch and then recognize it based on a conditional GAN structure, named SketchGAN.

Edge and contour detection. While there is an abundance of research in synthesizing images from sketches, we have not found any current examples of research in which the authors' intent is to explicitly generate sketch-like renderings of images. A similar area of research to the problem we are addressing can be found in edge and contour detection. While the differences between edge detection, contour mapping and approximating sketches may be subtle for some images, there are some important differences in their definitions. Edge detection refers to finding extreme gradient values relative to neighboring pixels, while contour mapping is typically derived from edge mapping but focused on accentuating the edges that correspond to object boundaries [154]. It is for this reason that Li et al.'s Photo-Sketching project [117] is the most closely-related research to the work presented in this paper. In addition, we tried to gain insights into some of the more successful models whose aim was to map sketches to photo-realistic images.

For edge detection, the type of Difference-of-Gaussians (DoG) operators has proved better performance, such as flow-based DoG (FDoG) [88, 89], and the variable thresholding DoG (XDoG) [206, 207]. Recently, an approach named holistically-nested edge detection (HED) [214] has been developed to detect good edge images for the holistic image training and prediction, as well as multi-scale and multi-level feature learning vision problems. It adopts an image-to-image translation approach based on a deep learning model.

Regarding contour generation, Ren and Bo [163] devised a contour detection algorithm which generates the state-of-the-art performance, while Lim et al. [119] proposed to learn a mid-level local contour-based image feature representation named sketch tokens by first extracting patches centered on contours from the hand-drawn sketches and then clustering

the patches to form a set of token classes.

Recently, Li et al. [117] proposed to generate a boundary-like contour drawing from an image to represent the outline of the visual scene of the image.

Generative adversarial networks (GANs). GAN-based approaches dominate the related research areas of the project. Since 2014, Goodfellow et al. [68] proposed the generative model Generative adversarial networks (GANs), many different GANs have been proposed by researchers either for further improvement or for different applications. Examples for the former include: Wasserstein Generative Adversarial Network (WGAN) [17], least squares Generative Adversarial Network (LSGAN) [132], and energy-based GAN (EBGAN) [233]. For the latter, we have deep convolutional GAN (DCGAN) [155], conditional GAN (cGAN) [84], CycleGAN [242], augmented CycleGAN [15], self-attention Generative Adversarial Network (SAGAN) [229], BigGANs [29] for high fidelity natural image synthesis, BigBiGAN [54] which combines BigGAN and BiGAN, and StyleGAN [91]. People also developed related approaches for GANs' visualization and understanding [23, 24], training [79,166,221], comparisons [99,129], as well as evaluation metrics, such as Inception Score (IS) [166], Fréchet Inception Distance (FID) [79] and intra FID [141].

Sketch style, abstraction, complexity and quality It is difficult to quantitatively measure the styles and abstraction levels of different human sketches. Therefore, most of existing related research works adopt a data-driven approach [26, 118] to learn different models for them. Muhammad et al. [143] regarded the sketch abstraction level of a sketch as a tradeoff between its recognizability and the number of strokes it contains, and proposed a sketch abstraction model through a stroke removal process guided by reinforcement learning. Snodgrass et al. [175] studied the visual complexity of 260 line drawings representing 260 different categories, and found that: (1) considerable variation in mean visual complexity exists across different categories, i.e., the low visual complexity in the fruit drawings versus the much higher visual complexity in the insect drawings; (2) the amount of details is consistent with the complexity of the real-life object. Kudrowitz et al. [98] proposed that we can measure the sketch quality of a sketch image based on its line work, perspective, and proportions and then found that higher quality sketches contribute to a higher ranking of their creativity levels. In our approach, we mainly use certain GAN evaluation metrics (see Section 5.3) to quantitatively compare the results of different GAN-based approaches.

Paired image-sketch datasets To train the GAN-based model, we need to provide paired image-sketch datasets, such as Berkeley Segmentation Dataset and Benchmark (BSDS500) [134], the Sketchy Database [167], the SketchyScene dataset [244], and the Photo-Sketching dataset [117]. We can also pair the 30 classes of scene images and sketches [223] in the Eurographics Shape Retrieval Contest (SHREC) 2019 2D Scene Sketch-Based [224] and Image-Based [8] 3D Scene Retrieval Benchmarks to form a new image-sketch pair dataset.

However, among all of the aforementioned datasets, only BSDS500 and the Photo-Sketching datasets have the best 2D image-2D sketch (in fact, sketch images are contour images) matching quality (i.e. accuracy in feature correspondence). For other datasets, either the matching quality is low such as the Sketchy and SketchyScene datasets, or the images and sketches are only matching at the category level, instead of at the appearance level, such as the SHREC-based generated one. BSDS500 has 500 natural images, while in average each image has five different early aligned contour images annotated by five subjects. The recently built Photo-Sketching dataset is much bigger. It has 5,000 roughly aligned contour images for 1,000 outdoor scene images.

2.11 Summary

3D scene analysis and processing is important for many applications such as autonomous driving cars and AR/VR industries. Recently, it has received more and more attentions. To improve the performance of related deep neural network models, a large amount of 3D scene data is required. With the increasing popularity and power of 3D scene sensing and capturing devices, it is more and more convenient to obtain more accurate 3D scene data.

This chapter aims to provide a comprehensive survey of most recent state-of-the-art 3D scene analysis and processing research methods. We summarize this research area from five directions: (1) 3D scene classification; (2) 3D scene recognition; (3) 3D scene retrieval; (4) 3D scene reconstruction; and (5) 3D scene generation. For each direction, we further classify the involved methods into data-driven and semantics-driven methods. In addition, we also review several most recent and popular 3D scene datasets in this research area. Each dataset meets the needs of one or more research directions in this area.

2.11.1 Challenges

- **Accuracy improvement for 3D Scene Analysis and Processing.** So far, the scholars and researchers have made great progress in the analysis and processing of single

3D object. However, the accuracy of the 3D scene analysis and processing is not as good as expected. Compared to 3D objects, 3D scenes are more complicated. 3D scenes usually contains multiple 3D objects, each object has spatial and semantics relatedness with each other. For example, in a 3D kitchen scene, there is a bowl in the sink, spatial relatedness exists between the bowl and the sink. And it is more likely that a table and a chair will appear in the kitchen scene together than a table and an elephant, because the table and chair have closer semantics relatedness [205]. Due to the complexity of the 3D scene, the 3D scene analysis and processing accuracy needs to be improved urgently.

- **Lack of a large-scale and/or multimodal 3D scene benchmark dataset.** As we know, the size of the training dataset has a great influence on the accuracy of the machine learning algorithms. According to our knowledge, at present, there is no such a 3D dataset which has large enough data volume, enough categories, and widely used by people. 3D scenes are basically derived from real life, it is very difficult to build or collect, this is also an important reason for restricting the development of 3D scene analysis and processing.

Chapter 3
BENCHMARKS BUILDING AND METHODS EVALUATION

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 [225], [4], [222], [5], while code for the evaluated methods are also available: `http://github.com/3DSceneRetrieval`.

3.1 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, 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 dissertation, 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 [201], [21], [186]. 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 [164] [187].

However, there is little research in 2D scene image-based 3D scene shape retrieval [136] [216] 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 [226],

[6], [224], [8] 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 [219] 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 [189]. Based on this benchmark, we organized the SHREC'18 2D scene sketch-based 3D scene retrieval track [226]. Considering the popularity of 2D scene images that also can be used as queries, we further collected 1,000 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 [6]. 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 red 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 [223] 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 3,000 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 [224] and the SHREC'19 extended 2D scene image-based 3D scene retrieval (SceneIBR2019) track [8]. The 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 Section 3.2, 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 3.3. Section 3.4 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 3.5 concludes

the chapter and lists several future research directions.

3.2 Benchmarks

In the SHREC'18 and SHREC'19 scene retrieval tracks [6, 8, 224, 226], 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.

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

Overview Our basic 2D Scene Sketch/Image-Based 3D scene Retrieval benchmark **Scene_SBR_IBR_2018** is publicly available [4, 225]. It utilizes: (1) the 250 2D scene sketches in Scene250 [219] as its 2D scene sketch query dataset; (2) 10,000 2D scene images selected from ImageNet [51] as its 2D scene image query dataset; (3) 1,000 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 (1,000 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 3.1**. 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 1,000 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/1,000 images and 100 models per class) for non-learning based approaches.

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

Table 3.1: 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	1,000	100
Total (all 10 classes)	250	10,000	1,000

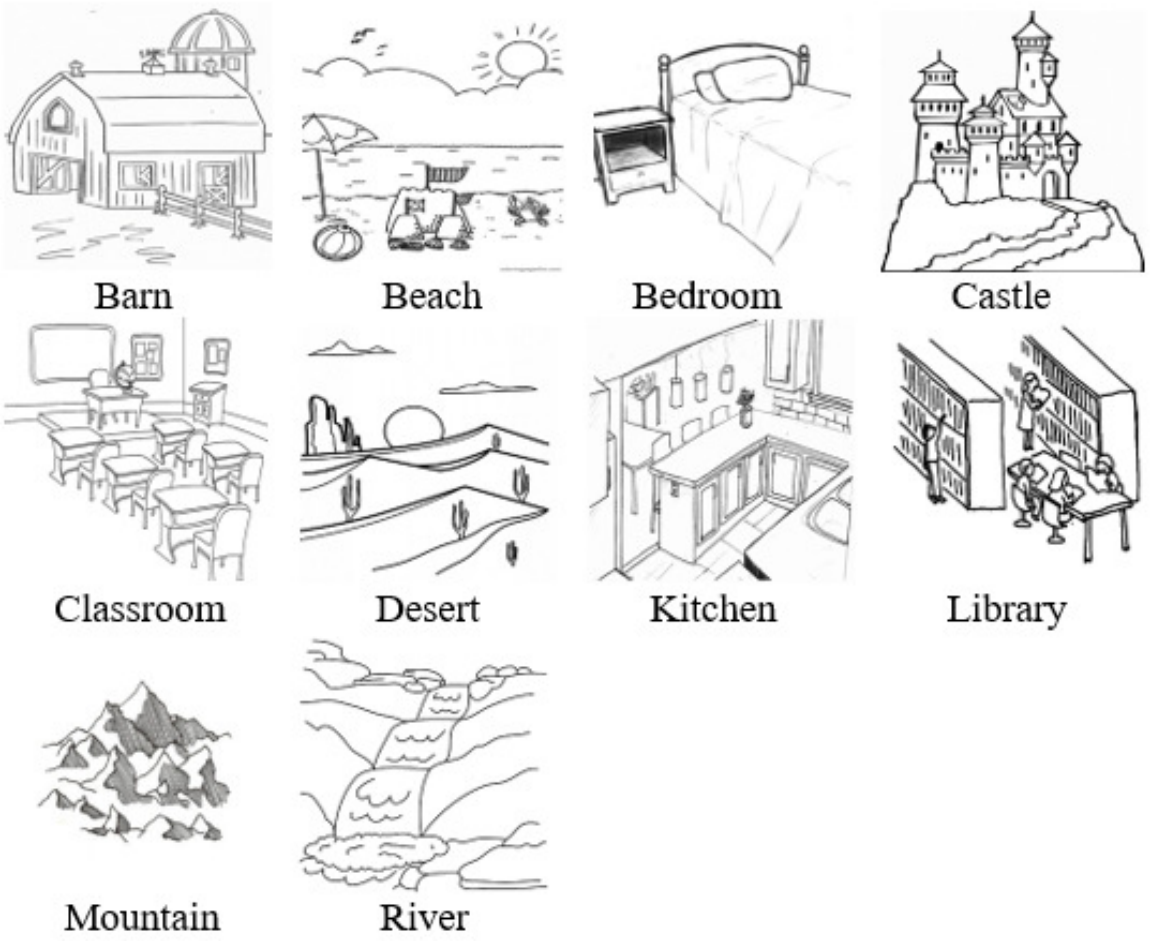


Figure 3.1: 2D scene sketch query examples [219] in our **Scene_SBR_IBR_2018** benchmark.

which are downloaded from 3D Warehouse [189]. One example per class is demonstrated in **Fig. 3.1**.

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 1,000 images) that are all from ImageNet [51]. One example per class is demonstrated in **Fig. 3.2**.

Figure 3.2: 2D scene image query examples in our **Scene_SBR_IBR_2018** benchmark.

3D scene model target dataset The 3D scene target dataset is built on the selected 1,000 3D scene models downloaded from 3D Warehouse. Each class has 100 3D scene models. One example per class is shown in **Fig. 3.3**.

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

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

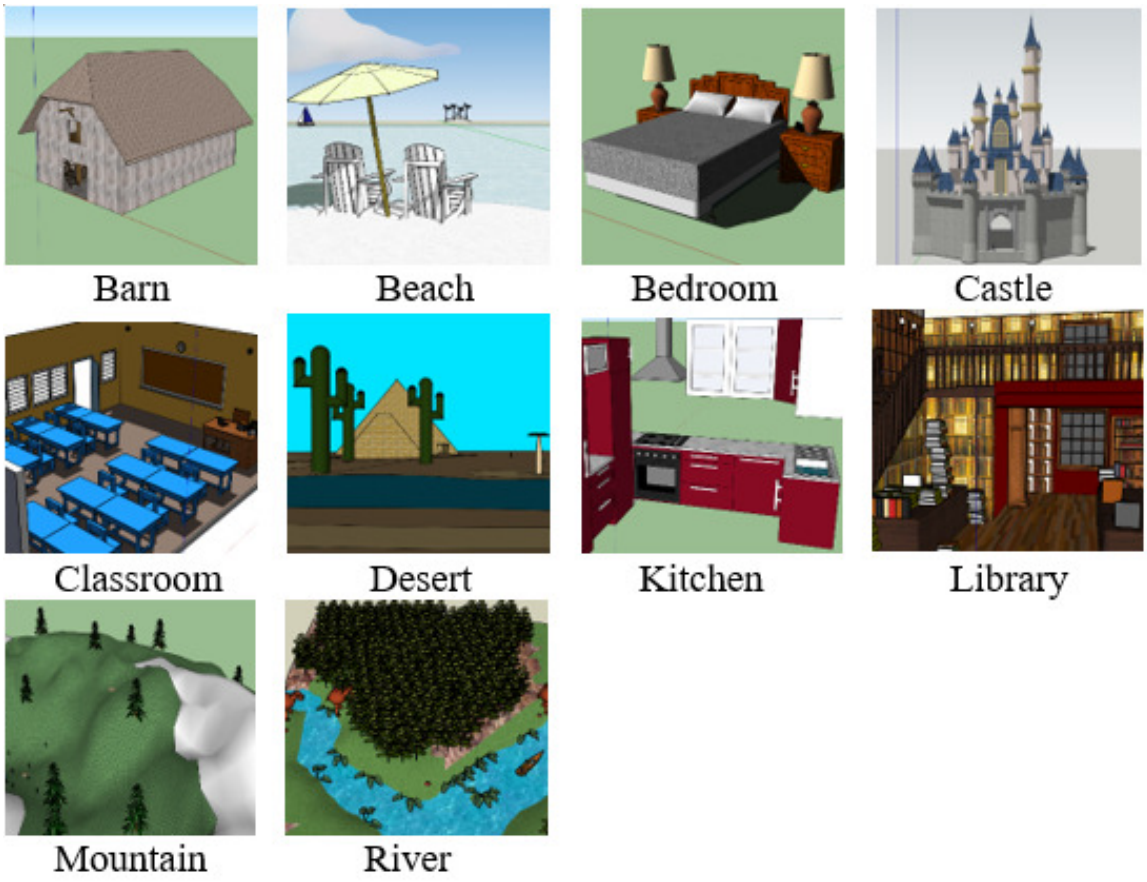


Figure 3.3: 3D scene model target examples in our **Scene_SBR_IBR_2018** benchmark.

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.

Motivation As mentioned in Section 3.2.1, to foster the research direction of sketch-based and image-based 3D scene retrieval, we built the first benchmarks **Scene_SBR_2018** and **Scene_IBR_2018** respectively and organized two related Shape Retrieval Contest tracks (SHREC) [6, 226]. 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.

Building process By referring to several of the most popular 2D/3D scene datasets, such as Places [238] and SUN [212], 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 [238]. It is worth noting that the 88 scene categories are already shared by ImageNet [51], SUN [210], and Places [238]. For the additional 20 classes' (sketches, images and models) data collection, we gathered their sketches and images from Flickr [62] as well as Google Images [69], and downloaded their SketchUp 3D scene models (in both the original ".SKP" format and our "transformed ".OBJ" format) from 3D Warehouse [189].

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.

Benchmark details Our extended 3D scene retrieval benchmark **Scene_SBR_IBR_2019** is publicly available [5, 222]. 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 2,000 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 (1,000), 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 (1,000 per class), and 3,000 3D scene models (100 per class) across 30 scene categories. Examples for each class are demonstrated in **Fig. 3.4**, **Fig. 3.5**, and **Fig. 4.1**.

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 3.2**. 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 3,000 scene model targets) dataset are needed.

Table 3.2: 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	1,000	100
Total (all 30 classes)	750	30,000	3,000

3.2.3 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) [171] and Average Precision (AP) [104]. For users' convenience, we also have developed an evaluation toolkit to compute them for each of the two benchmarks, and made them publicly available via the corresponding four tracks [225], [4], [222], [5]. 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

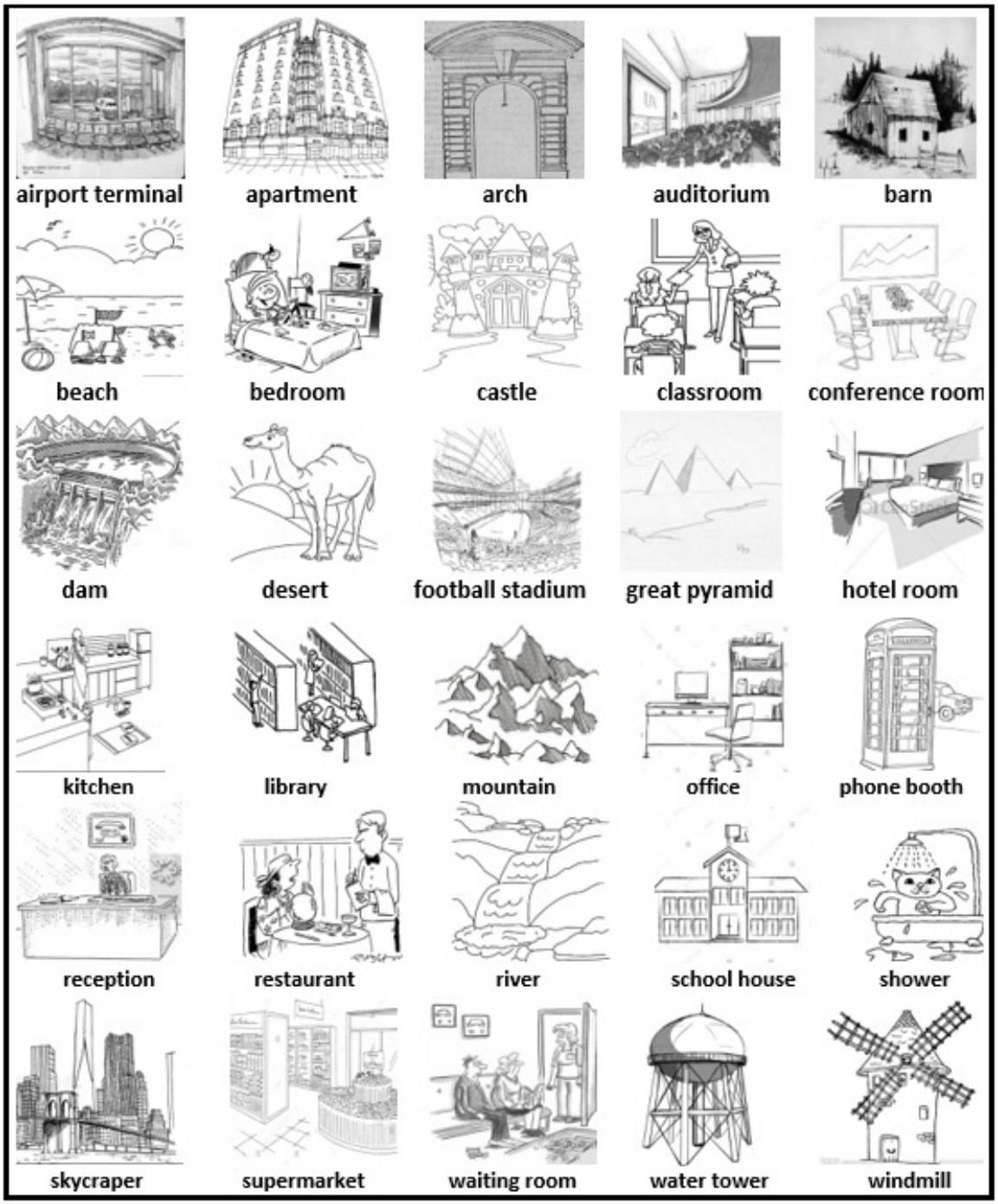


Figure 3.4: 2D scene sketch query examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown.

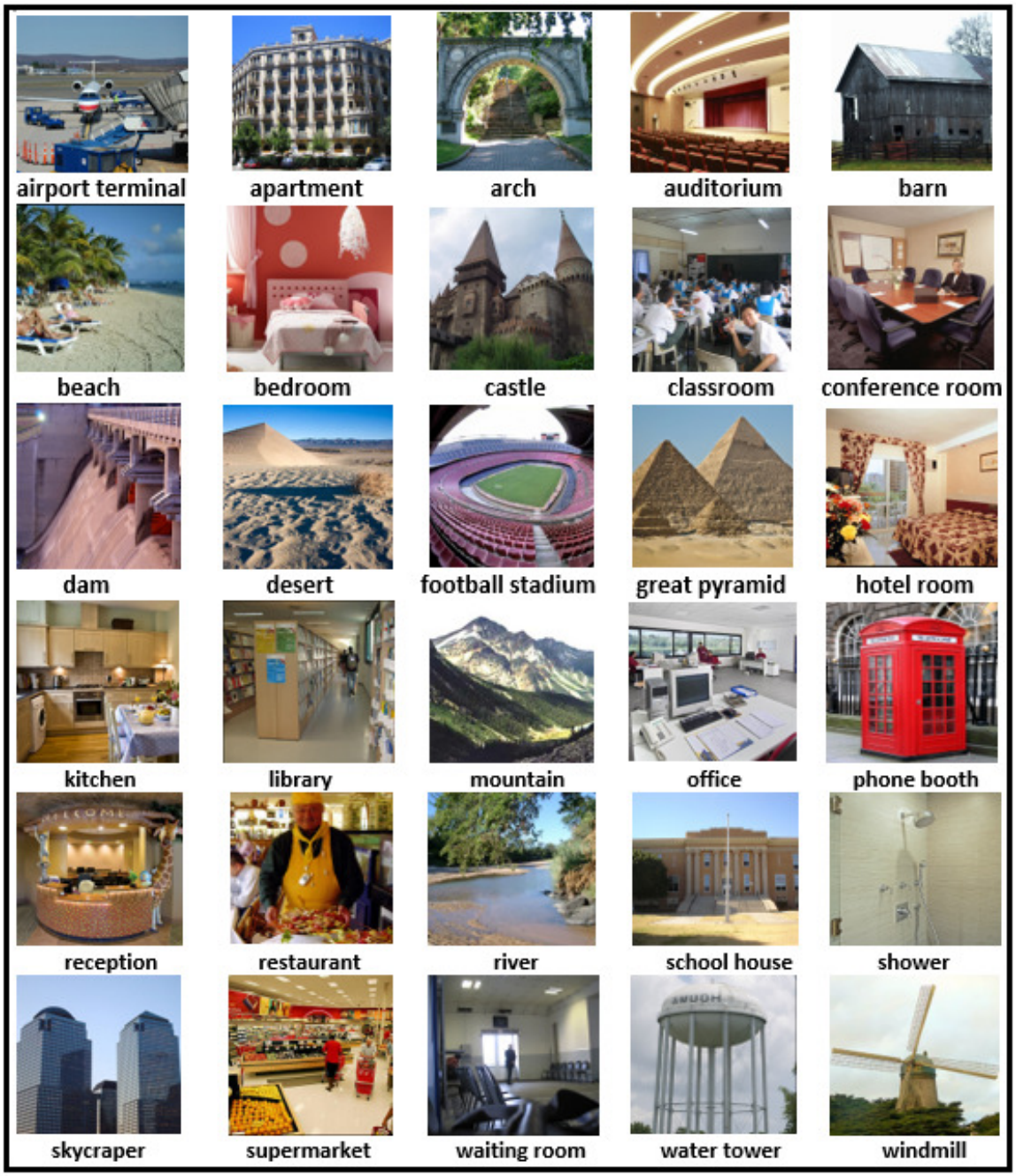


Figure 3.5: 2D scene image query examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown.

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

Figure 3.6: 3D scene model target examples in our **Scene_SBR_IBR_2019** benchmark. One example per class is shown.

$1 \leq K \leq n$. 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,

$$DCG_i = \begin{cases} G_1 & i = 1 \\ DCG_{i-1} + \frac{G_i}{\lg_2 i} & \text{otherwise} \end{cases} \tag{3.1}$$

  Finally, we normalize it by its optimum,

$$DCG = \frac{DCG_n}{1 + \sum_{j=2}^{C} \frac{1}{\lg_2 j}} \tag{3.2}$$

- **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.

3.3 Methods

We 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 3.3.1) has been incorporated in this project; while one and two new runs of the TCL method (Section 3.3.1) 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 3.3.2), other six Query-by-Image algorithms (i.e., *VGG* (Section 3.3.1), *MMD-VGG* (Section 3.3.1), *TCL* (Section 3.3.1), *VMV-VGG* (Section 3.3.1), *RNIRAP* (Sections 3.3.1∼
3.3.1), and *DRF* (Section 3.3.1)) are almost identical to their counterparts in the Query-by-Sketch category (*RNSRAP* for *RNIRAP*). Therefore, we merge their presentations only in Section 3.3.1 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 3.3 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).

3.3.1 Query-by-Sketch retrieval

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.

Table 3.3: 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 [109] 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	3.3.1	[174]
2	**MMD-VGG**	local	DFM	DA	VGG	No	3.3.1	[126, 174]
3	**TCL**	local	DFM	CNN	VGG, ResNet	No	3.3.1	[78]
4	**RNSRAP**	local	CBR	CNN	ResNet	Yes	3.3.1, 3.3.1	[162, 190, 238]
5	**VMV-AlexNet**	local	CBR	CNN	AlexNet	No	3.3.1	[223]
6	**VMV-VGG**	local	CBR	CNN	VGG	No	3.3.1	[223]
7	**DRF**	local	CBR	CNN	VGG	Yes	3.3.1	[228]
Query-by-Image								
8	**VGG**	local	DFM	No	VGG	No	3.3.1	[174]
9	**MMD-VGG**	local	DFM	DA	VGG		3.3.1	[126, 174]
10	**TCL**	local	DFM	CNN	VGG, ResNet	No	3.3.1	[78]
11	**VMV-VGG**	local	CBR	CNN	VGG	No	3.3.1	[223]
12	**BoW**	local	BoW	No	-	No	3.3.2	[120, 147]
13	**RNIRAP**	local	CBR	CNN	ResNet	No	3.3.1, 3.3.1	[162, 190, 238]
14	**CVAE**	local	DFM	VAE	-	No	3.3.2	[95]
15	**CVAE-VGG**	local	DFM	VAE	VGG	No	3.3.2	[95]
16	**DRF**	local	CBR	CNN	VGG	Yes	3.3.1	[228]

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. 3.7**.

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 [174] model pretrained on the Places [238] 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 [126] 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 [174] 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.

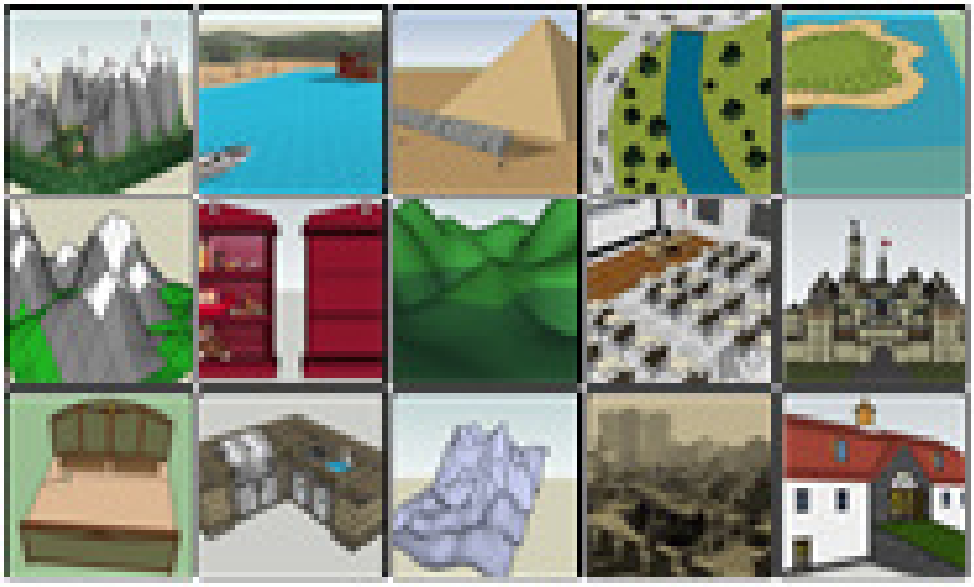


Figure 3.7: Several example representative views.

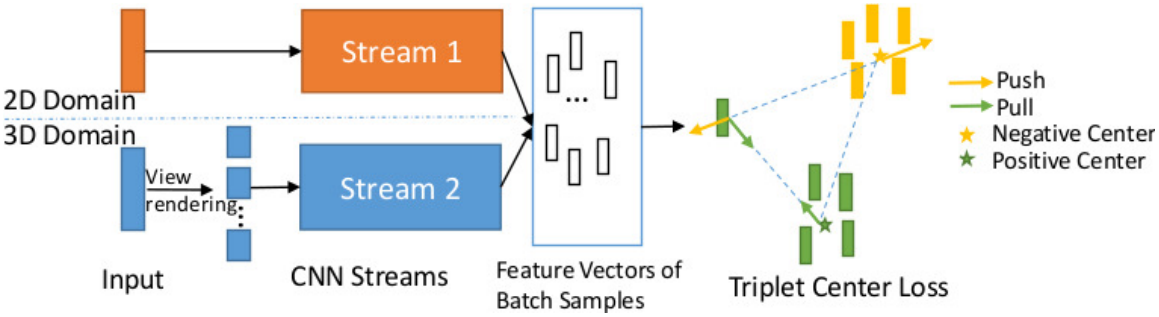


Figure 3.8: Illustration of the network architecture. Two separate CNN streams are used to extract features for the two domains. Triplet center loss along with softmax loss (not depicted here) is used to optimize the whole network.

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 [78] 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 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. 3.8**. 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 [184] 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 [174]). 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) [78] for feature learning. Given a batch of training data with M

samples, they define TCL as,

$$L_{tc} = \sum_{i=1}^{M} \max\left(D\big(f_i, c_{y_i}\big) + m - \min_{j \in C \setminus \{y_i\}} D\big(f_i, c_j\big), 0\right) \tag{3.3}$$

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 [22] 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 [174], ResNet50 [76] and ResNet101 [76] 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 project'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.

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 [76] is employed to encode a sketch into a feature vector of 2,048 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 to different entities in the image and synthesize other sketch images by matting these patches. By this way, they enrich the training dataset with 2,000 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 [92] 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 @ 2.30GHz, 8 GB Memory, and 1 x NVIDIA GeForce 920MX was used. The training time for a classification model is about 30 minutes. It takes less than 1 second 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

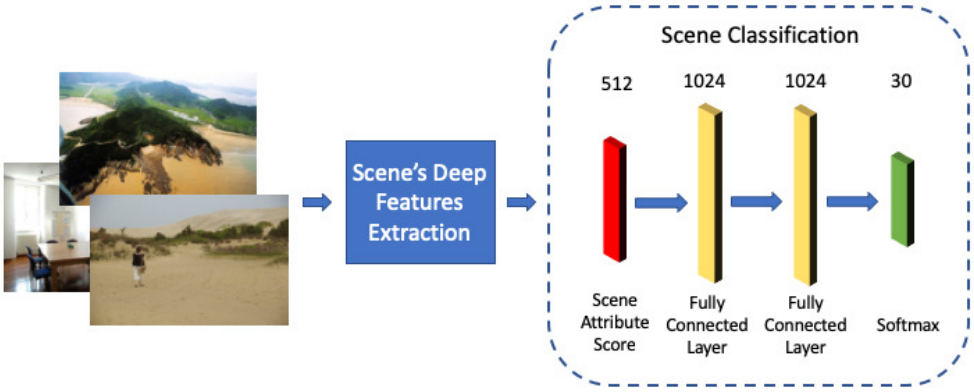


Figure 3.9: 2D scene classification with scene attributes.

Places365-CNNs [238] 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 7,000 training 2D scene images, they obtain a training data with a dimension of $7,000 \times 102$ 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. 3.9**). A training algorithm called Batch Gradient Descent with Adam optimizer [92] 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 [125] 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 [176, 190]. 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 [190], they propose their method to train the target mapping model so as to match the source distribution for place classification. **Fig. 3.10** 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

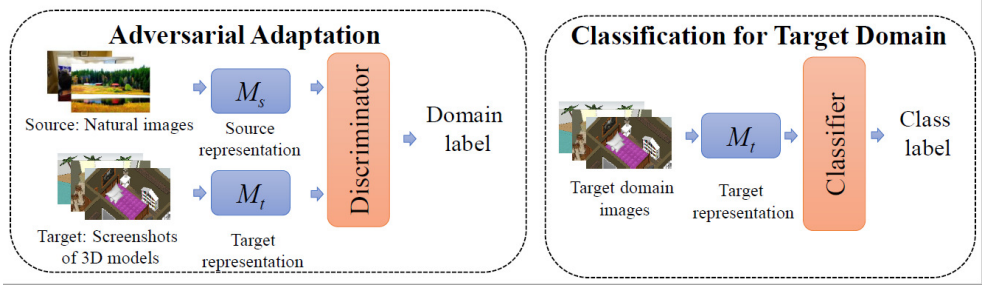


Figure 3.10: Place classification for screenshots of 3D models with adversarial discriminative domain adaptation.

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 @ 2.50GHz 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.

- 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.

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 [76] 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 3.3.1) to extract scenes' deep features using MIT Places API [238]. They train a simple network with the extracted features from Places API and use this network to classify an input image with 30 labels.

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 [76] 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 1,024 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.** 3.11. The network is trained on a server with 1 $\times$ 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.

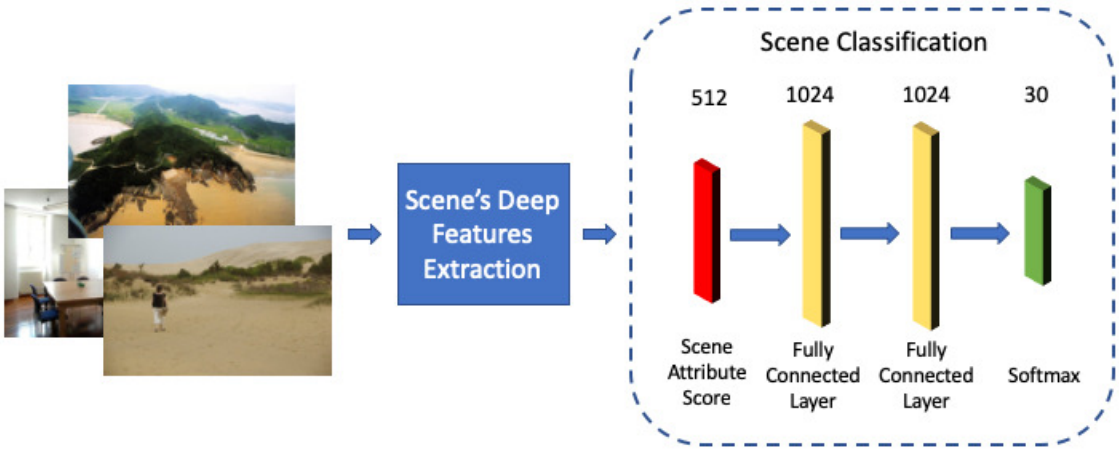


Figure 3.11: 2D scene classification with scenes' deep features.

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. 3.12**.

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 [51] and Places365 [238].

The first model is to extract feature vectors for each image using ResNet-50 [76] 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 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,

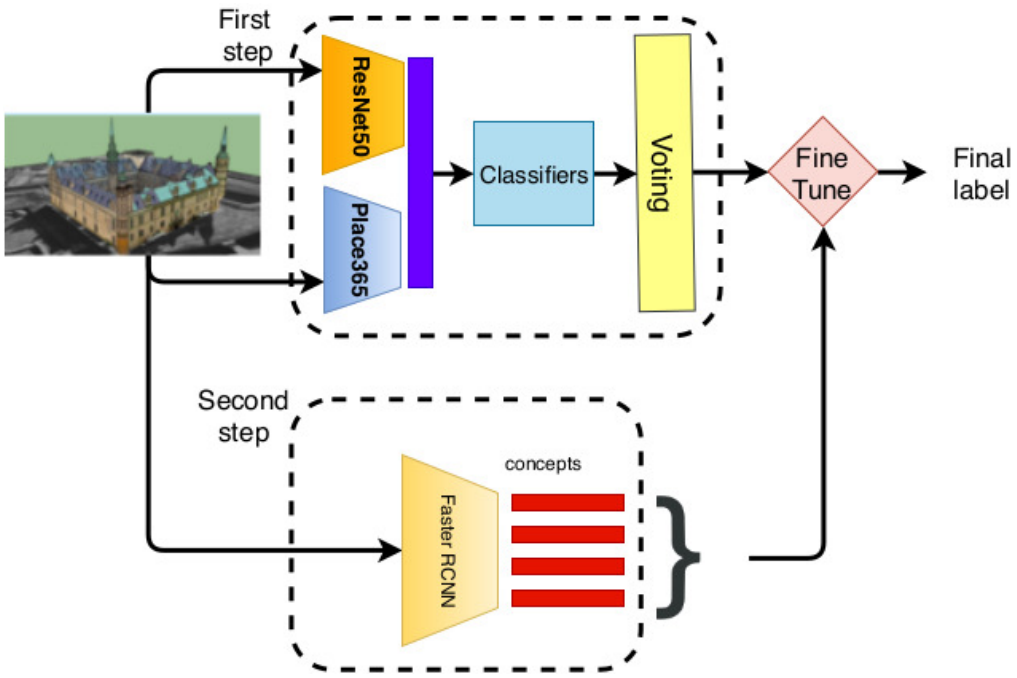


Figure 3.12: Two-step process of the 3D scene classification method.

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 1,000 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) [225, 226] 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 [162]. 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.

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) [223] by either employing the AlexNet (for Query-by-Sketch only) or the VGG-16 model. Its architecture is illustrated in **Fig. 3.13**.

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. 3.14**.

Data augmentation. To avoid overfitting issues, before each pre-training or training, they employ data augmentation technique (rotations, shifts and flips) [218] 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 [57] for 500 epochs, and pre-train AlexNet2/VGG2 on the Places scene image dataset [238] 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.

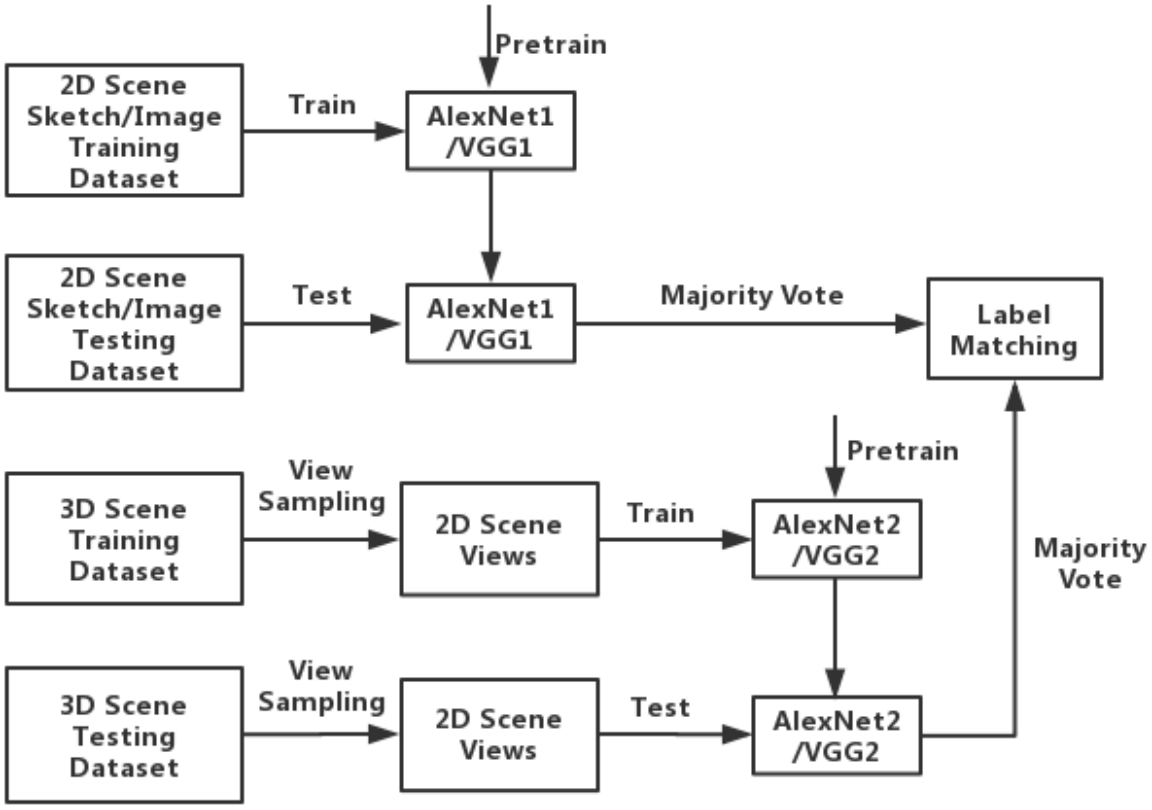


Figure 3.13: VMV architecture [223].

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, 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 [223], while the code is also publicly accessible[2].

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. [228], as illustrated in **Fig. 1.1**. The motivation of this retrieval algorithm is

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

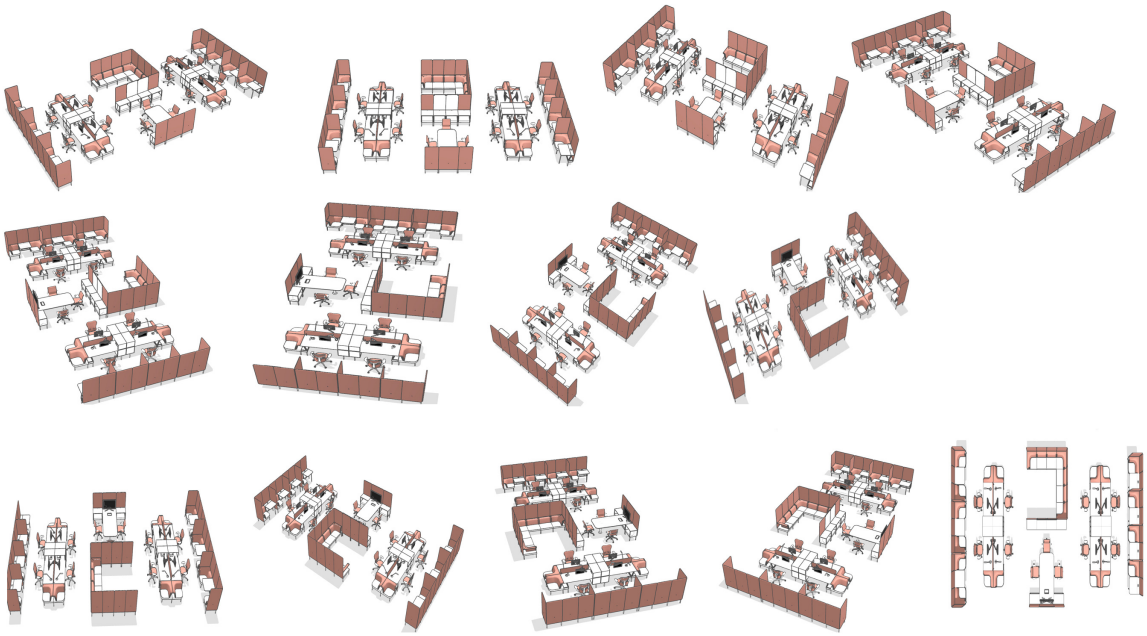


Figure 3.14: A 13 sampled scene view images example of an apartment scene model [223].

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 [140] 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 3.3.1, 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. [184]. 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 [160] 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. 1.1**.

(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,

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

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 [101]-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 3.3.1. Please refer to it for more details.

For the original DRF related code, data, and experimental results, please refer to [228].

3.3.2 Query-by-Image retrieval

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 3.3.1 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 [120, 147] 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 3.3.1).

For both 2D scene images and 3D model views, they follow the same retrieval process. First, they apply the Hessian Affine detector to detect the interest points N in each image, either a 2D scene image or a 2D view of a 3D model. They use RootSIFT [16] without angle

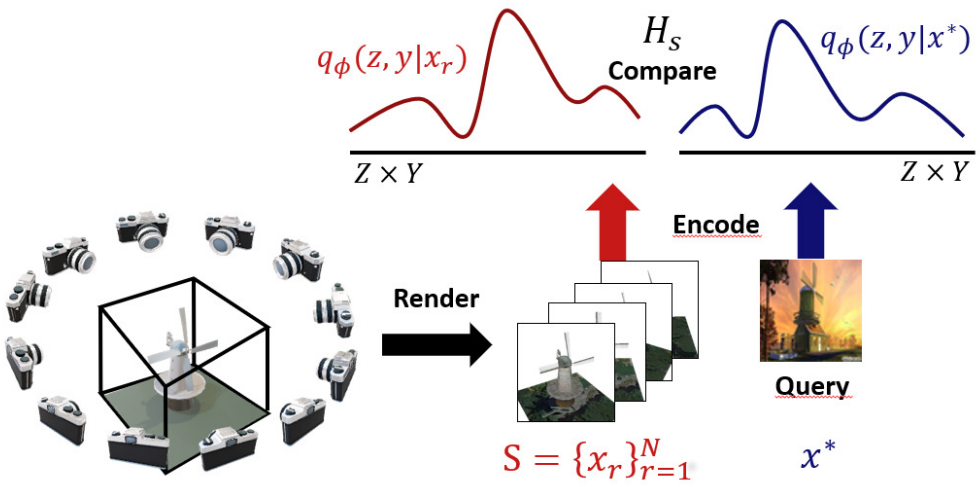


Figure 3.15: Overview of scene sampling and CVAE distribution learning.

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 [241], 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 hours 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 second 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.

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) [95], as shown in **Fig. 3.15**. 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 [184] 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 [95]. 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 [93, 95] 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. 3.15**).

$$D(x^*, S) \min_{r \in \{1,2,\ldots,13\}} H_s(q_\phi(z|x^*), q_\phi(z|x_r)) + \alpha H_s(q_\phi(y|x^*), q_\phi(y|x_r)). \quad (3.4)$$

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 [87] on the Places data set [238] as part of the encoder.

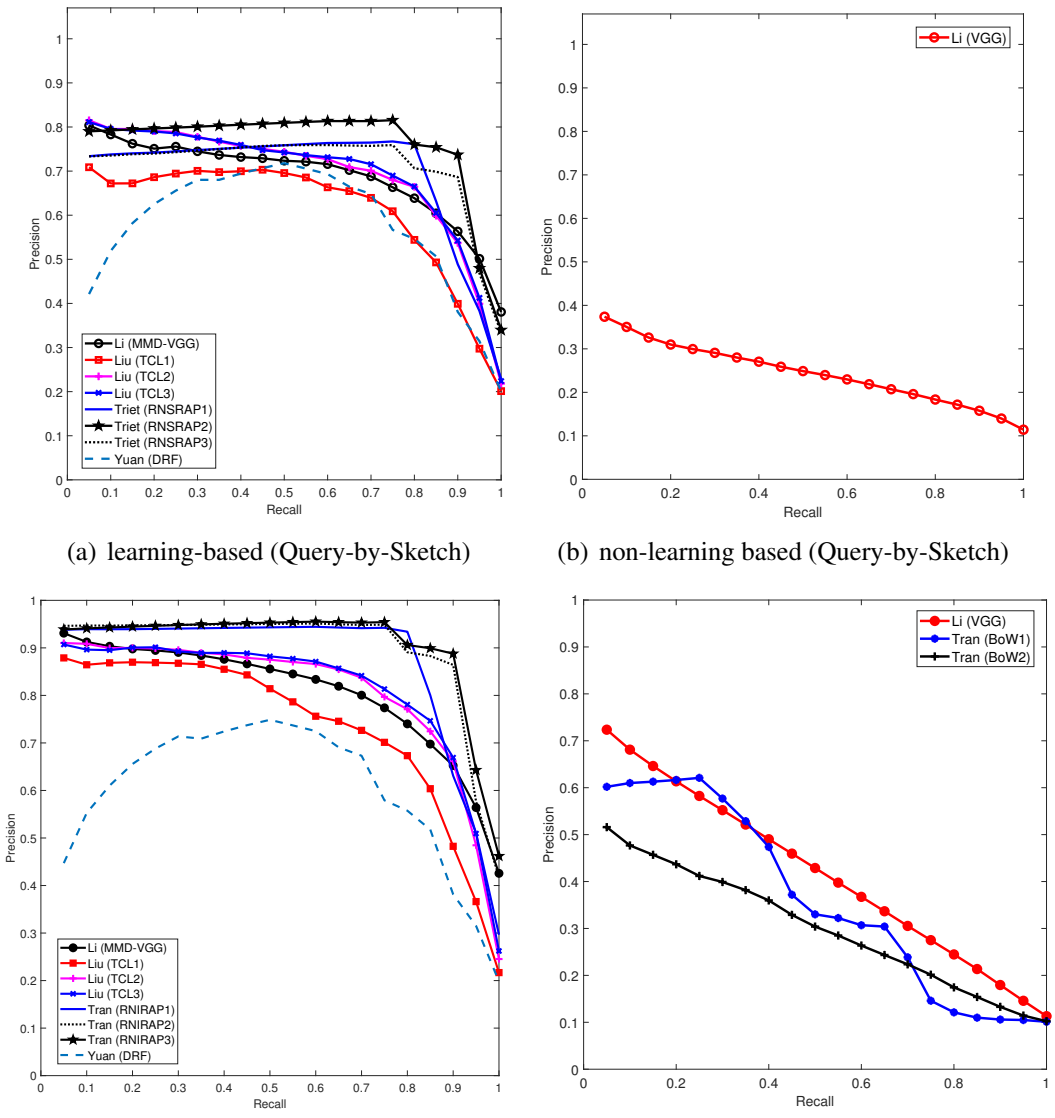
3.4 Results

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 3.2.3: PR, NN, FT, ST, E, DCG and AP.

3.4.1 Scene_SBR_IBR_2018 benchmark

Based on the the **Scene_SBR_IBR_2018** benchmark described in Section 3.2.1, 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. 3.16**

and **Table 3.4** 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)

Figure 3.16: Query-by-Sketch and Query-by-Image Precision-Recall diagram performance comparisons on our **Scene_SBR_IBR_2018** benchmark.

Table 3.4: 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

Peer performance evaluation **Query-by-Sketch retrieval.** We fist 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 [225].

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 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 [4].

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 [238] 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,

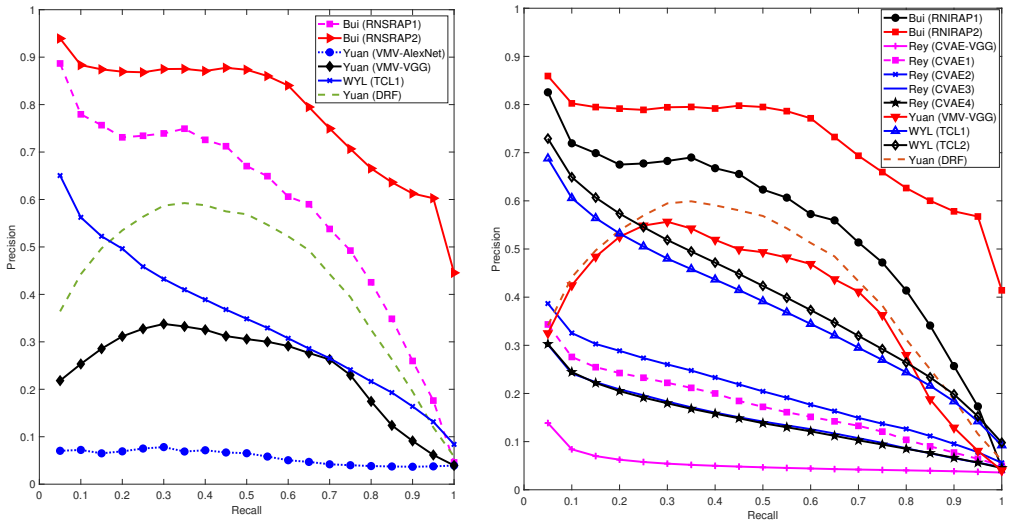
RNIRAP) from three participating groups (Li, Liu, Tran) utilize deep learning techniques to automatically learn local features. Therefore, all of the three groups have considered the deep learning framework for feature learning. DRF also adopts a deep learning-based approach to learn local features. 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.

Cross-track performance evaluation As can be seen from **Fig. 3.16** and **Table 3.4**, 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.

3.4.2 Scene_SBR_IBR_2019 benchmark

Similarly, based on the **Scene_SBR_IBR_2019** benchmark described in Section 3.2.2, 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. 3.17** and **Table 3.5** 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.



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

Figure 3.17: Query-by-Sketch and Query-by-Image Precision-Recall diagram performance comparisons on our **Scene_SBR_IBR_2019** benchmark.

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. 3.17** and **Table 3.5**, 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 [222]. 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

Table 3.5: 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

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 [5]. 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.

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 3.4.1: 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 3.4.1.

3.4.3 Timing performance evaluation

Table 3.6 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 hours 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 3.6. In a word, we think most evaluated algorithms have excellent scalability performance in terms of efficiency for large-scale 3D scene retrieval scenarios.

Table 3.6: 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) @3.3GHz (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 @2.3GHz; 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 @2.30GHz; GPU: 1 x NVIDIA GeForce 920MX) (for BoW only): CPU: Intel(R) Xeon E5-2660 @2.2GHz; Memory: 12 GB; OS: Windows 2008	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 @3.6GHz (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 @2.2GHz (4 processors, 20 cores); Memory: 256 GB; OS: Ubuntu 18.04)	CVAE	Ruby, Python	-	-	-	0.09
	CVAE-VGG	Ruby, Python	-	-	-	0.22

3.4.4 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 **Table 3.4** and **Table 3.5**, 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 3.3.1 and 3.3.1. 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) 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 [223]: 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.

3.5 Summary

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 [6, 226]. 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 [109, 114], SHREC'13 [102, 109], SHREC'14 [106, 112], SHREC'16 [107], SHREC'18 [6, 226]) by running another two related tracks [8, 224] 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 [4, 5, 222, 225]. We also review the related techniques and datasets, and provide a method description for each retrieval algorithm in the project. 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.

Chapter 4

SEMANTICS-BASED 3D SCENE RETRIEVAL

3D scene model retrieval plays an important role in the whole content-based 3D model retrieval research field. To promote this field of research, we organized two Shape Retrieval Contest (SHREC) tracks on 2D scene sketch/image-based and 3D scene model retrieval in 2018 and 2019. In addition, there is a lot of semantic information (i.e., object-object, object-scene, object parts, object groups) existing in 3D scene models. In this project, to further improve 3D scene model retrieval accuracy, we build a semantic scene tree to incorporate such helpful semantic information into the retrieval process. We propose a semantics-based 3D scene model retrieval approach. In our approach, the object-object semantics relatedness information can be learned automatically during the retrieval process. Experiments demonstrate that our semantics-based approach can capture the semantic information of 3D scene models effectively, measure their similarities accurately, and thus the retrieval performance has been significantly improved. The experimental results, code, data can be found on the project homepage: `https://sites.usm.edu/bli/Scene_SBR_IBR/`.

4.1 Introduction

3D model retrieval has always been a hot topic in computer vision, there is a lot of research on it currently. By providing a single sketch, image or model as an input query (Query-by-Sketch, Query-by-Image, Query-by-Model), 3D model retrieval is to retrieve a list of candidate 3D models which are related to the input. 3D scene model retrieval is a brand new research direction in this field of 3D model retrieval. In most of traditional 3D model retrieval research, each of the input query and the target 3D models only contains a single object. However, compared to traditional 3D model retrieval, 3D scene model retrieval is a new type of 3D model retrieval which is more challenging. The input query and the target 3D models are scene categories instead of one single object. Each scene involves multiple objects that may or may not overlap each other, and has spatial context configuration information as well. 3D scene model retrieval has a vast of important applications in real life, including robotics, autonomous driving cars, augmented reality (AR), virtual reality

(VR), 3D movie, 3D game production, etc. Therefore, this research topic is promising and deserves a further exploration.

To facilitate the research on 3D scene model retrieval field, we organized four Shape Retrieval Contest (SHREC) tracks on this research topic in 2018 and 2019. Each year contains a Query-by-Sketch and a Query-by-Image 3D scene model retrieval. In 2018, the dataset (Scene_SBR_IBR_2018) we built only contains 10 scene categories, each consisting of 25 2D scene sketches, 1,000 2D scene images and 100 3D scene models. In 2019, we tripled the size of Scene_SBR_IBR_2018, resulting in an extended dataset (Scene_SBR_IBR_2019). This benchmark has 30 distinct scene categories (10 categories from Scene_SBR_IBR_2018, and 20 extra categories), and has the same amount of 2D scene sketches, 2D scene images and 3D scene models in each category as Scene_SBR_IBR_2018. The performance of the participating methods in 2019 track has dropped apparently compared to those participating methods in 2018 track. The main reason for this is that the benchmark in 2019 has more scene categories, thus, it is more general and comprehensive. Therefore, it is urgent to develop a more robust and scalable method that is not affected by the size of the benchmark too much.

As a common sense, we know that there is a lot of semantic information existing in 3D scene models, such as object-object, object part-object part and object-group semantic information. Taking advantage of such helpful and important semantic information will have a significant and positive impact on the 3D scene model retrieval performance.

Motivated by the above facts, to further improve the retrieval accuracy, we propose a semantics based 3D scene retrieval approach. In this approach, the semantic relatedness between each object is captured automatically by using a Gradient Descent-based deep learning method during the retrieval process. Compared to utilizing the semantic relatedness based on the semantic ontology of WordNet, the semantic relatedness obtained by the Gradient Descent-based deep learning method is more adaptive to different types of benchmarks. Experimental results demonstrate a great improvement in the performance of the retrieval after utilizing the semantic relatedness information. In addition, it also proves that the semantic information of 3D scene models can be captured effectively and the similarities can be measured accurately by our approach during the retrieval process.

4.2 Semantics-driven 3D scene model retrieval

4.2.1 Overview

We propose a semantics-driven 3D scene model retrieval method in this project. We follow the retrieval framework of View and Majority Vote (VMV) [112] for 3D scene retrieval learning process, while we also incorporate semantics loss during the process and propose a deep random field (DRF) model. As illustrated in **Fig. 1.1**, our method includes the following five steps.

(1) **3D scene model view sampling**: for each 3D scene model, we sample 13 views by setting up 12 cameras on the equator of its bounding sphere as well as one camera on the top of the sphere. A QMacro script program is also developed to perform the operations of view sampling process on the SketchUp software automatically. One example is shown in **Fig. 3.14**.

(2) **Semantic object instances segmentation**: semantically segment each of the sampled 2D view images into a set of objects. For instance, as shown in **Fig. 1.1**, we semantically segment a 2D view image of a 3D classroom into the following instances: several desks, chairs, doors, windows, together with a white board. The *semantic* information of the scene view is composed of the category names of these segmented objects and their occurrences.

(3) **Semantic relatedness learning**: our method is data-driven, by utilizing the Stochasic Gradient Descent (SGD) algorithm. We obtain the semantically relatedness relationship between the labels of the segmented objects and the labels of the candidate scene category so as to incorporate this semantic information into the training and prediction process.

(4) **Semantic loss computation**: semantic similarity is computed between the semantics of each target scene category and that of the unknown 3D scene model, based on the semantic relatedness information of each 3D scene category pre-learned in Step (3), and the categorical names of the scene objects and the number of times they appear in the scene.

(5) **VGG-based joint loss retrieval (JLR)**: we adopt the VGG16 as our main framework to train our 3D scene retrieval model. The difference is that we not only use VGG's cross-entropy loss but also combine it with our semantics-based loss and the Triplet Center Loss (TCL) [78], whose details will be illustrated in Section 4.2.3. Finally, the trained JLR model is utilized to retrieve each 3D scene model on the testing dataset.

4.2.2 Semantic object instances segmentation and relatedness learning

Semantic object instances segmentation We utilize the YOLOv3 [160] model to detect all the possible objects that appear in each 3D scene model. YOLOv3 is an algorithm that can do object detection in image or video in real-time. However, the publicly available pre-trained YOLOv3 model can only detect 183 different categories of objects [122], which belong to the category list of COCO stuff [30], a widely used 2D scene image benchmark for large-scale object detection, captioning, and segmentation. Since our training and testing dataset contain some objects that are not included in these 183 categories, in order to better meet our requirements, 27 additional categories are added in our experiments, and you can find the names on our project homepage[1]. It is important and necessary to enlarge the dataset for training process by incorporating extra manually-annotated object instances for the additional 27 categories. The chance of the additional 27 categories appearing in certain scenes is also very high. For instance, desert is categorized into one of the 30 scene classes of our benchmark, and we know that cactus are very common objects in desert scenes. However, the original 183 categories do not contain cactus. Assume the set of our 210 (183 original categories plus 27 additional categories) object categories is O=$\{O_1, \ldots, O_n\}$. By utilizing the YOLOv3 framework, for each 2D scene view generated from a 3D scene model coming from the training dataset, every object category's number of occurrence c_i can be detected, thus the object occurrence statistics C=$\{c_1, \ldots, c_n\}$ information can be obtained. Based on the statistics, we train a 9-layer DNN model to learn the occurrence probability distribution for each object category, which is the probability of each object that appears in a 3D scene. We call this distribution the scene semantic information and will take advantage of it during the training process.

Object occurrence probability $\{P(O_i|S)\}$ is the conditional probability of a certain object O_i that appear in a a certain scene category S. For each layer of the DNN model, we respectively set the number of its nodes as 500, 625, 500, 400, 600, 300, 200, 120, and 210.

Scene Semantic Tree definition. WordNet [140] provides a broad and deep taxonomy with over 80K distinct synsets representing distinct noun concepts arranged as a directed acyclic graph (DAG) network of hyponym relationships (e.g., "table" is a hyponym of "furniture"). As shown in **Fig. 1.1**, a Scene Semantic Tree (SST) is a hierarchy of classes with corresponding 3D scene models organized based on the semantic hierarchy in WordNet

[1] `https://sites.usm.edu/bli/Scene_SBR_IBR/`

synsets. Each class (synset) of the Scene Semantic Tree has several attributes (i.e., via is-a, has-part, or is-made-of relations) according to its gloss defined in WordNet. Each leaf node of the Scene Semantic Tree has a number of 2D images belonging to the leaf node class. It also contains the scene semantics information (Object occurrence probability) learned in Section 4.2.2. Therefore, the Scene Semantic Tree forms a network of classes, attributes (i.e. scene object categorical names and their estimated distribution), and related 3D scene model files.

Semantic relatedness learning In the Scene Semantic Tree (SST) [228] work, we adopted the WordNet [140] as the tree structure, which is a hierarchical tree consisting of semantically-related concepts, to obtain the relatedness relationship between the labels of the objects and the labels of the scene category, this relatedness relationship is general and real world-based. However, the 2D scene sketch or 3D scene model datasets coming from diverse resources can be significantly different in appearance from those in the real world. For example, 2D scene sketches, which are iconic, and not realistic. WordNet is applicable in semantically hosting different types of scene data (sketches, images, and models) if we can accurately perform related scene recognition and object detection on those data. In order to obtain the semantic relatedness which fits our dataset, we propose a Gradient Descent-based deep learning method. We adopt the WordNet semantic relatedness as the initial values. During the training process, these semantic relatedness values will be adjusted automatically.

4.2.3 Joint loss definition and JLR model training

Joint loss in DRF model The standard way to classify the objects in a scene or an image is to treat each object independently and train a deep neural network (DNN) to classify each object. To improve the recognition accuracy, we incorporate the semantically relatedness relationships between the detected scene objects' labels and the candidate scene category labels into the training and prediction, as well, by utilizing the Scene Semantic Tree. For example, an object *table* detected from an unknown scene is more likely to help us classify the scene to be *kitchen* or *restaurant* rather than *phone booth* or *shower*, because *table* is a "PART_OF" *kitchen* or *restaurant* — they are more semantically related. We name our model as deep random field (DRF), because the way to encode the relationships resembles

Markov random fields [39]. The loss function of our DRF model is,

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

where, $\mathcal{L}_{\text{DNN}}$ and $\mathcal{L}_{\text{SST}}$ are the standard loss of a DNN classifier and the semantic loss based on the Scene Semantics Tree (SST), respectively, while both are defined based on the cross-entropy loss function (binary cross-entropy (BCE) for $\mathcal{L}_{\text{SST}}$); λ is a hyper-parameter that represents the strength of the standard DNN part; R_i is the WordNet-based semantic relatedness between two semantically related concepts: the object class name O_i and a candidate scene category S to classify the scene view. In our experiments, we adopt the Lesk [101] algorithm as the relatedness measurement; c_i is the detected number of occurrences of O_i in the scene view image; $\{P(O_i|S)\}$ is the scene semantics information of S learned in Section 4.2.2. The learning will be optimizing the loss function to jointly estimate the weights of DNN. Before loss combination, we scale both DNN and semantic losses to be in the range of [0, 1].

Joint loss in JLR model To improve the 3D scene retrieval accuracy, we first incorporate the gradient descent-based semantically relatedness, which is the relationship between the labels of the detected scene objects and the labels of the candidate scene categories, into the training and prediction processes. Similar to aforementioned DRF model, if an object detected in a scene query has closer semantic relatedness with one candidate scene category, then the scene query is more likely to be categorized into this candidate scene category. Secondly, we also utilize Triplet Center Loss (TCL) [78] in our model as well. The principle of TCL is that TCL can learn a center for each scene category, and the scene samples that belong to the same category have a closer distance from the center if compared with the samples that belong to different categories. The loss function of our JLR model is defined as follows,

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{DNN}} + \lambda_2 \mathcal{L}_{\text{GD}}(\{R_i * c_i\}, \{P(O_i|S)\}) + (1-\lambda_1-\lambda_2)\mathcal{L}_{\text{TCL}},$$

where, $\mathcal{L}_{\text{DNN}}$ and $\mathcal{L}_{\text{GD}}$ are the standard DNN classifier cross-entropy loss and the gradient descent-based semantic loss defined based on binary cross-entropy (BCE), respectively. While $\mathcal{L}_{\text{TCL}}$ is the TCL loss function; λ_1 and λ_2 are two hyper-parameters that represent the strength of the standard DNN part and the semantic part. R_i is the gradient descent-based semantic relatedness relationship between the object category label O_i and a target scene

category label S for the scene view classification; c_i is the number of occurrences of O_i detected in the view image. $\{P(O_i|S)\}$ represents the scene semantics information of S obtained in Section 4.2.2. During the training process, the loss function $\mathcal{L}$ is optimized by the three losses and thus jointly estimate the weights of our DNN model. We also scale all the three types of loss values in the range of [0, 1].

4.3 Experiments and discussions

4.3.1 Dataset

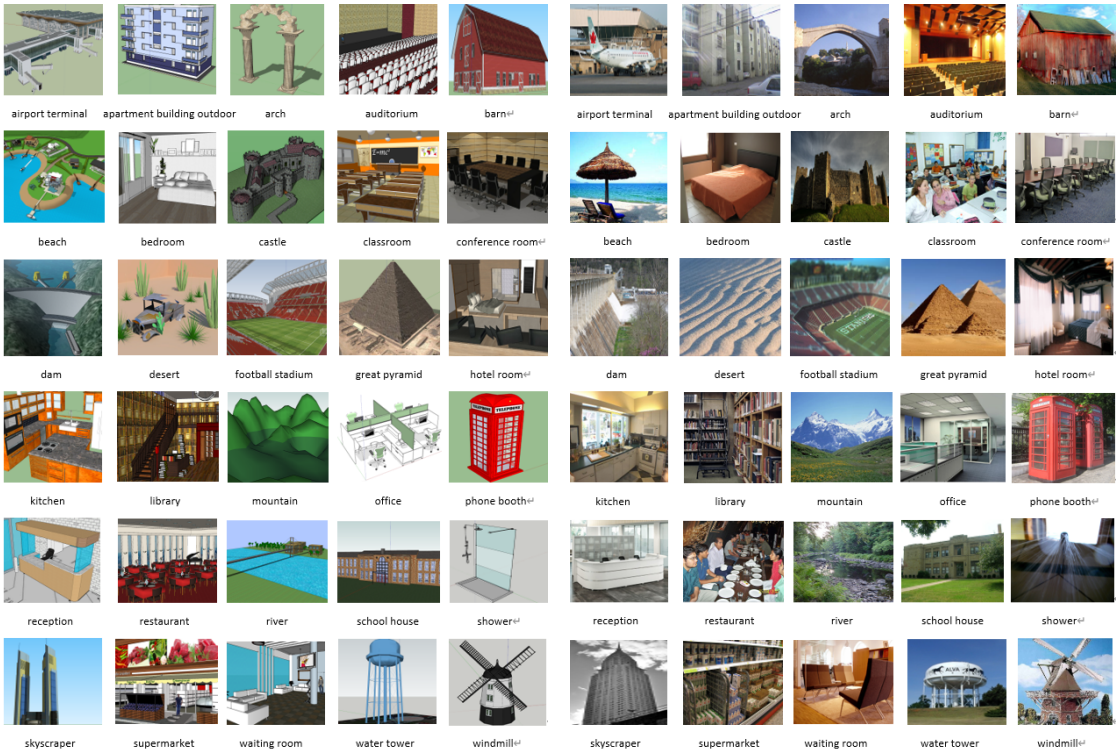
We conduct a comprehensive evaluation of our semantic scene recognition algorithm based on the latest sketch/image-based 3D scene retrieval benchmark built by us, named **Scene_SBR_IBR** [223]. **Scene_SBR_IBR** was also used by us in organizing two 2019 Eurographics Shape Retrieval Contest (SHREC'19) tracks on 3D scene shape retrieval. It contains three subsets: 750 2D scene sketches, 30,000 2D scene images, and 3,000 3D scene models. All the 2D sketches/images and 3D scene models are equally classified into 30 classes. For each class, 18 sketches, 700 images and 70 models were randomly chosen for training while the rest 7 sketches, 300 images and 30 models were kept for testing. We utilize its 3D scene subset (testing dataset portion) to evaluate our 3D scene recognition algorithm, while using its image subset (training dataset portion) for scene semantic information extraction (See Section 4.2.2 for details and results in Section 4.3.3), considering its much larger size than that of the 3D scene dataset, much higher overall accuracy in scene details, and much more realistic scene features. A 3D scene example and a 2D scene image instance for each class are demonstrated in **Fig. 4.1** (a) and (b), respectively.

4.3.2 Scene object categories

To learn the scene semantics information for the target 3D scenes in the **Scene_SBR_IBR** benchmark, we choose the maximum number of possible different object categories that may appear in the 3D scenes to be 210 by adding 27 additional classes, together with their manually annotated object instances to meet the needs of the **Scene_SBR_IBR** benchmark. The list of the 27 additional classes can be found on our project homepage.

4.3.3 Object occurrence probabilities

By following the approach presented in Section 4.2.2, for each scene category, we first adopt YOLOv3 [160] framework to detect the objects in each scene image within the



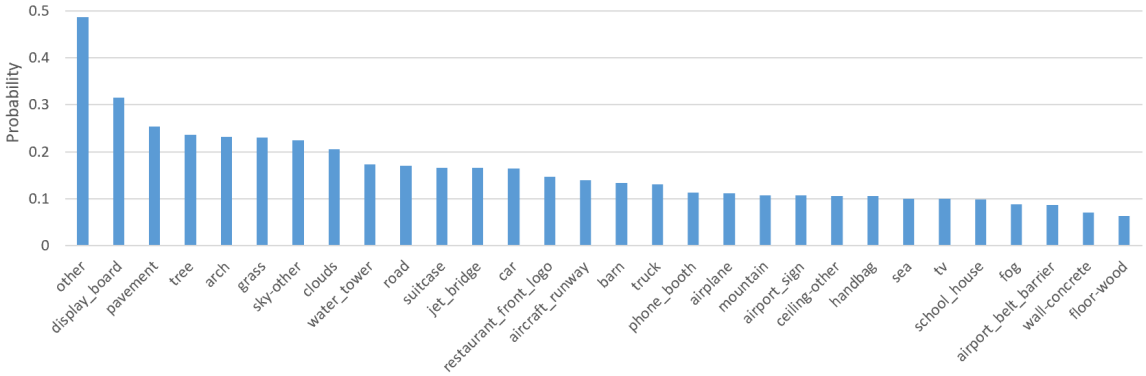
(a) models (b) images

Figure 4.1: 3D target scene model and 2D scene image examples in our **Scene_SBR_IBR** benchmark. One example per class is shown.

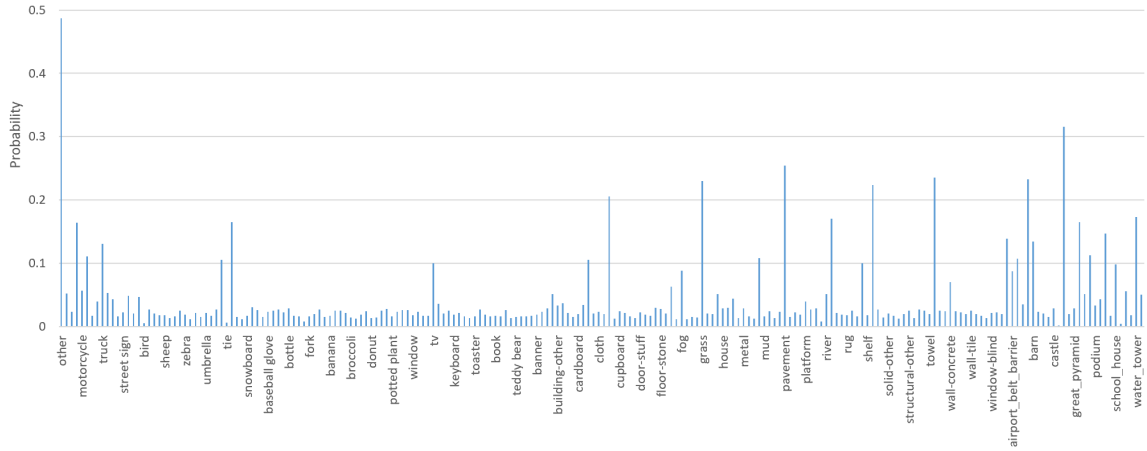
category to form the image's scene object statistics, and then individually employ a 9-layer deep neural network to train on all the obtained object statistics of the scene images to build the object occurrence probability for that scene category. **Fig. 4.2** shows an example result on the airport terminal scene class. Similarly, all the 30 scene categories' object occurrence probability distributions are available on the project homepage: `http://github.com/3DSceneRetrieval`.

4.3.4 3D scene retrieval results

DRF model results We evaluate our DRF approach based on the testing dataset of the 3D scene subset of the **Scene_SBR_IBR** benchmark, and compare with the adapted MVCNN [184] approach (i.e., using the Places365 [238] pretrained model for the VGG part) for 3D scene recognition, which was named scene-based MVCNN (sMVCNN) by us. As described in the Section 4.2, DRF shares with MVCNN in terms of the recognition



(a) top 30 classes



(b) all the 210 classes

Figure 4.2: Object occurrence probabilities for the airport terminal scene category.

framework but incorporates the additional semantic-tree based loss into the loss function definition. Since we are dealing with scene models, rather than single object models like MVCNN, we adopt the scene image recognition model Places365 which is also based on VGG.

Firstly, we respectively train sMVCNN and DRF based on the training dataset (sampled scene images) of the target 3D scene dataset **Scene_SBR_IBR**, by starting with the Places365 pretrained model [238] or a randomly initialized Places365 model. We search the best λ values based on a coarse-to-fine grid search with a search step of 0.1 and 0.01, respectively. The best λ values are 0.67 and 0.57 for DRF started with a pre-trained and randomly initialized Places365 model, respectively. Secondly, we test the trained sMVCNN and our DRF model with the corresponding testing dataset based on their scene images as well. **Table** 4.1 compares their recognition accuracies.

Table 4.1: Scene recognition accuracy comparison on the testing dataset of **Scene_SBR_IBR**.

Accuracy	**Pre-trained**	**Randomly initialized**
sMVCNN [184]	0.529	0.537
DRF (Our)	0.594	0.585

We can find that on the **Scene_SBR_IBR** dataset, for either way of model initialization, after incorporating the scene semantic information, our DRF has achieved an improvement of 12.3% and 8.94% in the accuracy, respectively, if compared to sMVCNN which does not consider the available scene semantic information. This demonstrates that our semantic-tree based approach has successfully captured the scene semantic information existing in the 3D scenes, and also accurately measured their similarities, thus significantly improved the 3D scene recognition performance.

JLR model results Our JLR approach is evaluated on the 3D scene testing dataset of the **Scene_SBR_IBR** benchmark. We compare our method with the DRF approach and the TCL approach for 3D scene retrieval. As described in the Section 4.2, JLR shares with DRF model in terms of the standard DNN classifier cross-entropy loss but learning the semantic relatedness information based on gradient descent instead of utilizing the WordNet relatedness relationship directly. In addition, we incorporate the Triplet Center Loss (TCL) in our loss function as well. Since both of the DRF and TCL approaches adopt VGG as the training model, we also utilize the VGG model in our approach so as to exclude the influence of the model on the results.

First, we train our JLR model and the TCL method on the 3D scene training dataset (sampled scene images) of the **Scene_SBR_IBR** benchmark respectively. For the JLR model, we set both λ_1 and λ_2 values to 0.33, which indicates the strength of three parts in the loss function is the same. Secondly, we test the trained JLR model and the TCL model on the corresponding 3D scene testing dataset as well. **Table** 4.2 compares and demonstrates their retrieval accuracies.

By comparing JLR (gradient descent only) results with DRF ones, we can find that its accuracy increased by about 2.8% in NN and 2.7% in AP. This is because the gradient descent-based learned semantic relatedness is more suitable for our benchmark than the semantic relatedness from WordNet, thus, it achieves a better performance. After incorporating the learned scene semantic information, compared to the original TCL method,

Table 4.2: 3D scene retrieval performance comparison on the 3D scene testing dataset of **Scene_SBR_IBR**.

Accuracy	**NN**	**FT**	**ST**	**E**	**DCG**	**AP**
DRF [228]	0.597	0.357	0.500	0.358	0.690	0.358
TCL [78]	0.632	0.375	0.521	0.376	0.706	0.378
JLR (GD only)	0.614	0.366	0.510	0.367	0.698	0.368
JLR (our)	**0.718**	**0.435**	**0.582**	**0.435**	**0.751**	**0.446**

our JLR has achieved an improvement of 12.3% and 15.3% in the accuracy of NN and AP, respectively. These demonstrate that our gradient descent-based JLR method successfully captures the semantic information of the 3D scene, more accurately measures their similarity, thereby significantly improves the performance of 3D scene retrieval.

4.4 Summary

In this chapter, we focus on the task of semantics-based 3D scene retrieval. We propose two VGG-based joint loss retrieval which incorporates the WordNet and gradient descent-based semantic relatedness information existing in the scenes into the training process, respectively. The occurrence information of the objects is also considered in this semantic information. Experiment results demonstrate that by jointly using semantic information and TCL in the loss function, compared to WordNet-based and other approaches, our Gradient Descent-based deep learning method can improve the retrieval performance by effectively capturing the semantic information of 2D scene images and 3D scene models and also can measure their similarities accurately.

Chapter 5

IMAGE TO SCENE SKETCH GENERATION

Image generation from sketch is a popular and well-studied computer vision problem. However, the inverse problem image-to-sketch (I2S) synthesis still remains open and challenging, let alone image-to-scene sketch ($I2S^2$) synthesis, especially when full-scene sketch generations are highly desired. In this project, we propose a framework for generating full-scene sketch representations from natural scene images, aiming to generate outputs that approximate hand-drawn scene sketches. Specifically, we exploit generative adversarial models to produce full-scene sketches given arbitrary input images that are actually conditions which are incorporated to guide the distribution mapping in the context of adversarial learning. To advance the use of such conditions, we further investigate edge detection solutions and propose to utilize Holistically-Nested Edge Detection (HED) maps to condition the generative model. We conduct extensive experiments to validate the proposed framework and provide detailed quantitative and qualitative evaluations to demonstrate its effectiveness. In addition, we also demonstrate the flexibility of the proposed framework by using different conditional inputs, such as the Canny edge detector.

5.1 Introduction

Image-to-Image (I2I) translation has received a lot of attentions [41, 84, 128, 155, 242] due to its many applications, including generating new data for training deep learning models. If we consider human-drawn sketches as a special type of image, then this problem comprises two subproblems: Sketch-to-Image (S2I) and Image-to-Sketch (I2S) translation. However, till now researchers have focused mainly on the S2I problem, including all the aforementioned research works, and also only considered single object-based sketches. According to our knowledge, there is no published research work in the Image-to-Scene-Sketch ($I2S^2$) research direction.

However, there is a urgent need to curate a large-scale scene sketch dataset for related applications, such as 2D scene sketch-based 3D scene retrieval [224]. Currently available and related scene sketch/contour datasets [134, 244] are either too small in terms of size or

Figure 5.1: Example sketches rendered by our method based on given images. Row 1: given images. Row 2: rendered sketches.

limited in within-class variations in terms of quality. For example, Berkeley Segmentation Dataset and Benchmark (BSDS500) [134] has only 500 natural images, and 2,500 contour sketches in total, while the Photo-Sketching dataset [117] has 5,000 contour images for 1,000 outdoor scene images. The SketchyScene dataset [244] composed each scene sketch by selecting among a limited number of pre-defined object sketches, thus it can not meet our requirement in generate realistic scene sketches. Due to lack of available high-quality 2D scene sketch data, collecting/generating a large number of scene sketches for training deep learning models for related applications is also a challenging task, even by using Amazon Mechanical Turk. Therefore, we are considering an automatic way to generate 2D scene sketches by using the existing large amount of 2D natural images by training a Generative Adversarial Network (GAN) model [68], that is developing a GAN-based Scene Sketch generation approach, dubbed SceneSketchGAN.

The main challenges involved in human-drawn scene sketch generation are mainly related to the inherent characteristics of human sketching: people draw sketches in different styles and at different levels of abstraction. This poses a highly under-constrained question for us. Motivated by the success of CycleGAN [242] in handling a similar problem: generating images from unpair data, we adopt a similar framework. However, we found it is still challenging to develop an end-to-end solution which generates satisfactory results. Then, to add more constraints to the CycleGAN model to solve this under-constrained

problem, we need to provide a conditional input, rather than the original image. Using different types of conditional inputs will generate human-drawn sketches with different styles and/or levels of abstraction. This motivates us to further investigate the role of conditional inputs in training a generative model for the problem of sketch generation. Finally, we utilize a feature selection process by providing the holistically-nested edge detection (HED) [214] map of a natural image as the conditional input, rather than using the raw natural image directly. Therefore, our framework can be generalized as **Edge Map + GAN**, as demonstrated in Fig. 5.3. We conduct extensive experiments including ablation studies to evaluate the proposed framework, and both quantitative and qualitative results demonstrate the effectiveness and competitiveness of our method. We illustrate several generated sketch examples in Fig. 5.1. More results can be found in the experiments section and here: `http://tinyurl.com/qrxq78o`.

In a word, our contributions can be concluded into three-fold:

- We propose a new research problem image-to-scene sketch (I2S^2) translation, which has a urgent need in meeting the large-scale benchmark requirements for scene sketch related applications.
- We evaluate different conditional inputs for image-to-sketch generation and demonstrate that edge-map is suitable for this task in terms of distribution mapping.
- We present a simple yet effective framework to leverage HED edge map-based feature selection (input conditioning) and a CycleGAN-based distribution mapping to generate appealing hand-drawn scene sketches.

5.2 Methodology

In this section, we firstly introduce our theoretical motivation of conditional input, and then discuss a feasible solution to extract desired conditional input. Secondly, we analyze the methods used for distribution mapping for sketch generation, and propose to use CycleGAN to perform such mapping. Finally, we present a framework to leverage each component for full-scene sketch generation.

5.2.1 Input conditioning theoretical analysis

Sketch generation from a given arbitrary input image can be regarded as a conditioned-generation task. Formally, given an input image x, the corresponding sketch y can be

Figure 5.2: Sketches generated by the CycleGAN [15] using the given images as direct inputs. Row 1: given images. Row 2: generated sketches.

obtained by mapping x to y using a distribution mapping function g, having $y = g(x)$.

Nevertheless, image to sketch generation is quite different from regular generation tasks. Using a regular input image directly may lead to poor performance. As an example, we adopt regular images as the inputs for a generative model (e.g. CycleGAN), and train the model to generate sketches. The results are unsatisfactory, as shown in Fig. 5.2. In traditional image generation tasks [41, 84, 128, 155, 242], the generated images contain ample information, and it is relatively less challenging to perform a mapping from randomly sampled inputs to the generated results in light of GAN [68] or Variational Auto-Encoder (VAE) [94] theories. However, as a sparse image, our target sketches usually contain much less clues than regular images and are far from sources. This makes the traditional image generation pipeline not an ideal candidate for image-to-sketch generation.

Here, we introduce a theoretical motivation of our conditional input. In the context, x is the source image, G and D denote generator and discriminator, respectively. f denotes a function applied on the source input to obtain conditional input.

Training a typical GAN will need a loss function that is shown in Eq. 5.1.

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G,D,X,Y) = {} & \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D(y)] \\ & + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1-D(G(x))].\end{aligned} \tag{5.1}$$

We would like to investigate how a conditional input will impact the loss function aiming to boost the performance of a GAN. To incorporate a conditional input, we replace $D(G(x))$ with $D(G(f(x)))$, which means a conditional input $f(x)$ will be fed into the generator. This conditional input is obtained by applying the function f to the original input x. According to the *Lanczos* Approximation of Composite Functions [45], we can have Eq. 5.2.

$$D(G(f(x))) \approx D(UQ_GW_Gf(x)), \tag{5.2}$$

where U, Q, and W are the matrices used in the *Lanczos* Approximation of Composite Functions [45]. Interested readers may refer to the original paper for details. Since D is a discriminator network that is non-linear, we can rewrite Eq. 5.2 to Eq. 5.3 that contains an identity mapping and a non-linear operator op.

$$D(G(f(x))) \approx D(UQ_GW_Gf(x)) \approx UQ_GW_Gf(x)\oplus op(x), \tag{5.3}$$

where $\oplus$ denotes an appropriate operation to combine the above two components. If G and D have similar architectures, we can replace G with D in Eq. 5.3, thus having Eq. 5.4.

$$D(G(f(x))) \approx D(UQ_GW_Gf(x)) \approx UQ_DW_Df(x)\oplus op(x). \tag{5.4}$$

We can further replace f with G if they are similar at functionality level, and Eq. 5.4 can be rewritten to Eq. 5.5. It is worth noting that G and f do not have exactly the same functionalities. In our work, G aims to generate sketches whereas f aims to extract an edge map from a given image.

$$D(G(f(x))) \approx D(UQ_GW_Gf(x)) \approx UQ_DW_DG(x)\oplus op(x). \tag{5.5}$$

The *Lanczos* theory tells us $D(G(x)) \approx UQ_DW_DG(x)$, and therefore we reformulate Eq. 5.5 to Eq. 5.6.

$$D(G(f(x))) \approx D(UQ_GW_Gf(x)) \approx D(G(x))\oplus op(x). \tag{5.6}$$

The Eq. 5.6 indicates that it actually incorporates a regularization term (e.g. $op(x)$) to the traditional GAN loss (Eq. 5.1) if we exploit a function f that has a similar functionality to the generator G. The regularized loss can provide a better training to the generative model, leading to appealing results illustrated in the experiments section. More importantly, this also motivates us to exploit a GAN-based edge detection algorithm as f because of

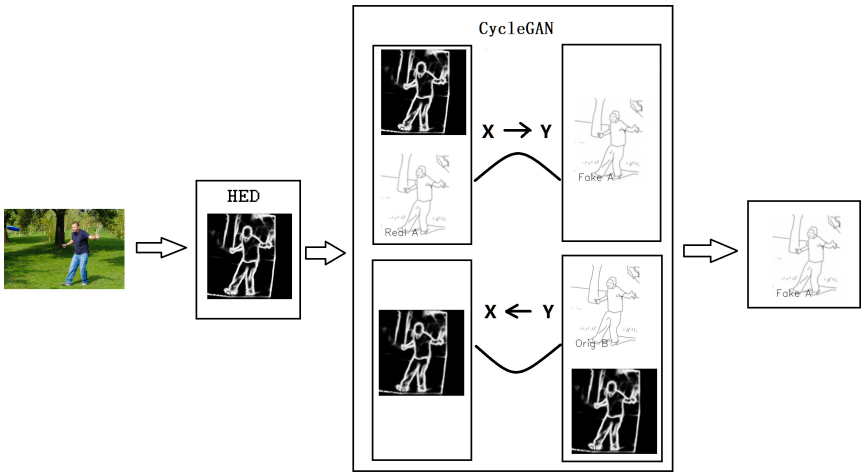


Figure 5.3: The proposed framework I2S^2 for full-scene Image-to-Scene Sketch translation. A natural images goes through two stages: HED edge detection-based feature selection and CycleGAN-based distribution mapping.

two aspects. Firstly, edge detection is well-studied and an edge map can be conveniently extracted from a given arbitrary input image. Secondly, edge detection is similar to sketch generation that is the role of the network G. As a result, we exploit GAN-detected edge map as conditional input for the generative model G in this work.

5.2.2 Edge detection-based conditional input

Edge detection is a well-studied and widely used technology in image processing. Typical methods include the Canny detector [33], Sobel detector [90], Prewitt detector [152], and etc. Technically, any edge detector can be employed to provide a conditional input for our task. However, these traditional edge detection methods have a common issue: lack of ability to produce edges at different scales and levels for images that may have a lot of variance in properties such as contrast and hue [142]. This problem becomes immediately apparent when one applies an edge detection method such as the Canny or Sobel detector to an entire dataset, as some images may yield a good edge map but others may not. In addition, the traditional methods, such as the Canny detector, may need additional thresholds to specify the sensitivity of edge detection, for example, may use the Otsu's method [148] to determine appropriate thresholds.

The Holistically-Nested Edge Detection (HED) method [214] endeavors to address the mentioned issues

by using multiple receptive fields of varying sizes to produce multiple edge maps in

parallel, and deep supervision to weight each output map appropriately. As a result, it can effectively extract edge features in image regions having sharp contrast, thus producing a more complete edge map. Such ability makes HED more suitable for our image-to-sketch generation task. Firstly, the training process of the discriminator during the adversarial learning will benefit from a more complete and accurate edge map, because the discriminator cannot be easily cheated unless the generated sketches are also complete. This will in turn boost the performance of the generator to generate better quality sketches. Secondly, with complete edge information, the generator is found to be more likely to produce reasonable full-scene sketches, while incomplete edge information often fails to provide sufficient conditioning and constraints for the generator's inference.

Based upon the above analysis, we adopt the HED method as the conditioning input function f (see Section 5.2.1). Specifically, to accommodate it in an end-to-end fashion, we utilize the pre-trained HED model to generate an edge map for a given input natural image, and then feed the produced edge map into the generative model which will be detailed in the following section. During training, we freeze the weights of the HED model, and only update the weights of the generative model.

5.2.3 Generative model-based distribution mapping

There are two branches of generative models, namely GAN and VAE. In this work, we exploit a GAN structure to generate sketches, but our framework can be easily changed to accommodate a VAE structure as the generative model. We employ a dataset in which one image corresponds to multiple sketch labels. One option is still using a GAN structure that favors a 1:1 match for the image pair, and designing a new loss to measure the average distance for all labels. The work in [15] takes this scheme. In our work, we argue that the CycleGAN [15] is more suitable for the selected dataset, because it was developed to map an image from an input domain to a target domain without having to be a 1:1 match for the image pair [15].

CycleGAN utilizes two pairs of generators and discriminators to map back and forth between image and target domain feature spaces. During training, the original input image is mapped to the target domain by generator A, and then back to the original input domain by generator B. Meanwhile, the target image is also being mapped to the input image domain and then back again to the target domain. The new cycle consistency loss introduced measures the L1 loss between the original image and target and their respective reconstructions via both generators.

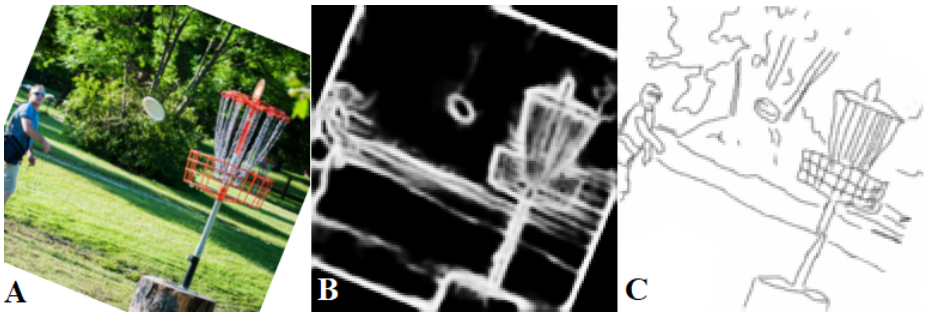


Figure 5.4: Sketch generation example with our model. (A) represents a given color image, (B) is the corresponding conditional input, and (C) is a generated full-scene sketch.

We empirically determine the optimal architecture and configuration of the CycleGAN for the purpose of generating appealing sketches. Each generator is implemented as a 9-layer ResNet [76].

The discriminators adopt the PatchGAN structure [84]. We train the entire generative model by following the CycleGAN pipeline.

It is worth noting that the original identity mapping is used to prevent the model from making any drastic changes when the image or target is close to their respective counterparts. However, we observe that it also helps to prevent producing too many details in our generated sketches. This is highly expected since sketches should be clear and simple, which is essentially different from edge detection. Moreover, it is convenient to control the quality of generated sketches by increasing the cycle loss weights, leading to more realistic sketches. We detail the analysis in Section 5.3.

5.2.4 Framework

Our entire framework for full-scene sketch generation is illustrated in Fig. 5.3. Although we exploit HED and CycleGAN in our framework, since it is general, it is easy to replace these two components with other methods for different purposes. For instance, we investigate the combination of Canny and CycleGAN in Section 5.3, and observe that it can also render acceptable sketches. We would like to highlight that our works aim to explore an effective framework for image to sketch generation. We illustrate an example of an input image, the conditional input, and the output result in Fig. 5.4.

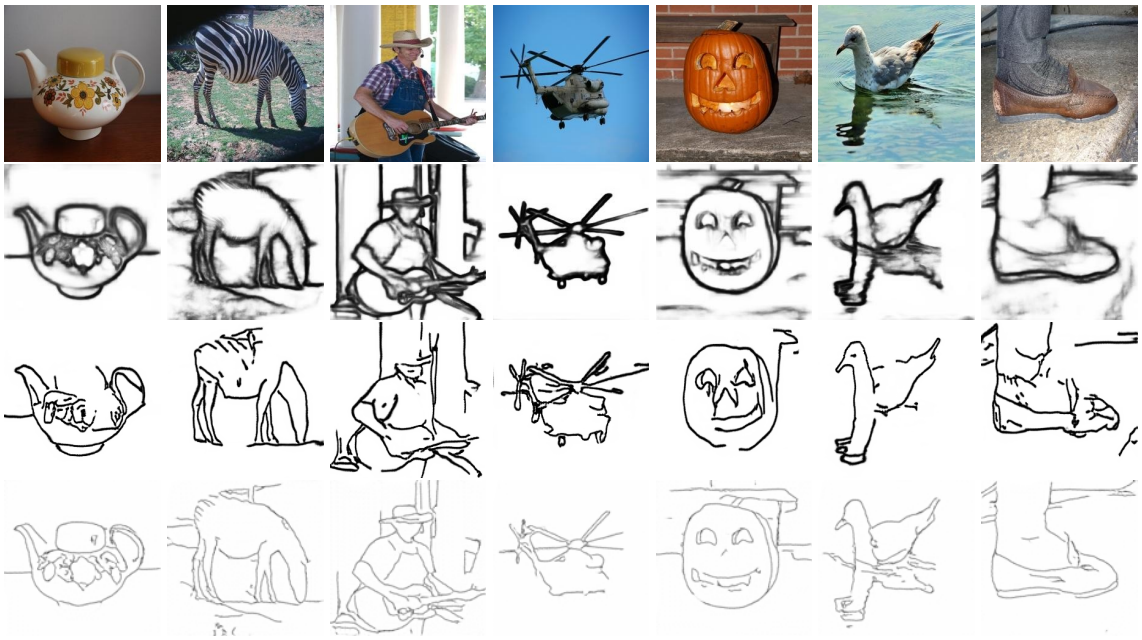


Figure 5.5: Qualitative evaluations of different methods. Row 1: given images. Row 2: results of HED [214]. Row 3: results of Photo-Sketching [117]. Row 4: our results.

5.3 Experiments and discussions

To demonstrate the effectiveness of our framework, we detail our extensive experiments in this section. We firstly introduce the dataset adopted for the experiments. Then, we introduce the evaluation metrics used to quantitatively evaluate the proposed method and training details, followed by a qualitative analysis of the results. Finally, we provide insights through discussions for the potential usage of our framework. Our code can be accessed via the project page.

5.3.1 The Photo-Sketching Dataset

We exploit the dataset curated by Li et al. [117] in our experiments. They crawled a dataset of 1000 outdoor images from Adobe Stock [11], each image is paired with 5 drawings. They selected 5000 high-quality drawings from this dataset. It is ideal for our task due to two main reasons. Firstly, each image in this dataset corresponds to five targets that include varying degrees of detail. This property is beneficial for full-scene sketch generation. Secondly, the contour maps covering almost all the objects in the corresponding images would have the benefit of encouraging our model to depict every significant object that is present in the

Figure 5.6: Generated sketches when different loss functions are employed to train the generative model. Row 1: given images. Row 2: results of the WGAN-loss [17] (**WGAN+**). Row 3: results of the CycleGAN-loss [15] (Our approach).

image. We follow general practices to augment the training images by flipping, rotation, and translation.

5.3.2 Evaluation metrics

To evaluate our method quantitatively, we adopt the Fréchet Inception Distance (FID) [79], Sørensen-Dice coefficient (Dice, a.k.a F-score, F-measure) [53, 181], sensitivity (SN, a.k.a 'recall' or 'hit rate'), and accuracy (Acc). Except FID, higher values are better. FID uses the output of the third layer of the Inception-v3 network trained on the ImageNet dataset [51] in order to measure the earth-mover distance between the generated distribution and target distribution. One main advantage of using FID for evaluation is that we compare related statistics in the feature space rather than doing that at a pixel-level. This is especially important for the image-to-sketch generation task, because a sketch image contains insufficient

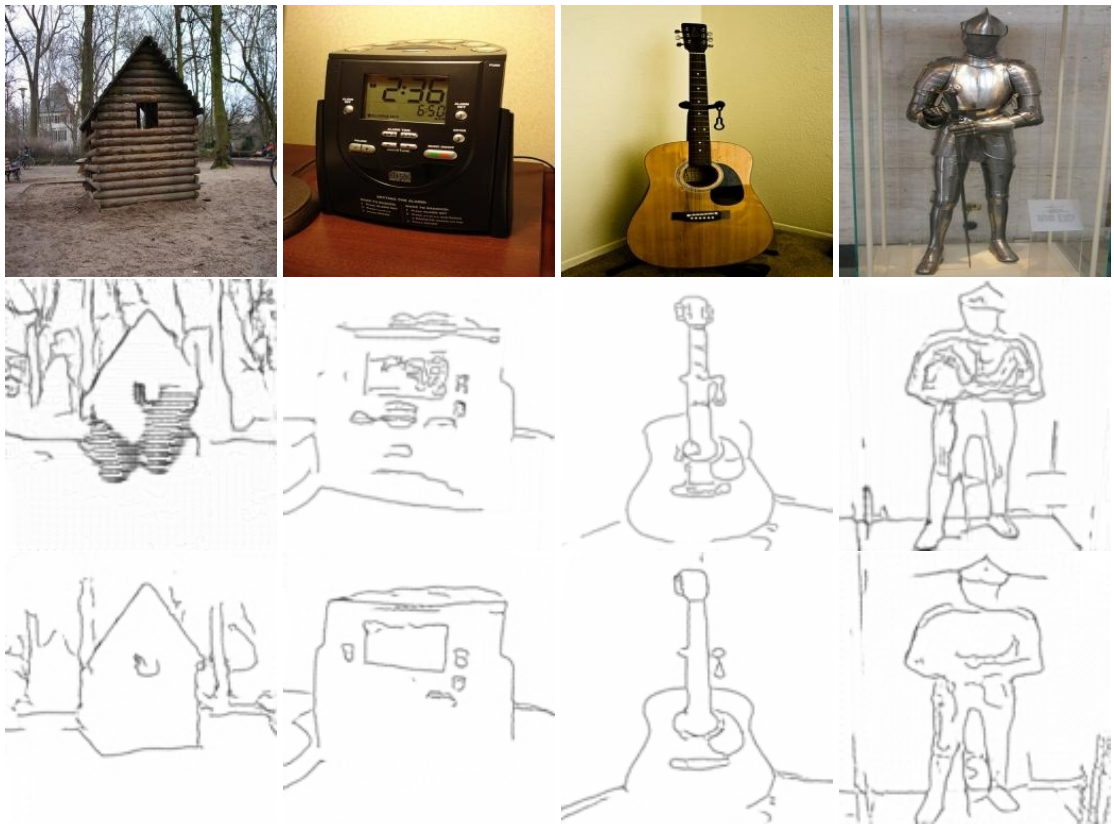


Figure 5.7: Generated sketches when different conditional inputs are used. Row 1: given images. Row 2: when the conditional input is provided by the Canny edge detector (**Canny+**). Row 3: when the conditional input is provided by the HED method [214] (Our method).

pixel information and most pixels are background. A lower FID score corresponds to a higher degree of similarity between images. The fluctuation due to differences in trained weights is small (less than 10% in instances mentioned in [17]), and the domain of object classes we use to train is also small, making FID an appropriate metric for the evaluation of generated sketches.

For Dice, sensitivity, and accuracy, true positive (TP) pixels represent target sketch pixels; false positive (FP) pixels represent background pixels incorrectly generated as sketch pixels. True negative (TN) and false negative (FN) refer to the truth of whether the pixel belongs to the background and is not part of the sketch. The Dice-Sorensen Coefficient (Dice), sensitivity (SN), and accuracy (Acc) are defined in Eqs. 5.7, 5.8, and 5.9, respectively.

$$Dice = \frac{2TP}{2TP + FP + FN} \tag{5.7}$$

Figure 5.8: Full-scene sketches generated by our method. Row 1: given images. Row 2: generated full-scene sketches.

$$SN = \frac{TP}{TP+FN} \tag{5.8}$$

$$Acc = \frac{TN+TP}{TN+TP+FN+FP} \tag{5.9}$$

Dice score can also be viewed as a ratio of intersection of predicted sketch pixels to union of predicted and actual sketch pixels. This metric is commonly used in image segmentation. Sensitivity measures the true positive rate, or recall of the generated sketch. Accuracy measures the ratio of correctly-placed pixels to the total number of pixels. These metrics are all pixel-wise evaluations to measure how well the generated sketch matches the target sketch. We strive to avoid using evaluation methods for boundary or contour detection, since for generating sketches our goal is the quality of the sketch, rather than simply extracting the locations and configuration of contours.

5.3.3 Training settings

It is worth noting that in our framework, the fist component used to provide conditional input is a pre-trained model, and its results are not subject to change with different training settings. Our training details will only affect the second component of the framework, that is, a CycleGAN. Here we only introduce the training settings which lead to the best results

Metric/Method	Photo-Sketching [117]	HED [214]	Canny+	CycleGAN	WGAN+	Ours
FID	103.268	255.942	54.549	47.516	121.575	**32.626**
Dice	0.330	0.293	0.246	0.128	0.197	**0.765**
Acc	0.916	0.842	0.871	0.883	0.912	**0.972**
SN	0.449	0.690	0.445	0.183	0.248	**0.994**

Table 5.1: Quantitative evaluations of generated sketches by different methods. For FID, the lower the better, and the higher the better for the other metrics. In **Canny+**, we adopt the Canny detector to detect the edge map from an image, and use this edge map as the conditional input. In **CycleGAN**, we directly use regular images as conditional inputs. In **WGAN+**, Wasserstein loss is used within our framework.

we have observed. We train the model using the Adam optimizer [92] with a learning rate of 0.0002. The batch size and the weight of identity loss are set to 1 and 0.5, respectively. We adjust the weights of the two cycle losses for the two generators to 20%. The model is trained for 30 epochs. For other settings, we strictly follow the practice of training a CycleGAN for the purpose of fair comparisons.

5.3.4 Results

To demonstrate the competence of our framework, we compare it with other methods, such as HED [214], and Photo-Sketching [117]. It is important to note that the HED method was not designed for sketch generation, and to our best knowledge there are very few works focusing on image to full-scene sketch generation. Therefore, we still add HED in our comparison, considering edge detection is one of the closest general image processing tasks to our image-to-sketch generation problem. We present the quantitative results of each method in Table 5.1, based on the Photo-Sketching dataset and the aforementioned metrics. As can be seen, our method always outperforms either Photo-Sketching or HED in terms of FID, Dice Score, Accuracy and Sensitivity. However, when a regular image is directly used as the input of the generator, that is, the **CycleGAN** only method, our framework without conditional input has inferior performance than the competitors in terms of all the four metrics except FID. It indicates the necessity of exploiting edge map as conditional input. But when the Canny edge detector is adopted to provide the conditional input, giving the **Canny+** approach, even though its performance is still competitive, but greatly falls behind our results. It can be observed that using different loss functions also have an impact on the results. The CycleGAN loss has demonstrated more robust and also better performance than the Wasserstein loss (**WGAN+**) for our framework.

We further compare the qualitative results by giving three sets of typical examples in Fig. 5.5, 5.6, and 5.7. We observe that the HED method [214] tends to generate too many

edge details, and the results are not like hand-drawn sketches. While the quality of the sketches generated by the Photo-Sketching method [117] is generally better, but they often miss a significant number of important feature lines, as well as critical visual cues. On the contrary, our results are much closer to hand-drawn sketches with necessary and proper level of details. Fig. 5.6 indicates that, compared with the WGAN loss (**WGAN+**), the CycleGAN loss is more helpful to robustly produce appealing results. In Fig. 5.7, **Canny+**) often generate inferior results than our HED-based approach. All of these further validate our best configurations for our proposed **Edge Map + GAN** image-to-sketch framework: **HED + CycleGAN**.

Differences from Sketch-RNN It deserves to notice that our method has different aims as those of Sketch-RNN [73]. As a sketch-to-sketch generation approach, Sketch-RNN focuses on constructing stroke-based drawings of objects, whereas our method concentrates on the image-to-sketch problem by generating a full-scene sketch from a natural image. That is, the inputs of the two methods are different. In Sketch-RNN, a common input is a hand-drawn sketch of a common object, while our method takes a full-scene color image as input. Considering of these, we do not compare our method and Sketch-RNN in this project.

5.3.5 Full-scene sketch generation

In this work, we aim to generate full-scene sketches, containing every important object in a given image. In fact, since our framework is based on a conditional input that covers every significant object in the scene, it is capable of generating the full-scene sketch. Sample results are illustrated in Fig. 5.8. As can be observed, our results well reflect the sketch of each apparent object in the given images.

5.4 Summary

We propose a flexible framework for image to full-scene sketch generation in this chapter. We demonstrate that different components can be exploited in this framework to achieve multiple levels of results. We investigate the impact of conditional input and demonstrate the necessity of edge map for appealing sketch generation from a regular image. We also analyze the distribution mapping problem in the context of sketch generation and demonstrate the suitability of CycleGAN for sketch generation. The effectiveness of the proposed framework

is validated through extensive experiments, and it is convenient to setup the framework to produce hand-drawn like sketches.

Like Berger et al. [26], we also have interest in generating sketches of diverse styles and also at different levels of abstraction. In addition, Augmented CycleGAN [15] is claimed to be able to get many-to-many mappings from CycleGAN instead of one-to-one, which is promising to further improve our algorithm, as well.

Chapter 6
CONCLUSIONS AND FUTURE WORK

In this chapter, we draw a conclusion and propose several future work. For the conclusions, we briefly present the main idea, results and contributions of our proposed algorithms. For the future work, we propose two new research directions to further extend the semantics-driven large-scale 3D scene model retrieval project.

6.1 Conclusions

Given a 2D scene sketch/image, 2D scene sketch/image-based 3D scene retrieval is to search for relevant 3D scene models from a 3D scene model dataset. The objective of my project is to provide a solution to deal with the challenges existing in this type of retrieval technique. The solution is composed of three components: (1) building a richly-annotated hierarchical 3D scene database; (2) proposing a semantic-based 3D scene retrieval framework, and (3) generating 2D scene sketches by utilizing adversarial networks.

(1) Literature review. I provided a review and critical evaluation on the most recent (i.e., within five recent years) and novel data-driven or semantics-driven 3D scene analysis and processing methods, as well as several involved 3D scene datasets. For each method, its advantage(s) and disadvantage(s) are discussed, after an overview and/or analysis of the approach.

(2) Benchmarks building and evaluation. We organized two Shape Retrieval Contest (SHREC) tracks on 2D scene sketch-based and image-based 3D scene model retrieval in 2018 and 2019, respectively. In 2018 tracks, 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. In 2019 tracks, we built a much larger extended benchmark for each type of retrieval which has thirty (30) classes. We perform a comprehensive comparison of all the participating retrieval methods by evaluating them on the two benchmarks. We also developed a deep learning-based baseline approach for the benchmarks in the two 2019 tracks.

(3) Scene semantics learning and related retrieval algorithm. It is dedicated to the challenge of semantic gap between 2D scene sketches/images and the 3D scene

models to further increase the retrieval accuracy. We developed semantic-tree and gradient descent-based approaches which incorporate the semantic relationships of the objects into the scene semantics learning process. The semantic information contains objects' occurrence, co-occurrence and spatial relations information.

(4) Image-to-scene sketch translation. We proposed a flexible framework for image to full-scene sketch generation. We demonstrate that different components can be exploited in this framework to achieve multiple levels of results. We investigate the impact of conditional input and demonstrate the necessity of edge map for appealing sketch generation from a regular image. The effectiveness of the proposed framework is validated through extensive experiments, and it is convenient to setup the framework to produce hand-drawn like sketches.

6.2 Future Work

6.2.1 Research Direction 1: Automatic Expansion of the Semantic Tree

We plan to classify the collected 2D/3D scenes either directly based on their available categorical label information or by developing an automatic deep learning-based scene classification algorithm for unlabeled scenes. The deep learning-based scene classification algorithm will also be used for automatic expansion of our scene semantic tree.

To allow automatic expansion of our semantic tree, we plan to develop a Semantics-driven Deep Embedding model (SDE) that maps all the 2D/3D scenes into the same latent (feature) space. When unlabeled new scenes join in, we can compute the similarity of their embeddings to the existing scenes in the tree, to identify an appropriate location. We can also combine with hierarchical clustering or semi-supervised clustering algorithms to create new branches for the tree. Our SDE model is designed as follows. For each particular type of scenes, e.g., 2D sketches, 2D images and 3D scene model views, we introduce a variational auto-encoder (VAE) [94] to learn their embedding. The VAE consists of an encoder that maps the original data to an embedding vector in a latent space, and a decoder that reconstructs the original data from the embedding. Both the encoder and decoder are DNNs, and the learning objective is to minimize the reconstruction error, which can be interpreted as maximizing a variational lower bound in the VAE framework. To ensure that the VAEs for each type of scenes map to the same latent space, we enforce their embeddings to have the same dimension. Furthermore, to guide the learning of the embeddings with the semantic information, we encourage that their embeddings are close to each other if

they are near in the semantic tree. Given the set of scenes in the tree $\{S_1, \ldots, S_N\}$ and their corresponding nodes in the tree $\{n_1, \ldots, n_N\}$, the loss function of our SDE model is,

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}_{\mathrm{VAE}_{c_i}}(S_i) + \lambda \cdot \sum_{1 \le j,k \le N, j \ne k} \frac{\|\mathbf{u}_j - \mathbf{u}_k\|}{\mathrm{tree\_dist}(n_j, n_k)}$$

where c_i is the type of the scene S_i, $\mathcal{L}_{\mathrm{VAE}_{c_i}}(\cdot)$ is the loss function of the VAE for the scenes of type c_i, λ is the hyper-parameter that controls the strength of the semantic part, and $\mathbf{u}_j$ and $\mathbf{u}_k$ are the embeddings of the scenes S_j and S_k, respectively.

Data Collection

We plan to refer to the following 2D/3D datasets for data collection methods as well as direct reuse of their data for 3D semantic tree building: Places [237], COCO [122], SUNCG [179], SUN RGB-D [177] and SUN [210], ObjectNet3D [208], ScanNet [47], ImageNet [51], and ShapeNet [38]. Augmenting our dataset with existing datasets is much more feasible than creating one from scratch. For example, COCO [122] and Places2 [237], offer excellent semantic labels for 2D scene images. ObjectNet3D [47] and SUN RGB-D [177] provide thousands of 3D scenes comprised of diverse 3D models across numerous classes. The annotation and segmentation toolkits included in these datasets will allow us to extend existing classes and create our own dataset as well. Since we have a lot of images available online and related datasets as well, collecting 2D scene images will be not an issue. Let's consider the rest two cases: 3D scene models and 2D scene sketches data collection.

(1) 3D Scene Models Data Collection. As the main data sources, we will develop web crawlers to automatically download free 3D scenes from popular online public 3D repositories such as 3D Warehouse [189] which hosts more than 4M free 3D models, as well as GrabCAD [1] (2.84M), Sketchfab [2] (1.5M), and Yobi3D [3] (1.0M). All of the above datasets together provide scene models from a diverse number of categories, like generic, CAD, architecture, watertight and RGB-D types, as well as 3D printer models.

(2) 2D Scene Sketches Data Collection —SceneGAN: Automatic Scene Sketch Generation from Images by Using a Generative Adversarial Network (GAN) Model. As presented in Chapter 4, we have proposed a full-scene image-to-sketch synthesis algorithm with CycleGAN [242] using holistically-nested edge detection (HED) [214] maps. We plan to use the scene sketch data generated based on this approach to further improve our

model for 2D scene sketch-based 3D scene retrieval presented in Chapter 4, as well as to further enlarge our curated scene sketch-based 3D scene retrieval benchmark to make it a real large-scale one to further promote this research area.

6.2.2 Research Direction 2: Developing An Adaptive Approach Supporting Processing Different Kinds of Scene Data

Building an adaptive approach for different kinds of scene data is significant. Besides building a benchmark with various types of 2D and 3D scene data, we can propose a new machine learning model which is versatile enough to handle different modalities of scene data. This is challenging but promising since it has great potentials in related practical application scenarios which typically involve big data and cloud computing, and those data of such application scenarios may vary either in format or style.

(1) Scene Data Conversion. The first way is to convert all the data into the same type. For example, some sketches collected from online sources are too concise and have very little content, while other sketches contain more details. As we all know, the more detailed information that the sketches contain, the more accurate performance of the neural network training and prediction will have. Therefore, to improve retrieval performance, we will develop a method that can automatically enrich those concise sketches so as to make them contain more content information (e.g., more objects) like other sketches, such as scene sketch completion techniques.

(2) Adaptive Machine Learning Model. The second way is to train an adaptive machine learning model which can fit different types of scene data. This kind of model can deal with many types of scene data. For example, no matter whether these scene data are detailed or not, realistic or iconic. In order to achieve this goal, we will jointly use multiple neural networks in our approach, and train those networks on various types of large scale scene data. Meanwhile, to further improve the retrieval performance, we will also incorporate the scene semantic relatedness information of different types of scene data into the final loss function computation of the model.

LIST OF PUBLICATIONS

Refereed International Journals

- **Juefei Yuan**, Bo Li, et al. Semantic-driven Large-scale 3D Scene Retrieval, Under preparation, to be submitted, August, 2021.
- **Juefei Yuan**, Hameed Abdul Rashid, Bo Li. A Survey of Recent 3D Scene Analysis and Processing Methods, Multimedia Tools and Applications, Volume 80, pages 19491-19511, February 21, 2021.
- **Juefei Yuan**, Hameed Abdul Rashid, Bo Li, et al. A Comparison of Methods for 3D Scene Shape Retrieval, Computer Vision and Image Understanding, Volume 201, 103070, December 2020.
- Bo Li, **Juefei Yuan**, et al. 3D Sketching for 3D Object Retrieval, Multimedia Tools and Applications, Volume 80, pages 9569-9595, November 11, 2020.

Peer Reviewed Conferences

- **Juefei Yuan**, Tianyang Wang, Shandian Zhe, Yijuan Lu, Bo Li. Semantic Tree-Based 3D Scene Model Recognition. The IEEE 3rd International Conference on Multimedia Information Processing and Retrieval MIPR' 20 August 6-8, Shenzhen, China (Invited Paper), 2020, 1-6.
- Daniel McGonigle, Tianyang Wang, **Juefei Yuan**, Kai He, Bo Li. I2S2: Image-to-Scene Sketch Translation Using Conditional Input and Adversarial Networks. 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), pages 1-6, 2020.
- **Juefei Yuan**, Hameed Abdul Rashid, Bo Li, et al. SHREC'19 Track: Extended 2D Scene Sketch-Based 3D Scene Retrieval, Eurographics Workshop on 3D Object Retrieval 2019 (3DOR 2019), pages 33-39, May 5-6, 2019.
- Hameed Abdul Rashid, **Juefei Yuan**, Bo Li, et al. SHREC'19 Track: Extended 2D Scene Image-Based 3D Scene Retrieval, Eurographics Workshop on 3D Object Retrieval 2019 (3DOR 2019), pages 41-48, May 5-6, 2019.

- **Juefei Yuan**, Hameed Abdul Rashid, Bo Li, Yijuan Lu. Sketch/Image Based 3D Scene Retrieval: Benchmark, Algorithm , Evaluation The IEEE 2nd International Conference on Multimedia Information Processing and Retrieval MIPR'19, March 28-30, USA (Invited Paper), January 2019, 264-269.

- **Juefei Yuan**, Bo Li, et al. SHREC'18 Track: 2D Scene Sketch-Based 3D Scene Retrieval. Eurographics Workshop on 3D Object Retrieval 2018 (3DOR 2018), April 16, pages 29-36, 2018.

- Hameed Abdul Rashid, **Juefei Yuan**, Bo Li, et al. SHREC'18 Track: 2D Scene Image-Based 3D Scene Retrieval. Eurographics Workshop on 3D Object Retrieval 2018 (3DOR 2018), pages 37-44, April 16, 2018.

BIBLIOGRAPHY

[1] Grabcad. `http://grabcad.com/library`, 2018.

[2] Sketchfab. `http://sketchfab.com/`, 2018.

[3] Yobi3d. `http://www.yobi3d.com/`, 2018.

[4] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, and Yijuan Lu. SHREC'18 2D Scene Image-Based 3D Scene Retrieval Track Website. http://orca.st.usm.edu/~bli/SceneIBR2018/, 2018.

[5] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, and Yijuan Lu. SHREC'19 Extended 2D Scene Image-Based 3D Scene Retrieval Track Website. http://orca.st.usm.edu/~bli/SceneIBR2019/, 2019.

[6] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. SHREC'18: 2D image-based 3D scene retrieval. In *Eurographics Workshop on 3D Object Retrieval, 3DOR 2018, 16 April 2018, Delft, The Netherlands.*, pages 37–44, 2018.

[7] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh-Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. SHREC'18 track: 2D scene image-based 3D scene retrieval. In *3DOR*, pages 1–8, 2018.

[8] Hameed Abdul-Rashid, Juefei Yuan, Bo Li, Yijuan Lu, Tobias Schreck, Ngoc-Minh Bui, Trong-Le Do, Mike Holenderski, Dmitri Jarnikov, Tu-Khiem Le, Vlado Menkovski, Khac-Tuan Nguyen, Thanh-An Nguyen, Vinh-Tiep Nguyen, Van-Tu Ninh, Luis A. Pérez Rey, Minh-Triet Tran, and Tianyang Wang. 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*, pages 41–48, 2019.

[9] Acuity Laser contributors. Acuity laser. https://www.acuitylaser.com/, 2020.

[10] J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, June 2018.

[11] Adobe Inc. Adobe stock, 2020.

[12] Dror Aiger, Brett Allen, and Aleksey Golovinskiy. Large-scale 3D scene classification with multi-view volumetric CNN. *CoRR*, abs/1712.09216, 2017.

[13] R. Akase and Y. Okada. Automatic 3D furniture layout based on interactive evolutionary computation. In *2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*, pages 726–731, July 2013.

[14] Ryuya Akase and Yoshihiro Okada. Web-based multiuser 3D room layout system using interactive evolutionary computation with conjoint analysis. In *Proceedings of the 7th International Symposium on Visual Information Communication and Interaction*, VINCI '14, pages 178:178–178:187. ACM, 2014.

[15] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron C. Courville. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 195–204, 2018.

[16] Relja Arandjelovic and Andrew Zisserman. 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*, pages 2911–2918, 2012.

[17] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 214–223, 2017.

[18] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera, 2019.

[19] Iro Armeni, Sasha Sax, Amir Roshan Zamir, and Silvio Savarese. Joint 3D-3D-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017.

[20] Y. Alp Aslandogan, Chuck Thier, Clement T. Yu, Jon Zou, and Naphtali Rishe. Using semantic contents and WordNet in image retrieval. In *ACM SIGIR '97*, pages 286–295, 1997.

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

[22] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. GIFT: A real-time and scalable 3D shape search engine. In *CVPR*, pages 5023–5032. IEEE, 2016.

[23] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[24] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Visualizing and understanding GANs. In *Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*, 2019.

[25] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3D scene flow estimation in autonomous driving scenarios? In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[26] Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth J. Carter, and Jessica K. Hodgins. Style and abstraction in portrait sketching. *ACM Trans. Graph.*, 32(4):55:1–55:12, 2013.

[27] Maros Blaha, Christoph Vogel, Audrey Richard, Jan D. Wegner, Thomas Pock, and Konrad Schindler. Large-scale semantic 3D reconstruction: An adaptive multi-resolution model for multi-class volumetric labeling. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] Cédric Bobenrieth, Hyewon Seo, Arash Habibi, and Frédéric Cordier. Indoor scene reconstruction from a sparse set of 3D shots. In *Proceedings of the Computer Graphics International Conference*, CGI '17, pages 27:1–27:5. ACM, 2017.

[29] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[30] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *CoRR*, abs/1612.03716, 2016.

[31] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. 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*, pages 1209–1218, 2018.

[32] Ali Caglayan, Nevrez Imamoglu, Ahmet Burak Can, and Ryosuke Nakamura. When cnns meet random rnns: Towards multi-level analysis for rgb-d object and scene recognition, 2020.

[33] John F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[34] Doersch Carl. Tutorial on variational autoencoders, 2016.

[35] João Carreira and Cristian Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1312–1328, 2012.

[36] Rich Caruana. Multitask learning. *Machine Learning*, 29:41–75, 1997.

[37] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[38] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *CoRR*, abs/1512.03012, 2015.

[39] Rama Chellappa and Anil Jain. Markov random fields. theory and application. *Boston: Academic Press, 1993, edited by Chellappa, Rama; Jain, Anil*, 1993.

[40] Bao Xin Chen, Raghavender Sahdev, Dekun Wu, Xing Zhao, Manos Papagelis, and John Tsotsos. Scene classification in indoor environments for robots using context based word embeddings. 05 2018.

[41] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9416–9425, 2018.

[42] Wongun Choi, Yu-Wei Chao andc Caroline Pantofaru, and Silvio Savarese. Understanding indoor scenes using 3D geometric phrases. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 33–40, 2013.

[43] Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas A. Funkhouser, and Szymon Rusinkiewicz. Where do people draw lines? *Commun. ACM*, 55(1):107–115, 2012.

[44] Forrester Cole, Kevin Sanik, Douglas DeCarlo, Adam Finkelstein, Thomas A. Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? *ACM Trans. Graph.*, 28(3):28, 2009.

[45] Paul G. Constantine and Eric T. Phipps. A Lanczos method for approximating composite functions. *Applied Mathematics and Computation*, 218(24):11751–11762, 2012.

[46] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[47] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2432–2443, 2017.

[48] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3D scans. *CoRR*, abs/1712.10215, 2017.

[49] Douglas DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Trans. Graph.*, 22(3):848–855, 2003.

[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[51] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[52] Leon Derczynski. Complementarity, f-score, and NLP evaluation. In *LREC*, 2016.

[53] Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, July 1945.

[54] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *CoRR*, abs/1907.02544, 2019.

[55] Siyan Dong, Kai Xu, Qiang Zhou, Andrea Tagliasacchi, Shiqing Xin, Matthias Nießner, and Baoquan Chen. Multi-robot collaborative dense scene reconstruction. *ACM Transactions on Graphics*, 38(4):Article 84, 2019.

[56] Hossein Ebrahimnezhad and Hassan Ghassemian. Robust motion from space curves and 3d reconstruction from multiviews using perpendicular double stereo rigs. *Image and Vision Computing*, 26(10):1397 – 1420, 2008.

[57] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH)*, 31(4):44:1–44:10, 2012.

[58] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[59] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, September 2010.

[60] Matthew Fisher, Manolis Savva, and Pat Hanrahan. Characterizing structural relationships in scenes using graph kernels. *ACM transactions on graphics (TOG)*, 30(4):34, 2011.

[61] Matthew Fisher, Manolis Savva, Yangyan Li, Pat Hanrahan, and Matthias Niessner. Activity-centric scene synthesis for functional 3D scene modeling. *ACM Trans. Graph.*, 34(6):179:1–179:13, October 2015.

[62] Flickr. Flickr website. `https://www.flickr.com/`, 2018.

[63] John Flynn, Michael Broxton, Paul E. Debevec, Matthew DuVall, Graham Fyffe, Ryan S. Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. *CoRR*, abs/1906.07316, 2019.

[64] Takahiko Furuya and Ryutarou Ohbuchi. Learning part-in-whole relation of 3D shapes for part-based 3D model retrieval. *Computer Vision and Image Understanding*, 166:102–114, 2018.

[65] Boyong Gao, Herong Zheng, and Sanyuan Zhang. An overview of semantics processing in content-based 3D model retrieval. In *International Conference on Artificial Intelligence and Computational Intelligence (AICI '09)*, volume 2, pages 54–59, Nov 2009.

[66] Chengying Gao, Qi Liu, Qi Xu, Jianzhuang Liu, Limin Wang, and Changqing Zou. Sketchy-coco: Image generation from freehand scene sketches. *CoRR*, abs/2003.02683, 2020.

[67] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[68] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[69] Google. Google Images. `https://www.google.com/imghp?hl=EN`, 2018.

[70] Stéphane Grabli, Emmanuel Turquin, Frédo Durand, and François X. Sillion. Programmable rendering of line drawing from 3d scenes. *ACM Trans. Graph.*, 29(2):18:1–18:20, 2010.

[71] Hewei Guo and Fusheng Guo. Urban scene 3D reconstruction optimization leveraged by line information. In *Proceedings of the 2Nd International Conference on Innovation in Artificial Intelligence*, ICIAI '18, pages 92–96. ACM, 2018.

[72] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[73] David Ha and Douglas Eck. A neural representation of sketch drawings. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[74] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding realworld indoor scenes with synthetic data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4077–4085, 2016.

[75] James Hays and Alexei A. Efros. IM2GPS: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[76] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016.

[78] Xinwei He, Yang Zhou, Zhichao Zhou, Song Bai, and Xiang Bai. Triplet center loss for multi-view 3D object retrieval. In *CVPR*, 2018.

[79] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6626–6637, 2017.

[80] Nguyen Vu Hoàng, Valérie Gouet-Brunet, Marta Rukoz, and Maude Manouvrier. Embedding spatial information into image content description for scene retrieval. *Pattern Recognition*, 43(9):3013–3024, 2010.

[81] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. SceneNN: A scene meshes dataset with annotations. In *3DV*, pages 92–101. IEEE Computer Society, 2016.

[82] Shengyu Huang, Mikhail Usvyatsov, and Konrad Schindler. Indoor scene recognition in 3d, 2020.

[83] Victoria Interrante, Henry Fuchs, and Stephen M. Pizer. Enhancing transparent skin surfaces with ridge and valley lines. In *IEEE Visualization '95, Proceedings, Atlanta, Georgia, USA, October 29 - November 3, 1995.*, pages 52–59, 1995.

[84] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5967–5976, 2017.

[85] Lesley Istead and Craig S. Kaplan. Stylized stereoscopic 3D line drawings from 3D images. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, Expressive 2018, Victoria, BC, Canada, August 17-19, 2018*, pages 20:1–20:2, 2018.

[86] Tilke Judd, Frédo Durand, and Edward H. Adelson. Apparent ridges for line drawing. *ACM Trans. Graph.*, 26(3):19, 2007.

[87] Grigorios Kalliatakis. Keras-vgg16-places365. `https://github.com/GKalliatakis/Keras-VGG16-places365`, 2017.

[88] Henry Kang, Seungyong Lee, and Charles K. Chui. Coherent line drawing. In *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering 2007, San Diego, California, USA, August 4-5, 2007*, pages 43–50, 2007.

[89] Henry Kang, Seungyong Lee, and Charles K. Chui. Flow-based image abstraction. *IEEE Trans. Vis. Comput. Graph.*, 15(1):62–76, 2009.

[90] N. Kanopoulos, N. Vasanthavada, and R. L. Baker. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-State Circuits*, 23(2):358–367, April 1988.

[91] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4401–4410, 2019.

[92] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[93] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[94] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[95] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[96] Michael Kolomenkin, Ilan Shimshoni, and Ayellet Tal. Demarcating curves for shape illustration. *ACM Trans. Graph.*, 27(5):157, 2008.

[97] Tao Ku, Remco C. Veltkamp, Bas Boom, David Duque-Arias, Santiago Velasco-Forero, Jean-Emmanuel Deschaud, Francois Goulette, Beatriz Marcotegui, Sebastián Ortega, Agustín Trujillo, José Pablo Suárez, José Miguel Santana, Cristian Ramírez, Kiran Akadas, and Shankar Gangisetty. Shrec 2020: 3d point cloud semantic segmentation for street scenes. *Computers & Graphics*, 93:13 – 24, 2020.

[98] Barry M. Kudrowitz, Paula Te, and David R. Wallace. The influence of sketch quality on perception of product-idea creativity. *AI EDAM*, 26(3):267–279, 2012.

[99] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3581–3590, 2019.

[100] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 739–746, Sep. 2009.

[101] Michael Lesk. 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*, pages 24–26, 1986.

[102] B. Li, Y. Lu, Afzal Godil, Tobias Schreck, Masaki Aono, Henry Johan, Jose M. Saavedra, and S. Tashiro. SHREC'13 track: Large scale sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 89–96, 2013.

[103] Bo Li and Henry Johan. View context based 2d sketch-3d model alignment. In *IEEE Workshop on Applications of Computer Vision (WACV 2011), 5-7 January 2011, Kona, HI, USA*, pages 45–50, 2011.

[104] Bo Li and Henry Johan. 3D model retrieval using hybrid features and class information. *Multimedia Tools Appl.*, 62(3):821–846, 2013.

[105] Bo Li and Henry Johan. Sketch-based 3D model retrieval by incorporating 2D-3D alignment. *Multimedia Tools Appl.*, 65(3):363–385, 2013.

[106] Bo Li, Y. Lu, C. Li, Afzal Godil, Tobias Schreck, Masaki Aono, Martin Burtscher, Hongbo Fu, Takahiko Furuya, Henry Johan, Jianzhuang Liu, Ryutarou Ohbuchi, Atsushi Tatsuma, and Changqing Zou. SHREC'14 track: Extended large scale sketch-based 3D shape retrieval. In *3DOR*, pages 121–130, 2014.

[107] Bo Li, Yijuan Lu, Fuqing Duan, Shuilong Dong, Yachun Fan, Lu Qian, Hamid Laga, Haisheng Li, Yuxiang Li, Peng Liu, Maks Ovsjanikov, Hedi Tabia, Yuxiang Ye, Huanpu Yin, and Ziyu Xue. SHREC'16: 3D sketch-based 3D shape retrieval. In *3DOR 2016*, pages 47–54, 2016.

[108] Bo Li, Yijuan Lu, and Ribel Fares. Semantic sketch-based 3D model retrieval. In *2013 IEEE International Conference on Multimedia and Expo, San Jose, CA, USA, July 15-19, 2013*, pages 1–4, 2013.

[109] Bo Li, Yijuan Lu, Afzal Godil, Tobias Schreck, Benjamin Bustos, Alfredo Ferreira, Takahiko Furuya, Manuel J. Fonseca, Henry Johan, Takahiro Matsuda, Ryutarou Ohbuchi, Pedro B. Pascoal, and Jose M. Saavedra. A comparison of methods for sketch-based 3D shape retrieval. *CVIU*, 119:57–80, 2014.

[110] Bo Li, Yijuan Lu, and Henry Johan. Sketch-based 3D model retrieval by viewpoint entropy-based adaptive view clustering. In *Eurographics Workshop on 3D Object Retrieval, Girona, Spain, 2013. Proceedings*, pages 49–56, 2013.

[111] Bo Li, Yijuan Lu, Henry Johan, and Ribel Fares. Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information. *Multimedia Tools Appl.*, 76(24):26603–26631, 2017.

[112] Bo Li, Yijuan Lu, Chunyuan Li, Afzal Godil, Tobias Schreck, Masaki Aono, Martin Burtscher, Qiang Chen, Nihad Karim Chowdhury, Bin Fang, Hongbo Fu, Takahiko Furuya, Haisheng Li, Jianzhuang Liu, Henry Johan, Ryuichi Kosaka, Hitoshi Koyanagi, Ryutarou Ohbuchi, Atsushi Tatsuma, Yajuan Wan, Chaoli Zhang, and Changqing Zou. A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. *CVIU*, 131:1–27, 2015.

[113] Bo Li, Yijuan Lu, and Jian Shen. A semantic tree-based approach for sketch-based 3D model retrieval. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 3880–3885, 2016.

[114] Bo Li, Tobias Schreck, Afzal Godil, Marc Alexa, Tamy Boubekeur, Benjamin Bustos, J. Chen, Mathias Eitz, Takahiko Furuya, Kristian Hildebrand, S. Huang, Henry Johan, Arjan Kuijper, Ryutarou Ohbuchi, Ronald Richter, Jose M. Saavedra, Maximilian Scherer, Tomohiro Yanagimachi, Gang-Joon Yoon, and Sang Min Yoon. SHREC'12 track: Sketch-based 3D shape retrieval. In *Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 109–118, 2012.

[115] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion, 2020.

[116] Manyi Li, Akshay Gadi Patil, Kai Xu, Siddhartha Chaudhuri, Owais Khan, Ariel Shamir, Changhe Tu, Baoquan Chen, Daniel Cohen-Or, and Hao Zhang. Grains: Generative recursive autoencoders for indoor scenes. *ACM Trans. Graph.*, 38(2):12:1–12:16, February 2019.

[117] Mengtian Li, Zhe L. Lin, Radomír Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Waikoloa Village, HI, USA, January 7-11, 2019*, pages 1403–1412, 2019.

[118] Yi Li, Yi-Zhe Song, Timothy M. Hospedales, and Shaogang Gong. Free-hand sketch synthesis with deformable stroke models. *International Journal of Computer Vision*, 122(1):169–190, 2017.

[119] Joseph J. Lim, C. Lawrence Zitnick, and Piotr Dollár. Sketch Tokens: A learned mid-level representation for contour and object detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3158–3165, 2013.

[120] Frederico A. Limberger, Richard C. Wilson, Masaki Aono, Nicolas Audebert, Alexandre Boulch, Benjamín Bustos, Andrea Giachetti, Afzal Godil, Bertrand Le Saux, Bo Li, Yijuan Lu, H. D. Nguyen, V.-T. Nguyen, Viet-Khoi Pham, Ivan Sipiran, Atsushi Tatsuma, M.-T. Tran, and Santiago Velasco-Forero. SHREC'17: Point-cloud shape retrieval of non-rigid toys. In *3DOR*, 2017.

[121] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3D object detection with RGBD cameras. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 1417–1424. IEEE, 2013.

[122] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.

[123] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[124] Fang Liu, Xiaoming Deng, Yu-Kun Lai, Yong-Jin Liu, Cuixia Ma, and Hongan Wang. SketchGAN: Joint sketch completion and recognition with generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5830–5839, 2019.

[125] Nian Liu and Junwei Han. DHSNet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016.

[126] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S. Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, pages 2200–2207, 2013.

[127] Charles Loop. *Smooth Subdivision Surfaces Based on Triangles*. January 1987.

[128] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using contextual GAN. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, pages 213–228, 2018.

[129] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 698–707, 2018.

[130] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas Guibas, and Hao Zhang. Language-driven synthesis of 3D scenes from scene databases. *ACM Trans. Graph.*, 37(6):212:1–212:16, December 2018.

[131] Zhixin Ma, Xukun Shen, and Chong Cao. A hybrid CRF framework for semantic 3D reconstruction. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST '17, pages 14:1–14:4, New York, NY, USA, 2017. ACM.

[132] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2813–2821, 2017.

[133] Marcin Marszalek and Cordelia Schmid. Semantic hierarchies for visual object recognition. In *CVPR*, 2007.

[134] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 2*, pages 416–425, 2001.

[135] Ravish Mehra, Qingnan Zhou, Jeremy Long, Alla Sheffer, Amy Ashurst Gooch, and Niloy J. Mitra. Abstraction of man-made shapes. *ACM Trans. Graph.*, 28(5):137, 2009.

[136] Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. *ACM Transactions on Graphics (TOG)*, 30(4):87, 2011.

[137] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[138] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[139] Ondrej Miksik, Vibhav Vineet, Morten Lidegaard, Ram Prasaath, Matthias Niessner, Stuart Golodetz, Stephen L. Hicks, Patrick Pérez, Shahram Izadi, and Philip H.S. Torr. The semantic paintbrush: Interactive 3D mapping and recognition in large outdoor spaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3317–3326. ACM, 2015.

[140] George A. Miller. WordNet: A lexical database for English. *Commun. ACM*, 38(11):39–41, 1995.

[141] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[142] Nathan Moroney, Mark D. Fairchild, Robert W. G. Hunt, Changjun Li, M. Ronnier Luo, and Todd Newman. The CIECAM02 color appearance model. In *The Tenth Color Imaging*

Conference: Color Science and Engineering Systems, Technologies, Applications, CIC 2002, Scottsdale, Arizona, USA, November 12-15, 2002, pages 23–27, 2002.

[143] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. Learning deep sketch abstraction. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8014–8023. IEEE Computer Society, 2018.

[144] Eric Müller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pages 575–592, 2018.

[145] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *ECCV*, 2020.

[146] Muzammal Naseer, Salman Hameed Khan, and Fatih Porikli. Indoor scene understanding in 2.5/3D: A survey. *CoRR*, abs/1803.03352, 2018.

[147] Vinh-Tiep Nguyen, Thanh Duc Ngo, Minh-Triet Tran, Duy-Dinh Le, and Duc Anh Duong. A combination of spatial pyramid and inverted index for large-scale image retrieval. *IJMDEM*, 6(2):37–51, 2015.

[148] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[149] Genevieve Patterson and James Hays. 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*, pages 2751–2758, 2012.

[150] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2):59–81, 2014.

[151] Eric Penner and Li Zhang. Soft 3D reconstruction for view synthesis. *ACM Trans. Graph.*, 36(6):235:1–235:11, November 2017.

[152] Judith M. S. Prewitt. Object enhancement and extraction. In *Picture processing and Psychopictorics*. Academic Press, 1970.

[153] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85, 2017.

[154] Muthukrishnan R. Edge detection techniques for image segmentation. *International journal of computer science and information technology*, 3:259–267, 12 2011.

[155] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[156] Mohammad Ramezani and H. Ebrahimnezhad. A novel 3d object categorization and retrieval system using geometric features. 2011.

[157] José Carlos Rangel, Miguel Cazorla, Ismael García-Varea, Jesus Martínez-Gómez, Élisa Fromont, and Marc Sebban. Scene classification based on semantic labeling. *Advanced Robotics*, 30(11-12):758–769, 2016.

[158] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[159] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6517–6525, 2017.

[160] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[161] Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, and Steve Seitz. Soccer on your tabletop. In *CVPR*, 2018.

[162] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[163] Xiaofeng Ren and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 593–601, 2012.

[164] Renault. Renault SYMBOIZ Concept. `http://www.renault.co.uk/vehicles/concept-cars/symbioz-concept.html`.

[165] Daniel Ritchie, Kai Wang, and Yu-An Lin. Fast and flexible indoor scene synthesis via deep convolutional generative models. *CoRR*, abs/1811.12463, 2018.

[166] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2226–2234, 2016.

[167] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The Sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4):119:1–119:12, 2016.

[168] Nikolay Savinov, Christian Häne, Lubor Ladicky, and Marc Pollefeys. Semantic 3D reconstruction with continuous regularization and ray potentials using a visibility consistency constraint. *CoRR*, abs/1604.02885, 2016.

[169] Manolis Savva, Angel X. Chang, and Maneesh Agrawala. Scenesuggest: Context-driven 3D scene design. *CoRR*, abs/1703.00061, 2017.

[170] Yifei Shi, Angel Xuan Chang, Zhelun Wu, Manolis Savva, and Kai Xu. Hierarchy denoising recursive autoencoders for 3D scene layout prediction. *CoRR*, abs/1903.03757, 2019.

[171] Philip Shilane, Patrick Min, Michael M. Kazhdan, and Thomas A. Funkhouser. The Princeton shape benchmark. In *SMI*, pages 167–178, 2004.

[172] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, pages 746–760, Berlin, Heidelberg, 2012. Springer-Verlag.

[173] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 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*, pages 746–760, 2012.

[174] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[175] Joan Gay Snodgrass and Mary Vanderwart. A standardized set of 260 pictures: Norms for name agreement, image agreement, familiarity, and visual complexity. *JOURNAL OF EXPERIMENTAL PSYCHOLOGY: HUMAN LEARNING AND MEMORY*, 6(2):174–215, 1980.

[176] Kihyuk Sohn, Sifei Liu, Guangyu Zhong, Xiang Yu, Ming-Hsuan Yang, and Manmohan Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. *CoRR*, abs/1708.02191, 2017.

[177] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 567–576, 2015.

[178] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. *CoRR*, abs/1611.08974, 2016.

[179] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 190–198. IEEE Computer Society, 2017.

[180] Gaurav Sood. *clarifai: R Client for the Clarifai API*, 2015. R package version 0.2.

[181] T.J. Sørensen. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*, volume 5 of *Biologiske skrifter*. 1948.

[182] D. Steinhauser, O. Ruepp, and D. Burschka. Motion segmentation and scene classification from 3D lidar data. In *2008 IEEE Intelligent Vehicles Symposium*, pages 398–403, June 2008.

[183] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[184] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *ICCV*, pages 945–953, 2015.

[185] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[186] Inside the Magic. 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, 2019.

[187] Linus Tech Tips. Driving a multi-million dollar autonomous car. `http://www.youtube.com/watch?v=vlIJfV1u2hM&feature=youtu.be`.

[188] Anne-Marie Tousch, Stéphane Herbin, and Jean-Yves Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, 2012.

[189] Inc Trimble. 3D Warehouse. http://3dwarehouse.sketchup.com/?hl=en, 2018.

[190] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017.

[191] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, abs/1908.00463, 2019.

[192] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray, S. Izadi, P. Pérez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 75–82, May 2015.

[193] Vibhav Vineet, Jonathan Warrell, and Philip H. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. *Int. J. Comput. Vision*, 110(3):290–307, December 2014.

[194] Nam N. Vo, Nathan Jacobs, and James Hays. Revisiting IM2GPS in the deep learning era. *CoRR*, abs/1705.04838, 2017.

[195] Krzysztof Walczak and Jakub Flotyński. Semantic query-based generation of customized 3D scenes. In *Proceedings of the 20th International Conference on 3D Web Technology*, Web3D '15, pages 123–131. ACM, 2015.

[196] Limin Wang, Sheng Guo, Weilin Huang, Yuanjun Xiong, and Yu Qiao. Knowledge guided disambiguation for large-scale scene classification with multi-resolution cnns. *CoRR*, abs/1610.01119, 2016.

[197] Peng-Shuai Wang, Yang Liu, and Xin Tong. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion, 2020.

[198] Xinying Wang, Tianyang Lv, Shengsheng Wang, and Zhengxuan Wang. An Ontology and SWRL based 3D model retrieval system. In *AIRS*, pages 335–344, 2008.

[199] Diana Werner, Ayoub Al-Hamadi, and Philipp Werner. Truncated signed distance function: Experiments on voxel size. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, pages 357–364. Springer International Publishing, 2014.

[200] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet - photo geolocation with convolutional neural networks. *CoRR*, abs/1602.05314, 2016.

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

[202] Wikipedia contributors. Jaccard — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Jaccard_index, 2020.

[203] Wikipedia contributors. Leap motion — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Leap_Motion, 2020.

[204] Wikipedia contributors. Lidar — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Lidar, 2020.

[205] Wikipedia contributors. Semantic similarity — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Semantic_similarity, 2020.

[206] Holger Winnemöller. XDoG: advanced image stylization with extended Difference-of-Gaussians. In *Proceedings of the 9th International Symposium on Non-Photorealistic Animation and Rendering 2009, Vancouver, BC, Canada, August 5-7, 2011*, pages 147–156, 2011.

[207] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C. Olsen. XDoG: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):740–753, 2012.

[208] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. ObjectNet3D: A large scale database for 3D object recognition. In *ECCV*, pages 160–176, 2016.

[209] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Bongsoo Choy, Hao Su, Roozbeh Mottaghi, Leonidas J. Guibas, and Silvio Savarese. ObjectNet3D: A large scale database for 3D object recognition. In *ECCV (8)*, volume 9912 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2016.

[210] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. SUN database: exploring a large collection of scene categories. *IJCV*, 119(1):3–22, 2016.

[211] Jianxiong Xiao, Krista A. Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *Int. J. Comput. Vision*, 119(1):3–22, August 2016.

[212] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492. IEEE Computer Society, 2010.

[213] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*, pages 1625–1632, 2013.

[214] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, 125(1-3):3–18, 2017.

[215] Xuexiang Xie, Ying He, Feng Tian, Hock Soon Seah, Xianfeng Gu, and Hong Qin. An effective illustrative visualization framework based on Photic Extremum Lines (PELs). *IEEE Trans. Vis. Comput. Graph.*, 13(6):1328–1335, 2007.

[216] Kai Xu, Vladimir G Kim, Qixing Huang, Niloy Mitra, and Evangelos Kalogerakis. Data-driven shape analysis and processing. In *SIGGRAPH ASIA 2016 Courses*, page 4. ACM, 2016.

[217] Kai Xu, Lintao Zheng, Zihao Yan, Guohang Yan, Eugene Zhang, Matthias Niessner, Oliver Deussen, Daniel Cohen-Or, and Hui Huang. Autonomous reconstruction of unknown indoor scenes guided by time-varying tensor fields. *ACM Trans. Graph.*, 36(6):202:1–202:15, November 2017.

[218] Y. Ye, B. Li, and Y. Lu. 3D sketch-based 3D model retrieval with convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2936–2941, Dec 2016.

[219] Yuxiang Ye, Yijuan Lu, and Hao Jiang. Human's scene sketch understanding. In *ICMR '16*, pages 355–358, 2016.

[220] Li Yi, Lin Shao, Manolis Savva, Haibin Huang, Yang Zhou, Qirui Wang, Benjamin Graham, Martin Engelcke, Roman Klokov, Victor S. Lempitsky, Yuan Gan, Pengyu Wang, Kun Liu, Fenggen Yu, Panpan Shui, Bingyang Hu, Yan Zhang, Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Minki Jeong, Jaehoon Choi, Changick Kim, Angom Geetchandra, Narasimha Murthy, Bhargava Ramu, Bharadwaj Manda, M. Ramanathan, Gautam Kumar, P. Preetham, Siddharth Srivastava, Swati Bhugra, Brejesh Lall, Christian Häne, Shubham Tulsiani, Jitendra Malik, Jared Lafer, Ramsey Jones, Siyuan Li, Jie Lu, Shi Jin, Jingyi Yu, Qixing Huang, Evangelos

Kalogerakis, Silvio Savarese, Pat Hanrahan, Thomas A. Funkhouser, Hao Su, and Leonidas J. Guibas. Large-scale 3D shape reconstruction and segmentation from ShapeNet core55. *CoRR*, abs/1710.06104, 2017.

[221] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168:022022, feb 2019.

[222] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, and Yijuan Lu. SHREC'19 Extended 2D Scene Sketch-Based 3D Scene Retrieval Track Website. http://orca.st.usm.edu/~bli/SceneSBR2019/, 2019.

[223] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, and Yijuan Lu. 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*, pages 264–269, 2019.

[224] Juefei Yuan, Hameed Abdul-Rashid, Bo Li, Yijuan Lu, Tobias Schreck, Ngoc-Minh Bui, Trong-Le Do, Khac-Tuan Nguyen, Thanh-An Nguyen, Vinh-Tiep Nguyen, Minh-Triet Tran, and Tianyang Wang. 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*, pages 33–39, 2019.

[225] Juefei Yuan, Bo Li, and Yijuan Lu. SHREC'18 2D Scene Sketch-Based 3D Scene Retrieval Track Website. http://orca.st.usm.edu/~bli/SceneSBR2018/, 2018.

[226] Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. 2D scene sketch-based 3D scene retrieval. In *Eurographics Workshop on 3D Object Retrieval, 3DOR 2018, 16 April 2018, Delft, The Netherlands.*, pages 29–36, 2018.

[227] Juefei Yuan, Bo Li, Yijuan Lu, Song Bai, Xiang Bai, Ngoc-Minh Bui, Minh N. Do, Trong-Le Do, Anh-Duc Duong, Xinwei He, Tu-Khiem Le, Wenhui Li, Anan Liu, Xiaolong Liu, Khac-Tuan Nguyen, Vinh-Tiep Nguyen, Weizhi Nie, Van-Tu Ninh, Yuting Su, Vinh Ton-That, Minh-Triet Tran, Shu Xiang, Heyu Zhou, Yang Zhou, and Zhichao Zhou. SHREC'18 track: 2D scene sketch-based 3D scene retrieval. In *3DOR*, pages 1–8, 2018.

[228] Juefei Yuan, Tianyang Wang, Shandian Zhe, Yijuan Lu, and Bo Li. Semantic tree-based 3D scene model recognition. In *3rd IEEE Conference on Multimedia Information Processing and Retrieval, MIPR 2020, Shenzhen, Guangdong, China, April 9-11, 2020*, 2020.

[229] Han Zhang, Ian J. Goodfellow, Dimitris N. Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 7354–7363, 2019.

[230] Suiyun Zhang, Zhizhong Han, and Hui Zhang. User guided 3D scene enrichment. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1*, VRCAI '16, pages 353–362. ACM, 2016.

[231] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba. Open vocabulary scene parsing. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2021–2029, Oct 2017.

[232] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba. Open vocabulary scene parsing. In *ICCV*, pages 2021–2029. IEEE Computer Society, 2017.

[233] Junbo Jake Zhao, Michaël Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[234] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3D: A large photo-realistic dataset for structured 3D modeling. *CoRR*, abs/1908.00222, 2019.

[235] Wencan Zhong, Alex Noel Joseph Raj, Palaiahnakote Shivakumara, Zhemin Zhuang, Tong Lu, and Umapada Pal. A new shadow detection and depth removal method for 3D text recognition in scene images. In *Proceedings of the 2018 2Nd International Conference on Computer Science and Artificial Intelligence*, CSAI '18, pages 277–281. ACM, 2018.

[236] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, June 2018.

[237] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: a 10 million image database for scene recognition. *IEEE Trans. on PAMI*, 2017.

[238] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2018.

[239] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using Places database. In *NIPS*, pages 487–495, 2014.

[240] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *CoRR*, abs/1608.05442, 2016.

[241] Cai-Zhi Zhu, Herve Jegou, and Shin'ichi Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In *ICCV*, pages 1705–1712, 2013.

[242] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017.

[243] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengyi Gao, Baoquan Chen, and Hao Zhang. SketchyScene: Richly-annotated scene sketches. In *Proc. of ECCV*, 2018.

[244] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. Sketchyscene: Richly-annotated scene sketches. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, pages 438–454, 2018.