UNIVERSITY OF CALIFORNIA SAN DIEGO

Single-Image 2D to 3D Understanding

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Sainan Liu

Committee in charge:

Professor Zhuowen Tu, Chair
Professor Hao Su, Co-Chair
Professor Manmohan Chandraker
Professor Virginia de Sa
Professor Ravi Ramamoorthi

2021

The dissertation of Sainan Liu is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my dearest family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I want to acknowledge Professor Zhuowen Tu for his support as my advisor and committee chair, who has been incredibly understanding and supportive of my career, especially during difficult times throughout raising a newborn and a year of global epidemic disaster caused by COVID. In addition, I want to thank Professor Hao Su, Manmohan Chandraker, Virginia de Sa, and Ravi Ramamoorthi for serving on my thesis committee.

I also would like to thank my colleagues in the mIPC Lab at the Cognitive Science Department in UCSD, Weijian Xu, Yifan Xu, Justin Lazarow, Kwonjoon Lee for meaningful discussions, and Vincent Ngyuen, Zeyu Chen, Isaac Rehg, Wenlong Zhao, Yuan Gao, and Saining Xie for collaborating with me. I want to extend my special thanks to Saining Xie for being my first main collaborator, who helped me tremendously with my first publication, and his encouragement to pursue a Ph.D. career.

I am also honored to have worked as a research intern at Qualcomm, Amazon AWS, Waymo and Google Research. The support and mentorship I have received have made them truly rewarding experiences. I want to thank Intel Lab for the support to my research in my final Ph.D years during COVID, my mentor, Subarna Tripathi and my supervisor Han Lin for their support.

Finally, I would like to dedicate this thesis to my parents, my husband and my daughter. I feel truly fortunate to have tremendous love and support from them.

Chapter 2, in full, is a reprint of the material as it appears in the Proceedings of the Conference on Computer Vision and Pattern Recognition(CVPR), 2018. (Sainan Liu, Saining Xie, Zeyu Chen, Zhuowen Tu, "Attentional ShapeContextNet for Point Cloud Recognition"). The dissertation author was the co-primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in the Conference on Computer Vision and Pattern Recognition(CVPR), 2020. (Sainan Liu, Vincent Nguyen, Isaac Rehg, Zhuowen Tu, "Recognizing Objects From Any View with Object and Viewer-Centered Representations"). The dissertation author was the primary investigator and the author of this paper.

Chapter 4, in part, has been submitted for publication of the material by Sainan Liu, Yuan Gao, Vincent Nguyen, Subarna Tripathi, Zhuowen Tu. This dissertation author was the primary investigator and author of this paper.

Thanks to many anonymous reviewers for their constructive suggestions. Much of this work is supported by NSF and Intel.

VITA

2013-2016	Bachelor of Computer Science, University of British Columbia
2016-2018	Master of Science, University of California San Diego
2018-2021	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

Sainan Liu*, Saining Xie*, Zeyu Chen, Zhuowen Tu "Attentional ShapeContextNet for Point Cloud Recognition", *Conference on Computer Vision and Pattern Recognition*, 2018. (* Co-first Author)

Sainan Liu, Vincent Nguyen, Isaac Rehg, Zhuowen Tu "Recognizing Objects From Any View with Object and Viewer-Centere Representations", *Conference on Computer Vision and Pattern Recognition*, 2020

ABSTRACT OF THE DISSERTATION

Single-Image 2D to 3D Understanding

by

Sainan Liu

Doctor of Philosophy in Computer Science

University of California San Diego, 2021

Professor Zhuowen Tu, Chair
Professor Hao Su, Co-Chair

Visual perception plays an essential role in the human recognition system. We heavily rely on visual cues to accomplish daily tasks. Inspired by human vision and human recognition, computer vision has been widely studied in recent decades to assist human activities better. It has been proven to be highly beneficial to help everyday computer tasks, such as smartphone applications, robotics, and autonomous driving. The fundamental question of computer vision is to understand 3D information from 2D images. Over the years, using machine learning techniques, learning from a single image, research in this area has progressed from 2D recognition to predicting 2.5D images to 3D objects to complete room/street layout prediction. For computer

vision to apply to daily tasks, we believe this is the perfect time to introduce the concept of panoptic 3D parsing, which puts the long-studied sub-problems into unified metrics.

In this dissertation, we first decompose the problem into two subcategories: 1. How to learn better effective priors to recognize objects in 3D. 2. How to enable computer vision neural networks to recognize objects in 2D from unseen views using 3D prior information with techniques inspired by the cognitive science community. In the final chapter, we present a set of networks that unify the understanding of 3D information from a single image thanks to the exploding development in modeling and computing and the availability of large-scale datasets.

Chapter 1

Introduction

Human perception is a complex process that transforms multiple sensory data, such as touch, sound, and visual, into abstract concepts ranging from the position or size to the category or application of the object. It is an essential part of our daily life since it directly influences our behavior. Among the different types of perception, we heavily rely on visual perception, which is a challenging topic on its own due to the diversity of the image perception caused by illumination, viewing angles, the context of objects, memory, and emotion.

Inspired by human vision, thanks to advanced imaging technology and reduced storage costs, computer vision has become an essential part of modern technologies, such as smartphone applications, robotics, and autonomous driving. In contrast to vision problems studied in Cognitive Science or Neuroscience, which focus on understanding human perception via human eyes, computer vision investigates the fundamental question: how to extract 3D information (structurally and perceptually) from 2D images captured by cameras. In contrast to human eyes, the camera captures images with certain limitations in the color spectrum, size, resolution, and illumination range. Additionally, available images are often stochastic in terms of timelines, and stereo or multi-view images are often unavailable for computers. The majority of the computer vision research community has focused on two main aspects: recognition and reconstruction.

This dissertation focuses on 2D (RGB images) and 3D (point cloud) object recognition tasks and incorporates scene recognition and scene reconstruction in computer vision with single-view image input. We hope our results can inspire others to move in this direction.

1.1 Point cloud recognition

Convolutional neural networks (CNN) [LBD+89, KSH12, SZ15, SLJ+15, HZRS16a] and their improvements [SVI+16, HLvdMW17, XGD+17, ZL17] have significantly advanced the state-of-the-art for a wide range of applications in computer vision. Areas like classification, detection [GDDM14, RHGS15], and segmentation [LSD15, HGDG17] for 2D images have witnessed the a tremendous advancement. Extending 2D-based convolution to 3D-based convolution for 3D computer vision applications such as 3D medical imaging [PM15, DYC+17], though still effective, is arguably less explosive than the 2D cases. This observation becomes more evident when applying 3D convolution to videos [TBF+15, CZ17, TWT+17, XSH+17] where 2D frames are stacked together to form a 3D matrix. Innate priors induced from careful study and understanding of the task at hand are often necessary.

The development of large datasets of 2D static images like ImageNet [DDS+09] is critical in the recent development of deep learning technologies. Similarly, the emergence of 3D shape-based datasets such as ShapeNet [CFG+15] has attracted significnat attention and stimulated 3D shape classification and recognition advancement. As lidar technology matures, large-scale scanned point cloud datasets (*e.g.* ScanNet [DCS+17]) and autonomous driving point cloud datasets (*e.g.* KITTI [GLSU13] and Waymo [way19]) become available. The social impact of point cloud recognition in Computer Vision becomes increasingly prominent. The 3D shape classification and recognition problem have been extensively studied in computer graphics [CTSO03, GZC15, SYS+16] and robotics [RBB09, WP15]. Unlike 2D images, shapes encoded by 3D point clouds [WSK+15, YKC+16] do not have a well-positioned strict grid structure, nor

is there an intensity value associated with each point. The unstructured and unordered nature makes the point cloud recognition problem extremely challenging.

Previous works [BSA17, BGLA18] have converted point clouds to RGB or RGB-D images to take full advantage of 2D convolutional neural networks. They first convert the point cloud to multi-view representations. Then project back to 3D to acquire the 3D semantic labeling. Another group of work [JXYY13, YKC+16, SMKLM15, WSK+15, TCA+17] in which scattered 3D points are assigned to individual cells in a well-structured 3D grid framework. Instead of binary 3D volumetric data, [MGLM18] utilizes a radial basis function combined with a variational autoencoder to obtain enriched voxel representations. This type of conversion from 3D points to 3D volumetric data can facilitate the extension from 2D CNN to 3D CNN , but it also loses the intrinsic geometric property of the point cloud.

A pioneering work, PointNet [CSKG17], addresses the fundamental representation problem for the point cloud by obtaining the intrinsic invariance of the point ordering. Well-guided procedures are undertaken to capture the invariance within point permutations for learning an effective network, achieving state-of-the-art results with many desirable properties. However, one potential problem with PointNet is that the concepts of parts and receptive fields are not explicitly addressed, because the point features in PointNet are treated independently before the final aggregation (pooling) layer. An improved work, PointNet++ [QYSG17], has been developed to incorporate the global shape information using special modules such as farthest point sampling and geometric grouping. Many follow-up works build their base network structure on top of PointNet. Such as PointSIFT [JWL18], which designed a SIFT-like module for all PointNet-based models that provide orientation embedding in 8 directions. [LS17] uses PointNet structure as a basic embedding for its super point (represents relationships between object parts.). [YZW+18, WLS+19] both build on top of PointNet to achieve instance segmentation tasks. [LVC+19] utilizes PointNet structure to encode features for each column space for autonomous driving datasets.

Instead of using multi-layer perceptron as the main architecture [CSKG17], Convolutional Neural Networks have also been widely explored. [KZH19, LFXP19, WQL19, XSW+20, LBSC18] designed advanced operators to capture local neighborhood information. Additionally, many novel network architectures have been explored. For example, RNN network structures [YLH+18, HWN18] have been used to explore local features. Other structures are used to explore on non-local features, such as adaptive sampling modules[HYX+19, YZL+20], transformer structures [YZN+19, ZX19, GCL+20] and graph networks [LS17, WSL+18, THGZ18], or a combination of the two [WHH+19].

1.2 Single-view object recognition

Identifying a familiar object seems effortless for humans has proven to be a reasonably complex task in computer vision.

2D object single-view recognition is commonly defined as a classification problem in computer vision. The input is a single RGB image, and the output is a class label. Convolutional neural networks have been a driving force for tackling this problem [KSH12, HZRS16b, GPAM+14, GBC16]. As more and more single-view image data become available [DDS+09, LMB+14, COR+16], deep convolutional neural networks have shown remarkable performance on the 2D object single-view recognition task. However, human recognition often combines multiple sensory inputs other than a single RGB view of an object. At a high level, the other sensors provide a certain level of 3D perception that allows us to infer novel views of novel objects from the same seen category. In human recognition, such perception has been shown to be providing object-centric information.

Similarly, other sensory such as lidar or ultrasound in computer vision provides a certain degree of 3D information. Hence 3D object recognition is also an essential part of object recognition. Many forms of 3D object representations have been widely studied, such as ex-

plicit volumetric voxel[WSK+15], mesh [WZL+18], point cloud [CSKG17], implicit surface [XWC+19], and implicit functions [MST+20a].

Objects are three-dimensional in the physical world, but the recognition tasks in computer vision have been primarily performed on 2D natural images [DDS+09]. Despite the great success of the deep convolutional neural networks (CNNs) [KSH12, SLJ+15, SZ15, HZRS16a, XGD+17], a standard CNN model that represents images in the 2D image space only tends to suffer from a "mental rotation" [SM71] like effect [Bas93], as shown in Figure 3.3. Namely, when training a network with a limited number of views of an object instance, it may have difficulty recognizing the same object instance from an unseen viewpoint. There are two schools of thought regarding object representations. For biological vision systems, there has been a long-time debate [LPP95] in cognitive psychology about whether objects are fundamentally encoded by *object-centered* or *viewer-centered* representations [TV02, Hay12]. In David Marr's pioneering vision paradigm [Mar82], object recognition is carried out primarily in an object-centered manner in which objects are represented either by explicit 3D primitives (*e.g.* cylinders) [Bie87] or by features that are invariant to viewpoint changes [BS83]. However, the theory of object-centered representation has been challenged in the past. Psychophysical and computational neural studies have shown evidence that viewer-centered representations [RD87, LP95, DM97] play a significant role in object recognition.

Implementations of both viewpoint-independent [LPS07, LSS08] and viewpoint-dependent [Bas93, BBZ+16] systems are present in computer and machine vision literature. An object-centered system typically encodes and stores a representation with viewpoint-independent (object-centric) features [KFR03] that are invariant to viewpoint changes. During test time, representations with viewpoint-independent features are computed for a query object under a novel view to match the stored features. A viewer-centered system instead stores a set of viewpoint-dependent features from typical viewing angles. A new view of an object instance is matched to the known features of trained viewpoints during testing.

An object-centered representation has the advantage of maintaining rotation-invariant features that are insensitive to viewpoint changes; however, it relies on the presence of faithful 3D reconstructions or effective invariant features that are usually difficult to obtain from a single view image [Hay12]. Conversely, a viewer-centered representation typically stores features that are sensitive to viewpoint changes; viewpoint-dependent features are usually straightforward to compute and learn.

Studies that combine both object-centered and viewer-centered representations also exist [MF91, BM00, Mil12]. However, there has been limited success in the computer vision literature to build a hybrid system [KT03]. Additionally, systematic novel-view evaluation metrics are rarely used to evaluate the new state-of-the-art recognition systems.

1.3 Scene Reconstruction and Parsing

Humans have the remarkable capability of recognizing and understanding 3D objects and scenes in diverse environments and configurations. This capability has been attributed to effective representations that encode the intrinsic 3D world for the 2D projections [LPP95] (though still, the mechanisms of forming these representations are not fully understood). A grand challenge in computer vision is to reach the same capabilities through machine perception. Though the task of translating the perceptual abilities of humans to machines is deeply rooted in decades of development in computer vision [Sze10] and photogrammetry [Lin09], it has only recently become practically feasible thanks to the exploding growth in modeling and computing[KSH12, HZRS16b, GPAM$^+$14, GBC16] and the availability of large-scale datasets [DDS$^+$09, LMB$^+$14, COR$^+$16, CFG$^+$15, FJG$^+$20].

As subproblems of computer vision, such as 2D recognition, 3D recognition, and 3D reconstruction, have been widely studied, tremendous success has been acquired in recent years. Scene reconstruction and scene understanding have also undergone revolutionary progress.

Recent works have focused on indoor scenes or autonomous driving 3D reconstruction along with 3D instance recognition. They achieve multi-object reconstruction in a scene via shape retrieval [ISS17, HQZ+18, YLH+18, YLH+18, KALD20, ERLF21] or directly reconstruct a type of shape representations [TGF+18, GMJ19, NHG+20, SZW19, PBF20, ZCZ+21].

For retrieval-based methods, IM2CAD [ISS17] is one of the pioneer works that utilizes object detection and scene segmentation information to predict box layout estimation and pose of objects to assist the retrieval of CAD models directly from a database. The output is a scene-level CAD model that is ready to use for computer graphic applications. A non-differentiable second-stage optimization step will then match the rendered scene reconstruction with the input image. [HQZ+18] additionally models the scene as a hierarchical graph and optimizes reconstruction and input image with the estimated surface normal, the depth map, and the object mask. [ISS17] and [HQZ+18] use 2D bounding boxes for the detection step. In contrast, [KALD20] and [ERLF21] use center prediction instead, which makes the detection step simpler. Comparing to [KALD20], [ERLF21] does not require object depth at test time for object pose prediction. It supports object stretching, and it includes collision loss which respects more of reconstructed object boundaries. [ERLF21] has shown impressive results on the Pix3D dataset. However, [KALD20] and [ERLF21] do not provide an end-to-end solution for joint layout prediction. The methods developed for indoor scenes can work relatively well for close-range predictions and are rarely evaluated for long-range environments, possibly due to limited annotations for outdoor city scenes. The majority of work for long-range detection is done for autonomous driving datasets. One such work is 3D-RCNN [YLH+18], which represents the shape using a linear basis from the training dataset, suitable for categories that share significant similarities within each class, such as pedestrians and cars.

All retrieval methods tend to produce visually good results. However, it is less likely to be able to generalize towards novel categories and novel shapes. Hence, another group of researchers focuses on reconstructed shapes when it comes to 3D scene understanding. Factored3D [TGF+18]

first proposed a volumetric prediction network that combines 3D unoccluded layout depth with reconstructed 3D volumetric furniture for indoor synthetic single-view images. Later, Mesh R-CNN [GMJ19] provided an end-to-end system that detects and reconstructs the 3D instances for a scene. However, they did not offer layout 3D information due to dataset limitations, nor do their instance reconstructions have correct relative 3D positioning, given that they do not resolve the scale/depth ambiguity. Total3DUnderstanding [NHG+20] can predict 3D object reconstruction with better 3D relative positioning; however, only a simplified box prediction is used to estimate the indoor layout environment. In addition to [NHG+20], [ZCZ+21] uses implicit 3D representation, Signed Distance Function (SDF), and a scene graph convolutional network to model the relative relationships between objects. Comparing to [NHG+20], [ZCZ+21] is capable of generating watertight, and better-aligned scenes. However, both [NHG+20] and [ZCZ+21] heavily rely on an external 2D bounding box detector. In contrast, [PBF20] does not use any detector as the first step. With a fixed 128^3 voxel grid, the network can address reconstruction of multiple objects simultaneously. The voxel grid also enables quantitative evaluation for multi-object scenes with new pairs and triplets ShapeNet datasets. However, when it comes to natural scenes, it tends to predict holes and errors. These networks require heavy annotations on the 3D side during training, which makes the network pipeline complicated and hard to optimize with multi-modalities during training. Most recently, researchers have been focusing on unsupervised generative models. Implicit scene representations with neural rendering techniques have been widely explored. [SZW19] proposed scene representation networks, which map (x, y, z) world coordinates to a feature representation at that position and uses a neural renderer to facilitate novel synthesis. Later, a variety of NeRF-based [MST+20b] networks have been explored for implicit scene understanding tasks using multi-view inputs. NeRF models learn a multi-layer perceptron that maps (x, y, z) coordinates to color and density values via neural volume rendering techniques. For example, using dynamic surveillance video footage, [OMT+20] can model the background and individual cars on the street with different NeRF-based structures

and a shared volume renderer. Although NeRF-based models do not rely on 3D annotations, it requires multi-view inputs for the learning process. The camera information is often needed and generalization towards novel scenes are still an open topic. Additionally, it provides a continuous 3D representation, but it sacrifices the ready-to-use discrete geometry output without guaranteeing watertight object shapes. The separation between objects and their environment needs to rely on the availability of dynamic video sequences, 2D segmentation masks, or 3D bounding box predictions.

Scene reconstruction and 3D panoptic (non-overlapping semantic and instance) understanding for both long-range and short-range environments is still an significant step forward to providing unified metrics for both reconstruction and semantic/instance segmentation with indoor and outdoor scenes.

1.4 Overview of This Dissertation

In this dissertation, we explore the problems of computer recognition. The research first focuses on sub-problems of 3D and 2D recognition. Then it asks the fundamental question of computer vision: how to extract 3D information from natural 2D images in the wild.

Chapter 2 describes 3D recognition in computer vision. More specifically, we tackle the problem of point cloud recognition. Unlike previous approaches where a point cloud is either converted into a volume/image or represented independently in a permutation-invariant set, we develop a new representation by adopting the concept of shape context as the building block in our network design. The resulting model, called ShapeContextNet, consists of a hierarchy with modules not relying on a fixed grid while still enjoying properties similar to those in convolutional neural networks — being able to capture and propagate the object part information. In addition, we find inspiration from self-attention based models to include a simple yet effective contextual modeling mechanism — making the contextual region selection, the feature aggregation, and the

feature transformation process fully automatic. ShapeContextNet is an end-to-end model that can be applied to the general point cloud classification and segmentation problems. We observe competitive results on a number of benchmark datasets.

Chapter 3 describes 2D recognition assisted with both object-centered and viewer-centered representations. In this paper, we tackle an important task in computer vision: any view object recognition. In both training and testing, for each object instance, we are only given its 2D image viewed from an unknown angle. We propose a computational framework by designing object and viewer-centered neural networks (OVCNet) to recognize an object instance viewed from an arbitrary unknown angle. OVCNet consists of three branches that respectively implement object-centered, 3D viewer-centered, and in-plane viewer-centered recognition. We evaluate our proposed OVCNet using two metrics with unseen views from both seen and novel object instances. Experimental results demonstrate the advantages of OVCNet over classic 2D-image-based CNN classifiers, 3D-object (inferred from 2D image) classifiers, and competing multi-view based approaches. It gives rise to a viable and practical computing framework that combines both viewpoint-dependent and viewpoint-independent features for object recognition from any view.

Chapter 4 describes the first approach that tries to interpret 3D panoptic scene parsing in the wild from a single 2D input image. In this paper, we present Panoptic 3D Parsing (Panoptic3D), the first system of its kind (to the best of our knowledge) for a single-view natural image in the wild that jointly performs dense semantic segmentation, object detection, instance segmentation, object 3D shape reconstruction, and 3D layout estimation altogether. We combat the issue under the absence of complete sets of multi-modality ground-truths for segmentation/objects/3D shapes/3D layout by developing a stage-wise system to maximize the generalization and robustness of the system where ground-truths are separately available for training the individual modules. We also present an alternative Panoptic3D system that can be trained end-to-end using synthetic data where a complete set of multi-modality ground-truth annotations for the 2D segmentation and 3D reconstruction can be generated synthetically. We show results on both indoor and outdoor scenes

from COCO and Cityscapes as well as quantitative panoptic 3D results on a fully annotated synthetic indoor dataset. Our proposed Panoptic3D framework points to a viable direction in computer vision and it can be applied to a wide range of applications. A system demo[1] is available at `http://35.82.89.112:443`.

[1]The demo is put up for review purpose with light use.

Chapter 2

3D Point Cloud Recognition

2.1 Introduction

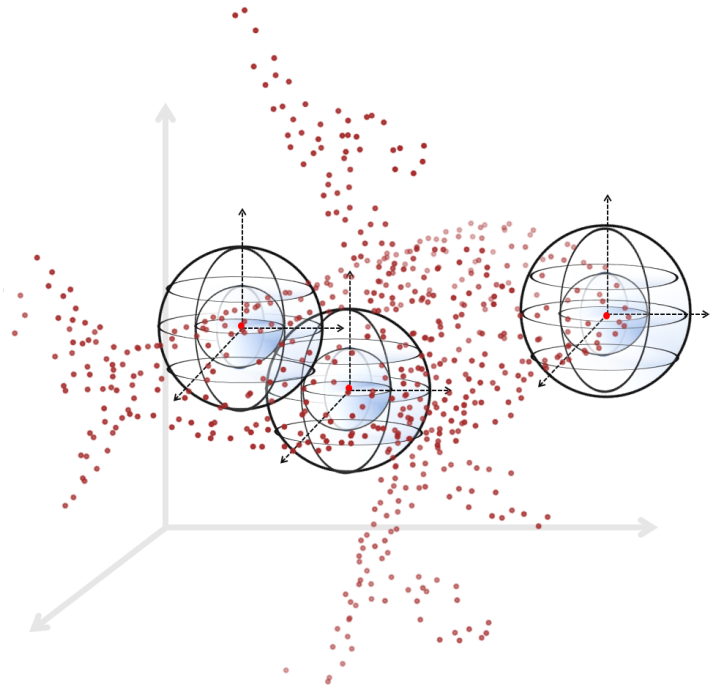
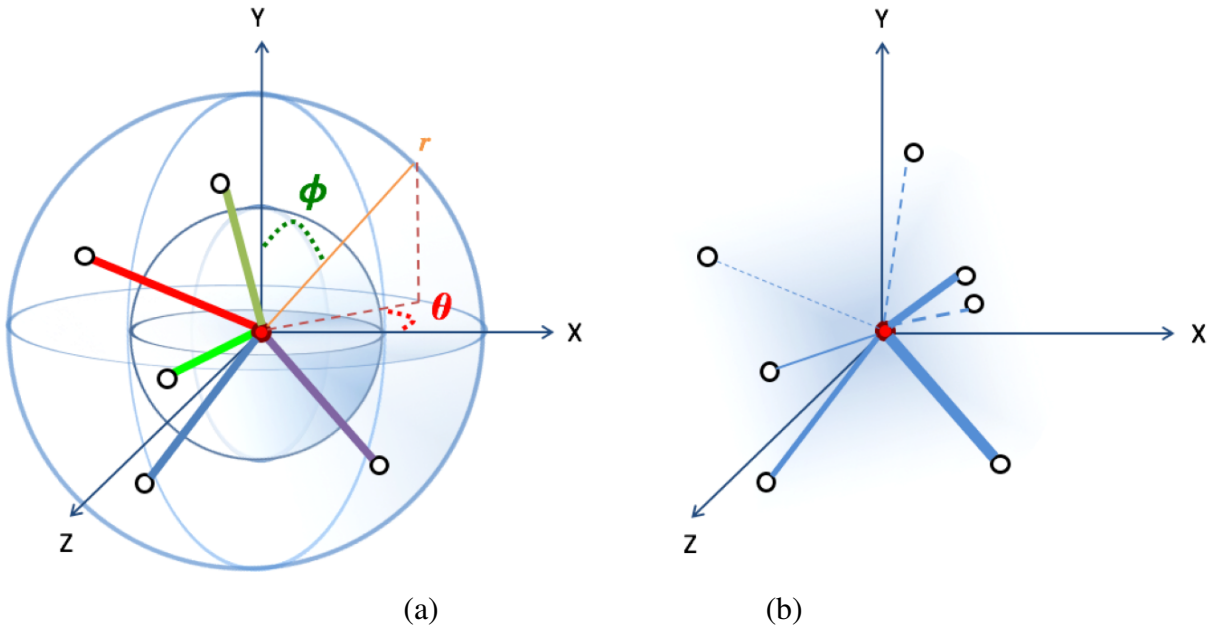


Figure 2.1: A motivating example to illustrate how the basic building block of our proposed algorithm, the shape context kernel, is applied to a 3D point cloud to capture the contexual shape information.

This chapter focuses on developing a deep learning architecture for point cloud classifica-



(a) (b)

Figure 2.2: An illustration of our shape context kernel displayed in a spherical coordinate system. (a) the shape context kernel, the number of bins on polar angle (ϕ), number of bins on azimuthal angle (θ) and number of bins on radial distance (r) are manually specified. Different colors of edges represent different binary affinity matrices indicating different bins. (b) the attentional shape context "kernel", where there is no predefined bins, and the soft affinity matrix, or attention weights (indicated by edge thickness) are learned during training.

tion that connects the classic idea of *shape context* [BMP02] to the learning and computational power of hierarchical deep neural networks [LBD$^+$89]. We name our algorithm *ShapeContextNet* (SCN) and a motivating example is shown in Figure 2.1.

Before the deep learning era [KSH12], carefully designed features like shape context [BMP02] and inner distances [LJ07] were successfully applied to the problem of shape matching and recognition. In *shape context*, an object is composed of a number of scattered points and there is a well-designed disc with unevenly divided cells to account for the number of neighborhood points falling into each cell; the overall features based on the occurrences of the points within every individual cells give rise to a rich representation for the object parts and shapes. *Shape context* was widely used before but kept relatively distant to the deep learning techniques.

Motivated by the rich representational power of *shape context* [BMP02], as well as the recent success in deep convolutional neural networks [KSH12], we propose a new method, ShapeContextNet (SCN) that adopts *shape context* as the basic building block acting like convolution in CNN. The basic network architecture of SCN is illustrated in Figure 2.4 with the basic shape context descriptor shown in Figure 2.2. We do not force a given set of points into volumetric data, nor do we remove the spatial relationship of the points. Instead, we build layers of shape context to account for the local and the global contextual information in a hierarchy learned by an end-to-end procedure. In order to incorporate the local shape context descriptor into a neural network, we break a *shape context block* into three key components, namely *selection*, *aggregation*, and *transformation*. For a point p_i in the point cloud $\{p_1, p_2, \ldots, p_i, \ldots, p_N\}$, the set of all $N-1$ points forms a rich context depicting the shape information centered at p_i. However, using all the neighborhood points might be computational and spatially unattractive. We instead design *shape context kernel* with distributed bins in the log-polar space, shown in Figure 2.2 which is inspired by the *shape context* descriptor [BMP02]. The *selection* operation thus decides a set of neighboring points of p_i to define coarse groups of neighboring points for p_i to attend to. The *aggregation* operation (such as histogram, or pooling) builds a robust descriptor that captures

the distribution over relative positions. The *transformation* operation projects the descriptor to a high-dimensional feature space by fusing features from different neighboring points or groups. Like in the standard CNN, SCN propagates the local part information through hierarchical layers to capture the rich local and global shape information.

Although the concept of building deep *shape context* is simple, we still face many implementation choices in practice: how to design the *shape context* bins and handle the additional computation cost for computing "point to bin" relationships, how to choose an aggregation operation that preserves feature discriminability, etc. We are inspired by the recent development in attention-based models that are mainly applied in natural language processing tasks such as sequence-to-sequence modeling [BCB15, XBK+15]. A self-attention approach is proposed in [VSP+17] and achieves state-of-the-art results on the neural machine translation task with an architecture design that consists of a stack of self-attention blocks. The dot-product self-attention block has no recurrence — keys, values and queries come from the same place and is highly efficient in computation. We connect the self-attention idea with *shape context* within a supervised learning setting. Self-attention combines the selection and aggregation process into a single soft alignment operation. The resulting model enjoys the property of *shape context* and is an end-to-end trainable architecture without the bells and whistles of a handcrafted selection operation (bins). We call it *Attentional ShapeContextNet* (A-SCN).

We apply SCN and A-SCN to 3D point shape classification and segmentation datasets [WSK+15, YKC+16] and observe improved results over the PointNet [CSKG17] model.

2.2 Our Approach

2.2.1 Revisiting the *Shape Context* Descriptor

We first briefly describe the classic *shape context* descriptor, which was introduced in a seminal work [BMP02] for 2D shape matching and recognition. One main contribution

in [BMP02] is the design of the shape context descriptor with spatially inhomogeneous cells. The neighborhood information for every point in a set is captured by counting the number of neighboring points falling inside each cell. The shape descriptor for each point is thus a feature vector (histogram) of the same dimension as the number of the cells with each feature dimension depicting the number of points (normalized) within each cell. The shape context descriptor encodes the rich contextual shape information using a high-dimensional vector (histogram) which is particularly suited for matching and recognition objects in the form of scattered points. For each point p_i in a give point set,

shape context computes a coarse histogram h_i of the relative coordinates of the neighboring point,

$$h_i(l) = \#\{p_j \neq p_i : (p_j - p_i) \in bin(l)\}. \tag{2.1}$$

Shape context uses a log-polar coordinate system to design the bins. Figure 2.3 shows a basic 2D shape context descriptor used in our method (note that we make the center cells larger which is slightly different to the original shape context [BMP02] design where the center cells are relatively small).

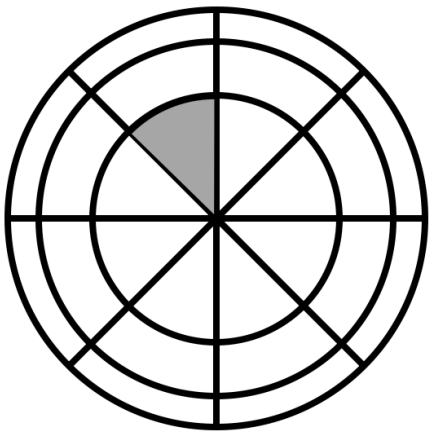


Figure 2.3: Example of a 2D shape context kernel with 24 bins ($n_r = 3$ and $n_\theta = 8$).

There were also attempts to extend *shape context* to 3D. In [KPNK03] concentric shells,

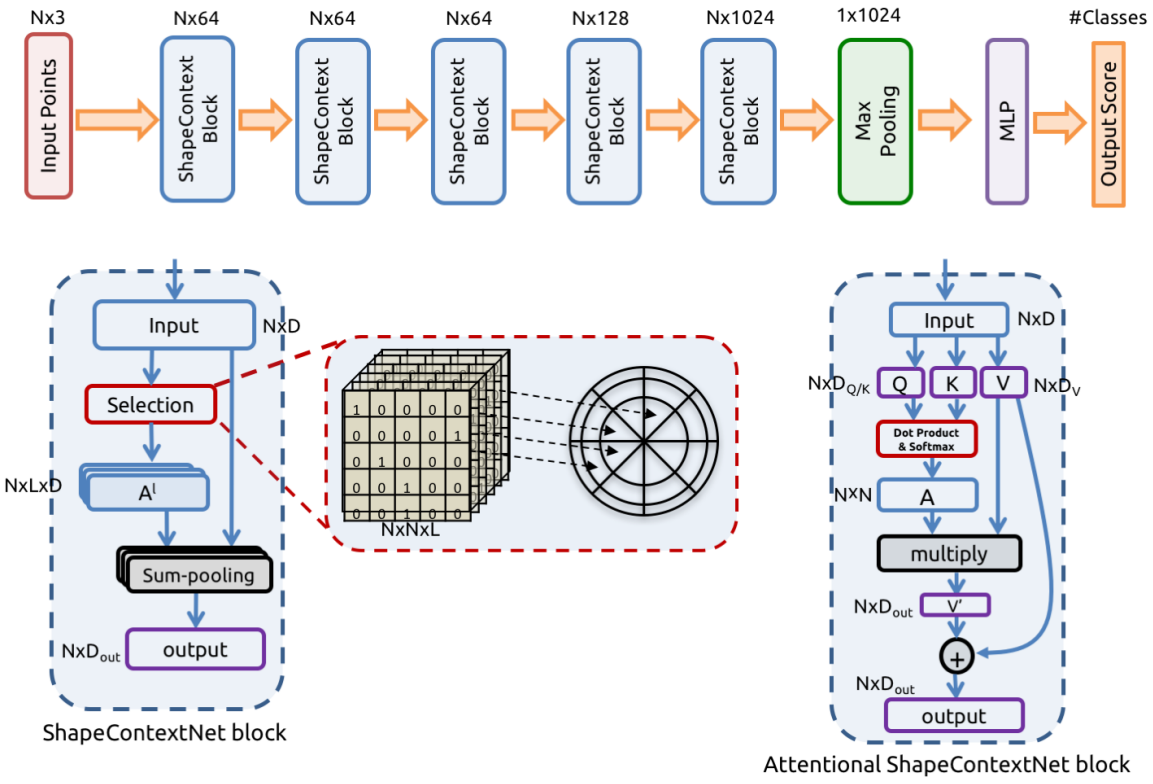


Figure 2.4: **ShapeContextNet (SCN) and Attentional ShapeContextNet (A-SCN) architectures.** The classification network has 5 ShapeContext blocks; each block takes N point feature vectors as input, and applies the selection, aggregation and transformation operations sequentially. The ShapeContext blocks can be implemented by hand-designed shape context kernels (SCN block), or a self-attention mechanism learned from data (A-SCN block). See text in Section 2.2 for details.

polar angle ϕ and azimuthal angle θ are considered to divide the space into different quadrants. We use a similar design for our bins, as is shown in Figure 2.2 (a). Although shape context is considered as one of the most successful descriptors in computer vision, its integration into the modern deep learning framework has been under-explored.

2.2.2 A General Formulation

In this section, we introduce a generalized formulation for *shape context* to build our deep ShapeContextNet. Let a given point set (cloud) for one shape be $P = \{p_1, p_2, \cdots, p_i, \cdots, p_N\}$.

Each $p_i \in \mathcal{R}^3$ is a point represented by its 3D coordinates. Our proposed ShapeContextNet (SCN) is a neural network architecture (shown in Figure 2.4) with its basic building block being SCN block (illustrated in Figure 2.2 (a)). Each SCN block consists of three operations: *selection*, *aggregation*, and *transformation*, which will be explained in detail below.

Selection. For a point cloud P of N points, the selection operation is to produce an affinity matrix $A \in \{0,1\}^{N\times N}$, where $A(i,j) = 1$ indicates that a point p_j has an edge to a reference point p_i, while $A(i,j) = 0$ indicates that a point p_j has no connection to point p_i. The connected component centered at point p_i is a representation of the global shape arrangement. In the original *shape context*, the selection operation first divides the space into L bins. In that case, instead of having a single affinity matrix, we build L disjoint affinity matrices simultaneously, and $A^l(i,j) = 1$ means $p_j \in bin(l)$ of the reference point p_i, for $l = 1,\cdots,L$. Note that the selection operations do not necessarily rely on any predefined partitioning of space, and can be automatically learned in the same vein as attention mechanism, where the A is the $N \times N$ attention weight. The attentional selection operation can either be hard or soft assignments.

Aggregation. After the selection operations, to form a compact representation of shape arrangement at a reference point p_i, we need to aggregate the information from the selected points. We denote an aggregation function as m. In original *shape context*, for N points and L bins, and a reference point p_i, we have L aggregation functions $m_i^l, l = 1,\cdots,L$, which together form the histogram representation. Each m_i^l is a counting function that counts the number of points in $bin(l)$, which can be represented as a sum pooling function $m_i^l = \sum_j \mathbb{1}[A^l(i,j) = 1]$.

In a more general form, m can be a weighted sum operator (dot product) such that $m_i = \sum_j A(i,j) \cdot \hat{p}_j$ using the learned attention weights A. $\hat{p}_j$ could be simply the input coordinates p_j, or any arbitrary feature vector associated with that point.

Transformation. Now we have an aggregated representation for the reference point p_i. It is natural to add a feature transformation function f to incorporate additional non-linearity and increase the capacity of the model. In the original *shape context*, after a local descriptor is built, a

discriminative classifier, e.g. a support vector machine, can be added for the final classification task. The transformation can be realized by a kernel function such as a radial basis function. In the context of deep neural networks, an MLP, or convolutional layer with a non-linear activation function can be used for the feature transformation purpose.

Shape context block. After we introduce the above three operations, the *shape context* descriptor *SC* can be formulated as,

$$SC_i = f(h_i) = f([h_i(1), \cdots, h_i(L)]) = f([m_i^1, \cdots, m_i^L]) \tag{2.2}$$

where $m_i^l = \sum_j \mathbb{1}[A^l(i,j) = 1]$. Note that every components in this formulation can be implemented by a backpropagatable neural network module, and thus, similar to a convolutional layer, *SC* is a compositional block that can be used to build a shape context network,

$$SCNet = SC_i(SC_i(SC_i(\cdots))) \tag{2.3}$$

2.2.3 ShapeContextNet

Shape context kernel. Similar to [KPNK03], we use concentric shells to design the shape context kernel. The kernel is adjustable with three parameters: polar angle ϕ, azimuthal angle θ and radial distance r (Figure 2.2 (a)). In our setting, ϕ and θ are evenly divided into different sectors, while for r, a logarithmic parametrization of the shell radii is used. We also set a maximum radius of the sphere max R, which defines the receptive field size for a single shape context kernel. Thus the design of the shape context kernel is paramerized by the maximum radius (max R), the number of bins for radius r (n_r), angles θ (n_θ) and angles ϕ (n_ϕ). The combined number of bins for a shape context kernel is equal to $n_r \times n_\theta \times n_\phi$.

Selection. With the L bins induced by a shape context kernel, the selection operation builds L disjoint affinity matrices $A^1, \cdots, A^L$, where each matrix is corresponding to a specific bin.

We generate the affinity matrices online during training and share them across different layers.

Aggregation. Following original *shape context*, the aggregation operation is simply a sum-pooling layer that aggregates points (associated with D-dimensional feature vectors) within each bin. Note that the sum-pooling layer can be implemented by L parallel matrix multiplications, as A^L is binary. The aggregation operation results in L sets of pooled features, thus the output is a tensor of shape $N \times L \times D$.

Transformation. Finally the transformation operation is realized by a convolutional layer with a $[L, 1]$ kernel that fuses L sets of feature points and projects them to (higher dimensional) output feature vectors of D_{out}. A ShapeContext block consists of above operations and our ShapeContextNet is a stack of ShapeContext blocks with increasing output dimensions of D_{out}. We follow the overall network configuration of PointNet and use $D_{out} = (64, 64, 64, 128, 1024)$ as the output dimensions for each ShapeContext block.

Limitations. While being conceptually simple and enjoying good properties of classic *shape context* descriptors such as translation-invariance, handcrafting shape context kernels are not straight-forward and hard to generalize across different point cloud datasets which usually have varying size and density. This motivates us to propose the following attention-based model.

2.2.4 Attentional ShapeContextNet

We now introduce a different approach inspired by research in natural language processing (sequence-to-sequence) tasks. Traditional sequence-to-sequence models usually adopt recurrent neural networks (e.g. LSTM[HS97]), external memory or temporal convolutions to capture the context information. The dot-product self-attention proposed in [VSP+17] is a model that handles long path-length contextual modeling by a light-weight gating mechanism, where the attention weight matrix is generated using a simple dot-product. It is worth-noting that self-attention is also invariant to the input ordering. Unlike traditional attention-based sequence-to-sequence models, in a self-attention block, *query* vector $Q \in \mathcal{R}^{D_Q}$, *key* vector $K \in \mathcal{R}^{D_K}$ (usually $D_Q = D_K$) and

value vector $V \in \mathcal{R}^{D_V}$ are learned from the same input. In a supervised classification setting, one can think Q, K and V are just three feature vectors learned by three independent MLP layers. Attention weights are computed by a dot product of Q and K, and then multiplied with V to obtain the transformed representation.

Figure 2.2 shows the similarities and differences between manually specified shape context kernels and the automatically learnable self-attention mechanism: They all aim to capture the distribution over relative positions; they are unified under the same formulation in Section 2.2.2; the selection operation in self-attention does not rely on hand-designed bin partitioning as it can be learned from data; self-attention has better modeling capability by adopting a weighted sum aggregation function, in contrast to using a simple sum-pooling function.

Selection and Aggregation. We consider computing self-attention on the whole point cloud P of size N. The selection operation produces a soft affinity matrix, which is the self-attention weight matrix A of size $N \times N$, the aggregation operation is transforming the value vector V with weight matrix A by a dot product,

$$\text{Attention}(Q,V,K) = \text{Softmax}(\frac{QK^T}{\sqrt{D_Q}}) \cdot V \tag{2.4}$$

Transformation. MLPs with ReLU activation function can be added as a feature transformation operation after each self-attention operation (Equation 2.4). To further improve the model expressiveness, we add a simple feature gating layer to the MLP, similar to [DFAG17, PSdV$^+$18].

2.3 Experiments

2.3.1 ShapeContextNets: 2D case

We first showcase the effectiveness of deep ShapeContextNet which has a stack of shape context blocks.

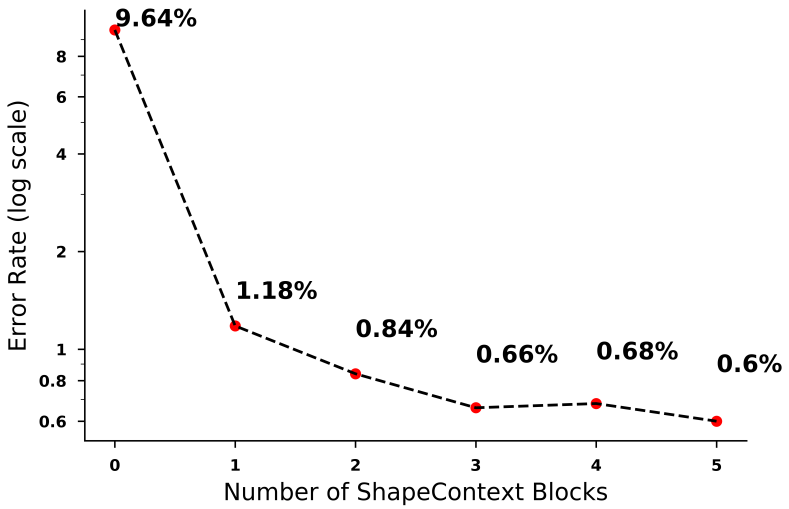


Figure 2.5: **Ablation analysis on the number of ShapeContext blocks.** The error rates obtained by increasing the number of ShapeContext blocks. Metric is overall accuracy on 2D MNIST test set (N = 256). The bin configuration is: max $R = 0.5$, $n_r = 3$, $n_\theta = 12$.

Table 2.1: 2D point cloud classification results on the MNIST dataset. ShapeContextNet achieves better performance than PointNet showing the effectiveness of contextual information; the *shape context* local model consists of only one shape context block.

Model	N	Error rate (%)
PointNet[CSKG17]	256	0.78
PointNet++[QYSG17]	512	0.51
shape context local	256	1.18
ShapeContextNet	256	0.60

2D point set is generated for MNIST dataset following the same protocol as used in PointNet[CSKG17], where 256 points are sampled for each digit. We use a shape context kernel with max $R = 0.5$, $n_r = 3$ and $n_\theta = 12$, thus 36 bins in total.

Table 2.1 shows that a simple 5-layer SCN achieves better performance than PointNet, showing that using the distribution over relative positions as a context feature is indeed helpful for the shape recognition task. The performance of SCN is also competitive to the recent PointNet++[QYSG17] model which uses 512 points as input. *shape context* local is a model that consists of only one shape context block, which resembles the "feature extraction and classifier learning" pipeline in traditional computer vision. To better understand the importance of hierarchical learning in ShapeContextNet, in Figure 2.5, we vary the number of shape context blocks from 0 to 5 in the network (Figure 2.4), where the 5-layer model is our ShapeContextNet, the 1-layer model is the *shape context* local model, and 0 means no shape context block. We observe that as the number of shape context blocks increases, the error rate decreases.

2.3.2 ShapeContextNets: 3D case

We evaluate the 3D shape classification performance of SCN on the ModelNet40 dataset [WSK$^+$15], with point cloud data from 12,311 CAD models in 40 categories. We use 9,843 for training and 2,468 for testing. Following [CSKG17], 1,024 points are sampled for each training/testing instance. Table 2.2 summarizes the impact of different shape context kernel design choices parametrized by max R, n_r, n_θ and n_ϕ.

We obtain the best results with max $R = 0.5$. Note that the coordinates of point cloud in ModelNet40 are normalized to $[-1, 1]$. This means the receptive field of a single shape context kernel covers around a quarter of the entire point cloud. With the same radius bin configuration, the test accuracy peaks when $n_r = n_\theta = n_\phi = 3$. Empirically, the number of r bins has the least impact on the test accuracy, whereas the number of θ bins appears to be crucial for the performance. With minimal change in architecture to a *vanilla* PointNet (by replacing the MLP layers to carefully

Table 2.2: **Ablation analysis on shape context kernel design in ShapeContextNet.** We evaluate SCN models with different kernel configurations (model (A)-(I)). max R is the maximum local radius for the sphere shape context kernel at each reference point. n_r, n_θ and n_ϕ are the number of different shell and angle bins. Unlisted values are identical to those of the preceding model. We report averaged and overall accuracy on ModelNet40 test set (N=1024).

	max R	No. of r bins	No. of θ bins	No. of ϕ bins	accuracy avg. class	accuracy overall
PointNet vanilla[CSKG17]	-	-	-	-	-	87.1
PointNet[CSKG17]	-	-	-	-	86.2	89.2
PointNet++[QYSG17]	-	-	-	-	-	**90.7**
(A)	0.25	3	3	3	86.2	89.3
(B)	1				84.8	88.6
(C)	0.5	2			86.7	89.6
(D)		4			86.5	89.6
(E)		3	2		81.4	84.8
(F)			4		82.2	84.2
(G)			3	2	85.5	88.9
(H)				4	87.5	89.7
SCN (I)	0.5	3	3	3	**87.6**	**90.0**

designed shape context kernels), ShapeContextNet (model (I)) achieves better or competitive results compared to full PointNet model (with additional input/feature transformation layers), and the recent PointNet++ model (with special sampling/grouping modules).

Table 2.3: **Ablation analysis on the Attentional ShapeContextNet architecture.** We evaluate the Attentional ShapeContextNet model on ModelNet40 dataset with different hyperparameter settings (model (A)-(G)). We report class-averaged and overall accuracy on test set. Unlisted values are identical to those of the preceding model. Q, K and V here represent the feature vectors learned in an A-SCN block (Figure 2.4).

A-SCN	Q=K?	ReLU Q/K/V	BN Q/K/V	residual connect.	Num of heads	accuracy avg. class	accuracy overall
(A)	✓	✓/✓/✓	✓/✓/✓	✓	1	85.7	89.0
(B)				✗		28.2	36.7
(C)	✗			✓		85.7	89.1
(D)		✗/✗/✗				86.1	89.2
(E)			✗/✗/✓			**87.4**	**89.8**
(F)					2	86.3	89.2
(G)					4	87.2	89.8

Table 2.4: **Segmentation results on ShapeNet part dataset.** We compared the results with Wu [WSWL14], Yi [YKC+16], 3DCNN from [CSKG17], PointNet [CSKG17] and recent PointNet++[QYSG17] which uses additional normal direction features. The results are evaluated with mean IoUs(%) metric on points. Our A-SCN model achieves competitive performance for point cloud part segmentation.

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [WSWL14]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [YKC+16]	81.4	81.0	78.4	77.7	75.7	87.6	61.9	**92.0**	85.4	82.5	95.7	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN[CSKG17]	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
PointNet++[QYSG17]	**85.1**	82.4	79.0	**87.7**	77.3	**90.8**	71.8	91.0	85.9	**83.7**	95.3	**71.6**	**94.1**	81.3	58.7	**76.4**	**82.6**
PointNet[CSKG17]	83.7	83.4	78.7	82.5	74.9	89.6	**73.0**	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
A-SCN (ours)	84.6	**83.8**	**80.8**	83.5	**79.3**	90.5	69.8	**91.7**	**86.5**	82.9	**96.0**	69.2	93.8	82.5	**62.9**	74.4	80.8

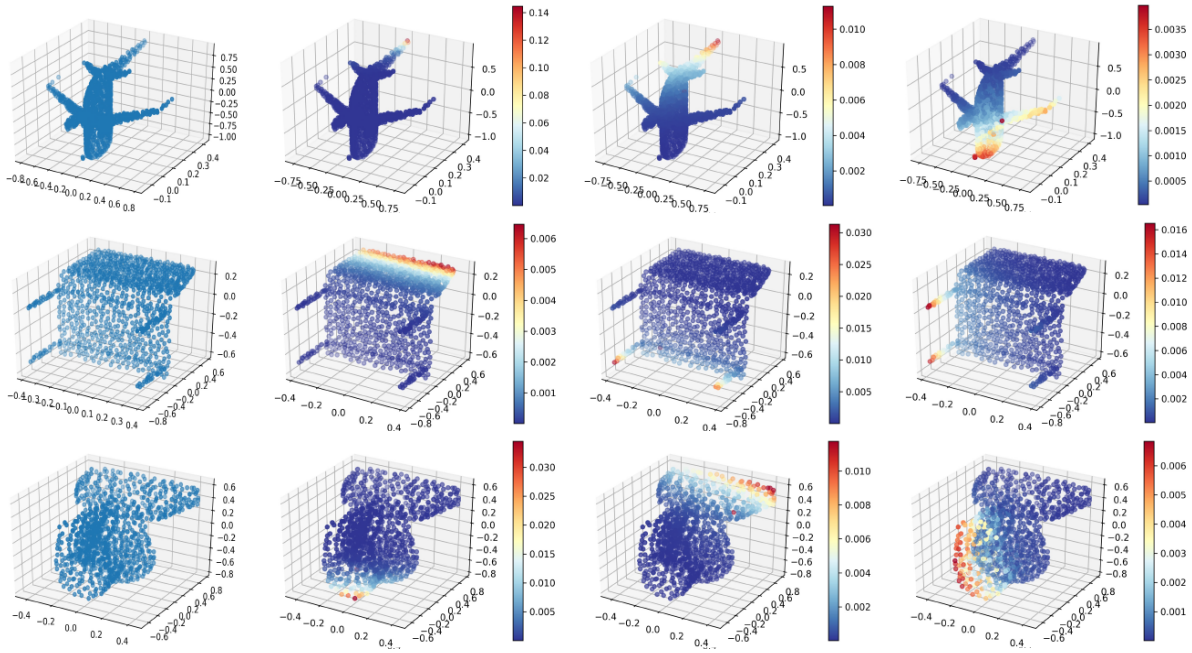


Figure 2.6: **Attention weights learned by A-SCN on three shape models: a plane, a chair and a toilet.** First column in each row shows the original point cloud. The other columns visualize learned weights for one randomly sampled reference point. Higher value indicates stronger connection to the reference point. Attention weights learned by A-SCN are diverse, sparse, and semantically meaningful and a reference point learns to attend to discriminative parts of a model.

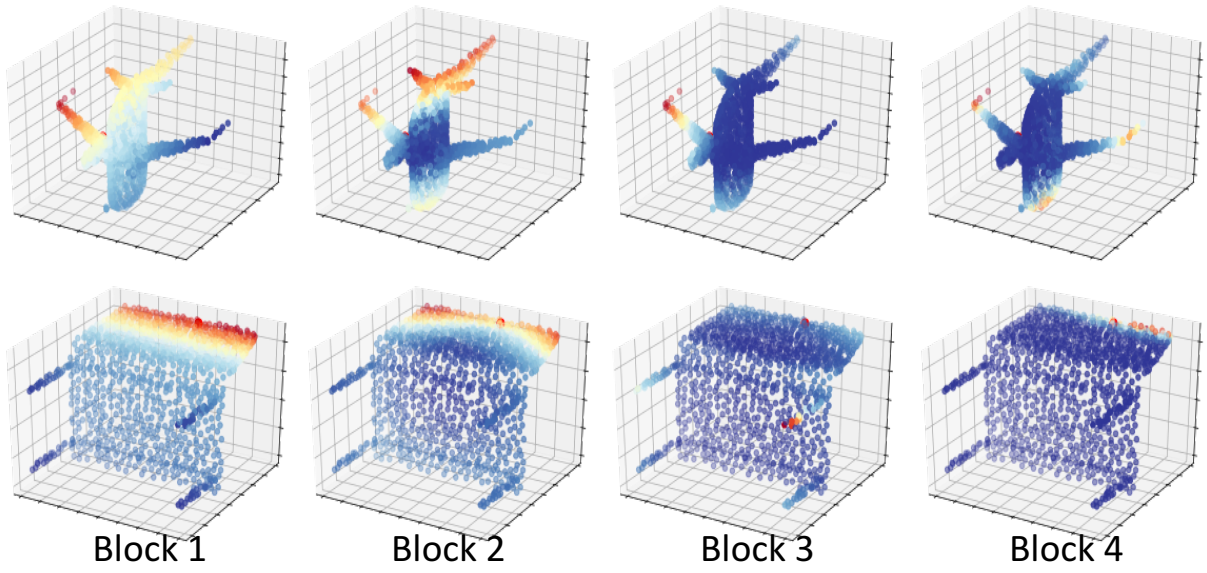


Figure 2.7: **Attention weights learned on different levels.** In A-SCN, shape information is propagated and condensed into a compact representation through a multi-level network structure. From left to right are attention weights, for a fixed reference point, learned in the first, second, third and fourth attentional shape context block. Attention becomes increasingly sparse, and focuses on smaller areas with compact representations.

2.3.3 Attentional ShapeContextNet

ModelNet40 Shape Classification. The architecture of Attentional ShapeContextNet (A-SCN) follows the general design of ShapeContextNet (SCN). In contrast to using hand-crafted shape context kernels, we adopt the self-attention module as the shape context block in the network (Figure 2.4). Q, K and V feature vectors are learned from the input using three MLPs. We use $D_K = D_Q = (32, 32, 32, 32, 64)$ and $D_V = D_{out} = (64, 64, 64, 128, 1024)$ for each block. Attention weight matrix of shape $N \times N$ is computed according to Equation 2.4. Table 2.3 summarizes the performance of A-SCN with different hyperparameters. The choices of different hyperparameters are generally aligned with those in [VSP+17] on the machine translation task. For example, the residual connection is necessary in order to learn a good model, and learning Q and K vectors independently is better than weight-sharing. Note that similar to SCN where L affinity matrices are used, we can also learn multiple attention weights in parallel for A-SCN. This is called *multi-head attention* in [VSP+17]. However, empirically we find that using multi-head

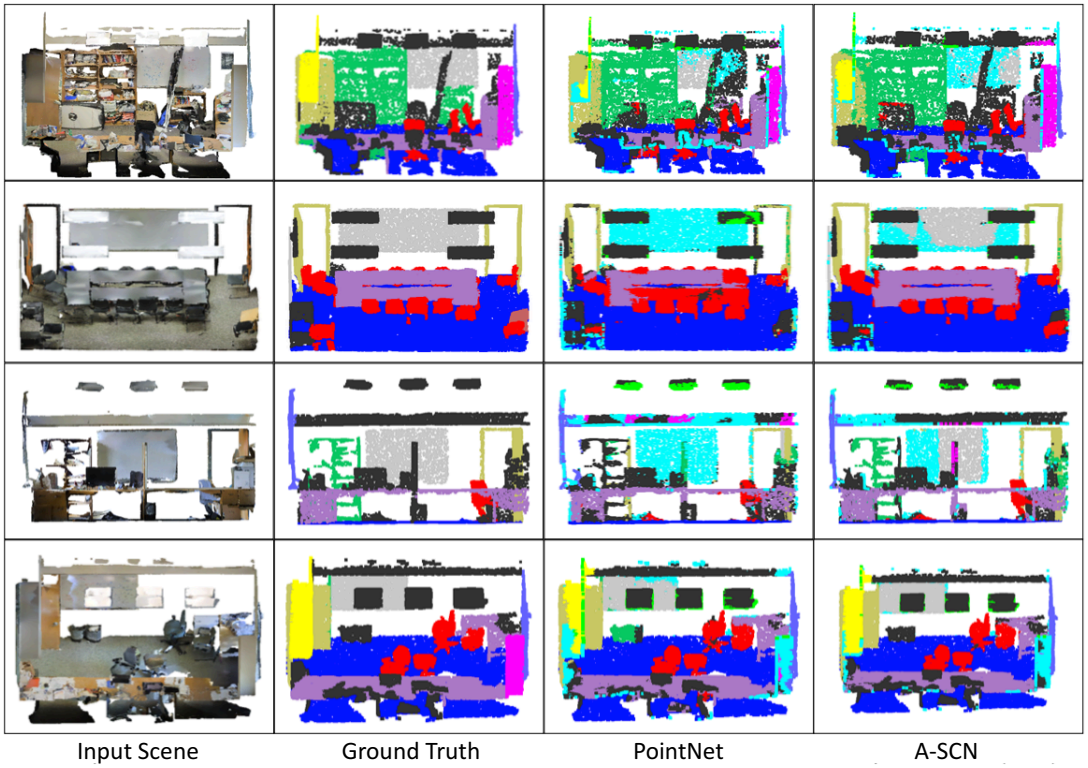


Figure 2.8: **Visualization of semantic segmentation results by A-SCN.** From left to right: original input scenes; ground truth point cloud segmentation; PointNet[CSKG17] segmentation results and Attentional ShapeContextNet (A-SCN) segmentation results. Color mappings are red: chairs, purple: tables , orange: sofa, gray: board, green: bookcase, blue: floors, violet: windows, yellow: beam, magenta: column, khaki: doors and black: clutters.

attention does not yield better performance comparing to the one-head model, and introduces additional computation overhead. Therefore, in this paper A-SCN refers to our one-head model (model (E)). A-SCN is able to achieve 89.8% overall accuracy, which is on par with SCN, but with a simpler design and fewer critical hyper-parameter to set.

In Figure 2.6 we show surprisingly diverse and semantically meaningful behavior of the learned attention weights. For a reference point, it oftentimes attends to areas far away to itself. The selected areas are usually descriptive and discriminative parts of a model, e.g. back or legs of a chair. Figure 2.7 visualizes how shape information is propagated and condensed into a compact representation in a multi-level neural network. For a fixed reference points, attention becomes increasingly sparse, and focuses on smaller areas when the level gets higher.

ShapeNet Part Segmentation. Part segmentation is a challenging task in 3D object recognition domain. Given a set of points of a 3D shape model (e.g. a plane), the part segmentation task is to label each point in the set as one of the model's part (e.g. engine, body, wing and tail). We follow the experimental setup in [CSKG17], and defines the task as a point-wise classification problem.

Our model (A-SCN) is trained and evaluated on ShapeNet part dataset following the data split from [CFG$^+$15]. ShapeNet part dataset [YKC$^+$16] consists of 16,881 object from 16 object categories, where each object category is labeled with 2-5 parts. During training, we randomly sample 1024 points from the 3D point cloud of each object and use cross-entropy as our loss function. We also followed the settings from [YKC$^+$16], which assume the object category labels are known and encoded by one-hot encoding. During testing, we test the model on all the points from each object and evaluated using point mean intersection over union (mIoU), which averages IoU across all part classes similar to [CSKG17]. Our A-SCN model outperforms PointNet over most categories, and is on par with the recent PointNet++ model which augment the input points with additional normal information. Full results for part segmentation are listed in Table 2.4.

S3DIS Semantic Segmentation. Stanford 3D indoor scene dataset[ASZ$^+$16] includes 6 large scale areas that in total have 271 indoor scenes. Each point in the scene point cloud is associated with one label in 13 categories. We follow [CSKG17] for data pre-processing, dividing the scene point cloud into small blocks. We also use the same k-fold strategy for training and testing. We randomly sample 2,048 points from each block for training and use all the points for testing. For each point, we use the XYZ coordinates, RGB value and the normalized coordinates as its input vector.

The evaluation results of our method are in Figure 2.5. By taking into account the global shape context in a hierarchical learning way, our A-SCN model achieves 52.72% in mean IoU and 81.59% in point-wise accuracy, improving the results by PointNet in both metrics. Some of

Table 2.5: **Results on scene semantic segmentation.** Mean IoU(%) on and point-wise accuracy are reported. Our Attentional ShapeContextNet model outperforms PointNet in both metrics.

	mean IoU(%)	overall accuracy (%)
PointNet [CSKG17]	47.71	78.62
A-SCN (ours)	**52.72**	**81.59**

our segmentation results are visualized in Figure 2.8.

2.4 Conclusion

To tackle the recognition problem for 3D/2D point clouds, we develop a new neural network based algorithm by adopting the concept of shape context to build our basic building block, shape context kernel. The resulting model, named as ShapeContextNet (SCN), consists of hierarchical modules that are able to represent the intrinsic property of object points by capturing and propagating both the local part and the global shape information. In addition, we propose an Attentional ShapeContextNet (A-SCN) model to automate the process for contextual region selection, feature aggregation, and feature transformation. We validated the effectiveness of our model on a number of benchmark datasets and observed encouraging results.

This chapter is based on the material as it appears in the Conference on Computer Vision and Pattern Recognition (CVPR), 2018 (Sainan Liu*, Saining Xie*, Zeyu Chen and Zhuowen Tu, "Attentional ShapeCOntextNet for Point Cloud Recognition"). The dissertation author is the co-primary investigator and author of this material.

Chapter 3

Unseen View 2D Recognition with 3D Prior

3.1 Introduction

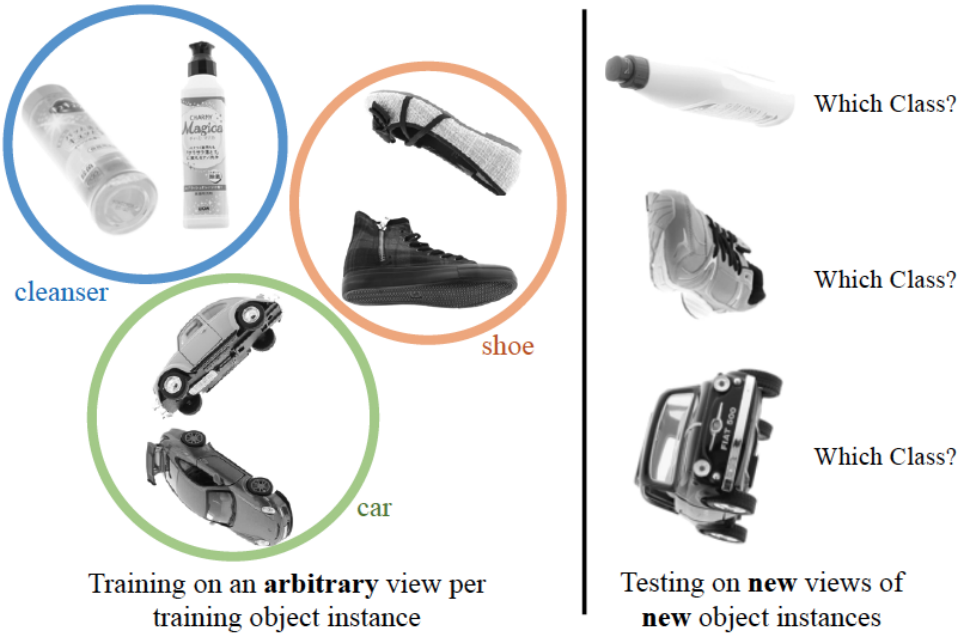


Figure 3.1: Problem illustration. Our task is to recognize an object from any view. In both training and testing, we only see 2D images without knowing the viewing angles and depth.

Inspired by the theories of object-centered and viewer-centered object recognition [Mar82, LPP95] as well as recent deep learning approaches for object recognition [SJS+18, CGKW18], in

this chapter, we propose a new algorithm: object and viewer-centered neural networks (OVCNet) for object recognition from any view. OVCNet has several attractive properties: 1) It adopts a pretrained Generalizable Reconstruction (GenRe) model [ZZZ$^+$18] to reconstruct 3D images from a single view image. We take advantage of the property of GenRe generalizing well to unseen object classes beyond the three classes ("plane", "car", and "chair") that it was trained on. Hence, we are able to infer the shape of a novel instance without additional object-specific 3D shape information. 2) OVCNet consists of three object recognition branches/modules by respectively implementing object-centric, 3D viewer-centric, and in-plane viewer-centric recognition to better perform the task. 3) We show that by adding sparse viewer-centered representations, we can further assist feature learning in the object-centered sub-module through spherical CNNs [CGKW18]. The resulting OVCNet is an integrated framework that learns viewpoint-independent and viewpoint-dependent features from an arbitrary view, and it can recognize novel views from both seen (familiar) and novel object instances.

In cognitive psychology, Marr initially proposed the definition [Mar82] of **object-centered** and **viewer-centered** representation for object recognition. Since then, further interpretations are provided in [Hay12, LPP95, LPP95, TV02] emphasizing that a viewer-centered representation captures shapes at a particular view, whereas an object-centered representation represents the intrinsic 3D shape. Inspired by these cognitive psychology findings, we ask for the following properties for an *object-centered* module in our network design: 1) *3D model based*

(*e.g.* volumetric, mesh, point-cloud or spherical maps);

2) *rotation invariant*; 3) *absent pose alignment*. Here, we characterize some of the methods [WSK$^+$15, CSKG17, SMKLM15, Kan18, CGKW18] referred in this paper in Table 3.1. Although these individual approaches in comparison have their own merits, our experiments show that each method alone does not produce satisfactory recognition result on 3D-reconstruction derived from an arbitrary view image.

To evaluate OVCNet, we use a real object grayscale multi-view dataset [Kan18], a

virtual object grayscale multi-view dataset generated from ShapeNet [CFG+15], and a natural-colored dataset (a subset of the Pascal VOC dataset [EVGW+]). We split the views of different object instances into training and testing. In training, the dataset consists of one 2D image per object instance from an unspecified viewing angle; in testing, we perform classification on two sets of images from novel viewpoints of both seen (familiar) and novel object instances, respectively. Compared to a 2D image-based object recognition system such as AlexNet [KSH12] and ResNet [HZRS16a] as well as several 3D object recognition methods [CGKW18, CSKG17, WSK+15] following a single-view reconstruction module, OVCNet shows its clear advantage in the performance observed, especially on the relatively larger dataset, gMIVO. Furthermore, we also show that our algorithm outperforms standard ResNet18 by a large margin on a subset of Pascal VOC natural images.

In comparison with standard image classification tasks such as ImageNet [DDS+09], their metrics concern with generalization to novel instances, whereas our paradigm introduces generalization to novel views as well. Our contributions are listed as follows.

- We study the problem of object recognition from any view (single-arbitrary-view training and novel-view-novel-object-instance testing) by developing an algorithm that jointly encodes object-centered and viewer-centered representations.
- We create an object and viewer-centered network (OVCNet) with three branches, each specializing in either object-centered, viewer-centered (3D), or viewer-centered (2D) learning. The proposed OVCNet consists of a combination of spherical CNNs, ResNet, and attention structures.
- Between object-centered and viewer-centered 3D branches, we develop a new network structure that enables integrated learning of both object-centered and viewer-centered representations with a communicating pathway between the two.
- We provide a new multi-view dataset generated from a subset of models of ShapeNetCoreV2

3D models.

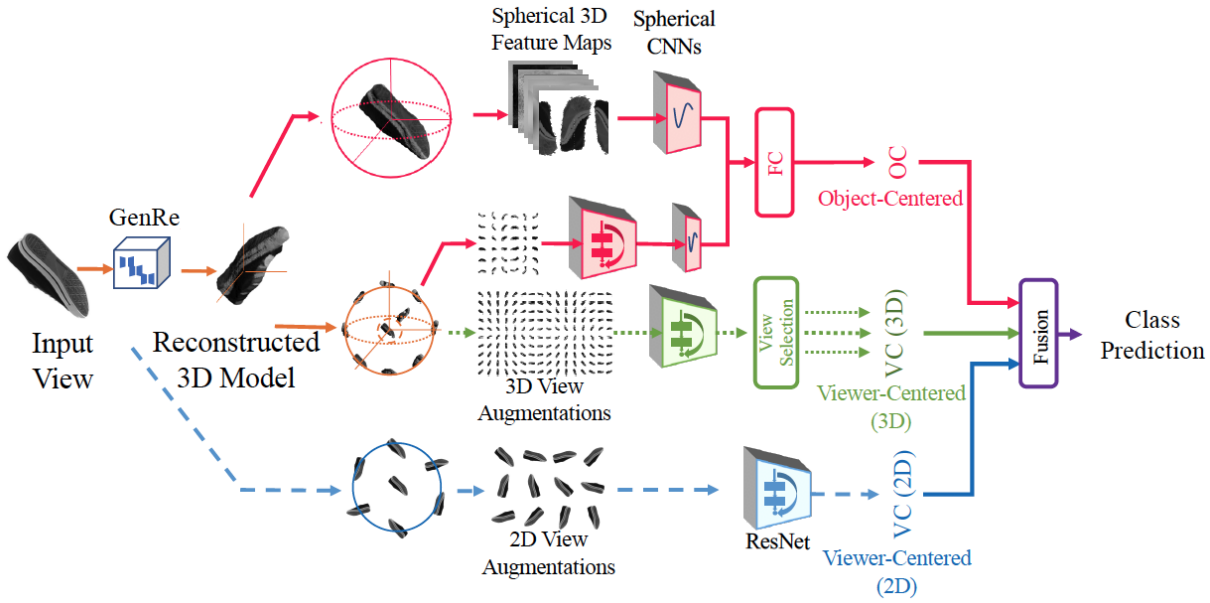


Figure 3.2: Network structure for our object and viewer-centered neural network, OVCNet. During training, each input is a 2D image of an object instance. OVCNet consists of 3 branches. For the top two branches, single-view 3D reconstruction using GenRe [ZZZ+18] is performed first. The first branch (Object-Centered) builds a representation using spherical maps [CGKW18]; the second branch (Viewer-Centered (3D)) builds a 2D CNN classifier with data augmentation using novel-view image syntheses. The third branch (Viewer-Centered (2D)) executes 2D based image classification with in-plane rotation for data augmentation. The final fusion layer provides a weighted sum of the outputs from the three branches/modules. Please see Section 3.3.5 for details about the three branches/modules, as well as the fusion layer.

3.2 Related Work

In this section, we briefly discuss the existing literature and methods related to object-centered and viewer-centered object recognition.

3D object recognition. With various 3D object datasets [CFG+15, WSK+15, XKC+16, CGKW18] being created and becoming increasingly popular, 3D object recognition [XRT12, SSFFS09, SMKLM15, Kan18, QSN+16, WSK+15, CSKG17, XLCT18, QYSG17, SJS+18,

Table 3.1: Properties as an **object-centered** representation for different methods.

Method	3D model based	Rotation-invariant	No pose alignment
3DShapeNet [WSK+15]	✓		
PointNet [CSKG17]	✓	✓	
MVCNN [SMKLM15]		✓	✓
RotationNet [Kan18]		✓	
Spherical CNNs [CGKW18]	✓	✓	✓

CGKW18] has become a highly discussed topic in computer vision. Existing systems rely on given ground-truth 3D data in the form of either volumetric shapes [WSK+15], point-cloud sets [CSKG17], spherical maps [CGKW18], or multi-view images [SSFFS09, SMKLM15, QSN+16, Kan18]. In contrast, we utilize these network structures as our recognition module following a single-view 3D reconstruction module.

2D Image-based object recognition. Viewer-centered feature learning has previously been addressed [Bas93]. Broadly speaking, the recent common practice of data-augmentation can be considered viewer-centered feature learning where no new views are generated since the augmentation is mainly implemented in the 2D image plane.

Hybrid 2D and 3D object recognition. SPLATNet [SJS+18] is a hybrid system that integrates both 2D and 3D features for object classification and segmentation and is closely related to ours. However, SplatNet takes two modalities of inputs: a point-cloud based 3D shape and 2D multi-view images. Hence the scope of SplatNet is very different from ours.

Data-augmentation for transfer learning. There have been recent works in transfer learning [SGMN13, RL17, LNH14, DKNV17, LWD+18] where data-augmentation is performed subject to certain domain adaption and regularization. These approaches address a fairly different problem compared to ours. We focus on the basic problem for 3D single image classification instead of a multi-task prediction problem.

Single-view 3D reconstruction. In the field of single-view 3D reconstruction, an object-centered network outputs 3D information in a canonical view of the object. In contrast, a viewer-

centered network's 3D output is relative to the input view [SFH18, TRR+19]. This definition is significantly different from what we define previously for recognition tasks. Nonetheless, for better reconstruction, Shin *et al.* and Tatarchenko *et al.* have shown that using 3D-supervision in a viewer-centered coordinate system tends to generalize better against unseen classes. Better generalization for unseen categories allows us to acquire 3D shape priors for new instances in an image without any 3D shape information during training. We adopt the state-of-the-art method for unseen class reconstruction, GenRe [ZZZ+18], to reconstruct 3D shape from a 2D single image, but GenRe itself does not perform image recognition.

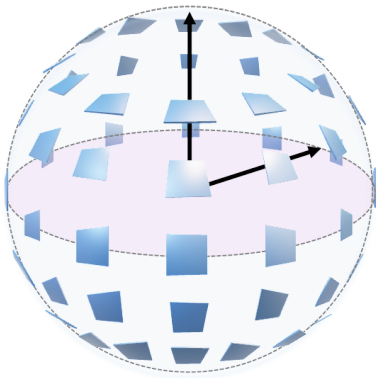
Spherical CNNs. We build our chosen object-centered representation based on spherical CNNs [CGKW18], which is an effective and efficient way to obtain 3D shape representation for the 3D object classification tasks. Spherical CNNs themselves do not perform object recognition from any view, and a 3D input is required to generate the spherical map that spherical CNNs need.

To summarize, we focus on a challenging problem setting for object recognition from any view using object and viewer -centered representations.

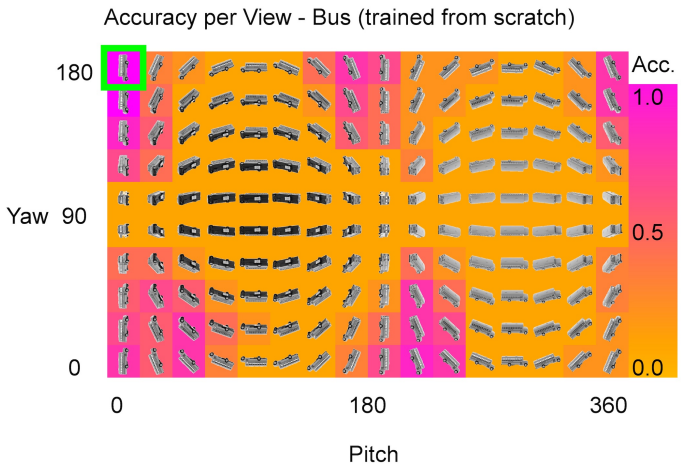
3.3 Our Approach

3.3.1 Problem Formulation

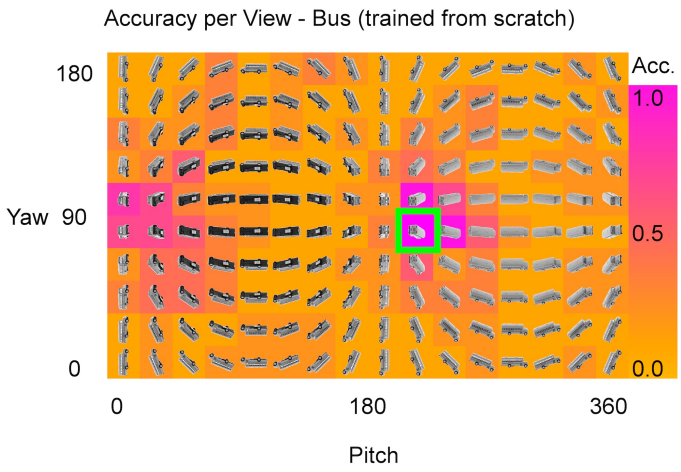
In this section, we focus on the any view object classification task. During training, the input is an arbitrary single view per training object instance, and the output is the ground truth class label. Every object instance is seen only once. We evaluate the effectiveness of OVCNet in two aspects: 1) SeenInstances: the ability to recognize novel views of seen (familiar) object instances (instances that are used in training) and 2) NovelInstances: the ability to recognize arbitrary views of novel/unseen object instances (instances absent from the training set). We present results from two experiments corresponding to these two aspects.



(a)



(b) trained on view 1



(c) trained on view 90

Figure 3.3: (a) is an example of viewpoints used for generating viewpoint dependent images for the Viewer-Centered (3D) module (Section 3.3.4) similar to [Kan18]. (b) and (c) show the classification accuracies across all viewpoints for a ResNet18 model trained only on view 1 (b) and view 90 (c) (highlighted) of the objects, respectively, on the MIRO dataset [Kan18]. Without seeing other views, classic 2D CNNs have unsatisfactory performances on novel views.

3.3.2 Single-view shape prior

Given a single view of an object instance, we first use a state-of-the-art algorithm, GenRe [ZZZ+18], to generate 3D object reconstruction from a 2D image. GenRe separates reconstruction into three sub-tasks: depth estimation, spherical map inpainting, and voxel refinement. The separation of these tasks enables reasonable reconstruction for unseen objects/classes. Therefore, no additional object-specific information is needed. The pretrained GenRe model is only trained on three object classes ("plane", "car", and "chair") for reconstruction, but GenRe has shown great potential when it is evaluated on a wide variety of unseen object categories [ZZZ+18]. In our classification task on the gMIVO dataset, we include plane, car, chair, as well as other object classes such as lamp, pistol, motorbike, knife, laptop, guitar, and table. We adopt the trained GenRe model [ZZZ+18] directly to perform 3D reconstruction for a 2D image and add texture information to the final 3D model. We sample the texture information from the seen side with the nearest neighbor search algorithm using a k-d tree. This approach may result in different texture patterns due to different vertex ordering. A better texture filling approach should be explored in future studies.

3.3.3 Object-centered representation (OC module)

We utilize existing 3D recognition network structures as our classification module following GenRe's 3D shape estimation. We evaluate all three 3D shape-based recognition networks in Table 3.1: 3D CNNs, PointNet [CSKG17], and spherical CNNs [CGKW18], respectively. 3D CNNs is a 3D convolutional network inspired by 3DShapeNet [WSK+15] and built on top of [Lin]. Among them, spherical CNNs match the most with our object-centered definition for the following reasons.

First, spherical CNNs model is a 3D shape-based method. Object classification is carried out based on distance spherical maps along with cosine and sine of surface signals from 3D

objects and their convex hulls. With spherical information of 3D models as input, the results of spherical CNNs on ShapeNet SHREC17 [CFG+15] are close to the state-of-the-art [CGKW18]. One can generate a spherical distance map by shooting a ray from the surface of a sphere (with a fixed radius) to the center of the object. The distance between the sphere surface and the object surface becomes the distance value captured by the spherical distance map [CGKW18]. *Second*, spherical CNNs use convolutions directly in the spherical harmonic domain, which keeps 3D rotation-equivariance of the spherical signals. See discussions about an empirical support for rotation-invariance in [CGKW18]. More discussion on its rotation-invariant capability is provided in Appendix A.2. *Third*, the network does not require any pose alignment.

In our overall model, we refer to the object-centered module branch with spherical CNNs as the **OC^b module**, where the superscript *b* indicates that it is a base module.

3.3.4 Viewer-centered representation (VC module)

For viewer-centered representations, different modules with two different inputs are used: 1) the original view **VC (2D) module**; 2) views re-projected using 3D viewpoint augmentation from the 3D output of the GenRe **VC (3D) module**. For both tasks, we find that ResNet18 works well as a 2D image classifier compared to other classic convolutional neural networks. To select augmented views, we implement three options for the view selection layer (discussed in detail in Section 3.3.5).

VC (2D) module. This module uses 2D augmentation with in-plane rotation. We evaluate ResNet18 with different angles of rotation augmentation, including intervals of 90, 30, 10, 5, and 1 degrees for gMIRO. We observe that the evaluation accuracy stops increasing as we provide denser angle augmentations. Rotation ablation studies (Appendix A.1) show that ResNet18's accuracy plateaus when we augment the input view with 2D in-plane rotations at 30-degree intervals for the gMIRO dataset. In contrast, for gMIVO, the network performance plateaus with augmentations of 90-degree intervals. We use these numbers in our later experiments. If

trained under identical views, ResNet18, as shown in Figure 3.3.b and c, experiences difficulties recognizing images from new angles for the same set of objects. We refer to this effect as " mental rotation".

VC (3D) module. This module uses 2D augmentation from 3D viewpoints. We augment images with 10 evenly divided elevation angles and 16 evenly divided azimuth angles, yielding 160 views per object. The viewpoint augmentation setting is shown in Figure 3.3.a [Kan18]. The viewpoint layout imitates the organization of object views in the dataset, starting from the input view. Additionally, we add in-plane (2D) rotation augmentations in 90-degree intervals to each augmented viewpoint.

For the VC (3D) module, we explore three types of view selection methods: 1) the nearest neighbor approach where the network only uses the augmented image that is closest to the input viewpoint for testing; 2) a simple selection layer where the network learns a set of weights for all augmented views; 3) an attention layer where the network learns a set of attention weights based on the input information. Option 1 is the most suitable for a dataset that has limited training views, such as gMIRO, and is the most efficient in terms of runtime. For options 2 and 3, we further divide the training views into a sub-training set1 and set2. We first use set1 for training ResNet18 and then use the set2 to train the selection network. We observe an improvement in average accuracy using a view selection network compared to a simple ensemble of all augmented views. However, given the limitation of the 3D reconstruction and size of the dataset, for the gMIRO dataset, using the input viewpoint alone outperforms the other options.

Other augmentations are also considered. We include 20 views taken from the 20 vertices of a dodecahedron around the object [SMKLM15, Kan18] for a GenRe [ZZZ$^+$18] + multi-view baseline. We also include 36 viewpoints from a sampling grid of spherical maps with a bandwidth of 3 [CGKW18] for a viewer-centered assisted object-centered module, OC (Section 3.3.5).

For the multi-view baseline, we include GenRe + multi-view CNN[SMKLM15] (MVCNN) and GenRe + RotationNet [Kan18]. A 20-view version of MVCNN is used due to memory con-

straint. The best performing backbones are VGG for MVCNN and ResNet18 for RotationNet. The results are encouraging for GenRe + MVCNN. However, MVCNN uses pretrained weights and requires 20-view augmentation during test. In contrast, we train our model with a single view and from scratch to avoid prior knowledge of unseen instances learned from the pretrained dataset.

3.3.5 Fused representation (OVCNet)

In summary, our overall network (Figure 3.2) includes 3 branches: OC^b branch ($GenRe^{text}$ + spherical CNNs [CGKW18]), VC (3D) branch ($GenRe^{text}$ + ResNet18 [HZRS16a] + view selection), and VC (2D) branch (ResNet18).

To fuse the OC^b base module with the VC (3D) module, we create an OC module (Figure 3.2). In this module, in addition to the 160-view set, we use the information from 36 augmented views to reduce the number of views needed for training. We then organize the learned ResNet features into a grid and pass them into an ancillary spherical CNNs with an input bandwidth of 3. This new branch is then trained with the original OC^b base module fused by a fully connected layer as the final OC module. The result of gMIRO is shown in Table 3.3.

To fuse the output of OC and VC modules, we experiment with 3 options. The first option is to train a fully connected fusion layer with or without each module frozen. The second option is to learn an attention layer to fuse the three results. The third option is to use a set of weights found through a grid search using a validation set. Our experiment has shown that the third option works the best for the gMIRO dataset. Two reasons may contribute to this: 1) different branches have different learning rates due to diverse input and module modalities; 2) Even with the three branches frozen, the simpler fusion method adapts better when we have limited training information. We find the learned weights from option 3 are stable, *e.g.*, around 0.2, 0.3, and 0.5 for combining OC module, VC (3D) module, and VC (2D) module on both gMIRO and gMIVO datasets.

Please see Appendix A.4 for details on runtime analysis.

3.4 Experiments

3.4.1 Baselines

Next, we report the results of various baseline classifiers as well as those by our OVCNet.

Traditional image classification networks. We learn 2D image classification using convolutional neural networks including AlexNet [KSH12], ResNet18 [HZRS16a], and ResNet152 [HZRS16a] directly on the input views. For AlexNet and ResNet18, the batch size for training is 96. For ResNet152, a batch size of 32 is used due to memory constraint. We start with an initial learning rate of 0.01 and decay by 10 every 30 epochs. ResNet18 seems to generalize better and has more efficient memory usage.

3D shape-based classification networks. We convert the reconstructed 3D object from GenRe to voxels ($30 \times 30 \times 30$ or $128 \times 128 \times 128$), point sets (2500 point samples), and distance spherical maps in order to run 3D CNNs [Lin], PointNet [CSKG17], and spherical CNNs [CGKW18], respectively, without texture information.

Re-projected viewer-centered classification networks. For re-projections from GenRe's output, as a baseline for VC (3D) module, we evaluate ResNet18 with a different number of view augmentations. Although our algorithm only uses a single view during testing in the overall model, we also show our results with 20 views during evaluation with GenRe [ZZZ$^+$18] + RotationNet [Kan18] and GenRe + MVCNN [SMKLM15] as a multi-view module baseline.

Object and viewer-centered network. Since OVCNet combines three modules, for a fair comparison, we include two ensemble strategies of three VC (2D) modules and report the results in Table 3.2 (ResNet18$^{\text{rot30/90}}$ Ensemble *I*, *II*). To compare with the ensemble results, we randomly select six VC (2D) modules and report the average over two sets of ensemble results. For a fair comparison with OVCNet, we randomly select one VC (2D) module from

the ensemble set to combine with our OC module and VC (3D) module. Ensemble *I* uses three equally weighted random models of the same type. Ensemble *II* trains additional fusing weights for the three random models.

Table 3.2: **Results summary.** ResNet18*: a standard 2D image data augmentation [KSH12]. ResNet18$^{\text{rot[d]}}$: 2D in-plane rotation augmentation with multiples of d degree rotation. GenRe^{tex}: texture is used for 3D viewpoint augmentation. RotationNet^{pre} and MVCNN^{pre}: using pre-trained weights. Ensemble *I* uses an equally weighted ensemble of three models. Ensemble *II* includes learned fusing weights for the three random models. Two repeats for OVCNet and the ensembles.The proposed OVCNet performs the best here.

	accuracy overall (%) SeenInstances	accuracy overall (%) NovelInstances
gMIRO		
AlexNet [KSH12]	24.61 $\pm$ 3.02	27.40 $\pm$ 2.13
ResNet152 [HZRS16a]	45.97 $\pm$ 1.08	43.68 $\pm$ 1.91
ResNet18 [HZRS16a]	51.34 $\pm$ 0.52	44.04 $\pm$ 1.31
ResNet18*	45.08 $\pm$ 0.98	38.70 $\pm$ 2.09
ResNet18$^{\text{r30}}$(VC (2D))	68.34 $\pm$ 1.57	53.27 $\pm$ 0.89
ResNet18$^{\text{rot30}}$ (Ensemble *I*)	70.56 $\pm$ 0.56	54.91 $\pm$ 1.85
ResNet18$^{\text{rot30}}$ (Ensemble *II*)	70.91 $\pm$ 0.34	55.74 $\pm$ 2.52
GenRe [ZZZ^{+}18] + PointNet [CSKG17]	27.33 $\pm$ 0.48	27.67 $\pm$ 0.80
GenRe + 3D CNNs [Lin]	30.26 $\pm$ 0.62	30.01 $\pm$ 0.75
GenRe^{tex} + RotationNet^{pre} [Kan18]	46.55 $\pm$ 3.97	46.44 $\pm$ 4.54
GenRe^{tex} + MVCNN^{pre} [SMKLM15]	58.68 $\pm$ 0.59	54.56 $\pm$ 0.41
OVCNet (ours)	**73.24** $\pm$ 0.08	**65.85** $\pm$ 0.14
gMIVO (ShapeNetCoreV2 subset)		
ResNet18$^{\text{rot90}}$ (VC (2D))	64.40 $\pm$ 0.45	64.86 $\pm$ 0.43
ResNet18$^{\text{rot90}}$ (Ensemble *I*)	65.70 $\pm$ 0.25	66.25 $\pm$ 0.59
ResNet18$^{\text{rot90}}$ (Ensemble *II*)	65.73 $\pm$ 0.18	66.27 $\pm$ 0.44
OVCNet (ours)	**79.24** $\pm$ 0.12	**75.03** $\pm$ 0.30

3.4.2 Datasets

We adopt the following three datasets: a grayscale version of the MIRO dataset [Kan18] (**gMIRO**), our new dataset, grayscale multi-view images of virtual objects (**gMIVO**), and natural-colored images from Pascal VOC [EVGW^{+}].

gMIRO. We use preprocessed grayscale images from the MIRO dataset [Kan18] (gMIRO) as our primary dataset for ablation studies. This dataset contains 12 classes with 10 object instances for each class. For each object, there are 160 views (10 elevations $\times$ 16 azimuth angles) from real objects with empty backgrounds. We randomly select 80% of the instances as familiar object instances. For each object, we randomly select an arbitrary single view to use in the training set ($12\ classes \times 10\ objects \times 80\%\ seen\ split \times 1\ view = 96\ images$). We use the remaining views of the familiar instances as the first test set that evaluates how well the model generalizes towards unseen views of seen object instances (SeenInstances). The final test is done utilizing all the views from the remaining 20% new instances, where we can evaluate the generalization towards views from all 160 angles of the unseen object instances (NovelInstances).

gMIVO. gMIVO is a larger dataset with a similar setup as gMIRO. A subset of ShapeNetCore v2 is selected to generate this dataset. We do not use ModelNet [WSK$^+$15] directly for this paper because an aligned ModelNet40 was not available at the time the project first started. Additionally, most of the objects are lacking material and texture information. ShapeNetCore v2 includes materials and textures and all objects are aligned [CFG$^+$15]. We select a subset of the objects from ShapeNetCore v2 by referring to the 10 classes with the highest frequency from DensePoint [CN19] (which uses ShapeNetCore v2 objects with good material and texture information) and take 160 views of each object. This new dataset contains ten classes where each class has 110 objects. For each object, 160 views are generated using similar viewpoints from MIRO [Kan18] as shown in Figure 3.3.a. Our rendering tool is built on top of the Stanford ShapeNet renderer. During training, we randomly select 80% of the objects for every class as the familiar objects. The two test sets, SeenInstances and NovelInstances, are set up similarly to gMIRO.

Pascal VOC. We use a subset of Pascal VOC images [EVGW$^+$] to evaluate the capability of OVCNet with real color images with background. For training, to use GenRe, we obtain the masks for each object from [MCL$^+$14]. For testing, an object mask is first obtained through a

foreground segmentation algorithm using [HCH+19]. We choose images of aeroplane, bicycle, car, and motorbike because there are fewer occlusions in those images, which allows adequate 3D reconstructions. We randomly select 20% of the images from each category for training and the remaining for testing.

We start with grayscale images for gMIRO and gMIVO to illustrate the fundamental idea of OVCNet. We then experiment with colored inputs for MIRO and PASCAL images. Please see Section 3.4.4 for more details.

3.4.3 Metrics

For both the gMIRO and gMIVO datasets, we partition the data into familiar and novel instances with an 80%/20% train-test split. If not otherwise specified, we conduct three repeats for each experiment and averge the results. We report the overall class accuracy (the mean and standard deviation) for unseen views with seen objects (SeenInstances) and all views with unseen objects (NovelInstances).

3.4.4 Results and Discussions

Object-centered feature learning. For the object-centered branch, we compare the results of different representations of the 3D reconstruction using GenRe + 3D CNNs, GenRe + PointNet, and GenRe + spherical CNNs in Table 3.3. We find that the performance for GenRe + 3D CNNs increases as the voxel resolution increases; however, the network size increases as well. For GenRe + spherical CNNs, the performance increases as bandwidth increases and plateaus at $bandwidth = 112$ for gMIRO. Overall, our OC branch outperforms other combinations in terms of overall accuracy for both SeenInstances and NovelInstances with comparable network size. Additionally, the OC module that further integrates information learned from the VC (3D) branch (bw=3 sgrid) can give OC^b baseline module an extra 10% boost on the gMIRO dataset.

Table 3.3: **Ablation study for object-centered network structure (OC) on gMIRO**. GenRe^{tex} + spherical CNNs [CGKW18] (with additional approximated texture spherical map information) is chosen as our OC^b module in our OVCNet due to its relative performance advantage. *vx* indicates voxel representation, *pt* indicates point cloud representation, and *bw* indicates the bandwidth for spherical signals. The final OC model with an ancillary spherical pathway integrating the information learned from the VC (3D) module (bw=3 sgrid) performs the best.

Networks	accuracy overall (%)	
	SeenInstances	NovelInstances
GenRe + 3D CNNs [Lin] ($30 \times 30 \times 30$vx)	20.94 ± 0.41	21.74 ± 0.42
GenRe + 3D CNNs ($128 \times 128 \times 128$vx)	30.26 ± 0.62	30.01 ± 0.75
GenRe + PointNet [CSKG17] (2500pt)	27.33 ± 0.48	27.67 ± 0.80
GenRe + spherical CNNs [CGKW18] (bw=60)	40.79 ± 1.21	41.50 ± 0.44
GenRe + spherical CNNs (bw=112)	42.43 ± 1.24	40.80 ± 0.51
GenRe + spherical CNNs (bw=128)	40.94 ± 1.88	41.23 ± 0.77
GenRe^{tex} + spherical CNNs (bw=112) (OC^b)	44.62 ± 0.58	44.65 ± 0.53
OC branch	**54.62** ± 0.73	**54.21** ± 0.54

Viewer-centered feature learning. For viewer-centered network structures with re-projected 2D images (VC (3D) module), we compare different 3D viewpoint augmentations during training, shown in Table 3.4. For GenRe + ResNet18, the performance increases as the number of training viewpoints increases. Once we introduce texture in the re-projection, both GenRe + MVCNN and VC (3D) outperform other methods. GenRe + MVCNN uses all 20 different viewpoints for testing. In contrast, VC (3D) only uses one viewpoint during the evaluation. Hence, it is more efficient than GenRe + MVCNN.

We also experiment with the attention structure as our view-selection layer (Not shown in tables). Compared to a simple ensemble of all 160 views at test time, we do notice a performance gain from the attention view selection layer in the Pascal dataset. This result suggests that a more complex view selection module during inference may boost the performance with increased training data.

For viewer-centered network structures with original 2D images (VC (2D) module), we conduct an ablation study on 2D rotation augmentation. In Appendix A.1, we show that, for gMIRO, the performance of ResNet18 plateaus with rotations of 30-degree intervals (12

augmented images per input). For gMIVO, we find that the performance of ResNet18 plateaus with rotations of 90-degree intervals (4 augmented images per input). These results may indicate that with increasing number of training instances, random viewing angles of similar instances increase. Hence, less in-plane rotation is needed to boost performance.

Table 3.4: **Ablation study for viewer-centered network structures with gMIRO** by using different types of data augmentations. **3D-aug**: the number of re-projected images used during training. Section 3.3.4 offers viewpoint details. GenRe^{tex} + MVCNN^{pre} and GenRe^{tex} + RotationNet^{pre} use fine-tuned weights with pre-trained models and 20 views for evaluation, whereas other methods only use single view. The final VC (3D) model with GenRe^{tex} and ResNet18 trained from scratch performs the best.

	3D-aug 1/160/640	accuracy overall (%) SeenInstances	accuracy overall (%) NovelInstances
GenRe + ResNet18	1	32.49 ± 0.68	32.95 ± 0.93
GenRe + ResNet18	160	45.15 ± 0.46	40.20 ± 0.51
GenRe + ResNet18	640	51.24 ± 0.23	47.57 ± 0.55
GenRe^{tex} + RotationNet^{pre} [Kan18]	20	46.55 ± 3.97	46.44 ± 4.54
GenRe^{tex} + MVCNN^{pre} [SMKLM15]	20	58.68 ± 0.59	54.56 ± 0.41
scratch VC (3D) (ours)	640	**65.70** ± 0.44	**58.27** ± 0.04

Object and viewer-centered network. Finally, we combine the results from both object (OC) and viewer -centered modules (VCs) for both gMIRO and gMIVO datasets. Through a simple grid search on the validation sets, the fusion layer outputs a weighted sum of probabilities from OC, VC (3D), and VC (2D) branches. The results are shown in Table 3.5. Our results show that the three models are complementary to each other for both datasets.

The advantage of OVCNet over the ensemble of ResNet18s appears to be more significant for gMIVO. The test accuracy improves by $\sim 13.5\%$ for unseen views of familiar object instances and $\sim 9\%$ for novel object instances in Table 3.2. It suggests that training with more arbitrary views of instances from the same category helps with classifying views from other viewpoints. Interestingly, for gMIVO in Table 3.5, the test accuracy of the VC (3D) branch alone is already higher than that of VC (2D); this further validates the importance of inferring 3D reconstruction through which our 3D view augmentation is realized.

We also evaluate the average class accuracy for OVCNet and the corresponding ensemble

baseline (not shown in tables). For gMIVO, for all ten classes, the SeenInstances (other views from familiar instances) accuracy is raised by 13.41% from 65.89% to 79.36%. The NovelInstances (all views from novel instances) accuracy is raised by 8.68% from 66.65% to 75.33% (we list these numbers here in the text directly).

Table 3.5: **Ablation study over different model integrations**. gMIRO uses an OC module (see Section 3.3.5), whereas gMIVO uses an OC^b module (see Section 3.3.3). For the VC (3D) branch (see Section 3.3.4), gMIRO uses textured reconstructed 3D models from GenRe to generate 640 3D viewpoint augmentations per input view, whereas gMIVO uses 160 viewpoints. For the VC (2D) branch (see Section 3.3.4), gMIRO uses 30-degree intervals whereas gMIVO uses 90-degree intervals. The three modules are shown to be complementary to each other on both datasets.

Experiments	OC	VC (3D)	VC (2D)	SeenInstances accuracy (%)	NovelInstances accuracy (%)
gMIRO					
(1)	✓			52.65	53.02
(2)		✓		65.70	58.31
(3)			✓	69.74	54.11
(4)	✓	✓		67.24	61.48
(5)		✓	✓	72.47	62.99
(6)	✓		✓	72.04	58.57
OVCNet	✓	✓	✓	**73.25**	**65.99**
gMIVO (ShapeNetCoreV2 subset)					
(1)	✓			52.83	50.49
(2)		✓		77.00	70.53
(3)			✓	63.66	64.50
(4)	✓	✓		77.60	71.23
(5)		✓	✓	77.71	74.50
(6)	✓		✓	67.83	67.63
OVCNet	✓	✓	✓	**79.36**	**75.33**

Given that we use a pretrained GenRe model that is trained on three classes from ShapeNet and our gMIVO dataset is also a subset of ShapeNet, we additionally test on gMIVO after removing the three classes that are overlapping between the two datasets. Our model shows a slightly greater improvement compared to using all ten classes. The final OVCNet model outperforms the ensemble of VC (2D) by 14.45% for unseen views of seen objects and 9.3% for

Table 3.6: **Ablation study with different train-test split percentages**. Each column corresponds to a different train-test split for the gMIRO dataset. OVCNet* uses a less optimal configuration compared to the OVCNet used in Table 3.2. Under varying training sizes, the trend of OVCNet w.r.t. VC (2D) is consistent as in Table 3.5 and Table 3.2.

	$80\% - 20\%$	$50\% - 50\%$	$20\% - 80\%$
	test accuracy for SeenInstances (%)		
VC (2D)	68.34 ± 1.57	64.42 ± 0.43	64.53 ± 0.84
OVCNet*	69.95 ± 0.35	67.24 ± 0.08	69.13 ± 0.75
	test accuracy for NovelInstances (%)		
VC (2D)	53.27 ± 0.89	47.36 ± 0.83	36.66 ± 0.54
OVCNet*	59.57 ± 0.28	50.99 ± 0.31	42.09 ± 0.06

all views of unseen objects. We demonstrate that the effectiveness of OVCNet does not depend on the training classes from GenRe. The improvement may be due to the removed classes being harder to classify.

Ablation study for train-test split percentages. To evaluate our model's performance on the varying training data size, we experiment with two more train-test splits. In addition to the original split (80% familiar instances vs. 20% new instances), we also test $50\%/50\%$ and $20\%/80\%$ train-test splits. Table 3.6 shows the means and standard deviations for the test accuracies on seen instances and novel instances under multiple repeats. As the number of familiar instances decreases, the overall classification accuracy also declines, which is typical when trained on fewer data. However, we see a similar improvement as that in Table 3.5 and Table 3.2 for OVCNet w.r.t. VC (2D) module. These experiments are tested with an earlier version of OVCNet for gMIRO that uses a less optimal configuration than what is used in Table 3.5 and Table 3.2.

Color and Natural Images.

Our experiments in Table 3.3 show the results of combining approximated texture information with the grayscale input. In a similar spirit, we also provide results for colored input as follows (not shown in tables). We use color images from MIRO to train the VC (2D) module (ResNet18 with in-plane rotations) as a baseline; we keep OC and VC (3D) the same since they

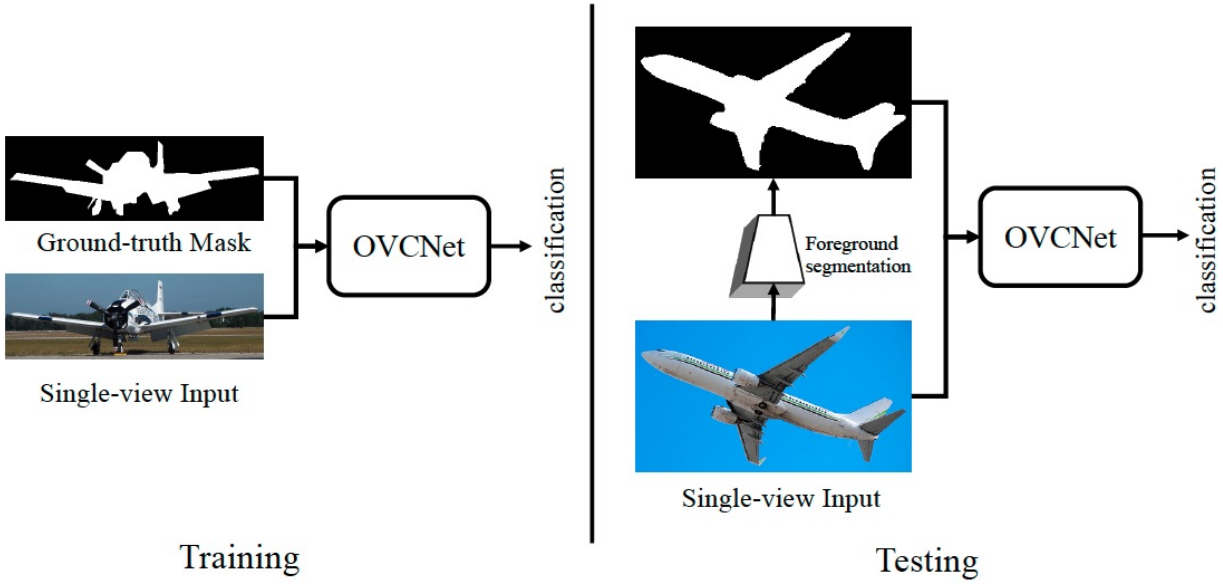


Figure 3.4: OVCNet algorithm pipeline for the PASCAL experiment.

mostly concern with shape. Nevertheless, our results show that, for gMIRO, OC, and VC (3D) modules still provide a consistent boost to the VC (2D) baseline trained with color images from MIRO. The accuracy improves from 73.23% to **75.64**% for SeenInstances (unseen views from familiar instances) and from 54.53% to **67.66**% for NovelInstances (unseen instances). This improvement validates the benefit of having an object- and viewer-centered representation for colored images as well.

Table 3.7: Test accuracy for Pascal VOC subset images for the aeroplane, bicycle, car, and motorbike classes.

	test accuracy (%)
OC^b (bw=112)	80.08
VC (3D) (160)	82.35
VC (2D)	72.84
VC (2D) (*Ensemble I*)	75.49
VC (2D) (*Ensemble II*)	75.91
OVCNet	**85.24**

An evaluation of natural-colored images with a background (a subset of Pascal VOC) also shows encouraging results. Experimental results are reported in Table 3.7. We see a 10%

improvement over the baseline. Random rotation does not improve the performance for VC (2D) here.

3.5 Conclusion

We have developed a new algorithm for any view object recognition that is inspired by the object and viewer-centered recognition theories. The resulting OVCNet is an integrated framework that learns viewpoint-independent and viewpoint-dependent features for an image from an unknown view, and it can be used to recognize novel instances from novel views. We show a clear advantage of OVCNet over the object-centered and viewer-centered baselines in Table 3.2 and 3.5. We also report results on natural-colored images in Table 3.7.

This chapter is based on the material as it appears in the Conference on Computer Vision and Pattern Recognition (CVPR), 2020 (Sainan Liu, Vincent Nguyen, Issac Rehg, Zhuowen Tu) The dissertation author is the primary investigator and author of this material.

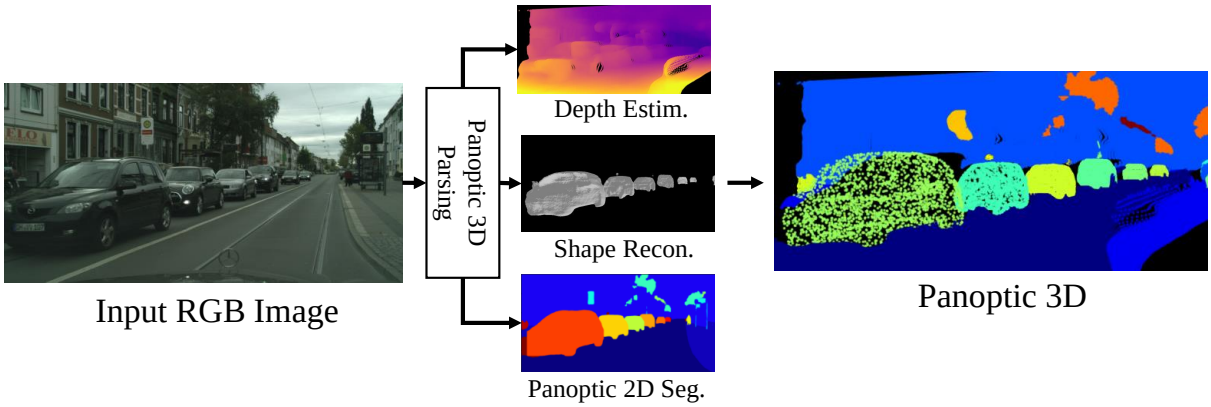
Chapter 4

Panoptic 3D Parsing

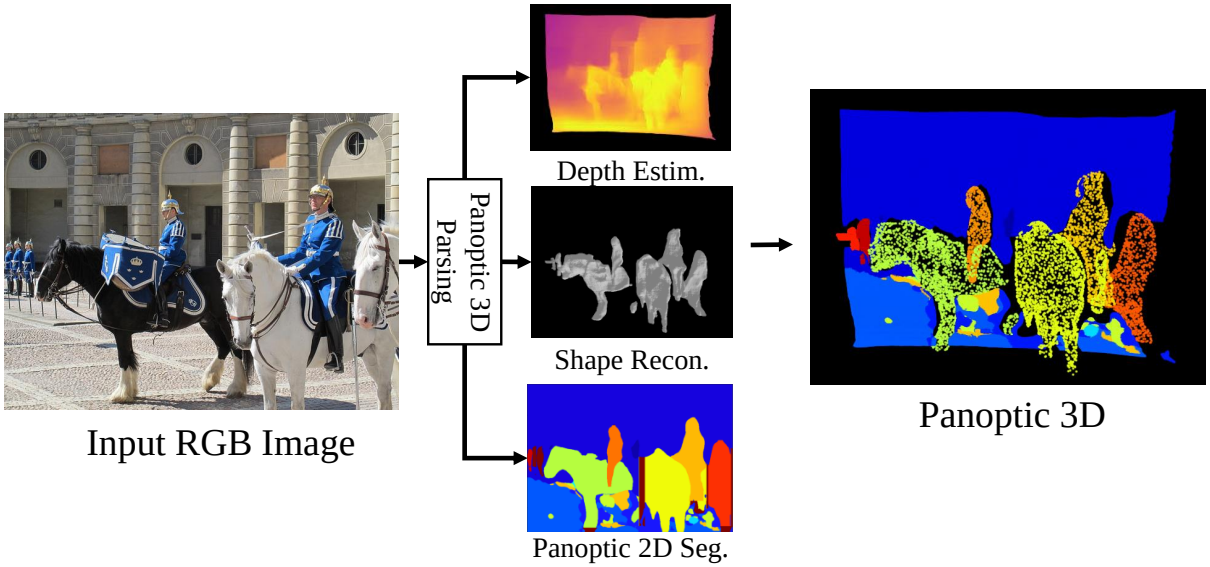
4.1 Introduction

This chapter focuses on a scientifically important but practically very challenging setting: single-view holistic 3D segmentation and shapes/layout reconstruction. While there are existing works for 3D shape reconstruction [WSK+15, ZZZ+18, WZL+18, GFK+18, KBJM18, XWC+19, CNH+20] and 3D layout estimation [TGF+18, ZCSH18], none have tackled the joint task of single-view 3D scene dense-segmentation/object-detection/shape-reconstruction/layout-estimation for the natural (both indoor and outdoor) scenes. The following motivations for panoptic segmentation, layout and depth reconstruction, and object instance reconstruction contribute to the overall strategy for 3D scene understanding proposed by our Panoptic3D paradigm (shown in Figure 4.2 and Figure 4.3). The contributions of our work are in three broad areas of problem definition, technical novelty, and datasets:

- We propose a new paradigm, Panoptic 3D Parsing (Panoptic3D) that, to the best of our knowledge, is the first system of its kind to perform joint panoptic segmentation and 3D shapes/layout reconstruction for indoor/outdoor scenes from single-view RGB images *in the wild*. Furthermore, we combat the issue under the *absence of complete sets of multi-*



(a) Illustration on a Cityscapes [COR+16] image.



(b) Illustration on a COCO [LMB+14] image.

Figure 4.1: Illustration of our **Panoptic 3D Parsing** (Panoptic3D) system. Given an input RGB image, Panoptic3D performs joint semantic segmentation, object detection, instance segmentation, depth estimation, 3D shape reconstruction, and 3D layout estimation.

modality ground-truths for segmentation/objects/3D shapes/3D layout by developing a *stage-wise system* to maximize the generalization and robustness where ground-truths are separately available for training the individual modules.

- We show that given fully annotated synthetic information, we can train a network in an *end-to-end* fashion and provide additional layout or stuff estimation in the 3D space for similar natural images.

- In addition, we generate a *synthetic dataset*, based on the 3D-FRONT dataset [FJG+20], with comprehensive multi-modality ground-truth annotations for panoptic segmentation and 3D shapes/layout reconstruction to facilitate training and evaluation.

Observing the experiments, we show new, encouraging results for the indoor and outdoor scenes [LMB+14, COR+16] for the natural and synthetic images [FJG+20] with both qualitative and quantitative metrics.

4.2 Related Work

We show the comparison with various related work in Table 4.1. Our panoptic 3D parsing framework produces more complete modalities and is more general than the existing segmentation, detection, and 3D shapes/layout reconstruction methods.

Single-view 3D scene reconstruction. Single image 3D reconstruction has a long history [Rob63, HZ04, TGF+18, ZCSH18, HQX+18, NHG+20]. Factored3D [TGF+18] is a closely related work to ours, which combines indoor scene layout (amodal depth) with 3D shape reconstructions. Still, no dense labeling is predicted for the scene layout ("stuff") [TGF+18], and the object shape reconstruction tends to overfit the canonical shape of known categories. Holistic3D [HQZ+18] performs 3D layout and object detection jointly, but it does not perform dense foreground and "stuff" segmentation or shape reconstruction for novel objects. Instead, it

Table 4.1: **Comparison for different 3D reconstruction methods**. †For Mesh-RCNN [GMJ19], training is based on a single object instance per image but its inference allows outputs of multi-object components; nevertheless, efforts are still required to enable the multi-object module in an end-to-end pipeline for both training and evaluation.

Method	3D	Single image	Layout 3D	Panoptic segmentation	Outdoor scenes	Multiple objects
[ZZZ+18, KBJM18, GFK+18, XWC+19]	✓	✓				
[GMJ19]	✓	✓				†
[HZ04]	✓	✓			✓	
[TGF+18, ZCSH18, HQX+18, NHG+20]	✓	✓	✓			✓
[LS18]	✓	✓			✓	
[AW18]	✓	✓	✓		✓	
[PKVG99, PNF+08]	✓		✓		✓	✓
[KHG+18, KGHD19, XLZ+19, LLT19]		✓		✓	✓	✓
Panoptic3DParsing (ours)	✓	✓	✓	✓	✓	✓

conducts a retrieval task from existing CAD models, something that would not scale to natural images of outdoor scenes. Total3DUnderstanding [NHG+20] generates 3D shape reconstruction results on natural indoor photos; however, it predicts a box layout without flexible structures and layout semantic segmentation. None of the methods above simultaneously perform holistic 3D shapes/layout reconstruction and panoptic segmentation for indoor and outdoor natural RGB images.

Single image depth estimation. The 2.5D depth representation was pioneered by David Marr [Mar82]. Depth estimation from a single image can be performed in a supervised way and has been extensively studied in the literature [SSN09, EPF14]. Development in deep learning [LSD15] has expedited the progress in depth estimation [BRG16, LS18].

However, these methods do not provide semantic information along side with the depth prediction or instance reconstruction for "stuff" that could be especially useful for reconstructing challenging scenes in the wild – a strategy that we employ for Panoptic3D.

Datasets. Natural scene image datasets are absent from existing comprehensive 2D and 3D annotations for semantic segmentation [LMB+14, COR+16, SLX15], amodal segmentation [QJL+19], and 3D shapes [SWZ+18]/layout reconstruction. Furthermore, the existing natural

image datasets either do not provide panoptic segmentation annotations or lack sufficient 3D ground-truth annotations. In this paper, we create a synthetic dataset for training/evaluating the end-to-end Panoptic3D pipeline [FJG+20].

Single-view single object 3D reconstruction.Existing single image single object 3D shape reconstruction methods can typically be divided into voxel-based [WSK+15, ZZZ+18, CNH+20], mesh-based [WZL+18, GFK+18, KBJM18], and implicit function based [XWC+19] methods. In this paper, we adopt unseen class reconstruction, GenRe[ZZZ+18], for multi-object reconstruction for natural image reconstruction when well-aligned ground truth 3D mesh models are unavailable. Moreover, inspired by Mesh R-CNN [GMJ19] for multi-object shape prediction, we can perform supervised end-to-end single image 3D panoptic parsing.

Panoptic and instance segmentation. There is a renewed interest in semantic and object segmentation (Image Parsing [TCYZ05]), called panoptic segmentation [KHG+18, KGHD19, XLZ+19, LLT19]. Panoptic segmentation [KHG+18] or image parsing [TCYZ05] combines semantic segmentation and instance detection/segmentation. However, existing panoptic segmentation methods [KHG+18, XLZ+19] are focused on performing 2D image segmentation/detection only. In our work, we adopt UPSNet as our panoptic segmentation module for the primary network and combine it with the work from Zhan *et al.* [ZPD+20] to better assist 3D reconstruction with amodal masks on natural images. For our end-to-end approach, we adopt a semantic segmentation head [KGHD19] to train amodal segmentation for stuff in the scene, enabling un-occluded layout segmentation. We also reference the panoptic head from UPSNet [XLZ+19] to assist unique object detection from input view. Additionally, we predict the un-occluded amodal masks and their corresponding 3D reconstructions for "things". The final end-to-end network enables the joint training for 3D "things" shape reconstruction and panoptic segmentation.

4.3 Our Approach

4.3.1 Stage-wise System

We design a stage-wise system for natural images (such as Cityscapes and COCO) where training data contain annotated ground-truths for panoptic segmentation but lack 3D information. The method comprises four main parts: 1). single-image depth estimation; 2). panoptic segmentation; 3). instance amodal completion; and 4). single object 3D shape reconstruction for unseen classes. Features from various state-of-the-art algorithms are adopted for the individual modules. We show the network pipeline in Figure 4.2. The system first predicts the panoptic segmentation (UPSNet [XLZ$^+$19]) and the depth (DenseDepth [AW18], a transfer learning depth predictor). It then passes the modal masks to the de-occlusion net [ZPD$^+$20] to acquire amodal masks. The instance meshes are then reconstructed based on the amodal masks using GenRe [ZZZ$^+$18]. Since GenRe only predicts normalized meshes centered at the origin, the final module aligns individual shapes in the z-direction using depth estimation. It then aligns each mesh instance in the x-y direction using the corresponding amodal mask. Finally, we place instance meshes and stuff point clouds in the same coordinate system to render the panoptic 3D parsing results for visualization. The inference time takes a few seconds for images of size 1048×2048.

The network demonstrates good generalizability on novel images even with unseen shapes. However, there is still room to improve, such as adding better modules for layout completion that would respect the mesh instances in the space. Adopting RGB completion results from the de-occlusion network does not facilitate a reasonable amodal layout depth and semantic segmentation completion at the moment. The segmentation completion appears unrealistic, and the layout depth completion tends to overlap with instance shapes in the 3D space. Furthermore, there is a lack of layout/stuff segmentation and depth annotation in the wild and the photorealistic synthetic datasets. Therefore, we introduce a synthetic indoor dataset with complete annotation and our end-to-end approach next.

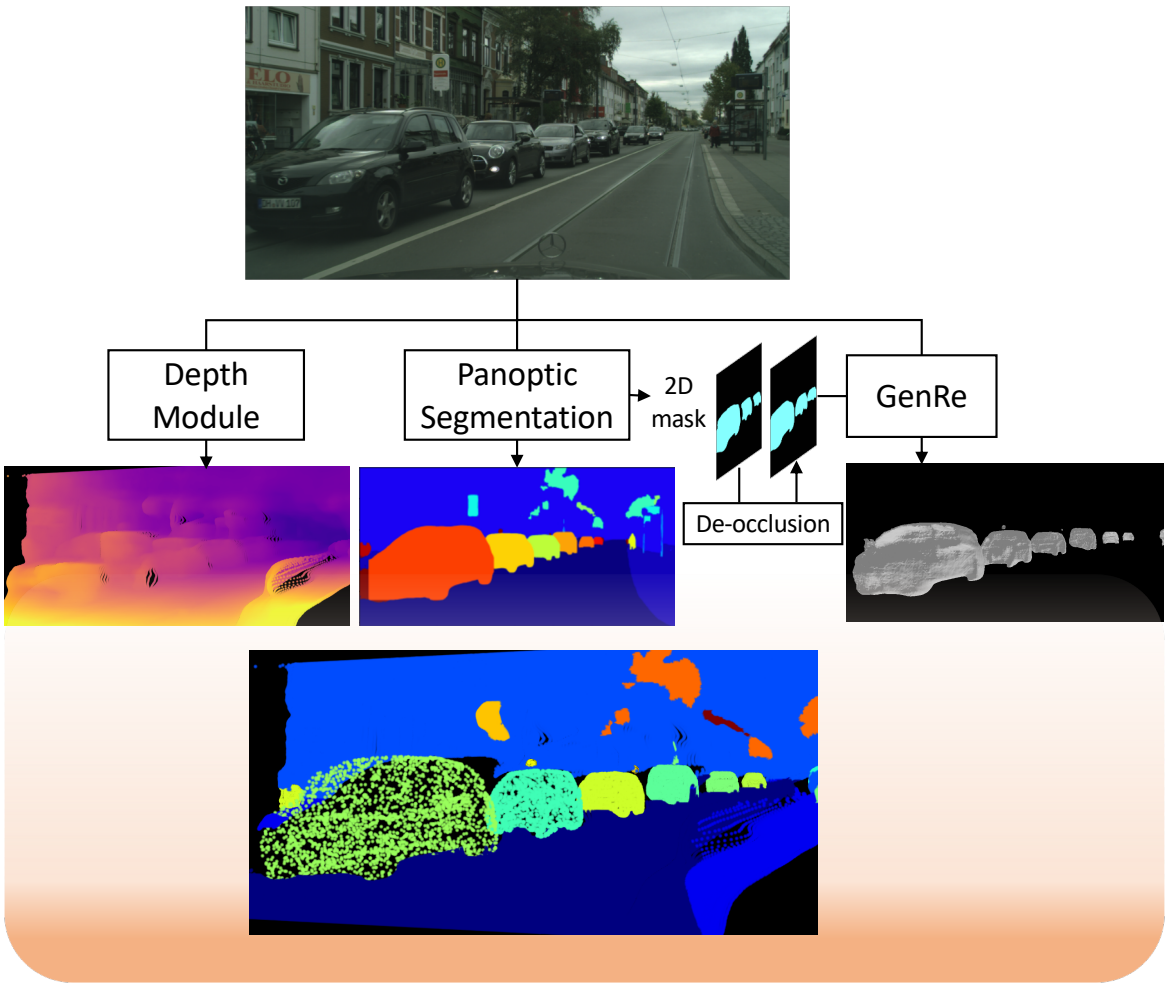


Figure 4.2: Network architecture for our **stage-wise system of panoptic 3D parsing**. Here we adopt DenseDepth [AW18] for depth prediction, UPSNet [XLZ+19] for panoptic segmentation, de-occlusion network [ZPD+20] for amodal mask completion, and GenRe [ZZZ+18] to perform instance based single image 3D reconstruction. The layout alignment module outputs the image on the bottom. The network produces meshes for the individual objects, shown as point clouds for illustration.

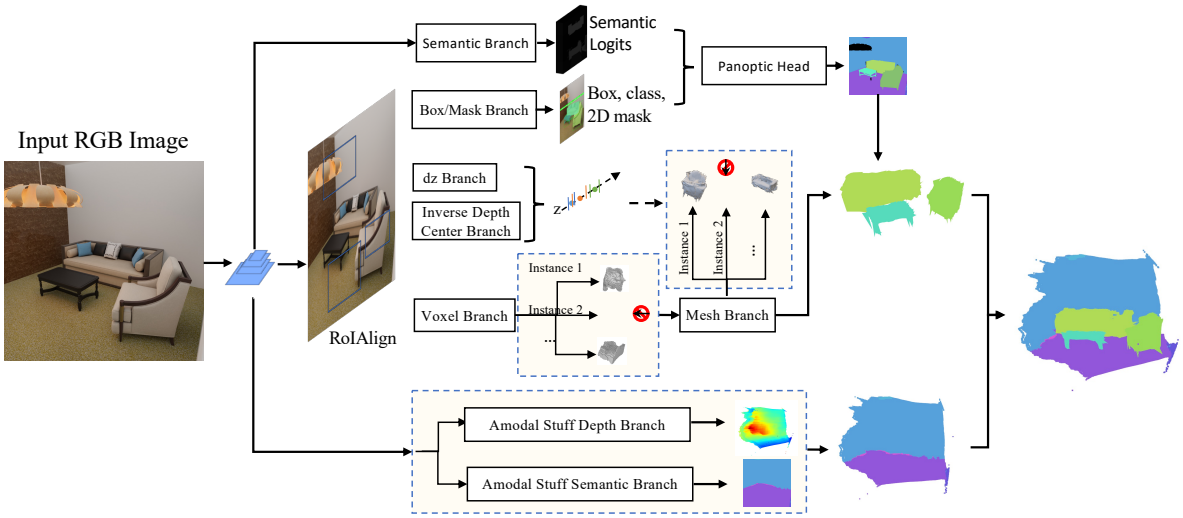


Figure 4.3: Network architecture for our **end-to-end system pipeline of panoptic 3D parsing**. dz: means depth extent. Red stop sign indicates that during training, only predictions with valid ground truth shapes are used for regression. Dotted connection indicates optional path in this end-to-end network.

4.3.2 End-to-end System

We develop an end-to-end system that can be trained on a set of ground-truth annotations for 3D scenes. Our system has 6 main components: 1). instance segmentation head; 2). multi-object training enabled shape heads; 3). amodal "stuff" semantic segmentation head; 4). amodal "stuff" depth, 5). relative object z center prediction branch, and 6). panoptic segmentation head. The network is trained end-to-end. The overview of the network structure is shown in Figure 4.3.

For instance segmentation and multi-object shape prediction, we enable the Mesh R-CNN model for multi-object training and evaluation and it comprises the first two components. Then, we add a FPN module, which is commonly used for modal semantic segmentation [KGHD19], to perform amodal "stuff" semantic segmentation. The joint training demonstrates that the network is capable of "hallucinating" the semantic information for "stuff" structures that are unseen without disrupting other tasks. We use inverse depth value, which is similar to Factored3D [TGF+18], for layout depth instead of the absolute value as it is imperially more effective during

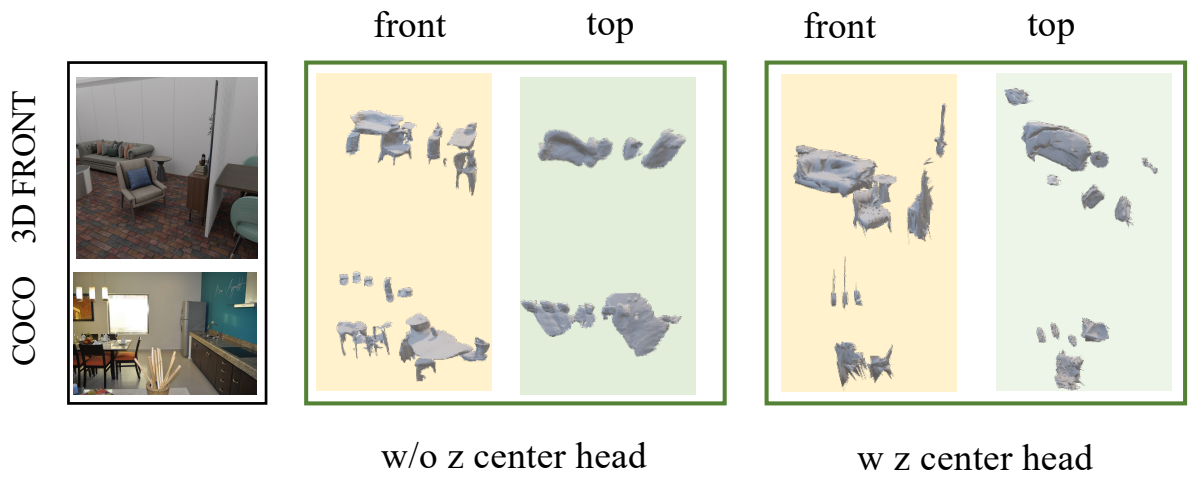


Figure 4.4: **Qualitative results of z center head ablation studies for panoptic 3D parsing.** The front and top view comparison of the predicted geometries. By adding z center prediction for each instance, our model can place objects in a relatively reasonable depth in both natural and synthetic images. In contrast, Mesh R-CNN uses a simple estimation of z center based on dz prediction; the top view from Mesh R-CNN shows that predicted instances tend to clutter into a line.

joint training. The amodal "stuff" depth is estimated using an encoder-decoder structure. In our network, we use the lower level features from the backbone network as our "shared context" for amodal depth prediction and the inverse depth [TGF+18] value for training. Mesh R-CNN [GMJ19] defines their scale-normalized depth extent as: $\bar{dz} = \frac{d_z}{z_c} \cdot \frac{f}{h}$. Here h is the height of the object's bounding box, f is the focal length, d_z is the depth extent, and $\bar{dz}$ is the prediction. This allows the network to recover an estimated scale-normalized thickness of the object in the z direction. However, Mesh R-CNN is not able to predict relative positions of objects in the same scene. We add the fifth component, a z center head, to predict where objects are on the z axis. The inverse z center value is used for training, described as $\frac{1}{z_c}$ in the equation for $\bar{dz}$. In Figure 4.4 we can see that comparing to Mesh R-CNN (meshrcnn://meshrcnn_R50.pth), by adding z center head, our model predicts shapes that gives a more reasonable top view. Mesh R-CNN uses a simple z center assumption which tends to line up the shapes at an arbitrary depth.

With the first five components, we can perform a reasonable 3D panoptic prediction. However, there is no algorithm in place that reasons about the uniqueness of space occupancy

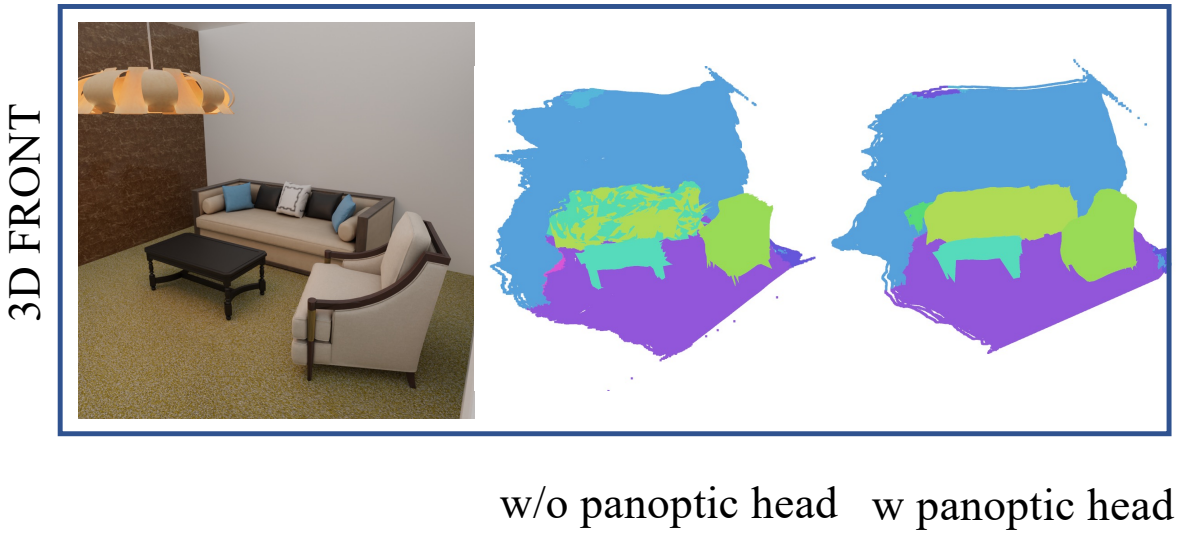


Figure 4.5: **Qualitative results of panoptic head ablation studies for panoptic 3D parsing.** The effect of panoptic head. With a panoptic head, our model can remove some duplicated objects.

in 3D. Multiple predictions may occur for the same object. Admittedly, it is non-trivial to calculate space overlap in 3D. The generalizability of the network will suffer if we choose panoptic 3D voxels as the scene representations. Additionally, the network does not guarantee a watertight mesh prediction, which prevents us from reasoning about space occupancy by converting the final predictions to solid voxels. Therefore, we introduce the last component with a panoptic segmentation head that help us predict unique objects in 3D that are linked to unique 2D detections in the input image. In Figure 4.5, we show that it effectively removed a duplicated couch prediction.

4.3.3 Datasets

Related Datasets To our best knowledge, no available dataset is accurately annotated with amodal instance segmentation, panoptic segmentation, amodal 2.5D information for "stuff", and 3D meshes for "things". Thanks to the availability of the 3D-FRONT dataset [FJG+20], we are

Table 4.2: **Datasets comparison**. For other benchmark datasets comparison for 3D-FUTURE/3D-FRONT, please refer to their report [FJG+20]. The last row shows the panoptic 3D 3D-FRONT dataset rendered and annotated by us. † available via purchase.

Dataset	Instance	Semantic	Panoptic	Depth	Amodal Semantic	Amodal Depth	3D "things"	3D "stuff"	Alignment
SUN-RGBD[SLX15]	✓	✓	-	✓	-	-	0	-	-
AI2Thor[KMH+]	✓	✓	✓	✓	-	-	100	✓	✓
ScanNet[DCS+17]	✓	✓	-	✓	-	-	14225/1160[ADD+19]	-	approx.[ADD+19]
3D-FUTURE[FJG+20]	✓	-	-	-	-	-	9992	-	✓
3D-FRONT[FJG+20]	-	-	-	-	-	-	9992	✓	✓
Hypersim[RP20]	✓	✓	✓	✓	-	-	approx. 50k†	✓†	✓
Panoptic 3D 3D-FRONT	✓	✓	✓	✓	✓	✓	9992	✓	✓

able to generate a first version of the panoptic 3D parsing dataset. We originally attempted to use a natural image dataset but met difficulties. The majority of natural image datasets either do not provide panoptic segmentation annotations or suffer from low diversity or low quantity for corresponding 3D mesh annotations. ScanNet [DCS+17] provides indoor images with diverse environment, it has large number of images annotated with both semantic and instance segmentation, and contains annotations for corresponding 3D meshes. Unfortunately, ScanNet does not contain amodal layout information. Also, the mesh annotations on ScanNet do not have good alignment with their masks, hence it is difficult to produce aligned amodal masks. Additionally, our attempt to generate panoptic segmentation information for ScanNet suffers from significant human errors that reside in both instance and semantic segmentation annotations. Therefore, we are not able to work on ScanNet for the current end-to-end supervised system. We are also aware of other existing 3D datasets such as SUN-RGBD [SLX15], AI2Thor [KMH+], ScanNet [DCS+17], Scan2CAD [ADD+19], 3D-FUTURE [FJG+20], and most recently Hypersim[RP20] (via Purchase). We show in Table 4.2 that the natural datasets, such as SUN-RGBD and ScanNet, do not precisely align 3D "stuff" or "things".

Generation Details The 3D-FRONT scenes and 3D-FUTURE models can be obtained for free by signing the license agreement from the official release website: `https://tianchi.aliyun.com/specials/promotion/alibaba-3d-scene-dataset`.

We build a panoptic 3D dataset from the 3D-FRONT dataset [FJG+20] with COCO-style annotations, which include 2D amodal instance and panoptic segmentations, modal and amodal

(layout) depth/semantic information, as well as corresponding 3D mesh information and voxelized 3D panoptic scene for every image. Referenced from the 3D-FUTURE dataset [FJG+20], we adopt 34 instance categories representing all of the countable objects as "things", and add 37 categories representing walls, ceilings, floors, etc. as "stuff".

We adopt the rendering pipeline from BlenderProc [DSW+19], which uses blender [Com18] to generate photorealistic images with realistic camera angles.

We have separated the dataset into 27 subsets with each set contains images from 250 scenes, and the last set contains 63 scenes all from the 3D-FRONT dataset. The training set contains 10 subsets, and test set contains 3 subsets. A total of 15031 images are used for training our final model, there are, on average, 2 centered and 2.7 boundary objects per image. We used 3 subsets for evaluation. Object shapes that are used for training and that touch the boundary of the image are excluded from the test sets. More details of the dataset is shown in the supplementary.

In terms of energy consumption. The rendering for each subset takes approximately 4 days on a single Titan X GPU, and the post-processing step that mostly happens on CPU takes approximately half a day.

We use binvox [NT03, Min19] and Open3d [ZPK18] for scene voxelization. Binvox is used to perform solid voxelization of each object with 128 as an edge size. The voxelized objects are then converted to dense point cloud. Each pixel in the layout depth map is also mapped to a point cloud. Finally, Open3D is used to voxelize the entire scene using the combined points from both things and stuff. The scene voxelization is done referencing the metrics used in SSCNet[SYZ+16]. The scene is bounded within a $240 \times 240 \times 340$ volume with a unit voxel size of 0.02. We exclude occluded objects from the voxelization. Each scene is in its own camera coordinate system, so the corresponding panoptic 2D annotation can be used to color each point. We modified Open3D so that color sampling is based on the maximum points occupying each voxel. The voxelized scenes are only used for panoptic 3D evaluation.

We also use the COCO [LMB+14] and Cityscapes [COR+16] mainly for qualitative

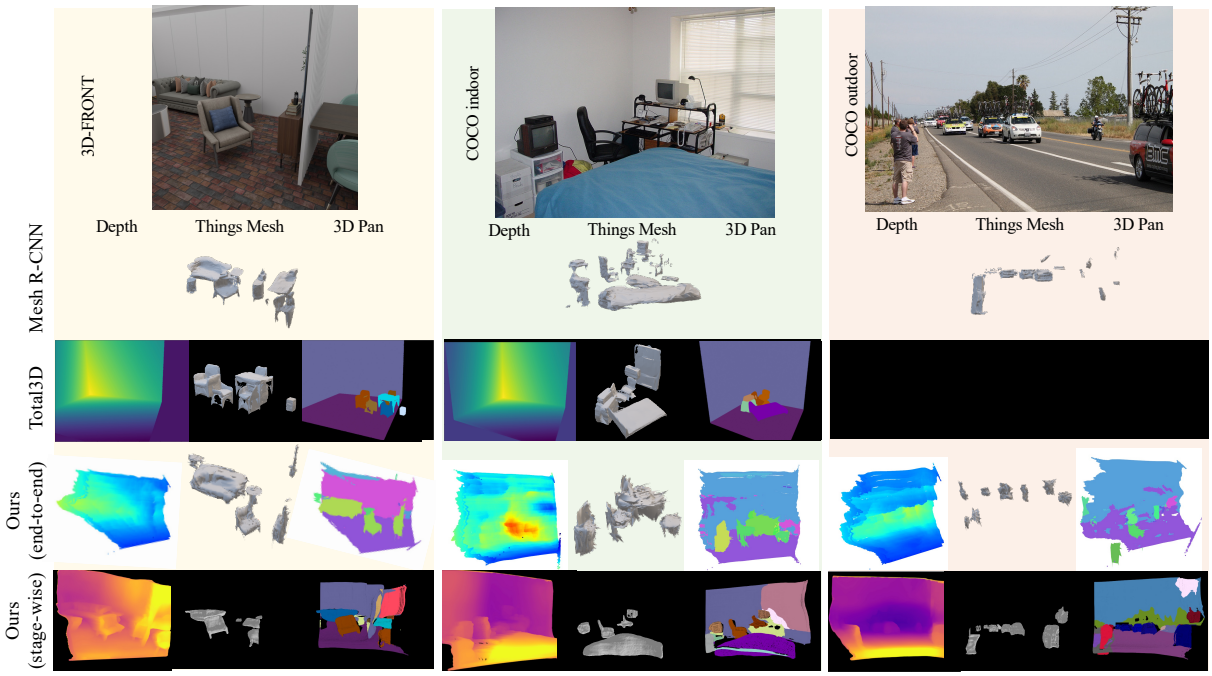


Figure 4.6: **Qualitative comparison for cross-domain evaluation on indoor and outdoor images**. We show amodal/modal depth, mesh predictions, and 3D panoptic estimations from three types of images: **Left**: an 3D-FRONT image [FJG+20], **Middle**: a COCO Indoor image [LMB+14], **Right**: a COCO outdoor image [LMB+14]. From top-down, we compare our models with the state-of-art methods: Mesh R-CNN [GMJ19] and Total3DUnderstanding [NHG+20]. We leave the area empty if no results can be acquired. Comparing to Mesh R-CNN [GMJ19] and Total3DUnderstanding [NHG+20], our stage-wise model(last row) is the only one that can provide inference on all three types of input images with reasonable output.

evaluation. COCO panoptic [KHG+18], 3D-FRONT [FJG+20] and 4000 SUNRGBD [SLX15] images are also used for quantitative evaluation.

4.4 Experiments

4.4.1 Stage-wise Network

Experiment Details The stage-wise pipeline is shown in Figure 4.2. We utilize the pretrained weights for UPSNet [XLZ+19], de-occlusion network [ZPD+20], depth network [AW18], and unseen classes object reconstruction network [ZZZ+18] for our pipeline. This

stage-wise framework takes an RGB image and predicts the 3D panoptic parsing of the scene in point cloud (for "stuff") and meshes (for " things"). First, the network takes panoptic results from UPSNet and depth estimation from DenseDepth. It then passes the modal masks to the de-occlusion net to acquire amodal masks. GenRe is then used to reconstruct the instance meshes based on the amodal masks. Then, the module maps panoptic labels to depth pixels and uses the camera intrinsics estimation to inverse project depth into point clouds. For the Cityscapes dataset, we compute its camera intrinsics with FOV = 60, height = 1024 and width = 2048 [COR$^+$16]. For the COCO dataset, since it doesn't provide its camera information, we estimate its FOV to be 60 based on heuristics. We pick images with the size of 480×640, which is compatible with every sub-module of the stage-wise network. GenRe only predicts normalized meshes centering at the origin. The final module aligns individual shapes in the z-direction using depth estimation and in the x-y direction using the mask. The module takes the mean of the $98th$ percentile and the $2nd$ percentile of the filtered and sorted per-pixel depth prediction within the predicted mask region to estimate the z-center depth of an object. Finally, it places meshes and point cloud in the same coordinate system to render the panoptic 3D parsing results. The general inference time is within seconds.

Qualitative Evaluation

We show the reconstructed Panoptic3D scene in colors that corresponds to the panoptic categories defined in 2D annotations for the qualitative measure. Although the pipeline outputs mesh shapes, here we sample point cloud to show the 3D effect in Figure 4.1 from COCO and Cityscapes, respectively. More results are demonstrated in Figure 4.7.

We compare our stage-wise and end-to-end models with state-of-art methods, such as Total3DUnderstanding and Mesh R-CNN in Figure 4.6. [1] We show that our stage-wise network

[1] *Note that Total3DUnderstanding [NHG$^+$20] was trained on SUNRGBD [SLX15] + Pix3D [SWZ$^+$18]; our end-to-end system was trained on the dataset extended from 3D-FRONT [FJG$^+$20]; our stage-wise system was trained for its individual modules based on separate datasets (e.g. panoptic segmentation on COCO [LMB$^+$14]), but not on 3D-FRONT [FJG$^+$20] nor on SUNRGBD [SLX15], Pix3D [SWZ$^+$18].* Our end-to-end model in the second last row performs less well for outdoor images. Total3DUnderstanding [NHG$^+$20] takes preprocessed images containing proposed labels known to their pre-trained model. We cannot pass unknown categories from outdoor

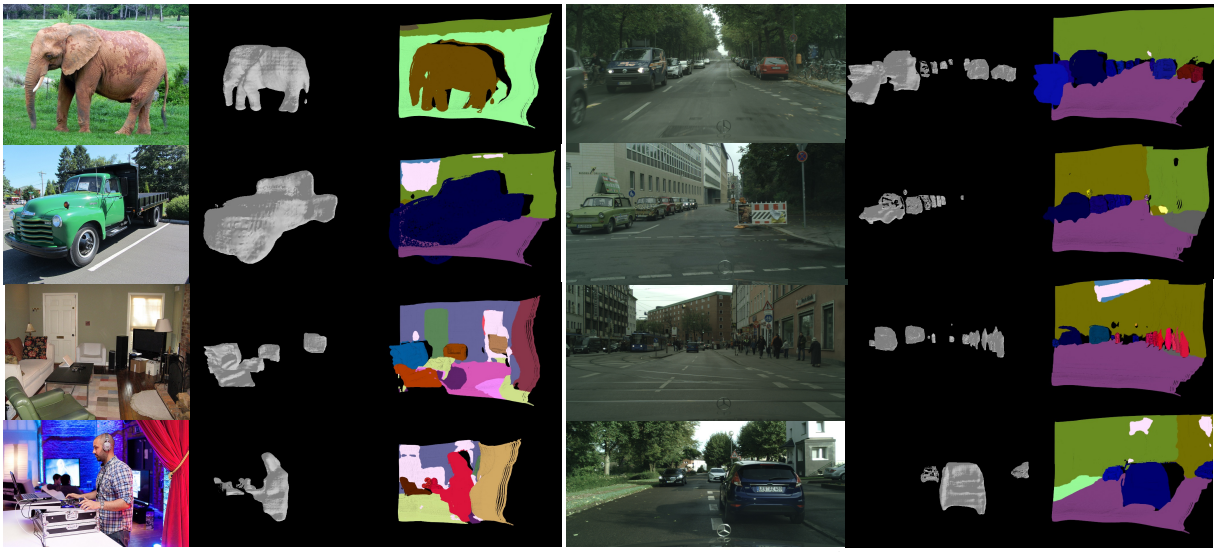


Figure 4.7: **Qualitative results of our stage-wise Panoptic3D system for single-view images in the wild** (COCO [LMB+14] and Cityscapes images [COR+16]). Results are taken from an off-angle shot to show the difference between depth and 3D panoptic results. The point cloud representation is to help with better visualization of 3D structures. They are sampled from result object meshes.

generalize well on both synthetic and natural images for both indoor and outdoor scenes. Our end-to-end model is able to generalize to natural images as well. Admittedly, the end-to-end model does not perform as well on the outdoor image. We hypothesize that adding diverse synthetic data to the training process will help improve its generalizability towards outdoor scenes. The third row of Total3DUnderstanding is left blank because the network is a stage-wise pipeline, and the released model can only be used for preprocessed proposals that contain NYU labels. We can acquire outdoor proposals, but it is unclear what is a fair way to convert outdoor labels to NYU labels. Therefore, we are not able to provide a result on the outdoor image from Total3DUnderstanding. Our instance shape prediction quality is closer to the results from Mesh R-CNN, which is expected. Total3DUnderstanding produces smoother surfaces. However, we find that their shape prediction is more class-dependent. For example, in the top image, a 3-person sofa is predicted as a one-seater. The predicted sofa would be significantly narrower, which

images to acquire a reasonable output due to this label requirement. The depth image inconsistency is introduced by different rendering software.

Table 4.3: **Comparison of re-projected 2D panoptic qualities from a subset of coco indoor images between Total3DUnderstanding and Stage-wise network**. For Total3DUnderstanding, the re-projection uses inferred camera extrinsic and we change the predicted layout box into meshes for wall, ceiling, and floor. Our stage-wise method outperforms Total3DUnderstanding on every metrics.

	PQ ↑			SQ ↑			RQ ↑		
Methods	IOU@.5	IOU@.4	IOU@.3	IOU@.5	IOU@.4	IOU@.3	IOU@.5	IOU@.4	IOU@.3
Total3DUnderstanding	0.043	0.06	0.077	0.046	0.063	0.081	0.065	0.101	0.15
Stage-wise (ours)	0.168	0.176	0.181	0.177	0.184	0.181	0.21	0.220	0.226

becomes visibly inconsistent with the input image. Different depth rendering here is due to the usage of different rendering software throughout the experiments.

Quantitative Evaluation

We evaluate the network from three perspectives. In the first aspect, using the COCO dataset, we can project the panoptic 3D results back to the input view and evaluate it against their ground truth 2D panoptic annotation to show its image parsing capability. We acquired around 300 images from the COCO test set that contains overlapped panoptic labels Total3DUnderstanding. In Table 4.3, we show that our pipeline outperforms Total3DUnderstanding on reprojected panoptic segmentation metrics. Additionally, to show 2D to 3D projection accuracy, we use the predicted masks from SUNRGBD provided by Total3DUnderstanding and evaluate the average 3D bounding box IoUs against their ground truth. The results are in Table 4.4. Our prediction does not indicate the orientation or rotation of the object; in fairness, we evaluate both our network and Total3DUnderestanding using camera axis-aligned 3D bounding boxes. Although our model has never been trained on the SUNRGBD dataset, we show that our stage-wise pipeline performs reasonably well compared to Total3DUnderstanding [NHG$^+$20]. We provide more qualitative and quantitative evaluation results in the supplementary material. In terms of 3D object reconstruction, we compared our stage-wise network with Mesh R-CNN with Chamfer distance in the supplementary.

Additionally, we generate voxelized 3D panoptic ground truth, which enables evaluating panoptic metrics [KHG$^+$18] in the 3D space provided in supplementary. A reasonable panoptic score is hard to acquire given that neither network produces watertight meshes (no solid voxels can

Table 4.4: **Comparison of 3D Bounding Box IOU between our stage-wise network and Total3DUnderstanding [NHG$^+$20]**. The Stage-wise network takes in the 2D mask information provided by [NHG$^+$20], and final 3D Boxes are adjusted to align with the camera axis.

	3DBBox IOU (mean) ↑	3DBBox IOU (max) ↑
Total3DUnderstanding [NHG$^+$20]	0.26	0.88
Stage-wise (ours)	0.144	0.71

Table 4.5: **The average precisions for boxes, masks and meshes increase** as more subsets are used during training for the end-to-end network. Refer to [GMJ19] for metrics details.

# subset	boxAP ↑	maskAP ↑	meshAP ↑
1	0.28510	0.25038	0.09610
2	0.34118	0.30417	0.15018
3	0.37947	0.31998	0.17670

be derived from the mesh). We voxelize sampled points from the voxel prediction for panoptic 3D evaluation during postprocessing. Our voxelization method may cause sparse voxel representation of things in the predicted space, which affects the panoptic 3D evaluation. For additional scene level reconstruction comparison, we evaluate point cloud from the voxelized scene with scene level F1 score and panoptic class average F1 score [TRR$^+$19] in Table B.2. We show that at the categorical level, the end-to-end model outperforms the stage-wise pipeline, which is expected since the stage-wise pipeline is not trained on the 3D-FRONT dataset. For scene-level F1 score, however, using iterative closest point (ICP) [ZPK18] adjusted results from the end-to-end model does not perform as well as the stage-wise model.

Table 4.6: **Comparison on semantic level 3D reconstruction F1 score on 3D-FRONT dataset between the stage-wise and end-to-end (E2E) networks**. ICP [ZPK18] and fscore threshold 0.01 and 0.02 are used for the stage-wise pipeline. Only overlapped categories are shown here. The last column refers to the scene level 3D reconstruction F1 score for the stage-wise pipeline on the 3D-FRONT dataset.

	Ceiling-merged	Floor-merged	Wall-merged	Chair	Couch	Bed	Dining table	Cabinet_thing	Lamp_thing	Scene-level F1 score
ICP + fscore threshold 0.02	0.056	0.055	0.06	0.13	0.47	0.29	0.12	0.10	0.006	0.173
ICP + fscore threshold 0.01	0.016	0.016	0.019	0.06	0.199	0.126	0.055	0.041	0.002	0.07
No ICP + fscore threshold 0.02	0.019	0.011	0.019	0.034	0.202	0.099	0.039	0.016	0.0	0.085
No ICP + fscore threshold 0.01	0.012	0.009	0.015	0.031	0.192	0.094	0.037	0.014	0.0	0.078
E2E model ICP + fscore threshold 0.02	0.22	0.37	0.106	0.42	0.29	0.16	0.375	0.41	0.15	0.063

4.4.2 End-to-end Network

Experiment Details

We train our final network with a base learning rate of 0.0002 for 30000-50000 iterations with Adam optimizer. Higher learning rates cause the model to diverge. Lower learning rates prolong the training time, which incurs high energy consumption. We use PyTorch [PGM+19] for model development and PyTorch3D [RRN+20] for rendering amodal instance masks. The majority of the experiments are performed on 4 GPUs. For a larger dataset with larger models, 8 GPUs are used for approximately two weeks. The final model uses 16 images per batch which takes 18 hours to run on 8 Titan X GPUs. The backbone of our network uses ResNet50. Our input size for the detection backbone is 1024×1024. During inference, if not provided, we assume the focal length is 27.7, and the field of view is 60.

Mesh R-CNN uses the voxel and meshes representation in a cuboid camera space, which may deform the edges beyond image boundaries to infinity. Hence we only use non-boundary objects for training and testing. Our earlier experiments suggest including boundary instances for training the detection head and the voxel head while only skipping the loss regression for the mesh head helps the network to predict boundary objects better. However, it also significantly increases the training time. We include the experiment results in the supplementary.

Qualitative Evaluation

Figure 4.6 shows the qualitative evaluation of natural images (indoor and outdoor) and synthetic test sets compared to our stage-wise model and other state-of-art models. The end-to-end network can detect and show promising results on natural images. However, the quality of layout estimation and shape estimation is visibly less appealing compared to synthetic image prediction. Admittedly, our end-to-end network cannot predict outdoor scenes and objects that do not exist or exist less frequently in the training dataset. As diverse synthetic datasets become available, we expect the network to improve as well.

Quantitative Evaluation

We evaluate our instance shape reconstruction with meshAP [GMJ19] and show that as we use more subsets for training, the accuracy increases for the same test set. There is a trade-off between multi-modality training and meshAP score. The best meshAP can be achieved by training one object at a time without any other modalities, such as panoptic, z center, and layout predictions. The model converges faster, which scores a higher meshAP value with less time. We show in Table 4.5 that the accuracy increases as more subsets are used for training. The results shown here are trained with a learning rate of 0.02 for 11000 iterations with 2D detection and shape predictions. However, when we add more modalities, the model does not converge at this learning rate. As we drop the base learning rate by 100 times, the training time to achieve similar reconstruction quality grows significantly. We have the training cut-off at 50k iterations for our final model. With a longer training time, the model can achieve smoother shape predictions.

We provide 3D voxel IoU based panoptic evaluation results as is shown in the supplementary. A meaningful panoptic3D result is hard to acquire for stuff class, given that they are created from layout depth, and only a thin layer of voxels can be derived. A slight offset in depth would introduce significant errors. We additionally, added semantic class average F1score and scene-level F1score to add additional quantitative measures as is shown in Table B.2. We provide our baseline metrics using RMSE and RMSE_log for layout estimation and mean IoU, mean Accuracy for layout semantic segmentation in the supplementary.

4.5 Conclusion

This paper has developed the first (to our best knowledge) practical system for predicting panoptic 3D scene parsing from a single-view image in the wild. We provide a dataset that allows 3D panoptic evaluation and an end-to-end system that can generalize to natural images similar to the synthetic dataset. Panoptic 3D parsing for single-view images in the wild points to an exciting

direction in computer vision. In terms of limitations, the end-to-end system has limitations when applied to natural outdoor scenes; our systems are still in their early development stage and they still have a large room to improve.

This chapter is based on the material that has been submitted for publication authored by Sainan Liu, Yuan Gao, Vincent Nguyen, Subarna Tripathi, Zhuowen Tu. The dissertation author is the co-primary investigator and author of this material.

Chapter 5

Discussion

In conclusion, we have described an attentional shapecontext net for point cloud recognition. We validate that we can also improve 2D recognition by combining viewer-centered and object-centered representations. Finally, we provided the first approach for panoptic 3D scene parsing in the wild.

Next, we discuss what we have learned from our experiments and what we think could be potential future directions for panoptic 3D parsing in the wild.

View dependent and view independent representations for panoptic 3D scene parsing The panoptic 3D scene parsing problem boils down to two main issues: 1. amodal shape prediction (self-occlusion and external occlusion), 2. alignment (3D to input 2D and relative positioning). For amodal shape prediction in the wild, view-dependent and view-independent representations should both be considered to achieve the best outcome. Here the concepts are slightly different from the definitions we used for recognition tasks. Given that the output is a 3D geometry, here we adopt similar definitions with [TRR+19]. There is a trade-off between the two types of representations in terms of generalization across categories. On the one hand, view-dependent representation can reconstruct the shape from the viewing angle to be consistent with the input view but may lack details for images from unseen viewpoints. On the other hand,

a view-independent representation may reconstruct a reasonable shape from all views but may be less consistent with the input view. To infer images with known categories, which have abundant synthetic 3D shape annotations, acquiring view independent representation could help capture the canonical features of the class, especially at the time of occlusion. Prior knowledge of the canonical shape of the same category could help complete the missing geometry in natural images, especially if amodal masks are not readily available. On the other hand, to infer objects from unknown categories or known categories with few 3D annotations or amorphous stuff, a view-dependent representation (often depth-based) would be beneficial for generalization on novel geometries. Studying latent code that allows interpolation between view-dependent and independent representations would be an exciting direction for future research. Additionally, an attention-based network structure can be applied to automatically adopt the correct combination of view-dependent and view-independent representations based on the input image content.

Hybrid 3D geometric representations for panoptic 3D scene parsing Inspired by computer graphics, discrete 3D geometries, such as voxel grid, point cloud, and meshes, are commonly used for scene reconstruction tasks in earlier studies. Voxel grid can naturally adopt convolutional neural network architectures[TGF+18], and in contrast to the other representations, within the defined resolution, one can easily reason the uniqueness of 3D occupancy. Therefore, we can use IoU-based panoptic 3D metrics for evaluation even without watertight prediction. However, post-processing is needed to acquire a smooth surface. If a 3D convolution network structure is present, the network does not scale sufficiently when high-resolution output is required. Point cloud representation is often used as input, such as inferred, ultrasound, and lidar data. As the reconstruction output, the inverse projection of the depth image is in the form of a point cloud. The majority of work is on object-level reconstruction using view synthesis with depth inverse projection [FSG17, MNB18, JSQJ18, MB19]. Mesh representation can offer a smooth surface on objects [WZL+18], but the predicted mesh is often not watertight if it's not a template-based method [GFK+18]. It is nontrivial to trim or grow mesh scenes, whereas voxel grid and point

cloud representation allow easy cut or extension while working with multi-view or video frames.

In recent years, implicit surface predictions have been proposed, such as truncated signed distance function (TSDF) [PFS+19, XWC+19] or occupancy volumes [CZ19, MON+19], which enabled the revolution towards continuous 3D shape representations. Such a method sacrifices ready-to-use discrete 3D shape output for continuous representation, which allows flexible resolution. At object-level reconstruction, such a method can generate a watertight prediction [MON+19], but it still requires 3D shape supervision. If one can achieve watertight free space estimation at scene-level reconstruction, it may be very beneficial for indoor robotic navigation tasks. Joint prediction of panoptic segmentation along with TSDF would be an interesting direction for future research.

As the Computer Vision and its inverse topic, Computer Graphics, evolve, interdisciplinary problems, such as novel view synthesis, or image-based rendering, provided research space for novel architecture development. In line with implicit representation in continuous space, NeRF-based studies exploded within a short half-year period. In addition to novel view synthesis tasks, the network also learns an implicit shape representation. NeRF utilizes neural rendering techniques to learn a 3D scene structure from multi-view images with camera intrinsic information. It freed the network from requiring 3D ground truth geometry data since it can represent 3D geometry based on training images alone. Multi-view images are needed here to compensate for the lack of annotated 3D geometry. Hence with fewer views, the network struggles to generalize to novel instances with reasonable 3D shape. Researchers have investigated directions where geometric priors are used as a 3D scaffold to provide priors for NeRF, which provided better synthesis results [RMBF21]. Scene decomposition has been studied by [NG21, XPMB21, YLSL21] and additionally segmentation image synthesis can be learned [SKK21]. However, without ground truth layout [XPMB21], dynamic instances [OMT+20], or ground truth 3D geometry, it is hard to inference watertight instance shapes from a scene-level image alone. How to best integrate object-level shape priors with scene-level multi-view learning would be an

interesting future direction to achieve better panoptic 3D parsing results.

Despite novel representations, another future research direction is to use hybrid geometric representations in tandem or parallel. Mesh R-CNN [GMJ19] studied a two-stage representation system where the shape is first predicted in voxel grid, then to mesh, which enabled an efficient learning pipeline that can detect and reconstruct multiple objects in a scene. Volumetric scene representation is often combined with TSDF [DT20] for efficient scene-level surface reconstruction. [MON+19] produces implicit continuous watertight 3D surface via decision boundary from either image, point cloud, or voxel grids at object-level. [RMBF21] can condition on discrete voxel grid output and produces continuous NeRF-based representation at object-level. At the scene level, although amortized rendering techniques have been developed [KSZ+21] to combat the problem of long inference time of NeRF, representing a complex scene with panoptic prediction concurrently is still a challenging task for in-the-wild images. Nonetheless, NeRF-based hybrid models could be a promising direction for panoptic 3D parsing representations.

Open Synthetic Dataset Obtaining diverse and photo-realistic synthetic images with complete corresponding panoptic 3D annotations in 3D is an expensive task. Due to the lack of such a dataset, as mentioned in Chapter 4, the research community tries to navigate it and produce impressive results. However, there are still tremendous merits in providing more open free synthetic indoor and outdoor datasets. For one thing, a synthetic dataset can provide precise annotations which allow us to evaluate our networks holistically when such annotations are missing from the real-world dataset, such as the 3D panoptic metrics we mentioned in Chapter 4. Additionally, as mentioned before, ShapeNet has been proven to be a comprehensive dataset for learning shape priors to assist object reconstruction tasks with view-independent representations. Networks trained with ShapeNet have also demonstrated a surprising amount of generalization capabilities on real images[ZZZ+18]. Furthermore, predictions such as amodal layout geometries from in-the-wild images can only be estimated via continuous video frames or static scenes with dynamic object instances. Providing synthetic layout geometries can significantly improve

disjoint predictions between things and stuff with common scenes [XPMB21]. Finally, if more and more hybrid 3D geometric representations are used in tandem, having ground-truth 3D scene-level geometries could help bridge the gap between 3D object reconstruction and scene-level reconstruction.

In addition to the datasets mentioned in Chapter 4, new datasets that reproduce the layouts from real-world scenarios, such as OpenRooms [LYS+21] would be beneficial to panoptic 3D parsing tasks if more amodal information is released. Additionally, potential combinations of OpenRooms and PartNet [MZC+19] which produces super panoptic 3D parsing annotations, would be an exciting future dataset direction that provides even more real-life scenarios.

Appendix A

Object and Viewer-Centered Representation

A.1 2D In-plane Rotation Ablation Study

We evaluate ResNet18 with different angles of rotation augmentation and observe that the evaluation accuracy stops increasing as we provide denser angle augmentations as is shown in Figure A.1.

A.2 Rotation Invariant Analysis

For spherical CNNs, Cohen *et al.* has shown empirical support for rotation-invariant learning problems. Here we show in Figure A.2 that the features of spherical CNNs (without any 3D rotation data augmentation) on the 3D reconstruction of a "bus" do demonstrate a certain level of rotation invariant property.

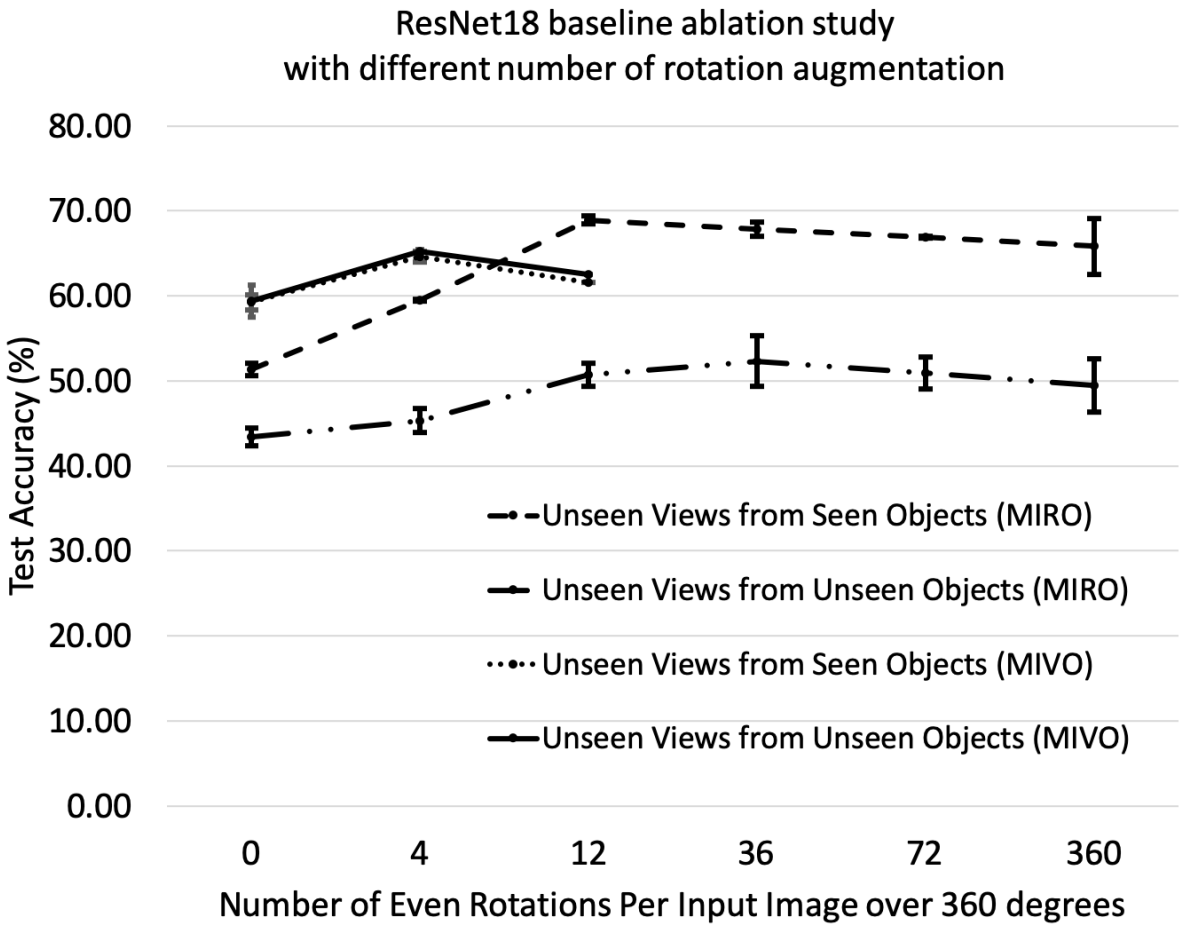


Figure A.1: Rotational ablation study for ResNet18. X-axis: the number of rotation over 360 degrees; 0 means no rotation augmentation is applied to the original input view within the 2D plane. Means and standard deviations are reported over two repeats each. The test accuracy plateaus as the number of 2D in-plane rotations increases. The accuracy plateaus around 30 degrees for gMIRO and 90 degrees for gMIVO.

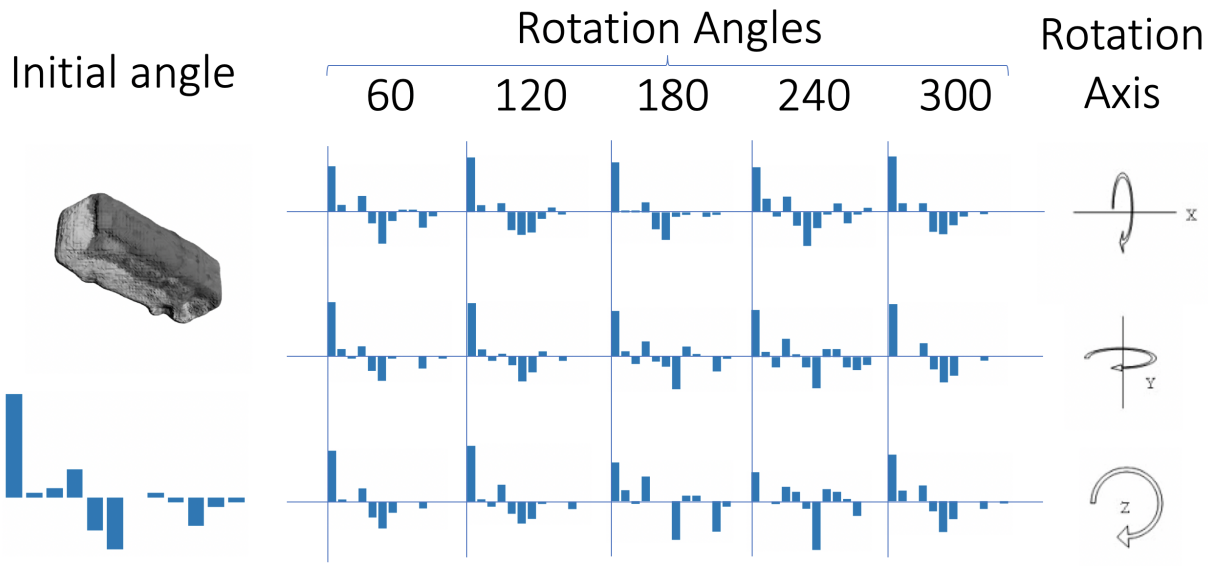


Figure A.2: Demonstration of the **achieved** rotation-invariance property of spherical CNNs on 3D reconstruction of an object ("bus") instance. We train our model with spherical maps generated from a reconstructed 3D object in its initial orientation from GenRe. The reconstructed object (top) and features from the trained spherical maps (bottom) are shown in the left-most column. We then rotate the reconstructed object along 3 different axes over 5 different angle variations (in degrees) to generate a spherical map test set. The last column shows the axis of rotation.

A.3 Experiment details

With the gMIRO dataset, in the OC module (Figure A.3) we train the spherical CNNs for 300 epochs, with the batch size 12, learning rate 0.1 (decay factor=10 for every 100 epochs for 300 epochs), and bandwidth 112; for the ResNet18 part, we use learning rate 0.1, batch size 10 for 500 epochs, and bandwidth of 3.

For the VC (3D) branch, we use 640 3D viewpoint augmentations with texture. We train ResNet18 for 500 epochs with batch size 512 and learning rate 0.01. For view selection, we use a nearest neighbor approach. The augmented image that is closest to the input viewpoint is used for evaluation on gMIRO dataset. When an attention selection layer is used, we first train a ResNet with 80% of the training data, and then 20% of the remaining training data is used to train a weighted or attention layer for selection. During the inference time, we use the ResNet model trained on the entire training set and the trained selection layer.

For the VC (2D) branch, we use online 2D in-plane augmentation with 30-degree rotations, and train ResNet18 for 1250 epochs with the batch size 512 and learning rate 0.01.

On the gMIVO dataset, for the OC branch, we train the spherical CNNs for 300 epochs, with the batch size 12, learning rate 0.1 (decay factor=10 for every 100 epochs for 300 epochs), and bandwidth 112. For the VC (3D) branch, we use 160 3D viewpoint augmentations with texture, and train ResNet18 for 120 epochs with the batch size 2048 and learning rate 0.1 (decay factor=10 for every 50 epochs). For the VC (2D) module, we use an online 2D in-plane augmentation with 90-degree rotations. We train ResNet18 for 1250 epochs with the batch size 512 and learning rate 0.1 (decay factor=10 for every 350 epochs).

A.4 Runtime Analysis

Each ResNet18 consists of around 11 million trainable parameters, whereas the largest spherical CNN has about 1.4 million trainable parameters. The space complexity for training is

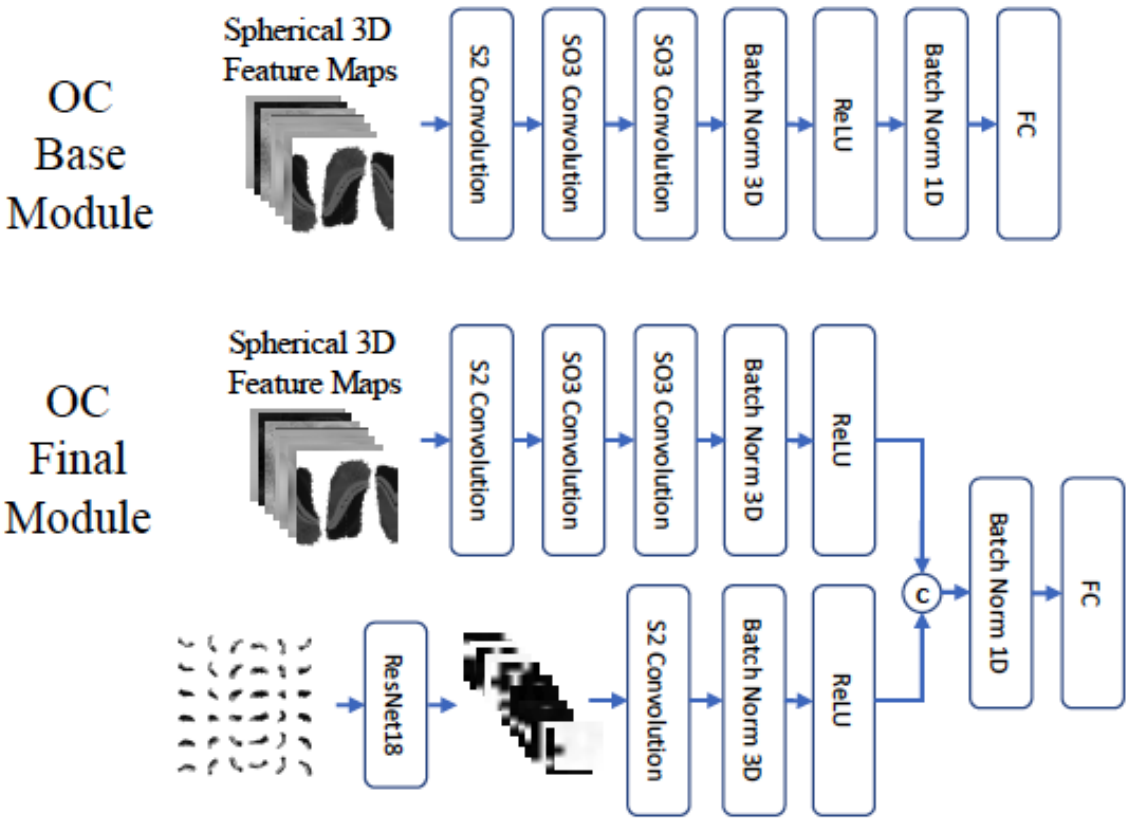


Figure A.3: Network structure for OC baseline module vs. final OC module. c here means the concatenation of the two branches.

approximately $O(Cn)$ where C equals 7 spherical signal maps + 3D to 2D projection of input view $\times$ 160/640 + 36 (with 3D-rotation augmentation) + 2D input view $\times$ 12 (with every 30 degree 2D-rotation augmentation). During testing, C equals 9 (7 spherical signal maps + 3D to 2D projection of input view $\times$ 1 (or 160 if using attention) + 36 + original input view $\times$ 1) spherical signal maps. The average inference time is always within minutes for the ResNet18s and spherical CNNs with the gMIRO dataset; it takes 10's of milliseconds for evaluation. It takes approximately a day to generate the reconstructions for all of the images from the gMIRO dataset with 3 Titan Xp GPUs. The texture estimation process uses a k-d tree structure for the nearest neighbor search, which takes on average $O(logV)$ in terms of time and $O(V)$ in terms of space. V here is the number of voxels in the volumetric representation ($128\times128\times128$) obtained from GenRe. It takes less than a second on average to process one object on a CPU. Both the texture mapping processes for images and the individual module training can be sped up by parallel processing.

Appendix B

Panoptic 3D Parsing

B.1 Indoor Scene Qualitative Ablation Results

Here we provide more indoor scene qualitative ablation comparisons between Total 3D Understanding (Total3D) and our methods. Note that Total3D uses natural indoor datasets (SUNRGBD and Pix3D) for training. Our end-to-end approach uses ten subsets of the 3D-FRONT dataset, but it has never seen any natural images. Some submodules of the stage-wise pipeline use the COCO 2D dataset, but it never uses 3D information regarding either 3D-FRONT or the COCO dataset. Additionally, to have a fair comparison, we post-process the results from Total3DUnderstanding by assigning floor, wall, and ceiling labels to the six sides of the predicted layout bounding box. Total3D does not predict 3D panoptic outputs.

In Figure B.1, with images a, c, d, and e, we show that Total3D's 3D layout bounding box predictions often cut through predicted objects, whereas our instance shape predictions align with the layout better. In image b, we show that Total3D's shape predictions do not represent the shape in the input image. For example, a three-seat sofa is predicted as a one-seat in image b, whereas our methods predict 3D shapes that are visibly consistent with the furniture in the input image. The results of images c, d, and e show that our models predict the shape orientation more

consistently with the input image, whereas Total3D predicts 3D shapes and orientations that are visibly significantly different from the input image.

Admittedly, there is a trade-off between the quality of the reconstruction and the layout consistency. Total3D's prediction closely resembles a retrieval method: each instance reconstruction can be of high quality; However, the orientation could be off, and the shape may not be consistent with the input image. For example, the shapes of chairs and beds predicted by Total3D appear very similar across the images a, b, and c, whereas the input images contain very different beds and chairs.

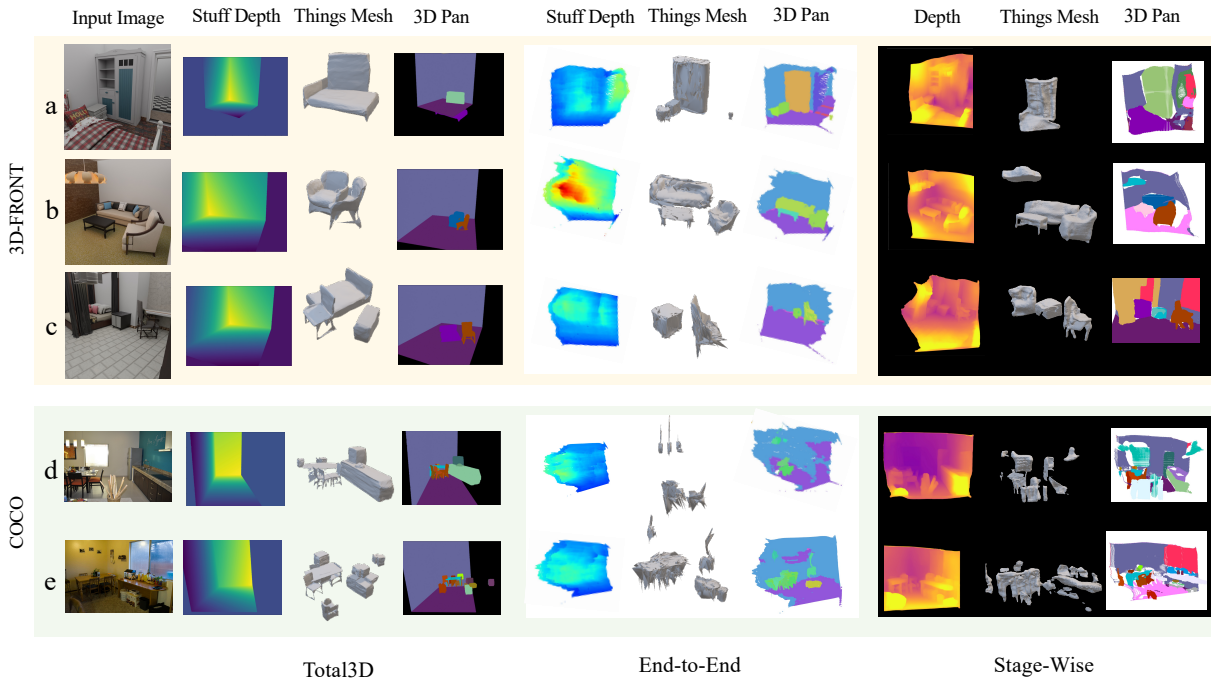


Figure B.1: Qualitative comparison from synthetic and natural indoor images between Total3DUnderstanding and our models. Our models offer object reconstruction that is less intersected with the layout, and is more consistent with the input image.

B.2 Additional Quantitative Results

B.2.1 Panoptic 3D Evaluation Results

In order to quantitatively evaluate scene level reconstruction, we evaluate three scene-level metrics: 1). 3D panoptic calculated on voxelized predictions, 2). categorical level and 3). scene level F1 score between point cloud sampled from the prediction and the ground truth scene voxels on the 3D-FRONT dataset.

For 3D panoptic evaluation, we design our panoptic metrics similar to the 2D panoptic metrics following Gkioxari et al. The results in Table B.1 shows the 3D panoptic results between the end-to-end and stage-wise pipelines. The panoptic 3D metrics are low at IOU threshold = 0.5 for both methods. Given that the stuff results for stage-wise are much lower, we also present the results for stage-wise at IOU threshold = 0.1. There are two main reasons for low panoptic 3D scores: 1. the stage-wise pipeline has never seen the 3D-FRONT dataset, and it does not contain camera pose prediction. In the paper we show that by using iterative closest point (ICP) alignment, the F1 score can be improved. Here we did not use ICP for fair comparison. 2. As is mentioned in the paper, the voxelization method for prediction may cause sparse voxel representation of things in the predicted space, which affects the panoptic 3D evaluation.

Table B.1: **3D Panoptic evaluation with 3D FRONT voxelized scenes for our end-to-end network**. 3D panoptic results are significantly lower than its 2D counter part. We did not consider free space as a category in this calculation.

	E2E (IOU=0.5)			Stage-wise(IOU=0.5)			Stage-wise(IOU=0.1)		
	PQ ↑	SQ ↑	RQ ↑	PQ ↑	SQ ↑	RQ ↑	PQ ↑	SQ ↑	RQ ↑
All	0.19	5.87	0.35	0.22	0.3	0.36	1.9	2.1	7.1
Things	0.36	9.77	0.66	0.38	0.5	0.6	3.5	4.0	13
Stuff	0.02	1.85	0.03	0.0	0.0	0.0	0.02	0.02	0.01

One drawback of the panoptic 3D metric is that it does not consider 3D geometric distance measures. Additionally, we use the equation provided by Tatarchenko and Richter et al. to calculate the F1 score with a threshold of 0.02. The results in Table B.2 use the direct results from the prediction without ICP. We show that for categorical F1 score, Our stage-wise model

outperforms Total3D, except the "Floor-merged" category. It is understandable given that Total3D predicts a layout box, which provides a flat surface, whereas the stage-wise pipeline only uses depth as an estimate. Our end-to-end model outperforms both models. It is expected since the end-to-end model is trained on 3D-FRONT's training set. Although the test set does not share scenes or non-boundary furniture, the test set could share similar image rendering quality. For the scene-level F1 score, we can see that both of our methods outperform Total3D.

Table B.2: **Comparison on semantic level 3D reconstruction F1 score on 3D-FRONT dataset between the stage-wise and end-to-end (E2E) networks**. Our methods outperform Total3D on almost all metrics, except the Floor-merged category. The fscore threshold is 0.02. Only overlapped categories are shown here. Total3D does not have Lamp class. The last column refers to the scene level 3D reconstruction F1 score for all three models on the 3D-FRONT test dataset.

	Ceiling-merged	Floor-merged	Wall-merged	Chair	Couch	Bed	Dining table	Cabinet_thing	Lamp_thing	Scene-level F1 score
Total3d	0.0	0.06	0.02	0.004	0.001	0.02	0.006	0.0	-	0.013
E2E	0.07	0.06	0.13	0.23	0.21	0.12	0.20	0.1	0.058	0.093
Stage-wise	0.02	0.017	0.02	0.058	0.2	0.17	0.035	0.04	0.002	0.089

B.2.2 Boundary Case

In an earlier version of the dataset, we also did a boundary case study. We find that including boundary cases for the mask, bounding box, and voxel initialization training while skipping the loss for meshes can help the overall performance.

The dataset we use for this ablation study is relatively more straightforward. The test set shares similar camera angles of the layout with the training set, and the lighting is a single light source from the camera. Table B.3 shows that the average precision for 2D boxes and masks increases as boundary cases are included during training. In addition, we notice an average prediction increase for mesh when we have 2D boundary losses, but we did not see an improvement when we include 3D boundary losses for the voxel branch.

Admittedly, when it comes to training with shapes that exceed the camera frustum, mesh representation may be disadvantageous if we maintain the prediction shape in a unit cuboid space.

Voxel or point cloud representations would be more suitable for this task since points that fall outside of a certain frustum can be easily filtered. In contrast, a mesh cut algorithm is required for mesh representations and leaves unnatural edges for the network to learn.

Table B.3: Mesh only model comparisons. Our baseline model is Mesh R-CNN with multi-object training and evaluation enabled for all valid annotations in all three heads (instance, voxel, mesh); (2): multi-object training with a partial loss on voxel and mesh heads, (3): multi-object training with expanded partial loss where voxel head includes objects that fell outside of camera frustum. N indicates the number of annotations used to regress each corresponding head during training.

	N instances	N voxels	N meshes	AP^{box}	AP^{mask}	AP^{mesh}
Mesh R-CNN*	16175	16175	16175	37.8 ± 1.4	34.2 ± 1.9	5.9 ± 0.4
(2)	55216	16176	16175	56.5 ± 0.9	52.6 ± 1.1	8.9 ± 1.5
(3)	55216	28182	16175	58.4 ± 1.2	54.6 ± 1.1	8.9 ± 0.5

B.3 Datasets details

Figure B.3 shows the instance distribution per categories. The top 10 categories for instances (ranked from most to least) are: dining chair, pendant lamp, double bed, nightstand, wardrobe, dining table, tea table and tv stand. The top 10 panoptic categories are (ranked from most to least by number of segments): floor, wallinner, baseboard, dining chair, pocket, pendant lamp, customizedceiling, double bed, nightstand, wardrobe.

B.3.1 Baseline Multi-modality Results

The RMSE and RMSE_log values for layout depth is 0.142 and 0.518 respectively for our final model. The 2D panoptic evaluation results for the panoptic branch of the end-to-end network is shown in Table B.4. Note that the numbers here are shown with $\times 100$.

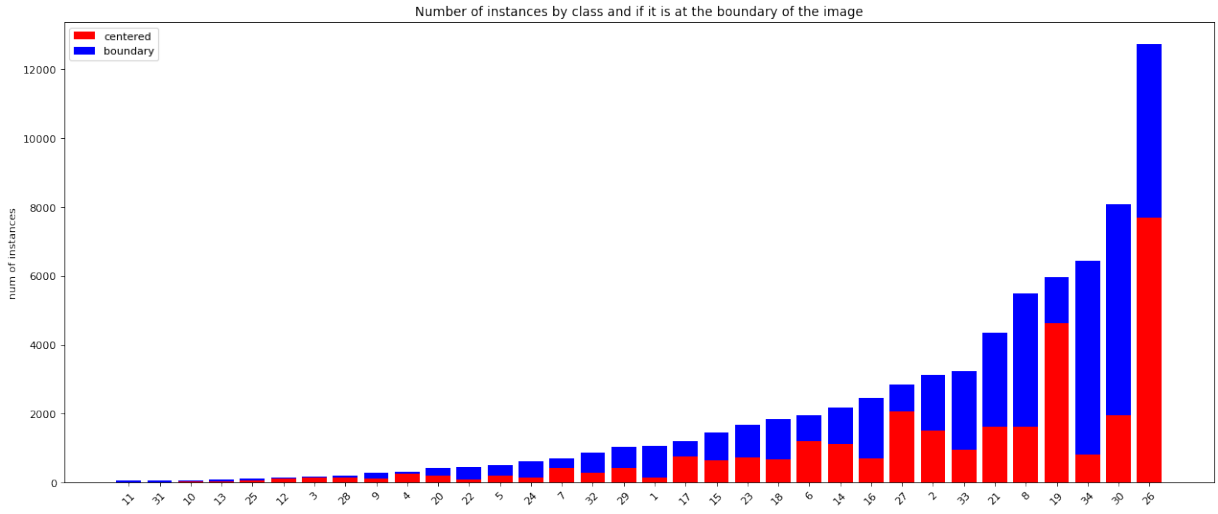


Figure B.2: Number of instances per class by centered (red) vs boundary (blue). Data is collected from the 10 subsets for the training dataset.

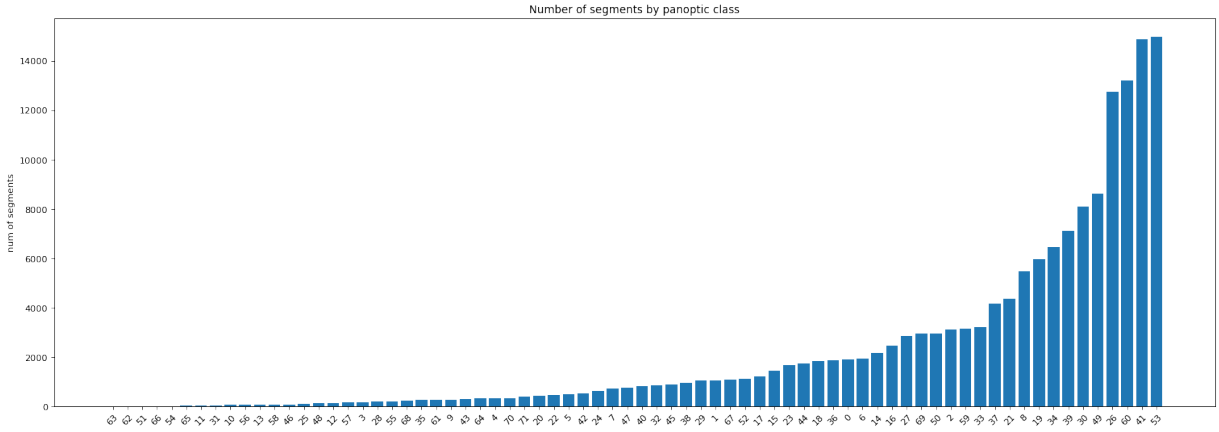


Figure B.3: Number of segments per class for panoptic segmentation. Data is collected from the 10 subsets for the training dataset.

Table B.4: The 2D panoptic evaluation results with 3D FRONT for our end-to-end network.

	PQ ↑	SQ ↑	RQ ↑
All	21.17	60.18	25.55
Things	22.93	69.98	28.89
Stuff	19.55	51.18	22.49

Bibliography

[ADD+19] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[ASZ+16] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[AW18] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv e-prints*, abs/1812.11941, 2018.

[Bas93] Ronen Basri. Viewer-centered representations in object recognition: A computational approach. In *Handbook of pattern recognition and computer vision*, pages 863–882. World Scientific, 1993.

[BBZ+16] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. GIFT: A real-time and scalable 3d shape search engine. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[BCB15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Int. Conf. Learn. Represent.*, 2015.

[BGLA18] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers and Graphics*, 71:189–198, 2018.

[Bie87] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115, 1987.

[BM00] E. Darcy Burgund and Chad J. Marsolek. Invariant and viewpoint-dependent object recognition in dissociable neural subsystems. *Psychonomic Bulletin & Review*, 7(3):480–489, 2000.

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[BRG16] Aayush Bansal, Bryan C. Russell, and Abhinav Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[BS83] Dana H. Ballard and Daniel Sabbah. Viewer independent shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):653–660, 1983.

[BSA17] Alexandre Boulch, B. L. Saux, and N. Audebert. Unstructured point cloud semantic labeling using deep segmentation networks. In *3DOR*, 2017.

[CFG+15] Angel X. Chang, A. Thomas 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 1512.03012*, 2015.

[CGKW18] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *Int. Conf. Learn. Represent.*, 2018.

[CN19] Xu Cao and Katashi Nagao. Point cloud colorization based on densely annotated 3d shape dataset. In *ACM Int. Conf. Multimedia*, 2019.

[CNH+20] Qimin Chen, Vincent Nguyen, Feng Han, Raimondas Kiveris, and Zhuowen Tu. Topology-aware single-image 3D shape reconstruction. *CVPR Workshop on Learning 3D Generative Models*, 2020.

[Com18] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[COR+16] 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. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[CSKG17] R Qi Charles, Hao Su, Mo Kaichun, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[CTSO03] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. In *Computer graphics forum*, 2003.

[CZ17] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[CZ19] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[DCS+17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 248–255, 2009.

[DFAG17] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017.

[DKNV17] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[DM97] Vaibhav A. Diwadkar and Timothy P. McNamara. Viewpoint dependence in scene recognition. *Psychological Science*, 8(4):302–307, 1997.

[DSW+19] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019.

[DT20] Maximilian Denninger and Rudolph Triebel. 3D scene reconstruction from a single viewport. *Eur. Conf. Comput. Vis.*, 2020.

[DYC+17] Qi Dou, Lequan Yu, Hao Chen, Yueming Jin, Xin Yang, Jing Qin, and Pheng-Ann Heng. 3d deeply supervised network for automated segmentation of volumetric medical images. *Medical Image Analysis*, 2017.

[EPF14] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inform. Process. Syst.*, 2014.

[ERLF21] Francis Engelmann, Konstantinos Rematas, Bastian Leibe, and Vittorio Ferrari. From Points to Multi-Object 3D Reconstruction. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[EVGW+] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[FJG+20] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3D-FUTURE: 3D furniture shape with texture. *arXiv preprint arXiv:2009.09633*, 2020.

[FSG17] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. MIT Press, 2016.

[GCL+20] Meng-Hao Guo, Junxiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: point cloud transformer. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/2012.09688, 2020.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[GFK+18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[GLSU13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[GMJ19] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. *Int. Conf. Comput. Vis.*, 2019.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Adv. Neural Inform. Process. Syst.*, 2014.

[GZC15] Kan Guo, Dongqing Zou, and Xiaowu Chen. 3d mesh labeling via deep convolutional neural networks. *ACM Trans. Graph.*, 35(1):3, 2015.

[Hay12] William G. Hayward. Whatever happened to object-centered representations? *Perception*, 41(9):1153–1162, 2012.

[HCH+19] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *Int. Conf. Comput. Vis.*, 2017.

[HLvdMW17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[HQX+18] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3D object, layout, and camera pose estimation. *Adv. Neural Inform. Process. Syst.*, pages 206–217, 2018.

[HQZ+18] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3D scene parsing and reconstruction from a single RGB image. *Eur. Conf. Comput. Vis.*, pages 187–203, 2018.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HWN18] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation on point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1802.04402, 2018.

[HYX+19] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1911.11236, 2019.

[HZ04] Feng Han and Song-Chun Zhu. Automatic single view building reconstruction by integrating segmentation. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 53–53, 2004.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[ISS17] Hamid Izadinia, Qi Shan, and Steven M Seitz. IM2CAD. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[JSQJ18] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia1. Gal: Geometric adversarial loss for single-view 3D-object reconstruction. *Eur. Conf. Comput. Vis.*, pages 802–816, 2018.

[JWL18] Mingyang Jiang, Yiran Wu, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *CoRR*, abs/1807.00652, 2018.

[JXYY13] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, 2013.

[KALD20] Weicheng Kuo, Anelia Angelova, Tsung-Yi Lin, and Angela Dai. Mask2CAD: 3D shape prediction by learning to segment and retrieve. *Eur. Conf. Comput. Vis.*, 2020.

[Kan18] Asako Kanezaki. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[KBJM18] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[KFR03] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In *Symposium on Geometry Processing (SGP)*, volume 6, 2003.

[KGHD19] Alexander Kirillov, Ross B. Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[KHG+18] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic Segmentation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[KMH+] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An interactive 3D environment for visual AI.

[KPNK03] M Kortgen, GJ Park, M Novotni, and R Klein. 3d shape matching with 3d shape contexts. In *In the 7th Central European Seminar on Computer Graphics*, 2003.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.*, 2012.

[KSZ+21] Adam R. Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Soňa Mokrá, and Danilo J. Rezende. Nerf-vae: A geometry aware 3d scene generative model, 2021.

[KT03] Takio Kurita and Takashi Takahashi. Viewpoint independent face recognition by competition of the viewpoint dependent classifiers. *Neurocomputing*, 51:181–195, 2003.

[KZH19] Artem Komarichev, Zichun Zhong, and Jing Hua. A-CNN: annularly convolutional neural networks on point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1904.08017, 2019.

[LBD+89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[LBSC18] Yangyan Li, Rui Bu, Mingchao Sun, and Baoquan Chen. Pointcnn. *Adv. Neural Inform. Process. Syst.*, abs/1801.07791, 2018.

[LFXP19] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1904.07601, 2019.

[Lin] Riccardo Lincetto. https://github.com/riclincio/3d-shape-classification.

[Lin09] Wilfried Linder. *Digital photogrammetry*. Springer, 2009.

[LJ07] Haibin Ling and David W Jacobs. Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):286–299, 2007.

[LLT19] Justin Lazarow, Kwonjoon Lee, and Zhuowen Tu. Learning instance occlusion for panoptic segmentation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[LMB+14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *Eur. Conf. Comput. Vis.*, 2014.

[LNH14] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(3):453–465, 2014.

[LP95] Nikos K. Logothetis and Jon Pauls. Psychophysical and physiological evidence for viewer-centered object representations in the primate. *Cerebral Cortex*, 5(3):270–288, 1995.

[LPP95] Nikos K Logothetis, Jon Pauls, and Tomaso Poggio. Shape representation in the inferior temporal cortex of monkeys. *Current Biology*, 5(5):552–563, 1995.

[LPS07] Simon Lacey, Andrew Peters, and Krish Sathian. Cross-modal object recognition is viewpoint-independent. *PLoS One*, 2(9):e890, 2007.

[LS17] Loïc Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *CoRR*, abs/1711.09869, 2017.

[LS18] Zhengqi Li and Noah Snavely. MegaDepth: Learning single-view depth prediction from internet photos. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[LSS08] Joerg Liebelt, Cordelia Schmid, and Klaus Schertler. Viewpoint-independent object class detection using 3d feature maps. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2008.

[LVC+19] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1812.05784, 2019.

[LWD+18] Bo Liu, Xudong Wang, Mandar Dixit, Roland Kwitt, and Nuno Vasconcelos. Feature space transfer for data augmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[LYS+21] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Milos Hasan, Ravi Ramamoorthi, and Manmohan Chandraker. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/2007.12868, 2021.

[Mar82] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., Inc., USA, 1982.

[MB19] Priyanka Mandikal and R. Venkatesh Babu. Dense 3d point cloud reconstruction using a deep pyramid network. *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, abs/1901.08906, 2019.

[MCL+14] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[MF91] Patricia A. McMullen and Martha J. Farah. Viewer-centered and object-centered representations in the recognition of naturalistic line drawings. *Psychological Science*, 2(4):275–278, 1991.

[MGLM18] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel VAE net with group convolutions for point cloud segmentation. *CoRR*, abs/1811.04337, 2018.

[Mil12] Branka Milivojevic. Object recognition can be viewpoint dependent or invariant– it's just a matter of time and task. *Frontiers in Computational Neuroscience*, 6:27, 2012.

[Min19] Patrick Min. binvox. `http://www.patrickmin.com/binvox` or `https://www.google.com/search?q=binvox`, 2004 - 2019. Accessed: 2020-06-29.

[MNB18] Priyanka Mandikal, K L Navaneet, and R Venkatesh Babu. 3D-PSRNet: Part segmented 3D point cloud reconstruction from a single image. 2018.

[MON+19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[MST+20a] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020.

[MST+20b] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934, 2020.

[MZC+19] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[NG21] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[NHG+20] Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. Total3DUnderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020.

[NT03] Fakir S. Nooruddin and Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.*, 9(2):191–205, 2003.

[OMT+20] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. 2020.

[PBF20] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. CoReNet: Coherent 3D scene reconstruction from a single RGB image. *Eur. Conf. Comput. Vis.*, 2020.

[PFS+19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox,

and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[PKVG99] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[PM15] Adrien Payan and Giovanni Montana. Predicting alzheimer's disease: a neuroimaging study with 3d convolutional neural networks. *arXiv preprint arXiv:1502.02506*, 2015.

[PNF+08] Marc Pollefeys, David Nistér, J-M Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, et al. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008.

[PSdV+18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018.

[QJL+19] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3014–3023, 2019.

[QSN+16] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[QYSG17] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst.*, 2017.

[RBB09] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Robotics and Automation*, 2009.

[RD87] Irvin Rock and Joseph DiVita. A case of viewer-centered object perception. *Cognitive psychology*, 19(2):280–293, 1987.

[RHGS15] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inform. Process. Syst.*, 2015.

[RL17] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *Int. Conf. Learn. Represent.*, 2017.

[RMBF21] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. ShaRF: Shape-conditioned radiance fields from a single view. *https://arxiv.org/pdf/2102.08860.pdf*, 2021.

[Rob63] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[RP20] Mike Roberts and Nathan Paczan. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. *arXiv*, 2020.

[RRN+20] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.

[SFH18] Daeyun Shin, Charless C. Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[SGMN13] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *Adv. Neural Inform. Process. Syst.*, 2013.

[SJS+18] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[SKK21] Karl Stelzner, Kristian Kersting, and Adam R. Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *CoRR*, abs/2104.01148, 2021.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[SLX15] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[SM71] Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

[SMKLM15] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Int. Conf. Comput. Vis.*, 2015.

[SSFFS09] Hao Su, Min Sun, Li Fei-Fei, and Silvio Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Int. Conf. Comput. Vis.*, 2009.

[SSN09] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: Learning 3D scene structure from a single still image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[SWZ+18] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and methods for single-image 3D shape modeling. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[SYS+16] Manolis Savva, Fisher Yu, Hao Su, M Aono, B Chen, D Cohen-Or, W Deng, Hang Su, Song Bai, Xiang Bai, et al. Shrec'16 track: large-scale 3d shape retrieval from shapenet core55. In *Eurographics Workshop on 3D Object Retrieval*, 2016.

[SYZ+16] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *arXiv preprint arXiv:1611.08974*, 2016.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015.

[Sze10] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[SZW19] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Adv. Neural Inform. Process. Syst.*, 2019.

[TBF+15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Int. Conf. Comput. Vis.*, 2015.

[TCA+17] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. *CoRR*, abs/1710.07563, 2017.

[TCYZ05] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *Int. J. Comput. Vis.*, 63(2):113–140, 2005.

[TGF+18] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2D image of a 3D scene. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[THGZ18] Gusi Te, Wei Hu, Zongming Guo, and Amin Zheng. RGCNN: regularized graph CNN for point cloud segmentation. *CoRR*, abs/1806.02952, 2018.

[TRR+19] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[TV02] Michael J. Tarr and Quoc C. Vuong. Visual object recognition. *Stevens' handbook of experimental psychology*, 2002.

[TWT+17] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *arXiv preprint arXiv:1711.11248*, 2017.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

[way19] Waymo open dataset: An autonomous driving dataset, 2019.

[WHH+19] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. 2019.

[WLS+19] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. *CoRR*, abs/1902.09852, 2019.

[WP15] Dominic Zeng Wang and Ingmar Posner. Voting for voting in online point cloud object detection. In *Robotics: Science and Systems*, 2015.

[WQL19] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1811.07246, 2019.

[WSK+15] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[WSL+18] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, abs/1801.07829, 2018.

[WSWL14] Zizhao Wu, Ruyang Shou, Yunhai Wang, and Xinguo Liu. Interactive shape co-segmentation via label propagation. *Computers & Graphics*, 38:248–254, 2014.

[WZL+18] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single rgb images. *Eur. Conf. Comput. Vis.*, 2018.

[XBK+15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[XGD+17] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[XKC+16] 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 *Eur. Conf. Comput. Vis.*, 2016.

[XLCT18] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[XLZ+19] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. UPSNet: A unified panoptic segmentation network. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[XPMB21] Christopher Xie, Keunhong Park, Ricardo Martin-Brualla, and Matthew Brown. Fig-nerf: Figure-ground neural radiance fields for 3d object category modelling. *CoRR*, abs/2104.08418, 2021.

[XRT12] Jianxiong Xiao, Bryan Russell, and Antonio Torralba. Localizing 3d cuboids in single-view images. In *Adv. Neural Inform. Process. Syst.*, 2012.

[XSH+17] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 2017.

[XSW+20] Qiangeng Xu, Xudong Sun, Cho-Ying Wu, Panqu Wang, and Ulrich Neumann. Grid-gcn for fast and scalable point cloud learning. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1912.02984, 2020.

[XWC+19] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3D reconstruction. *Adv. Neural Inform. Process. Syst.*, pages 492–502, 2019.

[YKC+16] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, ARCewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 35(6):210, 2016.

[YLH+18] Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. *Eur. Conf. Comput. Vis.*, 2018.

[YLSL21] Wentao Yuan, Zhaoyang Lv, Tanner Schmidt, and Steven Lovegrove. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. *arXiv preprint arXiv:2101.01602*, 2021.

[YZL+20] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/2003.00492, 2020.

[YZN+19] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1904.03375, 2019.

[YZW+18] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas. GSPN: generative shape proposal network for 3d instance segmentation in point cloud. *CoRR*, abs/1812.03320, 2018.

[ZCSH18] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. Layoutnet: Reconstructing the 3D room layout from a single rgb image. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[ZCZ+21] Cheng Zhang, Zhaopeng Cui, Yinda Zhang, Bing Zeng, Marc Pollefeys, and Shuaicheng Liu. Holistic 3d scene understanding from a single image with implicit representation. *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8833–8842, June 2021.

[ZL17] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *Int. Conf. Learn. Represent.*, 2017.

[ZPD+20] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020.

[ZPK18] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[ZX19] Wenxiao Zhang and Chunxia Xiao. PCAN: 3d attention map learning using contextual information for point cloud based retrieval. *IEEE Conf. Comput. Vis. Pattern Recog.*, abs/1904.09793, 2019.

[ZZZ+18] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B Tenenbaum, William T Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. *Adv. Neural Inform. Process. Syst.*, 2018.