Holistic Indoor Scene Understanding, Modelling and Reconstruction from Single Images

by

Yinyu Nie

National Centre for Computer Animation

Faculty of Media & Communication

Bournemouth University

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of *Doctor of Philosophy*

Jan. 2021

Copyright Statement

Acknowledgements

I would like to thank my supervisors, Prof. Jian Chang and Prof. Jian J Zhang, for their continuous supports and guidance during my entire Ph.D. life. Their priceless suggestions and supervisions motivate me to dive deeper into my topic and inspire me to go further with my own research philosophy.

I also show my deep gratitude to Dr. Shihui Guo from Xiamen University and Dr. Xiaoguang Han from the Chinese University of Hong Kong (Shenzhen). During each milestone of my Ph.D. project, their precious expertise and insights towards cutting-edge knowledge stimulate me to chase better research quality and help me to build my own aesthetics for valuable work.

Besides my supervisors and advisors, I really appreciate the days I have had with my labmates and friends. I express my sincere thanks to Jinglu Zhang for all the days and nights working together before deadlines, and for all the fun we have had in the past years. I would like to thank Dr. Yunfei Fu, Zhangcan Ding, Yao Lyu and Yanran Li for all the interesting ideas and academic supports during my entire Ph.D. life. I also extend my most gratitude to Yukun Wang, Xinglong Wang and Ruibin Wang. Their vision and altitude toward life always broaden my horizons and inspire me to be a better person.

My deep gratitude also goes to our faculty staff, Tanesha Duff, Tomasz Knutel, Sonia Ashby, Cansu Kurt Green, Sunny Choi and Jan Lewis, for their generous help during my research.

Lastly, I would like to express my sincere gratitude to my family and my girlfriend for their continuous support, and I am also

grateful and really thank the China Scholarship Council for the financial funding which enables me to finish my project and contribute to the research community.

Abstract

3D indoor scene understanding in computer vision refers to perceiving the semantic and geometric information in a 3D indoor environment from partial observations (e.g. images or depth scans). Semantics in a scene generally involves the conceptual knowledge such as the room layout, object categories, and their interrelationships (e.g. support relationship). These scene semantics are usually coupled with object and room geometry for 3D scene understanding, for example, layout plan (i.e. location of walls, ceiling and floor), shape of in-room objects, and a camera pose of observer. This thesis focuses on the problem of holistic 3D scene understanding from single images to model or reconstruct the indoor geometry with enriched scene semantics. This challenging task requires computers to perform equivalently as human vision system to perceive and understand indoor contents from colour intensities. Existing works either focus on a sub-problem (e.g. layout estimation, 3D detection or object reconstruction), or addressing this entire problem with independent subtasks, while this thesis aims to an integrated and unified solution toward semantic scene understanding and reconstruction.

In this thesis, scene semantics and geometry are regarded intertwined and complementary. Understanding each part (semantics or geometry) helps to perceive the other one, which enables joint scene understanding, modelling & reconstruction. We start by the problem of semantic scene modelling. To estimate the object semantics and shapes from a single image, a feasible scene

modelling streamline is proposed. It is backboned with fully convolutional networks to learn 2D semantics and geometry, and powered by a top-down shape retrieval for object modelling. After this, We build a unified and more efficient visual system for semantic scene modelling. Scene semantics are divided into relational (i.e. support relationship) and non-relational (i.e. object segmentation & geometry, room layout) knowledge. A Relation Network is proposed to estimate the support relations between objects to guide the object modelling process. Afterwards, We focus on the problem of holistic and end-to-end scene understanding and reconstruction. Instead of modelling scenes by top-down shape retrieval, this method bridges the gap between scene understanding and object mesh reconstruction. It does not rely on any external CAD repositories. Camera poses, room layout, object bounding boxes and meshes are end-to-end predicted from an RGB image with a single network architecture. At the end, We extend our work by using a different input modality, single-view depth scan, to explore the object reconstruction performance. A skeleton-bridged approach is proposed to predict the meso-skeleton of shapes as an intermediate representation to guide surface reconstruction, which outperforms the prior-arts in shape completion.

Overall, this thesis provides a series of novel approaches towards holistic 3D indoor scene understanding, modelling and reconstruction. It aims at automatic 3D scene perception that enables machines to understand and predict 3D contents as human vision, which we hope could advance the boundaries of 3D vision in machine perception, robotics and Artificial Intelligence.

Contents

1 Introduction **1**

1.1 Background and Motivation 1
1.2 Research Questions 5
1.3 Research Targets 6
1.4 Research Hypotheses 7
1.5 Contributions 8
1.6 Chapter Structure 10

2 Literature Review **12**

2.1 2D Scene Perception 12
2.1.1 2D Layout Estimation 13
2.1.2 Semantic Instance Detection 14
2.1.3 Depth Estimation 17
2.1.4 Support Inference 19
2.2 3D Scene Perception from 2D images 21
2.2.1 3D Layout Estimation 21
2.2.2 3D Object Detection 22
2.2.3 Shape Recovery for Object Instances 23
2.2.3.1 Shape Generation 23
2.2.3.2 Shape Completion 23
2.2.3.3 Skeleton-guided Surface Generation 25
2.3 3D Scene Modelling and Reconstruction 25
2.3.1 Scene Modelling by Shape Retrieval 26
2.3.2 Scene Reconstruction 27
2.4 Datasets and Metrics 29

2.4.1 Datasets 29
2.4.2 Metrics 31

3 Semantic Scene Modelling **34**
3.1 Method Overview 34
3.2 Instance Segmentation with Fully Convolutions 36
3.3 Depth Estimation from an RGB Image 37
3.4 2D Layout Estimation 38
3.5 Prior-based Support Inference 38
3.6 Scene Modelling with 3D Shape Retrieval 41
3.6.1 Matching Problem Formulation 42
3.6.2 Global Searching 45
3.6.3 Local Matching 45
3.7 Experiments and Discussions 45
3.7.1 Performance Evaluation 46
3.7.2 Comparisons and Limitations 48
3.8 Summary 50

4 Unified Scene Understanding and Modelling **51**
4.1 Method Overview 52
4.2 Non-relational Semantics Parsing 53
4.3 Relational Support Reasoning 55
4.3.1 Relational Reasoning with Visual-Question Answering . 56
4.3.2 Formulation for Support Inference 56
4.4 Global Scene Optimization 58
4.4.1 Scene Initialization 59
4.4.2 Contextual Refinement and Scene Modelling 63
4.5 Experiments and Analysis 65
4.5.1 Efficiency Analysis 66
4.5.2 Qualitative Evaluation 66
4.5.3 Quantitative Evaluation 68
4.5.4 Discussions 71
4.6 Summary 73

5 Toward Total 3D Scene Understanding and Mesh Reconstruction 75
5.1 Method Overview . . . 76
5.2 3D Object Detection and 3D Layout Estimation . . . 78
5.3 Density-aware 3D Mesh Generation . . . 80
5.3.1 Density Definition on Mesh Points . . . 81
5.3.2 Mesh Generation Network . . . 82
5.4 End-to-end Learning for Total 3D Understanding . . . 83
5.5 Results and Evaluation . . . 84
5.5.1 Experiment Setup . . . 84
5.5.2 Qualitative Analysis and Comparison . . . 85
5.5.2.1 Object Reconstruction . . . 85
5.5.2.2 Scene Reconstruction . . . 86
5.5.3 Quantitative Analysis and Comparison . . . 87
5.6 Ablation Analysis and Discussion . . . 91
5.7 Summary . . . 92

6 Skeleton-bridged Shape Completion 94
6.1 Method Overview . . . 95
6.2 Learning Meso-Skeleton with Global Inference . . . 98
6.3 Skeleton-to-Surface Reconstruction with Non-Local Attention 98
6.3.1 Non-Local Attention . . . 98
6.3.2 Learning Surface from Skeleton . . . 100
6.3.3 Surface Adjustment with Local Guidance . . . 101
6.4 Loss Functions for End-to-end Training . . . 102
6.5 Experiment Setups . . . 104
6.5.1 Datasets . . . 104
6.5.2 Implementation . . . 104
6.5.3 Running Time . . . 104
6.5.4 Benchmark Configuration . . . 105
6.5.5 Comparisons with Point Completion Methods . . . 105
6.5.6 Comparisons with Mesh Reconstruction Methods . . . 106
6.5.7 Ablation Analysis . . . 107

6.5.8 Discussions . . . 109
6.6 Summary . . . 111

7 Conclusion and Future Work **112**
7.1 Conclusion . . . 112
7.2 Future Work . . . 114

Bibliography **117**

A Supplementary Material for Chapter 3 **140**
A.1 Parameter setting . . . 140
A.2 Room corner searching method . . . 141

B Supplementary Material for Chapter 4 **143**
B.1 Technical illustrations . . . 143
B.1.1 Indoor scene segmentation . . . 143
B.1.2 Model Retrieval . . . 144
B.1.3 Relation Network . . . 144
B.1.4 Global scene optimisation . . . 145
B.2 Priors for support inference and height estimation . . . 146
B.3 2D object segmentation comparisons with existing works . . . 146
B.4 Intermediate results in scene modelling . . . 148

C Supplementary Material for Chapter 5 **156**
C.1 Camera and World System Setting . . . 156
C.2 Network Architecture . . . 156
C.3 3D Detection on SUN RGB-D . . . 158
C.4 Object Class Mapping . . . 160
C.5 More Comparisons of Object Mesh Reconstruction on Pix3D . 160
C.6 More Samples of Scene Reconstruction on SUN RGB-D . . . 161

D Supplementary Material for Chapter 6 **163**
D.1 Network Architecture and Parameters . . . 163
D.1.1 Learning Meso-Skeleton with Global Inference . . . 163
D.1.2 Skeleton2Surface with Non-local Attention . . . 164

D.1.3 Surface Adjustment with Local Guidance 165
D.2 Data Preparation 165
D.3 More Qualitative Comparisons 167
D.4 More Quantitative Comparisons 167
D.4.1 Comparisons on Extra Categories and Metrics 167
D.4.2 Discussions on Normal Estimation 168

List of Figures

1.1 Lawrence Roberts's Ph.D. thesis in 1963. Given an input photo (a) as the input, it recovered the multiple shapes in a 3D scene, and rendered them to a novel viewpoint (b). . . . 2
1.2 Manhattan World Assumption. 2
1.3 Layout estimation. Left to right: input image; 2D room layout; 3D room layout and camera pose (coloured arrows) by fitting the 2D layout with a cuboid. 3
1.4 3D object detection from images. 4

2.1 2D layout estimation (Ren et al. 2016). The instance labels in the geometric context map indicate the surfaces of floor, ceiling or walls. 14
2.2 NYU V2 Dataset for support inference (Silberman et al. 2012). 19

3.1 Pipeline of our method 35
3.2 Object instance segmentation on an indoor image. 37
3.3 Point cloud retrieval 37
3.4 Room layout estimation 39
3.5 Support priors from SceneNN dataset 40
3.6 Searching instances with a support relationship. 41
3.7 Support hierarchy 42
3.8 Point cloud and the matched model 42
3.9 Ground truth data 46
3.10 Semantic modelling results. (a) Test images; (b) Instance masks; (c) Depth maps; (d) Layout edge maps; (e) 2D Projections of matched models; (f) Retrieved semantic scenes 47

3.11 Segmentation comparison with Zhang et al. (2015). 49
3.12 Scene modelling result comparison with Liu et al. (2017). . . . 49
3.13 Modelling with interactions when layout estimation fails. . . . 50

4.1 Pipeline of indoor scene modelling from a single image. The whole process is divided into three phases: 1. non-relational semantics parsing (e.g. room layout and object masks); 2. support relationship inference; 3. global scene optimization. . 52
4.2 Instance segmentation samples 54
4.3 CAD model candidates. For each object image, we search our model dataset and output five similar candidates for scene modelling. 55
4.4 Relation Network for support inference. The whole architecture consists of three parts. The vision part and the question part are responsible for encoding object images and related questions separately, and the Relation Network answers these questions based on the image features. 55
4.5 Questions and answers for training 58
4.6 Camera-layout joint estimation. The camera parameters and vanishing points are jointly optimized in Part I, which leads to generate room layout proposals in Part II. The optimal layout is decided by the maximal probability score in layout edge map. 59
4.7 3D room layout with camera orientation (left: original image, right: 3D layout). The coloured arrows represent the camera orientation. The gray arrows respectively point at the floor and walls, which indicates the room layout orientation. 61
4.8 Single-view geometry for object height estimation 61
4.9 Scene modelling with contextual refinement. The leftmost column presents the original RGB images and the corresponding segmentation. The median part shows the scene modelling results by iterations. The rightmost column illustrates the iteration trajectory of `IoU` values correspondingly. 65

4.10 Scene modelling samples on the SUN RGB-D dataset. Each sample consists of an original image (left), the reconstructed scene (raw mesh, middle) and the rendered scene with our estimated camera parameters (right). 67
4.11 Comparison with other methods. (a) and (d): The input images. (b) and (e): Reconstructed scenes from other works. The last row is provided by (Izadinia et al. 2017), and the remaining results are from (Huang et al. 2018b). (c) and (f): Our results. All the input images are from the SUN RGB-D dataset. 68
4.12 Orientation correction. (a) and (d): The object images. (b) and (e): Matched models from MVRN. (c) and (f): Corrected orientations. 72
4.13 Limitation cases. Objects that are segmented with rather few pixels (a), out of our model repository (b) or from 'other category' (right) may not get a proper geometry estimate. For 'non-Manhattan' room layout (d), we fit it with a cuboid. The green and blue lines in (d) respectively represent the 2D room layout and the projection of the 3D layout. 73

5.1 From a single image (left), we simultaneously predict the contextual knowledge including room layout, camera pose, and 3D object bounding boxes (middle) and reconstruct object meshes (right). 76
5.2 Overview of our approach. (a) The hierarchy of our method follows a 'box-in-the-box' manner using three modules: the Layout Estimation Network (LEN), 3D Object Detection Network (ODN) and Mesh Generation Network (MGN). A full scene mesh is reconstructed by embedding them together with joint inference. (b) The parameterisation of our learning targets in LEN and ODN (Huang et al. 2018a). 77
5.3 3D Object Detection Network (ODN) 79

5.4 Mesh Generation Network (MGN). Our method takes as input a detected object which is vulnerable to occlusions, and outputs a plausible mesh. 81
5.5 Mesh reconstruction for individual objects. From left to right: (a) Input images and results from (b) Mesh R-CNN Gkioxari et al. (2019), (c) AtlasNet-Sphere (Groueix et al. 2018a), (d, e) TMN with $\tau = 0.1$ and $\tau = 0.05$ (Pan et al. 2019a), (f) Ours. 87
5.6 Scene reconstruction on SUN RGB-D. Given a single image, our method end-to-end reconstructs the room layout, camera pose with object bounding boxes, poses and meshes. 88

6.1 Given a partial scan of an object (green points, backprojected from a depth image), SK-PCN estimates its meso-skeleton (grey points) to explicitly extract the global structure, and pairs the local-global features for displacement regression to recover the full surface points (blue points) with normals for mesh reconstruction (right). 95
6.2 Network architecture of our method. SK-PCN consists of a shape generator and a patch discriminator. The shape generator produces a meso-skeleton first, and uses it to aggregate the multi-resolution local features on the global surface space for surface completion. The patch discriminator measures the fidelity score of our completion results on the overlapped area with the input scan. The layer specifications are detailed in Appendix D.1. 97
6.3 Illustration of the multi-scale feature aggregation for our skeleton extraction (a) and the Non-Local Attention module to broadcast local details from the partial scan to skeletal points (b). 99
6.4 The pipeline of our surface adjustment module. 101
6.5 Point patch confidence prediction using Li et al. (2019b). Note that $C_d = 64$ and $C'_d = 256$. 101

6.6 Comparisons on point cloud completion. From left to right respectively are: a) input partial scan; b) DMC (Liao et al. 2018a); c) MSN (Liu et al. 2019); d) PF-Net (Huang et al. 2020); e) P2P-Net (Yin et al. 2018); f) ONet (Mescheder et al. 2019); g) PCN (Yuan et al. 2018); h) ours; i) ground-truth scan. 106
6.7 Comparisons on mesh completion. From left to right respectively are: a) input partial scan; b) DMC; c) ONet; d) IF-Net; e) P2P-Net*; f) ours; g) ground-truth mesh. 108
6.8 Mesh reconstruction with the configuration $\boldsymbol{C_1}$, $\boldsymbol{C_2}$, $\boldsymbol{C_3}$ and **Full**. 109
6.9 Skeleton v.s. Coarse points in shape completion. From left to right for each sample: input scan, results bridged with coarse points, and ours (bridged with skeletal points). 109
6.10 Skeleton extraction results. From left to right for each sample: input scan; predicted shape skeleton (2,048 points); and the ground-truth. 110
6.11 Limitation cases. From left to right for each sample: input partial scan, predicted skeleton, points and mesh, ground-truth mesh. 110
6.12 Tests on real scans (Choi et al. 2016). From left to right: image of the target object; input partial scan; predicted point cloud; predicted 3D mesh. 111

A.1 Searching the optimal corner on an edge map of the room layout 141

B.1 Reconstruction of 'thin' structures. 148
B.2 Height distribution for each object category. (1-8 categories) . 149
B.3 Height distribution for each object category. (9-18 categories) 149
B.4 Height distribution for each object category. (19-27 categories) 149
B.5 Height distribution for each object category. (28-37 categories) 150
B.6 Support relationship priors 150
B.7 Intermediate results in scene modelling. 155

C.1 Camera and world systems 157

C.2 Qualitative comparisons between the proposed method and TMN (Pan et al. 2019a) on object mesh reconstruction. From left to right: input images, results from TMN, and our results. 161
C.3 Reconstruction results of test samples on SUN RGB-D 162

D.1 Skeleton Estimation Network. 164
D.2 Network Architecture of Skeleton2Surface. 164
D.3 Upsampling layer in Skeleton2Surface. 165
D.4 Surface adjustment. 165
D.5 Ground-truth data preparation 166
D.6 Input data preparation 166
D.7 More qualitative comparisons on the testing set. 168
D.8 Reconstruction results with different configurations. 170

Chapter 1

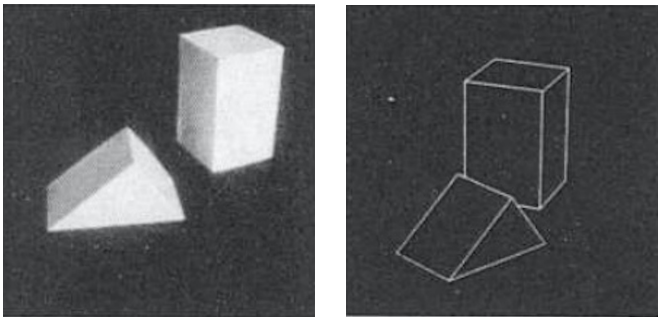
Introduction

1.1 Background and Motivation

3D scene understanding refers to perceive, analyse and interpret the 3D contents in scenes from visual observations. It is a significant branch in computer vision, graphics and robotics, which also has already been widely applied in industrial applications, e.g., virtual interior design, virtual tour, automatic navigation, and digital entertainment with VR/AR devices. By digitizing our surroundings into virtual environments, 3D scene understanding techniques assist people to interpret the 3D representation and infer the scene knowledge from their living world.

Understanding indoor scenes from single images takes a unique significance in computer vision and graphics. It requires our machines to perform equivalently as human vision to perceive and understand indoor contents with only image information. However, in 3D vision, the single-view scene understanding is more an ill-posed problem because of the depth ambiguity and object occlusion, which makes this topic particularly challenging.

The problem of single-view scene understanding can be traced back to 1963. Lawrence Roberts (1963) provided a system to infer multiple 3D objects from a single photo in his Ph.D. thesis (see Figure 1.1). In his work, edge clues from images are extracted to estimate the location and orientation of each single object in a scene. Robert's work demonstrated an early solution of how to perceive object existences and infer the 3D shapes with



(a) (b)

Figure 1.1: Lawrence Roberts's Ph.D. thesis in 1963. Given an input photo (a) as the input, it recovered the multiple shapes in a 3D scene, and rendered them to a novel viewpoint (b).

segmentations from a scene image.

Lawrence Roberts's milestone work presented a meaningful attempt for understanding multiple primitive shapes from single images. However, the cases in real scenes are much more complicated considering the object geometry, illumination and occlusion. To ease the difficulty of 3D scene understanding in real world and make this problem trackable. Coughlan and Yuille (1999 2001) designed a constraint called by 'Manhattan World' assumption. This assumption simplifies the room layout setting and imposes a general regularity: surfaces (e.g., walls, celling and floor) in a indoor scene are aligned with three dominant directions (typically corresponding to X,Y and Z axes, see Figure 1.2).

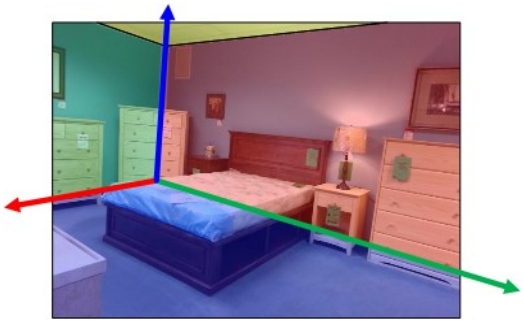


Figure 1.2: Manhattan World Assumption.

With Manhattan World assumption, many methods have been developed

to estimate the room layout for 3D scene understanding. Given the input images, layout estimation refers to predict the location of the ceiling, floor and walls of the indoor room (see Figure 1.3). It has been actively studied since 2007 (Hoiem et al. 2007, Hedau et al. 2009, Lee et al. 2009). The room layout provides boundary information by locating the floor, walls, and the ceiling on 2D images. By fitting the 2D room layout with a 3D cuboid (Nie et al. 2020a, Huang et al. 2018a), it further indicates the camera location relative to the room. In 3D indoor scene understanding, room layout reflects the scope that a camera can roam around, and the current viewpoint the camera is focusing at. It provides the camera coordinate system relative to the whole room, acting as a reference for many downstream tasks such as scene modelling or reconstruction (Izadinia et al. 2017, Nie et al. 2020a b, Huang et al. 2018c).

Figure 1.3: Layout estimation. Left to right: input image; 2D room layout; 3D room layout and camera pose (coloured arrows) by fitting the 2D layout with a cuboid.

3D room layout provides the room boundary geometry though (e.g., walls, floor, celling), it does not take account of the location of indoor objects. On this top, single-view 3D object detection has received a rising development since 2010 (Gupta et al. 2010, Xiao et al. 2012, Choi et al. 2013, Zhang et al. 2014, Huang et al. 2018a 2019). These methods located indoor objects by a 3D bounding box with semantic object label (e.g., chair, table, bed, see Figure 1.4). Although 3D detection approaches understand 3D scenes with instance-level semantics and box locations, the resulting object shapes are cuboids and do not depict the object geometry details.

Figure 1.4: 3D object detection from images.

To reconstruct geometric details of object instances, many works leveraged the advance of LiDAR technology and used active cameras to acquire 3D scans (Chen et al. 2015) as the input to model object shapes. For example, early works (Kim et al. 2012, Nan et al. 2012, Shao et al. 2012, Chen et al. 2014) retrieve the 3D objects with CAD models from shape repositories using numerical optimisation. After that, 3D deep learning reforms this process into a learnable manner that model retrieval can be replaced with deep feature matching (Avetisyan et al. 2019a b 2020, Ishimtsev et al. 2020). Beyond model retrieval, Hou et al. (2020) predicted object shapes with a geometry completion manner learned from 3D scans.

Comparing with using 3D scans as the input, instance shape reconstruction from a single image is particularly challenging. This is primarily because of the inherent ill-posed nature in the problem of single-view reconstruction and the absence of 3D geometric constraints in the input. Single-view shape reconstruction has been intensively studied for decades from traditional Structure-from-Motion (Schonberger and Frahm 2016) to current neural network approaches (Groueix et al. 2018a, Wang et al. 2018a, Tang et al. 2019). These methods focused on the geometry recovery of a single shape without considering the object and scene semantics. So far, previous works either aimed to understand the scene and object semantics, for example, room layout, 3D object labels and locations, or only focused on single shape reconstruction ignoring the scene context. Few works have been done towards both scene semantics understanding and geometry reconstruction in a joint way, which inspired the primary motivation of this thesis: how to jointly understand scene semantics (i.e. layout, object labels, locations) and recon-

struct object geometries - to holistically understand and reconstruct indoor scene contents from a single image.

1.2 Research Questions

The target of this thesis is to develop a 3D vision system that is able to understand and reconstruct the indoor contents (including object semantics and geometric structure) from a single image. In this thesis, instead of solving this problem in one go, we approach the solution from semantic scene modelling to holistic scene understanding and instance reconstruction. Specifically, we firstly aim to model the semantic scenes with object shape retrieval, and then focuses on understanding the scene contextual knowledge to help scene modelling. After that, an end-to-end network architecture is designed for joint semantic instance reconstruction. Lastly, we also attempted to use single depth images for object reconstruction, which indicates the future perspective of 3d scene understanding on hybrid input modalities. To achieve this target, there are four primary research questions that we attempt to address as discussed below:

- **How to estimate shapes from single-view images.** Single-view reconstruction is an inherently ill-posed problem for the absence of depth constraints and self occlusions. How to estimate the geometry of invisible object surfaces serves as the primary challenge, especially for indoor scenes with multiple object instances.

- **How to handle diverse objects in complex 3D scene context.** Different with single object reconstruction, indoor scenes are generally involved with objects in different categories, sizes, locations and diverse camera poses. Besides, the clutter and occlusions between objects makes our task even more complicated. How to locate and reconstruct different types of objects in various scene conditions is another challenge.

- **How to leverage contextual semantics for scene modelling.** Objects in an indoor room generally involve with relational semantics (e.g.,

beds are commonly supported by the floor) which present contextual knowledge and could help object placement. However, to learn and leverage the contextual relations between objects for scene reconstruction with only a single image is non-trivial.

- **How to jointly understand and reconstruct 3D scenes.** Semantic reconstruction of indoor scenes refers to both scene understanding and object reconstruction. Existing works either address one part of this problem (e.g., layout estimation, object detection, shape reconstruction) or focus on independent objects. Nevertheless, 3D scenes manifest contextual knowledge where understanding scene semantics could help object reconstruction, and vice versa. How to jointly understand scenes and reconstruct indoor contents with a single end-to-end network is undoubtedly a challenge for holistic scene understanding.

1.3 Research Targets

Motivated by above questions and challenges, there are four primary objectives we aim to address in this thesis.

- **Semantic scene modelling from single images.** The first objective is to recover the shape geometry of object instances from single images for scene modelling. Particularly, we aim at retrieving the 3D shapes of indoor objects for a single image, and assembling the object shapes into a 3D scene, making it consistent with the input image. The input image only indicates the colour intensities, and indoor objects are commonly occluded. Thus one objective here is to interpret the object features from images for shape retrieval, and another is to estimate physical constraints between objects to address occlusions for object placement. The related content is discussed in Chapter 3, 4.

- **Learning contextual semantics for scene modelling.** The second objective is to predict the relational (e.g. support relations) and non-relational (e.g. category labels, masks, shapes) features between objects

for holistic scene understanding. It extends the above objective by using a unified framework backboned with deep neural networks to learn both relational and non-relational features for scene modelling. The related content is discussed in Chapter 4.

- **Total 3D scene understanding and mesh reconstruction.** The third objective is to understand scene semantics and reconstruct object meshes with an end-to-end network architecture. Previous methods often decouple this problem into single tasks. We aim at bridging the gap between scene understanding and object mesh reconstruction, and proposing an end-to-end solution to jointly reconstruct room layout, object bounding boxes and meshes from a single image. The related content is discussed in Chapter 5.

- **Shape generation from a single depth image.** The fourth objective is to investigate the capability of shape reconstruction from single depth images. Different with colour images, depth maps do not contain object appearance features but provide geometric clues of the visible surface. This objective is to explore the future perspective of 3D scene understanding with multiple input modalities, which is discussed in Chapter 6.

1.4 Research Hypotheses

Before introducing the methods and contributions, I list the hypotheses or assumptions that our methods adapt to. It gives the problem domain that our methods aim at to address the research targets above.

- **Input modality** This thesis focuses on how to understand the semantics and geometry of 3D scenes from single images. The image modality includes RGB images (Chapter 3, 4 and 5) and depth images (Chapter 6). From Chapter 3 to 5, we aims to address the problem of indoor scene understanding, modelling and understanding from single RGB images. Chapter 6 is an attempt of our future work in multi-modality

indoor scene understanding and reconstruction. In this chapter, we recover 3D object geometry from depth images.

- **Room layout** As mentioned in Section 1.1, we follow the Manhattan assumption (Coughlan and Yuille 1999 2001) that walls should be perpendicular to the floor and ceiling. Besides, in our method, we approximate the room layout with a 3D bounding box to present a world coordinate system, as the prior works (Huang et al. 2018a b).

- **Object category** In our 3D object detection or segmentation, we conclude all object categories into 40 classes, following NYU v2 dataset Silberman et al. (2012). It is also introduced in Chapter 2.4.1.

- **Evaluation** All the contributions this thesis claimed are evaluated from two aspects: 1. qualitative results by visual comparisons; 2. quantitative results by numerical comparisons. The metrics used in numerical evaluation are listed in Chapter 2.4.2.

1.5 Contributions

The contribution of this thesis lies on addressing the research questions and targets under the hypotheses above. For each objective in this thesis, we summarize our contributions as follows.

- We attempt to address the problem of semantic scene modelling via deep learning networks and non-linear optimisation (Chapter 3). We propose a system based entirely on fully convolutional networks (FCN) for object feature extraction, and a data-driven support inference approach for hierarchical scene modelling. We have demonstrated that this approach shows effectiveness in modelling objects with severe occlusions. The work leading to this contribution is published in

  Nie, Y., Chang, J., Chaudhry, E., Guo, S., Smart, A. and Zhang, J.J., 2018. Semantic modeling of indoor scenes with support inference from a single photograph. Computer Animation and Virtual Worlds, 29(3-4), p.e1825. (Nie et al. 2018).

- We further target at dense scene modelling with a unified vision system (Chapter 4). We learn support relations between objects with a Relation Network module to address the object occlusions and improve the performance in object placement. We also provide a global optimization strategy for indoor scene synthesis. It incorporates the outputs from former networks and iteratively optimize the 3D scenes to make them contextually consistent with the scene context. Extensive experiments demonstrate the feasibility of our method in understanding and modelling semantics-enriched indoor scenes with different complexity. The work leading to this contribution is published in

  Nie, Y., Guo, S., Chang, J., Han, X., Huang, J., Hu, S.M. and Zhang, J.J., 2020. Shallow2Deep: Indoor scene modeling by single image understanding. Pattern Recognition, 103, p.107271. (Nie et al. 2020a)

- Apart from semantic scene modelling, we provide a solution to jointly reconstruct room layout, object bounding boxes, and meshes from a single image (Chapter 5). It is designed with an end-to-end network architecture for comprehensive 3D scene understanding with mesh reconstruction at the instance level. This integrative approach shows the complementary role of each component. Extensive experiments demonstrate that this method consistently outperforms previous methods on layout estimation, 3D object detection and mesh reconstruction. The work leading to this contribution is published in

  Nie, Y., Han, X., Guo, S., Zheng, Y., Chang, J. and Zhang, J.J., 2020. Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor Scenes from a Single Image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 55-64). (Nie et al. 2020b)

- We design a novel shape completion network to predict the full shape meshes from single depth images (Chapter 6). Existing works usually estimate the missing shape by decoding a latent feature encoded from the input scan. However, real-world objects are usually with diverse topologies and surface details, which a latent feature may fail to represent. To this end, we propose a skeleton-bridged point completion

network (SK-PCN) for shape completion. It predicts the shape skeleton as a global representation to guide shape completion on surface details. Extensive experiments on point and mesh completion show that our approach outperforms the existing methods on various object categories. The work leading to this contribution is published in

Nie, Y., Lin, Y., Han, X., Guo, S., Chang, J., Cui, S. and Zhang, J., 2020. Skeleton-bridged Point Completion: From Global Inference to Local Adjustment. Advances in Neural Information Processing Systems, 33. (Nie et al. 2020c)

Besides the above publications, I have also contributed to other coauthored works as follows. These works are fulfilled during my Ph.D., but not straightforward to tackle the challenges in Section 1.2.

- Du, D., Zhu, H., Nie, Y., Han, X., Cui, S., Yu, Y. and Liu, L., 2020, December. Learning Part Generation and Assembly for Sketching Man-Made Objects. In Computer Graphics Forum.
- Zhang, J., Nie, Y., Lyu, Y., Li, H., Chang, J., Yang, X. and Zhang, J.J., 2020, October. Symmetric Dilated Convolution for Surgical Gesture Recognition. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 409-418). Springer, Cham. (Zhang et al. 2020a)
- Ren, T., Lin, L., Guo, S., Lin, J., Liao, M., Deng, S., Xu, P. and Nie, Y., 2020. Salient object segmentation for image composition: A case study of group dinner photo. Neurocomputing. (Ren et al. 2020)

1.6 Chapter Structure

The remaining chapters in this thesis are organized as follows:

- Chapter 2 presents a thorough review of previous approaches in 3D perception, modelling and reconstruction of 3D scenes.

- Chapter 3 elaborates our method of semantic scene modelling with support inference from a single photograph. Given a single RGB photo, it retrieves object CAD models and organizes their spatial placement into a 3D scene.
- Chapter 4 introduces a unified vision system to holistically perceive, understand and retrieve 3D object models backboned with convolutional neural networks.
- Chapter 5 introduces our architecture 'Total3DUnderstanding'. It jointly learns room layout, camera poses, object bounding boxes and surface meshes with a single end-to-end network.
- Chapter 6 presents our work 'Skeleton-bridged Point Completion'. It discusses how to complete object meshes from only a depth image.
- Chapter 7 concludes the entire thesis and presents several future directions in 3D scene understanding.

Chapter 2

Literature Review

This chapter presents a comprehensive literature review of previous works in scene perception, object modelling and reconstruction that are closely related to our research questions. Specifically, I start from reviewing the related works on 2D scene perception in Section 2.1 including 2D layout estimation, instance detection/segmentation, depth estimation and support inference. These methods produce 2D scene semantics and geometry that lay a foundation for 3D scene understanding. The recent advances in 3D scene perception from 2D images (e.g., layout estimation, object detection and shape prediction in 3D) are detailed in Section 2.2. Lastly, I summarise the previous methods and milestones of 3D scene modelling and reconstruction in Section 2.3.

2.1 2D Scene Perception

2D scene perception refers to estimating the 2D scene semantics and geometry from observations. It is often used as preliminary steps to predict 3D contents. In 2D perception of scene contents, there are four research topics related to this thesis: 1. 2D layout estimation; 2. instance detection/segmentation; 3. depth map estimation; 4. 2D support inference. I summarise the milestone works towards the four topics as follows.

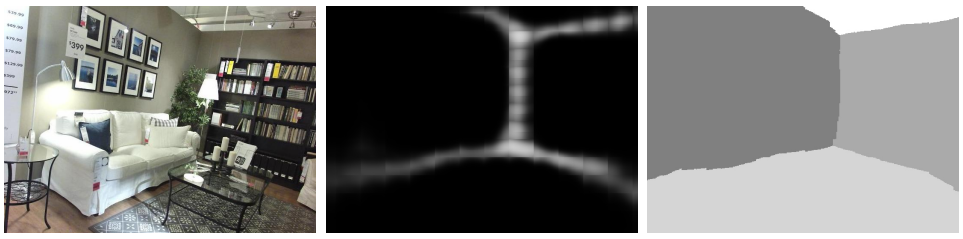
2.1.1 2D Layout Estimation

It is widely accepted that modern layout estimation methods start from Hedau et al.'s work (Hedau et al. 2009). The authors firstly adopted a 3D bounding box to estimate the room structure under the assumption that most faces in a room align with the room directions. With the techniques provided by Hoiem et al. (Hoiem et al. 2007), they labelled pixels with five geometric contents (i.e. left wall, middle wall, right wall, ceiling and floor). Features extracted from these geometric contents are used to train a customized structured learning algorithm, thereby to rank 3D box candidates and obtain the best fitted solution.

This milestone work built feasible framework for modelling room layout as a cuboidal box. Many of existing methods were developed following it to enhance this framework, where the improvements can be divided into two branches: 1. higher inference efficiency (Urtasun et al. 2012, Schwing and Urtasun 2012) and 2. improved feature descriptors (Lee et al. 2009, Ramalingam et al. 2013). Gupta et al. (2010) proposed a volumetric reasoning method, where the 3D volume of indoor objects is parametrised for optimization. It infers the relations between indoor content and room layout faces to reason the structure of a scene. Ramalingam et al. (2013) used Manhattan junctions to deduce room layouts. They provided a voting scheme to label and classify these junctions in a single RGB image and built a conditional random field to infer the room layout. Each room layout was represented by a cuboid by aligning those corners to fit the Manhattan junctions. There are also several works attempted to recover the objects and room layout simultaneously based on the intuition that detecting indoor clutter helps to recover layout structure (Hedau et al. 2010, Wang et al. 2013, Schwing et al. 2013).

With the rising of deep learning techniques, the state-of-the-art layout estimators generally chose convolutional features to guide the bounding box prediction. Mallya and Lazebnik (2015) started this trend. They adopted a Fully Convolutional Network (FCN) to jointly extract an edge map along with geometric context (i.e. pixels labelled with walls, the floor and the ceiling) of room layout from RGB images (see Figure 2.1). The edge map

feature was used to generate layout candidates, followed by a structured regressor to pick out the best layout structure. Similarly, Ren et al. (2016) adopted FCNs to predict edge maps and geometric context jointly. They utilized the edges to guide the generation of vanishing lines to produce layout candidates. Apart from learning edge maps, Dasgupta et al. (2016) adopted FCN to estimate belief maps (each map corresponds to a layout label) for pixel labelling. Based on those belief maps, Dasgupta et al. (2016) provided an post-processing refinement to keep the geometric consistency in indoor labels, as the CNN output usually contains ambiguous edges with a wavy boundary. Since most layout estimation methods require a post-processing step to propose the optimal layout candidate with vanishing points, Zhang et al. (2019) designed a novel pixel-level refinement method to improve the layout proposal quality. Besides, since this post-processing is commonly time-consuming, Lin et al. (2018b) provided a layout-degeneration augmentation method that realises real-time layout prediction.



(a) Input image (b) Edge map (c) Geometric context

Figure 2.1: 2D layout estimation (Ren et al. 2016). The instance labels in the geometric context map indicate the surfaces of floor, ceiling or walls.

2.1.2 Semantic Instance Detection

Semantic instance detection or segmentation on a single image has been intensively researched for decades (see Zaitoun and Aqel (2015), Thoma (2016) and references within). With the rapid progress in convolutional networks, novel architectures have been continuously refreshing the accuracy. In regard to the whole history of massive CNN-based methods in instance recognition, I refer readers to the elaborated survey and related references from Geng

et al. (2018). In this section, I only review the milestone studies and relevant deep learning methods in instance detection and segmentation.

In modern detection or segmentation networks, many state-of-the-art approaches shared, to some extend, a common underlying architecture (i.e. meta-architectures) on the basis of different varieties of feature extractors (or backbones). There are three milestone meta-architectures, Faster R-CNN (Ren et al. 2015), R-FCN (Dai et al. 2016) and SSD (Liu et al. 2016), acting as building blocks in many leading CNN systems (Huang et al. 2017). Meanwhile, these meta-architectures were usually combined with alternative CNN feature extractor such as AlexNet (Krizhevsky et al. 2012), VGG-16 (Simonyan and Zisserman 2014), Resnet-101 (He et al. 2016), Inception v2 (Ioffe and Szegedy 2015) and v3 (Szegedy et al. 2016), to construct a detection or segmentation system.

In the family of Faster R-CNN architectures, an early object detection network is the so-called Region-based ConvNet (RCNN) (Girshick et al. 2014). This architecture was formed by a natural heuristic region proposal search method (i.e. selective search) and backboned by AlexNet. Beginning with an input image, around 2,000 bounding box proposals were generated by selective search. Then those proposals were individually cropped and warpped to feed the AlexNet to extract 4096 feature vectors for each warpped proposal. After that, a support vector machine was trained to classify these objects. As there were repeated convolutions among proposals, it took exhaustive computation. Based on R-CNN, fast R-CNN (Girshick 2015) as an upgraded version, it adopted Region of Interest Pooling (RoIPooling) to share the forward layers of the VGG16 network across each region proposal instead of handling these proposals individually. As only one forward pass was required in CNN feature extraction, it took 213x speed compared with R-CNN at test time Girshick (2015). After all these improvements, the Selective Search method for region proposal generation was still not efficient enough in Fast R-CNN. It was an external algorithm outside of the CNN framework. Therefore, in Faster R-CNN (Ren et al. 2015), the heuristic region proposal method (i.e. selective search) was replaced by region proposal network (RPN), which was based on the insight of reusing those CNN

features from the backbone for region proposal generation instead of relying on the slow selective search method. That enabled box regression and object classification share the same convolutional features, improving detection efficiency to five frames per second. Apart from locating and classifying each object bounding boxes, Mask R-CNN (He et al. 2017) extended Faster R-CNN to segment each object instance at the pixel level. Instead of using VGG16, Mask R-CNN leveraged Feature Pyramid Network (FPN) (Lin et al. 2017) and ResNet101 (He et al. 2016), and appended a branch for estimating object masks, which parallels with the bounding box regressor and object classifier in Faster R-CNN.

Considering the fact that, in Faster R-CNN family, the region proposal extractors (e.g. selective search, RPN) were massively used per image, Dai et al. (2016) developed the region-based fully convolutional network (R-FCN) to minimize the computation in each regions. In this method, regions were cropped at the last feature layer before the prediction, and a position-sensitive cropping method was used to keep translation variance for object detection instead of using ROI pooling. They claimed that the R-FCN can reach the accuracy of Faster R-CNN with higher time efficiency, and the R-FCN-like model (Li et al. 2017a) has won the COCO 2016 instance segmentation challenge.

SSD-like network was another popular meta-architecture (Liu et al. 2016), which referred to those single feedforward convolutional networks to directly predict object labels and anchors without considering proposal generation operations. With this structure, many other designs like Multi-Box (Szegedy et al. 2014) and RPN mentioned above used it to predict box proposals, and some other approaches leveraged it to predict final class labels (e.g. Sermanet et al. (2013), Redmon et al. (2016)) or even poses Poirson et al. (2016).

The new layer structure, Transformer (Vaswani et al. 2017), presented overwhelming performance in natural language translation. In the recent year, it was proven distinctly effective that even exceeds the convolutional networks in traditional vision tasks, e.g., image recognition, classification, super-resolution and object detection (Khan et al. 2021, Han et al. 2020). The work *DETR* (Carion et al. 2020) and its variant *Deformable DETR* (Zhu

et al. 2020) regarded the object detection as a set prediction problem, which removed the redundant hand-crafted components (i.e. non-maximal suppression and anchor box generation) in the Faster R-CNN like architectures and facilitates end-to-end training. Alternatively, they employed a transformer-based encoder-decoder network that reasons the relations between objects and directly outputs the object detections parallelly. Massive experiments on COCO dataset demonstrated the outstanding performance of Transformer against the prior-art architectures.

2.1.3 Depth Estimation

Depth estimation from RGB images has been studied for several decades. The most classical and traditional approach mainly used Multi-view stereo or Structure from Motion (Wang 2011). It was based on the principles of retrieving 3D geometry of objects via estimating depth from disparities between feature pairs in images from different views. While these methods heavily relied on image quality (rich surface textures without transparent or reflective material), and normally needed dozens of images captured around the target object. Its application scope was very narrow considering these requirements, let alone estimation from a single image. Nevertheless, there were several solutions to handle this kind of ill-posed problem. Before the era of deep learning, Markov Random Field (MRF) or Conditional Random Field (CRF) was known as a main paradigm in regressing depth or 3D model structure from pixels or superpixels (Saxena et al. 2006), wherein, the Make3D (Saxena et al. 2009) was known as a famous system. Apart from CRF and MRF, Ladicky et al. (2014) modelled the depth estimation in a joint way, where semantic classes and depth labels were classified simultaneously to improve each other and finally gave a pixel-wise image segmentation and depth labelling. However, this method relied on handcrafted features and a sensitive procedure (i.e. scale alignment) for foreground and background objects, and it showed inability in handling low-resolution images.

Normally, these traditional feature descriptors were too raw and ambiguous for a depth regressor to estimate a compact and smooth depth map. Since

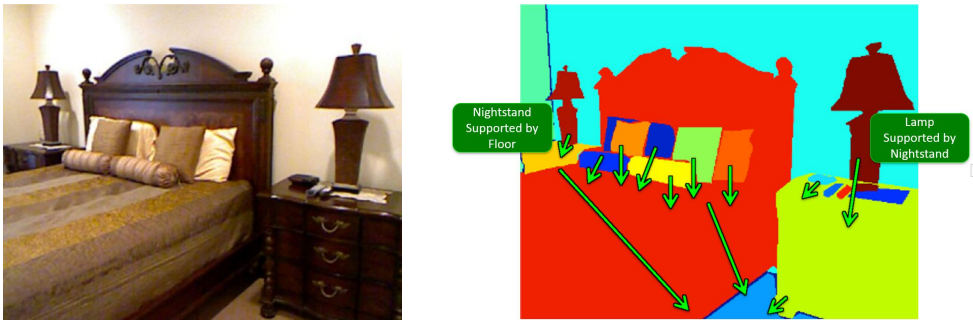
the convolutional networks showed a convincing ability in 2D image perception, it also demonstrated an outstanding performance in understanding the depth from a scene image. In earlier works of using deep learning in depth estimation, Eigen et al. (2014) developed a depth regressor with two network stacks which were accordingly responsible for a coarse estimation and a local fine-tuning. It presented great results both in indoor and outdoor cases. Furthermore, Eigen and Fergus (2015) proposed a multi-scale CNN architecture on that basis to address three computer vision tasks (i.e. depth prediction, surface normal estimation, and semantic labelling), and this architecture outperformed the prior arts in all the three tasks. This work was refined by Ummenhofer et al. (2017), where a depth map along with surface normal and optical flow were calculated simultaneously. Besides that, by using the layer features from convolutional networks, Liu et al. (2015) demonstrated the value of leveraging CNNs as feature extractors for traditional depth regressor (i.e., CRF in their paper). With the success of Fully Convolutional Networks in image segmentation Long et al. (2015), Laina et al. (2016) proposed an FCN architecture for an end-to-end training without the needs of any post-processing. Their network ran in real-time and fewer parameters are required. Furthermore, based on the FCN, Cao et al. (2016) used the Residual Network as the backbone to formulate the depth estimation as a classification problem to classify the depth range instead of regression and outperformed the state-of-the-arts.

In addition to the above supervised learning, there were also several works focused on semi-supervised (Kuznietsov et al. 2017) or unsupervised depth estimation (Garg et al. 2016, Godard et al. 2017) without any (or dense) ground-truth data. Garg et al. (2016) developed a convolutional encoder to predict the depth of the source image, where a stereo pair of images (i.e. source image and target image) were considered with the known camera motion. They adopted a inverse warping method in the estimation process with the target image to help reconstruct the source image and optimise its depth. Similarly, Godard et al. (2017) applied the unsupervised learning on the basis of binocular stereo with epipolar geometry constraints. They advanced the

work of Garg et al. (2016) using bilinear sampling to make the reconstruction loss fully differentiable. Started from Garg et al. (2016), Zhan et al. (2018) expanded its application to visual odometry from monocular stereo sequences. On the other hand, Yu et al. (2020) argued that the challenges of unsupervised depth estimation lie on the massive non-texture areas in 3D scenes. They encoded the piece-wise plane priors into the network and demonstrated its effectiveness in indoor scene depth estimation.

2.1.4 Support Inference

Support relationship provides a sort of geometric constraint between indoor objects to build scenes more robustly. This originates from our daily experience that an object requires some support to counteract the gravity. Support inference from RGB images is an ambiguous problem without knowing the 3D geometry, where occlusions usually make the supporting parts invisible in the field of view (see Figure 2.2). However, the arrangement of indoor furniture generally follows a set of interior design principles and living habits (e.g. tables are mostly supported by the floor; pictures are commonly on walls). These latent patterns behind scenes make the support relationship a kind of priors that can be learned by viewing various indoor rooms.



(a) Original RGB image (b) Support labels

Figure 2.2: NYU V2 Dataset for support inference (Silberman et al. 2012).

Support inference presents higher level geometric clues from the ordinary object semantics (e.g., object category) for scene understanding. The pioneer work of support inference can be traced to Silberman et al. (2012).

They aimed at parsing indoor scenes (captured by RGB-D devices) into support surfaces and object segmentations to interpret the support relations within. In this work, Silberman et al. (2012) firstly aligned the point cloud of the room by searching three dominant orthogonal directions. RANSAC was used to detect planes for RGB-D image segmentation, then integer programming was designed to infer support relationships between these planes. In this paper, three types of support relationship were defined: 1. support from below; 2. support from behind; 3. support from hidden region, which lays a convention in support inference. Silberman et al. (2012) built this early milestone with traditional optimisation strategy, and many subtasks still had bottlenecks at that time (e.g. image segmentation). Following this work, Xue et al. (2015) used SIFT features from segmented regions for area classification (i.e. ground, furniture, object or structure) and provided an energy function for simultaneously classifying regions and support relationships with considering physical stability. Apart from the physical support inference above, there were also a large amount of works focused on inferring support conditions on the perspective of object stability and indoor safety. We direct the readers to Zheng et al. (2015) and Jia et al. (2013) for details. Aiming at predicting support surfaces, Guo and Hoiem (2013) provided an approach to predict horizontal support planes with spatial scale and height. As support inference generally requires a large dataset to learn geometric and semantic priors, building this sort of dataset is rather an arduous manual work. Wong et al. (2015) designed a user-friendly interactive system for RGB-D image annotation. This system facilitated bunches of tedious repetitive work to be automatic, like grouping pixels, recommending object labels, and estimating 3D bounding boxes. Its performance can be improved with the accumulation of human interventions. Inspired by the insights of support relations heavily relying on object semantics, without using any depth clues, Zhuo et al. (2017) provided a pipeline to jointly segment objects, estimate their labels with the involved support relationships under a MRF structure with parameters learned by an SVM (support vector machine). Yang et al. (2017), Huang et al. (2018b) implemented the support inference along with a scene graph, which offers more clues to enhance contextual relations between

each pair of supporting objects. As scene graph is another higher and abstract semantics, it usually can provide a hierarchical affiliation relationship between indoor areas or objects for scene understanding. We refer readers to the definition of scene graph in Liu et al. (2014).

2.2 3D Scene Perception from 2D images

3D scene perception from 2D images refers to reason the 3D contents in indoor scenes. For scene understanding, it mainly contains three tasks, i.e., 3D layout estimation, object bounding box detection and shape reconstruction, which will be reviewed as follows.

2.2.1 3D Layout Estimation

Different from 2D layout map estimation, 3D layout estimation obtains the 3D boundaries of rooms as the output. One popular solution is to extend 2D layout map estimation, and fit the 2D layout with 3D bounding boxes as the outputs (Nie et al. 2020a, Huang et al. 2018b, Izadinia et al. 2017). Since the problem of 2D layout estimation from images is ambiguous in depth, Zhang et al. (2020b) incorporated depth map estimation into networks to provide geometric cues in locating layout surfaces, and the results showed considerable gains in both cuboidal and non-cuboidal rooms. Some methods parameterised the 3D layout cuboid into a 7-DoF vector (e.g. 3-D center, 3-D size and 1-D heading angle). Thus the 3D layout estimation problem can be converted into regressing the 7-DoF vectors from images (Huang et al. 2018a, Nie et al. 2020b). In contrast, Tulsiani et al. (2018) formulated 3D layout into inverse depth estimation, they predicted the room layout as the disparity map of scenes where no objects exist. Avetisyan et al. (2020) characterised the layout surfaces (floor, ceiling and walls) into planer elements, and predicted the corner junctions and edges of these planes to represent the 3D layout. With a similar modality, Stekovic et al. (2020) formulated it into a discrete optimisation problem to obtain the optimal 3D polygons that

construct the layout structure. Besides, they also provided a 'render-and-compare' approach and improve the layout estimate iteratively to address the occlusion issues between indoor contents.

2.2.2 3D Object Detection

3D object detection from single images has been actively studied in the recent decade. Prior to deep learning era, Gupta et al. (2010) leveraged the traditional image processing tools (line segment and vanishing point detection) to propose object cube hypothesises, followed by a post-evaluation to decide the final scene configuration. Not relying on vanishing points and camera intrinsics, Xiao et al. (2012) developed a part-based discriminative detector that describes object boxes with corners and internal edges. As this manner presents compatibility across different object sizes, it was proven effective on images with different viewpoints, aspect ratios and object categories. Beyond standard cameras, Zhang et al. (2014) extended the 3D detection on panoramas. Similar to Gupta et al. (2010), object hypothesises were proposed from bottom to up using the image features, i.e. vanishing points, edges and segmentations. Then a trained support vector machine (SVM) was adopted to rank these proposals and choose the optimal one. To holistically understand a 3D scene, Choi et al. (2013) proposed a hierarchical scene model, namely 3D geometrical phrases, to reason the semantic and geometric relationships between 3D objects. This relational scene configuration presented better explanation on scene semantics and geometry towards the holistic 3D scene understanding.

Image-based 3D object detection receives rising development with the advent of deep learning and the open-sourcing of indoor scene datasets (e.g., NYU V2 (Silberman et al. 2012), SUN-RGBD (Song et al. 2015), ScanNet (Dai et al. 2017a)). To make the 3D object boxes learnable, Huang et al. (2018a) parameterised the bounding boxes into 7-DoF vectors (i.e. 3-D center, 3-D size, 1-D orientation). Then the bounding boxes of objects and room layout can be jointly regressed with ResNet He et al. (2016). Nie et al. (2020b) extended this 3D detector and encoded the relational and geometric

features between objects in estimation. To compensate the 3D information loss from the input image, Huang et al. (2019) leveraged the perspective points from the 3D Manhattan keypoints to provide the 3D geometric constraints. This configuration reduced the ambiguity in 3D depth and improves the consistency between the 2D image plane and 3D world system, making it outperformed the state-of-the-art on image-based 3D object detection.

2.2.3 Shape Recovery for Object Instances

In this section, I review the recent development of 3D deep learning on shape generation, shape completion and skeleton-guided surface generation. It is a fundamental step in this project for object geometry reconstruction from images or scans.

2.2.3.1 Shape Generation

Shape generation aims at predicting a visually plausible geometry from object observations (e.g., images, points and depth maps). Some architectures support shape generation conditioned on various input sources by changing the encoder, where 3D shapes are decoded from a latent vector and represented by points (Fan et al. 2017), voxels (Firman et al. 2016, Choy et al. 2016a, Dai et al. 2017b), meshes (Wang et al. 2018a, Groueix et al. 2018b, Tang et al. 2019, Pan et al. 2019b) or an SDF (Mescheder et al. 2019, Chibane et al. 2020, Liao et al. 2018a). They share the similar modality, that is to decode the equal-size bottleneck feature for shape prediction. This implicit manner reveals the limitation of producing an approximating shape to the target. In 3D scene reconstruction, a single-view shape generation method is proposed in this project to reconstruct multiple object instances from a scene image.

2.2.3.2 Shape Completion

Shape completion aims to recover the missing shape from a partial scan. Deep learning methods attempt to achieve this target with various representations, e.g., points (Sinha et al. 2017, Yuan et al. 2018, Tchapmi et al.

2019, Huang et al. 2020, Wang et al. 2020, Liu et al. 2019, Yin et al. 2018, Wen et al. 2020), voxels (Firman et al. 2016, Brock et al. 2016, Wang et al. 2017, Dai et al. 2017b, Han et al. 2017) or implicit fields (Liao et al. 2018a, Stutz and Geiger 2018, Mescheder et al. 2019, Chibane et al. 2020). Voxels discretize the shape volume into 3D grids. It preserves shape topology but fine-detailed voxel quality relies on high resolution, improving which exponentially increases the time consumption. Implicit fields represent shapes with a signed distance function (SDF). Theoretically it can achieve arbitrary resolution though, learning an accurate SDF still relies on the quality of voxel grids, and these methods require massive spatial sampling to obtain an SDF for a single object, which distinctly increases the inference time (Liao et al. 2018a, Stutz and Geiger 2018, Mescheder et al. 2019, Chibane et al. 2020). Besides, both voxel and SDF methods do not preserve the surface information and present defective results on complex structures. Point cloud is a natural representation of shapes that discretizes the 2-manifold surface. Comparing with voxels and SDFs, 3D points are more controllable, scalable and efficient for learning, which makes it popular for shape completion. However, existing methods commonly adopt an encoder&decoder to parse 3D points (Yuan et al. 2018, Tchapmi et al. 2019), making them struggle to keep shape topology and produce coarse results. Mesh-based methods recover ordered surface points, but current methods predict object meshes by deforming templates (e.g., meshed spheres or planes (Groueix et al. 2018b)), making it restricted from recovering complex structures. For these reasons, many works complete shapes with point clouds (Yuan et al. 2018, Tchapmi et al. 2019, Yin et al. 2018, Huang et al. 2020, Liu et al. 2019, Wang et al. 2020, Wen et al. 2020), especially after the pioneer work PointNet and PointNet++ (Qi et al. 2017a b). However, as mentioned above, directly decoding the bottleneck feature from encoders shows inadequacy in expressing details. From this point, Yuan et al. (2018), Wang et al. (2020), Wen et al. (2020) used skip or cascaded connections to revisit the low-level features to extend shape details. Liu et al. (2019), Yuan et al. (2018) adopted a coarse-to-fine strategy to decode a coarse point cloud and refine it with dense sampling or deforming. PF-Net (Huang et al. 2020) designed a pyramid decoder to recover the missing geometries on

multiple resolutions. However, implicitly decoding a latent feature does not take into account the topology consistency. The recent P2P-Net (Yin et al. 2018) learned the bidirectional deformation between the input scan and complete point cloud. It achieved compact completion results but still struggled to recover the topology especially on invisible areas. In this project, the shape completion is implemented in an explicit manner. The structure of 3D shapes is preserved with skeletal points which guide the surface completion to predict globally and locally consistent shapes.

2.2.3.3 Skeleton-guided Surface Generation

In this project, a skeleton-bridged method is developed for object mesh completion. Before the advent of deep learning, using shape skeletons to guide surface recovery has been well developed with traditional optimisation strategy, wherein Tagliasacchi et al. (2009), Cao et al. (2010), Wu et al. (2015) associated the surface points with its skeleton to represent a compact and smooth surface. Deep learning methods receive rising attention with the advance of shape representation. However, previous methods more focus on learning the skeleton of a specific shape (e.g., for hand pose estimation (Baek et al. 2018) or human body reconstruction (Jiang et al. 2019)). The recent work (Tang et al. 2019) provided a solution to infer 3D skeleton from images which also bridges and benefits the learning of single-view surface reconstruction. P2P-Net (Yin et al. 2018) supports bidirectionally mapping between skeletal points and surface points. In this project, shape skeletons are learned from partial scans as an intermediate representation to guide surface completion.

2.3 3D Scene Modelling and Reconstruction

In this section, I review the related works on semantic 3D scene recovery with different modalities: 1) with shape retrieval and 2) with shape reconstruction. Both of the two manners are used in this project to build 3D semantic scenes.

2.3.1 Scene Modelling by Shape Retrieval

To predict indoor object shapes, early methods adopted cuboids (Deng and Latecki 2017, Huang et al. 2018a) to recover the orientation and placement of target objects without the need of querying CAD model datasets. However, these geometric details are weak because objects are only represented by a bounding box. Rather than using cuboidal shapes, some methods produced promising results in placement estimation of a single object by aligning CAD models with the object image (Lim et al. 2014, Wu et al. 2016). Other methods leveraged shallow features (e.g. line segments, edges and HOG features) (Zhang et al. 2015, Liu et al. 2017 2018) to segment images and retrieve object models, or used a scene dataset as priors to retrieve object locations based on co-occurrence statistics (Hueting et al. 2018). They either asked for human interaction or hand-crafted priors in parsing object features.

With the advent of deep learning, recent studies also considered CNNs to retrieve objects (Izadinia et al. 2017, Nie et al. 2018) with informative scene knowledge (Huang et al. 2018b). Huang et al. (2018b) estimated depth maps and surface normal maps from RGB images with scene grammar to optimise the object placement. However, depth prediction is sensitive if the input distribution is slightly different from the training data (Nie et al. 2018). Instead of tailoring scene grammar to improve the modelling results, Nie et al. (2020a) incorporated the relational reasoning to infer the object support relationship with a Relation Network. A parallel development (Izadinia et al. 2017) followed a Render-and-Match strategy to optimize object locations and orientations, which did not involve any depth clues and relational constraints. CAD scenes are iterated until their renderings are sufficiently close to the input image. However, involving rendering in optimisation iterations results in relative low efficiency for scene modelling. Since these approaches required iterations of rendering or model search from a dataset, the mesh similarity and time efficiency depend on the size of the model repository and raise further concerns.

In summary, 3D scene modelling from a single image is challenging as it requires computers to perform equivalently as human vision to perceive and

understand indoor context with only colour intensities. It generally requires for blending various vision tasks (Chen et al. 2015) and most of them are still under active development, e.g. object segmentation (Bu et al. 2016), layout estimation (Wei and Wang 2018) and geometric reasoning (Liu et al. 2018). Although machine intelligence has reached comparable human-level performance in some tasks (e.g. scene recognition (Zhou et al. 2018)), those techniques are only able to represent a fragment knowledge of full scene context. Let alone how to resolve the problem when indoor geometry is over-cluttered and complicated. In our view, there are three major challenges in scene modelling and reconstruction. First, complicated indoor scenes involve heavily occluded objects, which could cause missing contents in detection (Izadinia et al. 2017). Second, cluttered environments significantly increase the difficulty of camera and layout estimations, which critically affects the reconstruction quality Lee et al. (2017). Third, compared to the large diversity of objects in real scenes, the reconstructed virtual environment is still far from satisfactory (missing small pieces, wrong labelling). Existing methods have explored the use of various contextual knowledge, including object support relationship (Huang et al. 2018b, Nie et al. 2018) and human activity (Huang et al. 2018b), to improve modelling quality. However, their relational (or contextual) features are hand-crafted and would fail to cover a wide range of objects in cluttered scenes.

2.3.2 Scene Reconstruction

Different from the shape retrieval manner, scene reconstruction does not rely on a CAD shape dataset. Object shapes are predicted directly from images.

Scene reconstruction at the instance level remains problematic because of the large number of indoor objects with various categories. It leads to a high-dimensional parameter space of object shapes subjected to diverse geometry and topology. To first address single object reconstruction, some approaches represented shapes in the form of point cloud (Fan et al. 2017, Mandikal et al. 2018, Kurenkov et al. 2018, Navaneet et al. 2019), patches Groueix et al. (2018a), Wang et al. (2018b) and primitives (Tian et al. 2019,

Tulsiani et al. 2017, Paschalidou et al. 2019, Deprelle et al. 2019) which are adaptable to complex topology but require post-processing to obtain meshes. The structure of the voxel grid (Choy et al. 2016b, Liao et al. 2018b, Wallace and Hariharan 2019) is regular while suffering from the balance between resolution and efficiency, demanding the use of Octree to improve local details (Riegler et al. 2017, Tatarchenko et al. 2017, Wang et al. 2018b). Some methods produced impressive mesh results using the form of signed distance fields (Park et al. 2019) and implicit surfaces (Chen and Zhang 2019, Michalkiewicz et al. 2019, Xu et al. 2019, Mescheder et al. 2019). However, these methods are time-consuming and computationally intensive, making it impractical to reconstruct all objects in a scene. Another popular approach was to reconstruct meshes from a template (Wang et al. 2018a, Groueix et al. 2018a, Kato et al. 2018), but the topology of the reconstructed mesh was restricted. So far, the state-of-art approaches modified the mesh topology to approximate the ground-truth (Pan et al. 2019a, Tang et al. 2019). However, existing methods estimated 3D shapes in the object-centric system, which cannot be applied to scene reconstruction directly.

Previous works have attempted to address scene reconstruction via various approaches. **Scene understanding** methods (Schwing et al. 2013, Huang et al. 2018a, Choi et al. 2013) obtain room layout and 3D bounding boxes of indoor objects without shape details. **Scene-level reconstruction** methods recover object shapes using contextual knowledge (room layout and object locations) for scene reconstruction, but most methods currently adopt depth or voxel representations (Shin et al. 2019, Li et al. 2019a, Tulsiani et al. 2018, Kulkarni et al. 2019). Voxel-grid presents better shape description than boxes, but its resolution is still limited, and the improvement of voxel quality exponentially increases the computational cost, which is more obvious in scene-level reconstruction. **Mesh-retrieval** methods (Izadinia et al. 2017, Huang et al. 2018b, Hueting et al. 2017) improve the shape quality in scene reconstruction using a 3D model retrieval module. As these approaches require iterations of rendering or model search, the mesh similarity and time efficiency depend on the size of the model repository and raise further concerns. **Object-wise mesh reconstruction** exhibits the advantages in both

efficiency and accuracy (Wang et al. 2018a, Groueix et al. 2018a, Pan et al. 2019a, Kato et al. 2018, Gkioxari et al. 2019), where the target mesh is end-to-end predicted in its own object-centric coordinate system. For scene-level mesh reconstruction, predicting objects as isolated instances may not produce ideal results given the challenges of object alignment, occlusion relations and miscellaneous image background. Although Mesh R-CNN (Gkioxari et al. 2019) is capable of predicting meshes for multiple objects from an image, its object-wise approach still ignores scene understanding and suffers from the artifacts of mesh generation on cubified voxels. So far, to the best of authors' knowledge, few works take into account both mesh reconstruction and scene context (room layout, camera pose and object locations) for total 3D scene understanding.

The most relevant works are (Li et al. 2019a, Tulsiani et al. 2018, Kulkarni et al. 2019, Gkioxari et al. 2019), which took a single image as input and reconstructed multiple object shapes in a scene. However, the methods (Li et al. 2019a, Tulsiani et al. 2018, Kulkarni et al. 2019) were designed for voxel reconstruction with limited resolution. Mesh R-CNN (Gkioxari et al. 2019) produced object meshes, but still treated objects as isolated geometries without considering the scene context (room layout, object locations, etc.). Different from the above works, the method in Chapter 5 connects the object-centric reconstruction with 3D scene understanding, enabling joint learning of room layout, camera pose, object bounding boxes, and meshes from a single image.

2.4 Datasets and Metrics

In this part, I present an introduction of the datasets and relevant metrics used for evaluation in this thesis.

2.4.1 Datasets

NYU v2 NYU v2 offers 1449 indoor RGB-D images with densely segmented objects at the instance level. In Chapter 3 and 4, we use its object segmentations on RGB images to train and evaluate our method. The RGB

images are used as the input, and the instance segmentations and depth images are used as the ground-truth for evaluation. Since NYU v2 does not contain 3D models, we manually collect a small dataset with around 300 CAD models for shape retrieval task in Chapter 3. This small dataset is collected from 3D Warehouse[1] including 13 categories, i.e, bed, book, ceiling, chair, floor, furniture, object, picture, sofa, table, TV, wall, window (defined by Silberman et al. (2012)). Besides, the images in NYU v2 are also labelled with instance support relationships in 2D. We also use these information to learn object support context from RGB images in Chapter 4.

SceneNN The SceneNN (Hua et al. 2016) dataset contains 50 sophisticated 3D scenes segmented at the instance level. They use the same semantic labels as defined in NYU v2. In Chapter 3, we use this dataset to extract support priors between 3D objects for support inference.

SUN RGB-D The SUN RGB-D dataset (Song et al. 2015) is a 3D indoor scene understanding benchmark. It contains 10,355 RGB-D images labelled with oriented 3D object bounding boxes, room layout bounding boxes and camera poses. In Chapter 4 and 5, we only use its RGB images as the input for single view scene modelling and reconstruction, where the corresponding camera poses, object and room bounding boxes are used for supervision. The official train/test split is used for evaluation.

Pix3D The Pix3D dataset (Sun et al. 2018) contains 395 furniture CAD models with 9 categories, which are aligned with 10,069 images. We use this dataset for single-view object reconstruction in Chapter 5.

ShapeNetCore and ShapeNet-Skeleton The ShapeNetCore dataset is a subset of the full ShapeNet dataset (Chang et al. 2015). It contains 55 categories of objects with single clean CAD models and alignment annotations, where 51,300 unique 3D models are covered. The ShapeNet-Skeleton (Tang et al. 2019) dataset extracts the meso-skeleton points (Wu et al. 2015) of

[1] `https://3dwarehouse.sketchup.com/?hl=en`

objects from 13 categories of ShapeNetCore. These skeleton points provide clean topological clues of objects without surface details. In Chapter 4, we use the ShapeNetCore dataset for our shape retrieval task. In Chapter 6, we use ShapeNetCore + ShapeNet-Skeleton as datasets for our shape completion task. In shape retrieval, we search the object CAD models with the most similar appearance with the input object images. In shape completion, we predict the full shape meshes from a single depth scan of each object.

Other Datasets In Chapter 4, we augment the CAD model dataset ShapeNetCore with SUNCG (Song et al. 2017). SUNCG is a synthesized indoor scene dataset which contains 2644 unique object meshes covering 84 categories for our shape retrieval task. Besides, we also use ScanNet (Dai et al. 2017a) to obtain the object height priors in Chapter 4. ScanNet contains 1,513 real-scanned 3D scenes with point-wise annotated object instances.

2.4.2 Metrics

In this thesis, we use the standard metrics in our evaluation. All of them are widely used in various benchmarks.

For **instance segmentation** and **3D object detection** in Chapter 3,4,5, we use the general mean Average Precision (mAP) metric to evaluate our method (Hariharan et al. 2014, Huang et al. 2018a). mAP has been seriously defined in information retrieval[2]. Besides, we also adopt Pixel Accuracy (PA), Mean Accuracy (MA) and Intersection over Union (IoU) in Chapter 4 to evaluate the **semantic segmentation** by comparing our predictions with the ground-truth. The three metrics are extensively reviewed by Garcia-Garcia et al. (2017).

For **depth estimation** in Chapter 3, we adopt the mean absolute relative error (rel), root mean squared error (rms) and log10 error (Laina et al. 2016) to compare the predicted deptm map with the ground-truth. They are

[2] `https://en.wikipedia.org/?title=Mean_average_precision`

defined as below.

$$
\begin{aligned}
rel(D_{pred}, D_{gt}) &= \frac{1}{N_{pixel}} \sum \frac{|D_{pred} - D_{gt}|}{D_{gt}}, \\
rms(D_{pred}, D_{gt}) &= \sqrt{\frac{\sum |D_{pred} - D_{gt}|^2}{N_{pixel}}}, \\
log10(D_{pred}, D_{gt}) &= \frac{1}{N_{pixel}} \sum \frac{|log_{10}(D_{pred}) - log_{10}(D_{gt})|}{D_{gt}},
\end{aligned}
\tag{2.1}
$$

where D_{pred} and D_{gt} respectively denote the predicted and ground-truth depth maps. N_{pixel} is the number of all pixels on the image plane.

For **layout estimation**, we adopt the Intersection over Union (IoU) (Huang et al. 2018b) to calculate the 3D overlap between our predictions and the ground-truth. Its defined as below.

$$
IoU(V_{pred}, V_{gt}) = \frac{V_{pred} \cap V_{gt}}{V_{pred} \cup V_{gt}}, \tag{2.2}
$$

where V_{pred} and V_{gt} respectively denote the volume of predicted and ground-truth layout boxes.

For **camera pose estimation**, we adopt the mean absolute error (mae) defined below as the metric to calculate the distance between predicted and ground-truth camera angles.

$$
mae(\gamma_{pred}, \gamma_{gt}) = \frac{\sum |\gamma_{pred} - \gamma_{gt}|}{N_s}, \tag{2.3}
$$

where γ_{pred} and γ_{gt} are the predicted and ground-truth camera angles. N_s denotes the number of scenes.

For **shape reconstruction** and **completion** in Chapter 5 and 6, we adopt Chamfer distance and Earth Mover's distance as the metrics to calculate the distance between predicted and ground-truth shape surface points. The Chamfer distance between $\mathbf{P}_{pred}, \mathbf{P}_{gt} \subseteq \mathbb{R}^3$ is defined as:

$$
CD(\mathbf{P}_{pred}, \mathbf{P}_{gt}) = \sum_{x \in \mathbf{P}_{pred}} \min_{y \in \mathbf{P}_{gt}} \|x - y\|_2^2 + \sum_{y \in \mathbf{P}_{gt}} \min_{x \in \mathbf{P}_{pred}} \|x - y\|_2^2, \tag{2.4}
$$

where $\| * \|_2^2$ denotes the square of Euclidean distance. $\mathbf{P}_{pred}$ and $\mathbf{P}_{gt}$ respectively denote the surface points of predicted and ground-truth shapes.

Consider $\mathbf{P}_{pred}$ and $\mathbf{P}_{gt}$ have a equal size $s = |\mathbf{P}_{pred}| = |\mathbf{P}_{gt}|$. The EMD between $\mathbf{P}_{pred}$ and $\mathbf{P}_{gt}$ is defined as:

$$EMD(\mathbf{P}_{pred}, \mathbf{P}_{gt}) = \min_{\phi:\mathbf{P}_{pred}\to\mathbf{P}_{gt}} \sum_{x\in\mathbf{P}_{pred}} \|x - \phi(x)\|_2 \tag{2.5}$$

where $\phi : \mathbf{P}_{pred} \to \mathbf{P}_{gt}$ is a bijection. Hence, EMD is an optimization problem: to find a function ϕ mapping $\mathbf{P}_{pred}$ to $\mathbf{P}_{gt}$ to minimize the distance between $\mathbf{P}_{pred}$ and $\mathbf{P}_{gt}$, and the distance value is used as the metric.

In Chapter 6, we also use Normal Consistency to evaluate point normal estimation for mesh reconstruction, which is defined as below.

$$NCon(\mathbf{P}_{pred}, \mathbf{P}_{gt}) = \frac{\sum |\mathbf{n}_{pred} \cdot \mathbf{n}_{gt}|}{N_{pred}}, \tag{2.6}$$

where $\mathbf{n}_{pred}$ denote the normal vector of a point $\mathbf{p}_{pred} \in \mathbf{P}_{pred}$ on predicted surface. $\mathbf{n}_{gt}$ is the normal vector of the nearest point on ground-truth surface to $\mathbf{p}_{pred}$. N_{pred} is the total number of points on the predicted surface.

Chapter 3

Semantic Scene Modelling

In this chapter, I mainly discuss our work on semantic modelling of indoor scenes from a single photograph. We build the indoor scene modelling pipeline on feature maps extracted by deep neural networks. Three Fully Convolutional Networks (Long et al. 2015) are adopted to individually predict object instance masks, a depth map and an edge map of the room layout. Based on these intermediate features, support relationships between indoor objects (e.g. a lamp is supported by a table from below) are inferred with a data-driven manner. Constrained by the support context, a global-to-local model matching strategy is followed to model the whole indoor scene. We demonstrate that the proposed method can efficiently retrieve indoor objects including situations where objects are severely occluded.

3.1 Method Overview

The pipeline of our algorithm is presented in Figure 3.1. With only one indoor photo, our goal is to model a 3D scene with informative semantic context. We produce object masks (Li et al. 2017b), depth maps (Laina et al. 2016), and room layout edge maps (Mallya and Lazebnik 2015) using three FCNs with different architectures to guide object modelling. In object segmentation, a novel FCN architecture Li et al. (2017b) is adopted for training on the NYU v2 (Silberman et al. 2012) dataset, so that we can segment 40 common categories at the instance level. Combining the depth map with instance

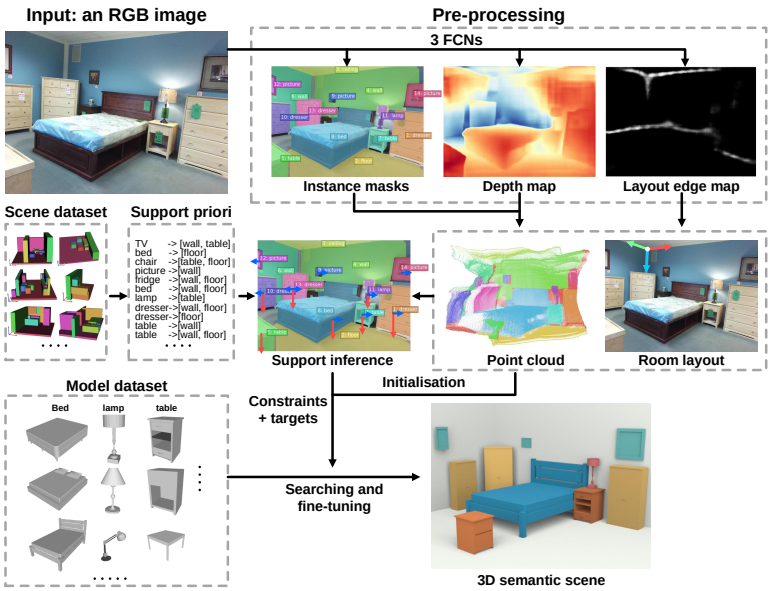


Figure 3.1: Pipeline of our method

masks, the segmented point cloud of the scene can be calculated given the camera parameters. Meanwhile, room layout is estimated from the layout edge map to provide a baseline for the subsequent support inference.

In support inference, we decide support between objects a prior. Parsing the SceneNN (Hua et al. 2016) dataset, and combining it with the point cloud and room layout allows the support relationships between objects to be inferred. We build the support context as a hierarchical structure. Beginning with the layout frame (floors and walls), Each object is modelled on the basis of its supporting objects (e.g. if a lamp is supported by a table, the table should be built first). In our experiments, this kind of hierarchical constraint ensures a robust modelling result.

In object modelling, we build each object with a search-to-match strategy using a model database. The segmented point cloud is used to estimate an initial position and size for each object. We set the orientation angle, the translation vector and 3D scales of each model as optimisation variables. The Intersection over Union (IoU) ratio between the model's perspective projection area and its mask is used as the maximization target. With two optimisation steps (global searching and local matching), the whole scene is

built following the derivation of the support hierarchy.

In summary, the key modules and contributions of this chapter are:

- A novel approach for indoor scene modelling based entirely on FCNs. The portability of FCNs also indicates that our modelling performance can be improved further using deep learning techniques.
- We provide a data-driven support inference approach to achieve hierarchical modelling, and have demonstrated that this approach shows great effectiveness in modelling badly occluded objects.

In the following sections, we will discuss the methods involved in object segmentation, depth estimation, and room layout estimation, which are used for the final semantic scene modelling in section 3.6.

3.2 Instance Segmentation with Fully Convolutions

We adopt the FCN architecture proposed by Li et al. (2017b) to segment a scene into instance-level objects. It offers an end-to-end solution with a great performance in instance segmentation. To use it for indoor scenes, the NYU v2 dataset is utilized. This offers 1449 indoor images with 40 fully segmented labels at the instance level. We use the official training/test split to evaluate the network. The mAP score (Hariharan et al. 2014) reaches 29.95% and 19.13% at IoU threshold of 0.5 and 0.7 respectively. We conduct training on the whole dataset (1449 images) to improve its performance. Figure 3.2b shows the segmentation result on an image (Figure 3.2a) from the dataset SUN-RGBD (Song et al. 2015).

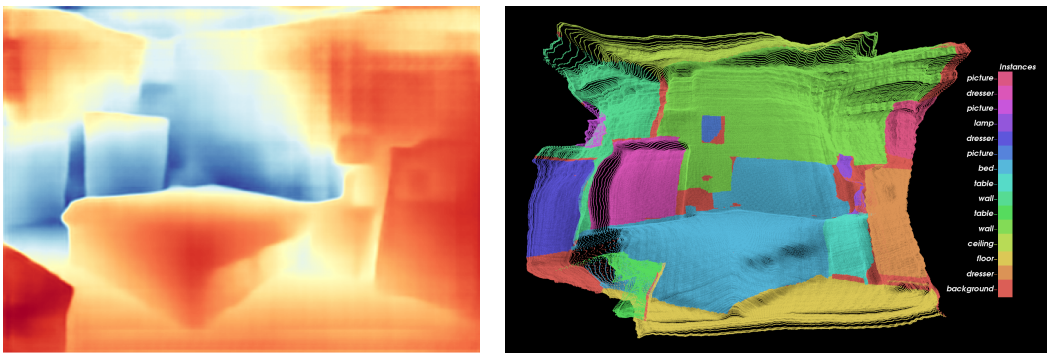
To refine those zig-zagged areas on the mask margin of the segmentation results, in the testing phase after training, we append a post-processing layer at the end of the FCN. The Grab-cut method (Rother et al. 2004) is adopted using the FCN masks as probable foreground and the other areas as probable background. The refined result is shown in Figure 3.2c.

(a) Original image (b) Initial result from FCN (c) Refined result from Grab-cut.

Figure 3.2: Object instance segmentation on an indoor image.

3.3 Depth Estimation from an RGB Image

For depth estimation, we adopt the network proposed by Laina et al. (2016). It also has an FCN architecture based on residual learning. Without any post-processing, only a small amount of training data is required. As this model contains fewer parameters, it runs fast in forward propagation. Since the model is trained on the benchmark dataset NYU v2 where Microsoft Kinect is used, we adopt the technical parameters of Kinect (Konolige and Mihelich 2011) to retrieve the point cloud (see Figure 3.3). Figure 3.3b presents the segmented point cloud using the object masks. Its clearly illustrates that the depth map is noisy especially for the margin area of the image. Therefore, support inference is considered to compensate for the geometric information.



(a) Depth map (b) Point cloud

Figure 3.3: Point cloud retrieval

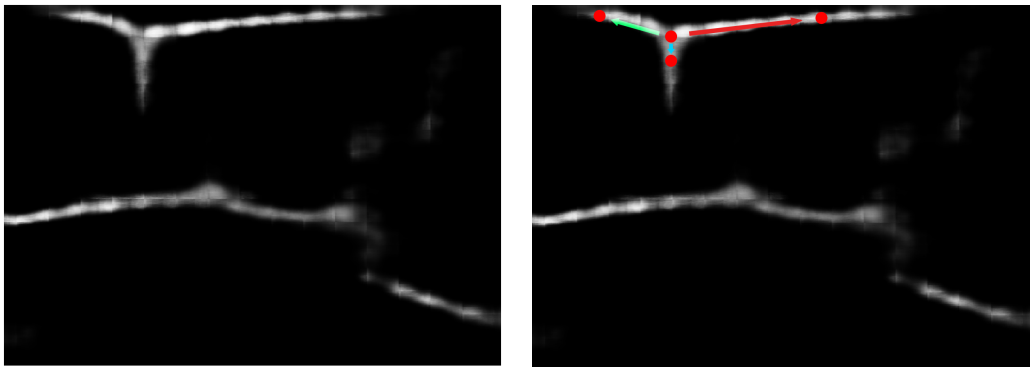
3.4 2D Layout Estimation

Layout estimation is intended to provide a unified reference coordinate system for the further support inference and object modelling. Unlike the general room layout estimation (Hedau et al. 2009) where a 3D parametric box is used to estimate the room layout, only a corner of the box is required to construct the reference system. In this part, we extract the edge map of the room layout following the work by Mallya and Lazebnik (2015) (see Figure 3.4a), where structured edge detection forests and an FCN are used to provide a probability map of layout edges. Their experiments present robust results in occluded cases.

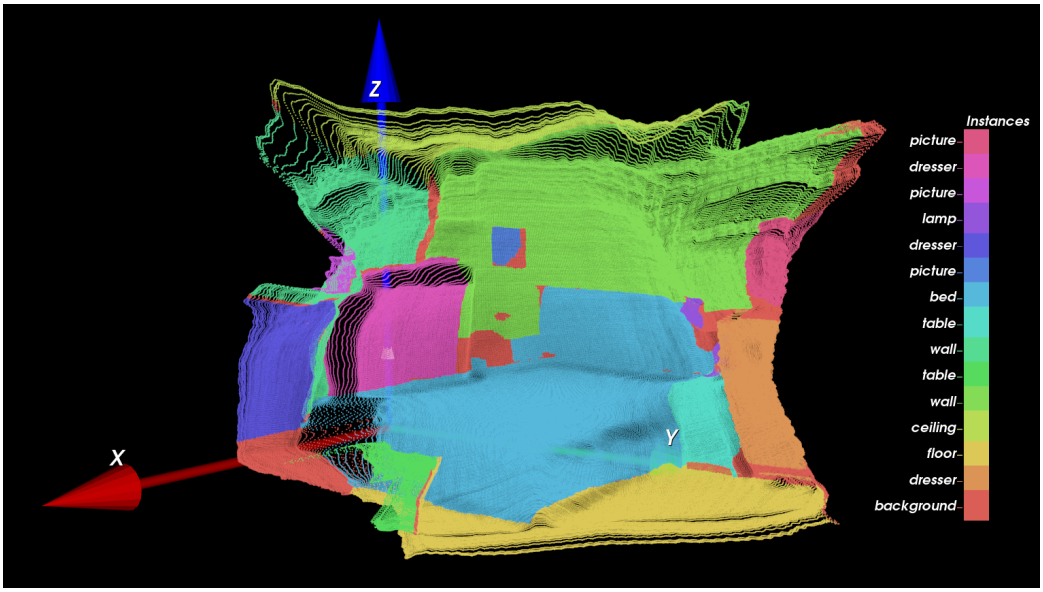
From the edge map, we adopt the RANSAC algorithm to search for a robust room corner (see Figure 3.4b and a detailed description in Appendix A.2). With the room corner and the point cloud, extrinsic parameters of the camera can be estimated by fitting the corner with an orthogonal system. We transform the point cloud into the new reference system, then align its x-y plane to the floor (the lowest plane) and its z-axis upwards (see Figure 3.4c). By the layout estimation, the ceiling, two walls and the floor have already been determined. Therefore in the segmentation step, we do not require floors, ceilings and walls to be accurately segmented.

3.5 Prior-based Support Inference

The layout information and the segmented point cloud above are used for support inference between object instances. Three support types are defined: 1. support from below; 2. support from behind; 3. support from the top. It should be noted that we generally only consider the first two support types as they are able to explain most scenarios. We first build basic support rules as priors at the object category level from the SceneNN (Hua et al. 2016) dataset, which contains 50 sophisticated scenes with the same semantic labels defined in NYU v2. The object co-occurrence map of the dataset is illustrated in Figure 3.5a, where the colour intensity indicates the frequency of two co-occurred objects. To infer support relationships at a general category level,



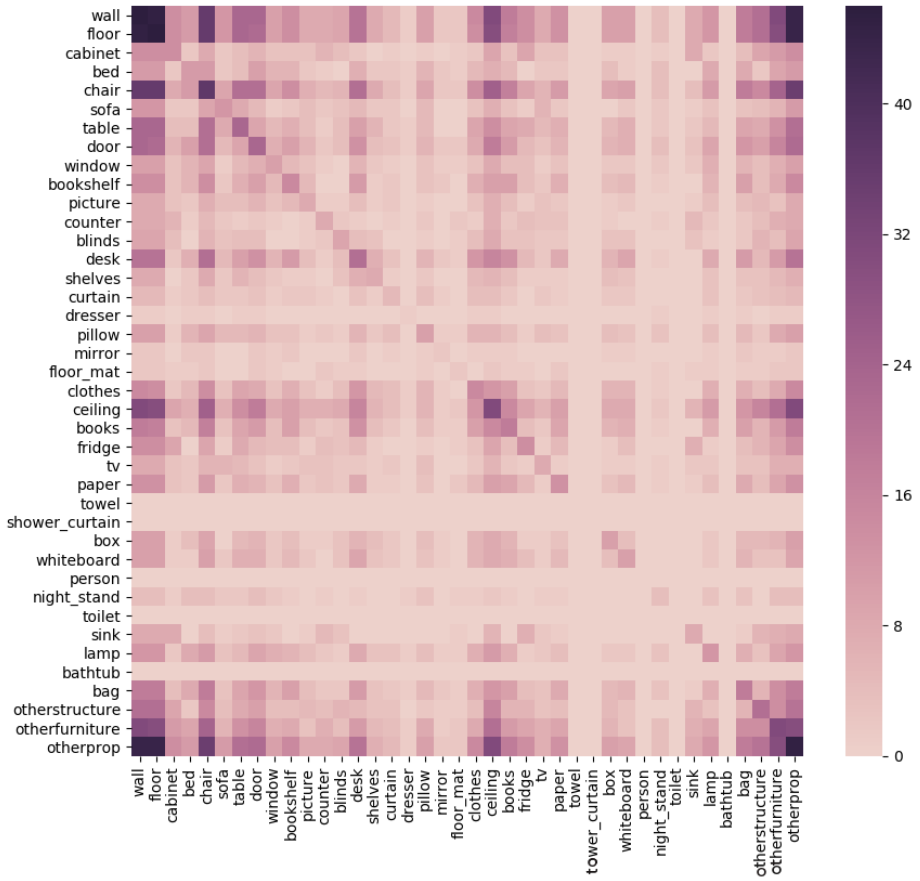
(a) Edge map of room layout (b) Identified room corner



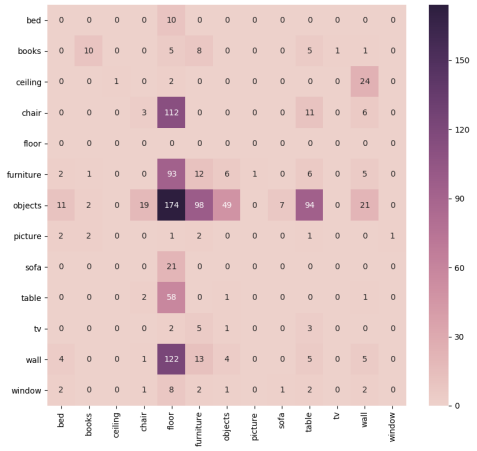
(c) Aligned point cloud

Figure 3.4: Room layout estimation

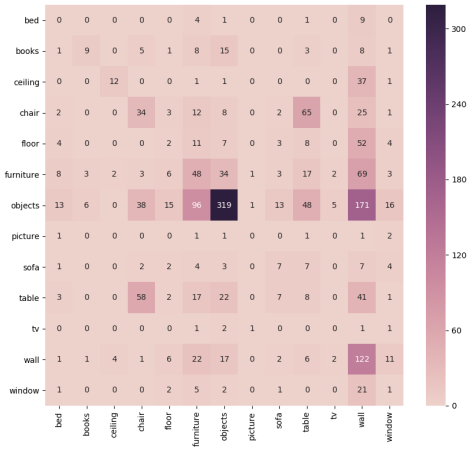
we merge the 40 object categories with the class mapping provided with NYU v2 (Silberman et al. 2012) resulting in 13 general categories. It is based on our observation that if object A is supported by object B, it could also be supported by others with a similar semantic label to object B (e.g. lamps can be supported by both desks and tables). The frequency of two co-occurred instances that have some support relationship (support type 1 or 2) are counted. The results are illustrated in Figure 3.5b and Figure 3.5c, where the block colour represents the number of cases when object i (in



(a) Objects co-occurrence map



(b) Cases with support from below



(c) Cases with support from behind

Figure 3.5: Support priors from SceneNN dataset

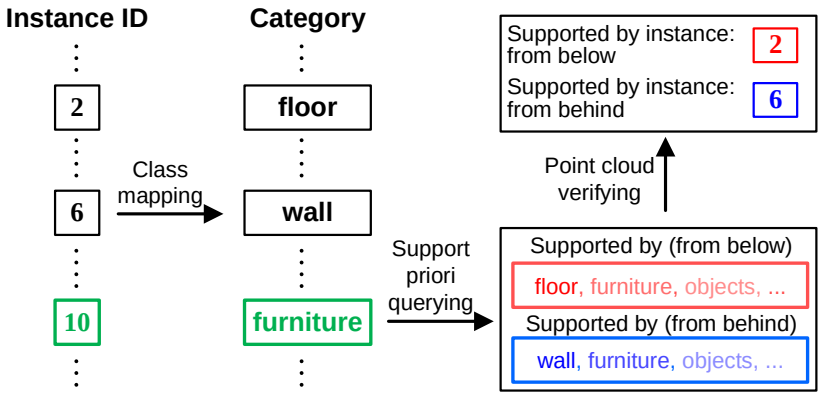


Figure 3.6: Searching instances with a support relationship.

row) is supported by object j (in column) from below or behind. These two matrices inform the support relationship priors.

For each instance in a scene, we query the support matrices to recommend other instances which could have a support relationship with. Taking the dresser (see No.10 instance in Figure 3.2c) as an example, as the Figure 3.6 shows, it belongs to the furniture category. From querying the support matrices, the furniture category is likely to be supported from below (by floors, furniture, etc.) and from behind (by walls, furniture, etc.). A subsequent search is undertaken to identify additional instances that belonging to these categories. The point cloud is subsequently used to verify whether they are indeed close to each other from below/behind and to exclude wrong instances when these are identified. Using the prior information can not only improve searching efficiency, it also avoids judgment mistakes that could occur when only using the noisy point cloud. To handle sophisticated cases, we usually use the priors to recommend all potential categories in querying. The inferred support context behind Figure 3.2 can be built as a hierarchical structure (see Figure 3.7).

3.6 Scene Modelling with 3D Shape Retrieval

A global-to-local approach is followed in this section to both search and fine-tune models in the database for semantic modelling. Our optimisation

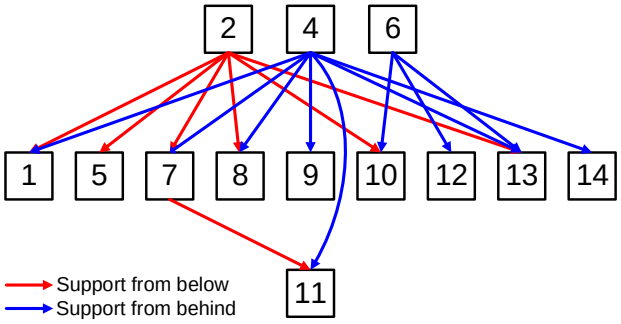


Figure 3.7: Support hierarchy

variables include the size, position and orientation of models and to match them with object masks to retrieve an indoor scene. Models with the most similar shape are firstly identified by global searching, and secondly fine-tuned. Before matching, models in the database should be categorized by labels and pre-processed with z-axis upwards, x-axis front-toward with their centre to the origin. We build this database with Google 3D Warehouse.

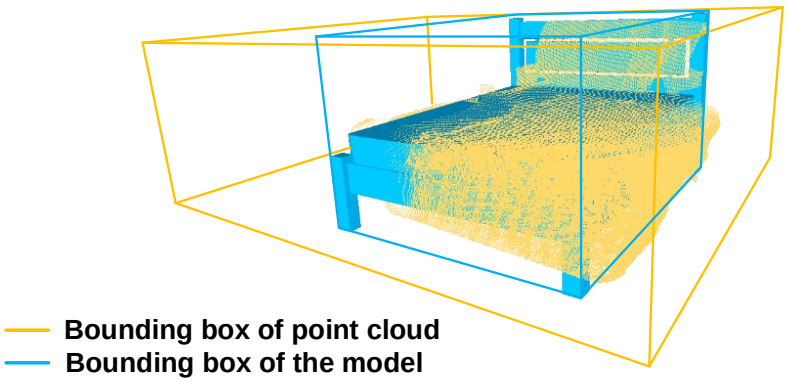


Figure 3.8: Point cloud and the matched model

3.6.1 Matching Problem Formulation

As the point cloud is noisy, it is difficult to use this information to estimate the object orientation. However, this data provides insights regarding the position and height clues. Therefore, we utilize the point cloud to initialize the model position and scales (see Figure 3.8 where height size is initialized by the point cloud). As the segmented mask provides edge and contour

clues, we use it as the optimisation target to obtain the object orientation, and subsequently refine the position and 3D scales.

In this routine, the floor and walls are firstly built by the layout estimation. We denote the point cloud of the target object by $\mathbf{P}_i$, its segmented image mask by $Mask_i$. $\mathbf{M}_i$, ${\mathbf{M}_i}^{\mathrm{H}}$ and ${\mathbf{M}_i}^{\mathrm{L}}$ respectively represents the 3D model for matching, the supporting model behind and the one below. $i = 1, 2, ..., n$, n is the number of object instances in the scene. The 3D model scales $\mathbf{S}$, the spatial position $\mathbf{p}$ and the orientation angle θ around z-axis are set as optimisation variables. With the camera parameters estimated in the layout estimation and point cloud estimation, the operator for projecting the 3D model onto the image plane can be calculated and we denote it by $\texttt{Proj}(*)$. Then we build the matching problem as to minimize

$$
\begin{aligned}
&\max_{\theta,\mathbf{S},\mathbf{p}} \texttt{IoU}\{\texttt{Proj}[\mathbf{R}(\theta)\cdot\mathbf{S}\cdot\mathbf{M}_i+\mathbf{p}], Mask_i\},\\
&\mathbf{R}=\begin{bmatrix}\cos(\theta) & -\sin(\theta) & 0\\ \sin(\theta) & \cos(\theta) & 0\\ 0 & 0 & 1\end{bmatrix},\\
&\mathbf{S}=\begin{bmatrix}s_1 & 0 & 0\\ 0 & s_2 & 0\\ 0 & 0 & s_3\end{bmatrix}, \mathbf{p}=\begin{bmatrix}p_1\\ p_2\\ p_3\end{bmatrix},
\end{aligned}
\tag{3.1}
$$

where IoU score is used to measure optimisation performance. We aim at that, on the image plane, the projection area of the optimal model can match with the corresponding mask. Besides, there are three types of constraints: 1. from supporting objects below; 2. from supporting objects behind; 3. from point cloud data.

1. From supporting objects below

   This type of constraints is to ensure the matched model placed on the upper surface of its supporting model (see Equation (3.2)).

$$
\begin{aligned}
\mathrm{mean}[\mathbf{R}(\theta)\cdot\mathbf{S}\cdot\mathbf{M}_i+\mathbf{p}]_{x,y} &>= \min[{\mathbf{M}_i}^{\mathrm{L}}]_{x,y}\\
\mathrm{mean}[\mathbf{R}(\theta)\cdot\mathbf{S}\cdot\mathbf{M}_i+\mathbf{p}]_{x,y} &<= \max[{\mathbf{M}_i}^{\mathrm{L}}]_{x,y}\\
\min[\mathbf{R}(\theta)\cdot\mathbf{S}\cdot\mathbf{M}_i+\mathbf{p}]_{z|x,y} &= \max[{\mathbf{M}_i}^{\mathrm{L}}]_{z|x,y}
\end{aligned}
\tag{3.2}
$$

2. From supporting objects behind

   The orientation of the target object generally has high relevance with its supporting object behind, and they should be attached close. Here we denote the orientation angle of its supporting model $\mathbf{M}_i{}^{\mathrm{H}}$ by $\theta_i{}^{\mathrm{H}}$. The constraints are written as:

$$\begin{aligned} &\theta \in \{\theta_i{}^{\mathrm{H}} + k \cdot \pi/4 | k = 0, 1, \ldots, 7\} \\ &\texttt{dist}(\mathbf{M}_i, \mathbf{M}_i{}^{\mathrm{H}})_{x,y,z} < \mathbf{d}_1 \end{aligned}, \tag{3.3}$$

   where `dist` is to get the shortest distance in x,y and z axis between $\mathbf{M}_i$ and $\mathbf{M}_i{}^{\mathrm{H}}$, and $\mathbf{d}_1$ is the threshold vector. For those objects that are not supported by any others from behind, we restrain the search domain of θ by $\theta \in \{k \cdot \pi/4 | k = 0, 1, \ldots, 7\}$. Especially, this type of constraint will not be used if a bidirectional support relationship exists between two objects.

3. From point cloud data

   The point cloud is used to initialize model position and 3D scales. The height scale of the model can be initialized by

$$s_3^* = \frac{\max(\mathbf{P}_i)_z - \max(\mathbf{M}_i{}^{\mathrm{L}})_z}{\max(\mathbf{M}_i)_z - \min(\mathbf{M}_i)_z}. \tag{3.4}$$

   To deal with cases when point cloud is partly occluded, $\max(\mathbf{P}_i)_z - \max(\mathbf{M}_i{}^{\mathrm{L}})_z$ is used to estimate the real height of the target object. We set the geometric centre of the point cloud $\mathbf{P}_i$ as $\mathbf{p}_i{}^{\mathrm{c}}$, and the constraints are designed as

$$\begin{aligned} &|\mathbf{p} - \mathbf{p}_i{}^{\mathrm{c}}| < \mathbf{d}_2 \\ &s_1 \in [\rho_1{}^{\mathrm{L}} \cdot s_3^*, \rho_1{}^{\mathrm{U}} \cdot s_3^*] \\ &s_2 \in [\rho_2{}^{\mathrm{L}} \cdot s_3^*, \rho_2{}^{\mathrm{U}} \cdot s_3^*] \\ &s_3 \in [\rho_3{}^{\mathrm{L}} \cdot s_3^*, \rho_3{}^{\mathrm{U}} \cdot s_3^*] \end{aligned}, \tag{3.5}$$

   where $\mathbf{d}_2$ is to ensure that the model $\mathbf{M}_i$ overlaps the point cloud $\mathbf{P}_i$, and $\{(\rho_k{}^{\mathrm{L}}, \rho_k{}^{\mathrm{U}}) | k = 1, 2, 3\}$ are used to adjust the model scales. The first line in Equation 3.5 is to guarantee that the retrieved object centre should be close to the centroid of its predicted point cloud, and the remaining items means that the 3D sizes of each object should be

in a reasonable interval. The specific parameter values are detailed in Appendix A.1.

The optimisation problem described above is built for both global model searching and local fine-tuning. Following the support hierarchy, every supported objects should be built after their supporting objects.

3.6.2 Global Searching

The global searching step generally requires an efficient method to find out the model with a similar semantic shape in the whole parametric space. Here we adopt the Dividing Rectangles (DIRECT) algorithm (Jones et al. 1993) to solve the nonlinear optimisation problem (Equation 3.1) and find the model with the highest IoU score. The DIRECT algorithm is a deterministic-search method, which can efficiently handle global optimisation problems with bound constraints. It starts by scaling the search domain to a hypercube then subdivide it into smaller hyperrectangles step by step to find the global optima. Since we only use it to search an appropriate model for the next local matching, only a few iterations are required.

3.6.3 Local Matching

After the model is identified, the BOBYQA algorithm (Powell 2009) is followed to decide the final size, position and orientation. This derivative-free approach performs an iteratively constructed quadratic approximation for the objective function, where bound constraints are acceptable. In practice, we use the optima from the global searching to initialize the optimisation variables and keep the constraints unchanged. The target object is built after the algorithm converges.

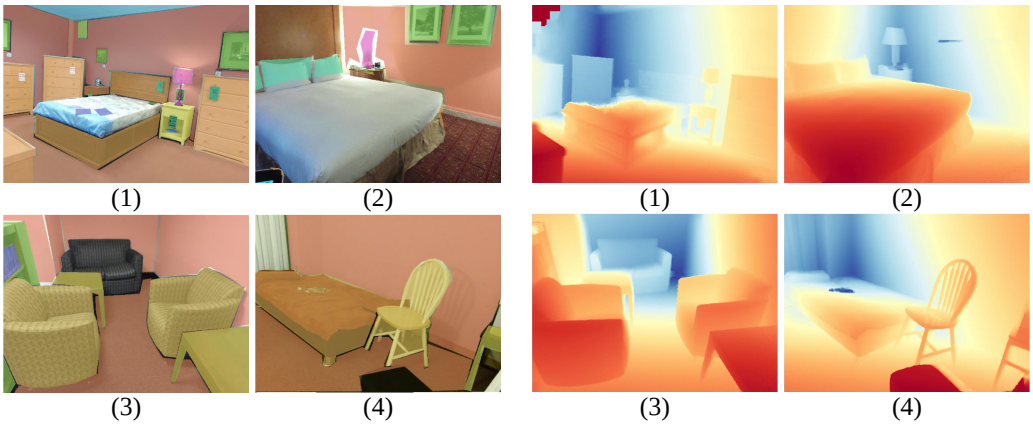
3.7 Experiments and Discussions

We present the modelling performance on a variety of indoor images from SUN-RGBD (Song et al. 2015). The whole algorithm is implemented on Ubuntu 16.04 with a GTX 1080 GPU and Intel Xeon CPU E5-1650 0 @

3.20GHz x 12. The modelling results are shown in Figure 3.10. The parameters involved in our algorithm are presented in Appendix A.1. The performance analysis, comparisons and limitations of our method are also discussed below.

3.7.1 Performance Evaluation

Since the ground-truth masks in SUN-RGBD dataset are only segmented at the class-level (see Figure 3.9a), a numerical comparison is not discussed here. Our segmentation results (Figure 3.10b) show that most objects in scenes are segmented with their masks refined. Figure 3.9b presents the corresponding ground-truth depth maps. The errors of the predicted depth maps (see Figure 3.10c) are evaluated by rel, rms and log10 scores (Laina et al. 2016) in Table 3.1, which shows that they are noisier than the test results on the NYU v2 dataset (Laina et al. 2016). Although a noisy depth map would result in errors in initializing model positions and scales, we have demonstrated that, with support constraints, these depth maps are sufficient for retrieving semantic scenes. Figure 3.10d gives the room corner searching results. With the searched room corner as a reference system, the object models (Figure 3.10f) are built by matching their projection areas (Figure 3.10e) with the corresponding masks (Figure 3.10b).



(a) Instance masks (b) Depth maps

Figure 3.9: Ground truth data

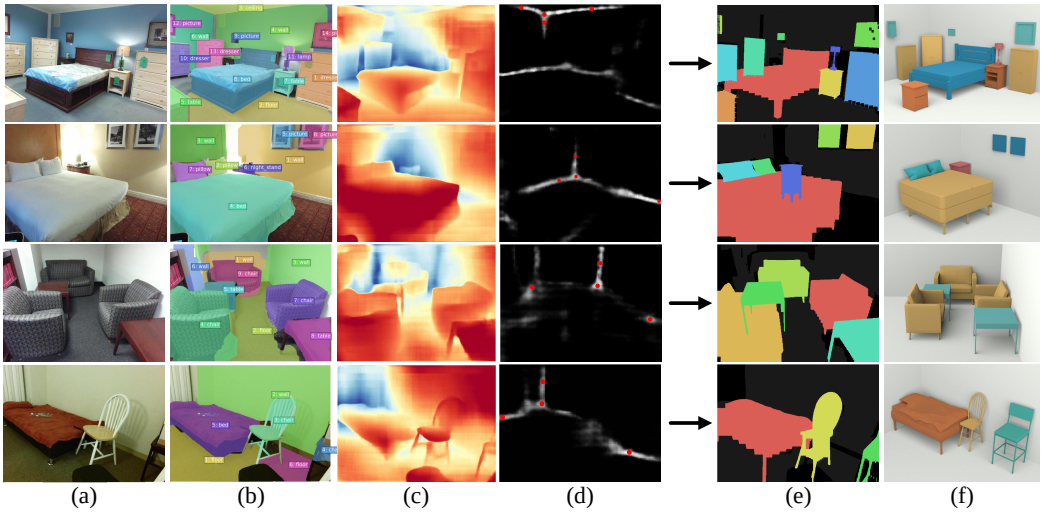


Figure 3.10: Semantic modelling results. (a) Test images; (b) Instance masks; (c) Depth maps; (d) Layout edge maps; (e) 2D Projections of matched models; (f) Retrieved semantic scenes

Table 3.1: Depth estimation errors

	rel	rms	log10
Avg	0.463	1.642	0.276

Time consumption details are listed in Table 3.2. Since the grab-cut algorithm processes masks by sequence, the quantity of objects determines the time cost in segmentation. Note that the No.3 scene in Table 4.1 consumes more time in scene modelling even its object number is not the largest one. It is because there are more chairs detected in this scene. 'Chair' is the largest category in our CAD model dataset, thus it will cost much more time in shape retrieval. The time consumed in layout estimation is distinct between cases as its efficiency is correlated to the sparsity of the layout edge map. For the modelling step, we calculate the average time cost of matching with a single model, and take the summation for all objects involved. Taking data loading, transferring and all the other factors into consideration, a semantic scene with dozens of objects can be built within five minutes.

From a visual point of view, Figure 3.10f illustrates that our algorithm can retrieve plausible indoor objects even for badly occluded ones (e.g. the

Table 3.2: Time consumption details (in seconds). ID: ID of test images; Num: Number of segmented objects; (a): Image segmentation; (b): Depth estimation; (c): Layout estimation; (d): Scene modelling

ID	Num	(a)	(b)	(c)	(d)
(1)	14	7.396	0.318	27.718	78.668
(2)	8	4.548	0.254	0.2398	78.137
(3)	9	5.713	0.304	1.1725	141.745
(4)	6	3.918	0.270	14.415	64.967

nightstand behind the bed in row 2, and the table behind the sofa in row 3, see Figure 3.10). This is mainly because we use the top point and the supported model of an object to deduce the height size. For occluded objects we can also obtain their spatial scope (see Figure 3.10e).

3.7.2 Comparisons and Limitations

We have compared our method with two closely related works (Zhang et al. 2015, Liu et al. 2017). There are some similarities within the modelling approach. All of our works have a single RGB image using a model database as the input and with the scene modelling completed in a data-driven manner. However, several differences exist. Firstly, our work benefits from high-level features with the three trained FCNs. There are fewer parameters in our method compared with hand-crafted methods that require features to be defined beforehand. Besides, in the segmentation part, different from (Liu et al. 2017), we do not require users to give a semantic label for objects. Also unlike (Zhang et al. 2015) where only main objects are segmented, more objects like pictures, blinds can be segmented in our work (see Figure 3.11). Secondly, these methods do not provide support semantics between objects. Offering the support context along with the reconstruction can provide cues for retrieving more objects that are supported by others (see Figure 3.12). In addition, we mainly handle scene modelling from photographs. For recovering scenes from rendered images, we need to loosen the constraints from point cloud and append extra placement priors.

(a) Input (b) Zhang et al. (2015) (c) Our method

Figure 3.11: Segmentation comparison with Zhang et al. (2015).

(a) Input (b) Liu et al. (2017) (c) Our method

Figure 3.12: Scene modelling result comparison with Liu et al. (2017).

There are some limitations in our work. Although we have tested that our method appears robust in handling noisy inputs, it could possibly fail when the pre-processing step does not work well. The weakest part is the layout edge map generation. For images whose layout edge map is not clear or the layout frame is occluded as Figure 3.13a shows, the corner searching algorithm could fail (see Figure 3.13b). In these cases, however, only a few manual interactions are required. As Figure 3.13c presents, four points on the original image are picked to correct the result. This can be used to improve the final performance (see Figure 3.13d). For cases with an extremely complicated support context, which cannot be parsed with some dataset, a novel support inference method, based on point cloud and image features, is required.

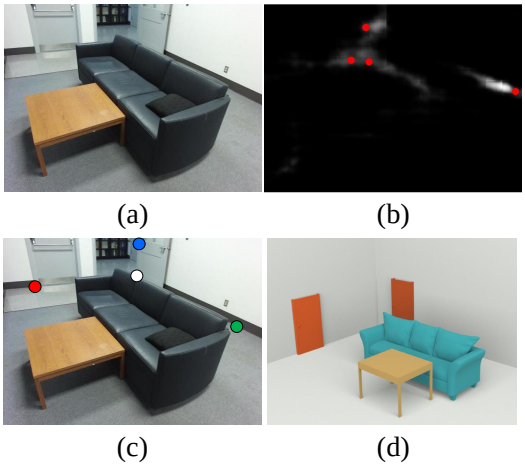


Figure 3.13: Modelling with interactions when layout estimation fails.

3.8 Summary

We propose an indoor scene modelling method with only a single photograph. Three FCN architectures are blended to produce different feature maps. We have shown that these high-level features can provide informative geometric clues and instance-level semantics for objects. Based on these features, support relationships between instances can be reasonably estimated in a data-driven manner. This offers an effective hierarchical constraint for the model matching, enabling our method to reconstruct objects with noisy inputs. The experiments show that we can retrieve reliable geometry with detailed support context for indoor scenes even when poorly occluded objects exist. From this work, we also observed many modules that bottlenecks the final performance, for example, the depth estimation module and support inference module. The failure of these modules would have a large impact to the scene modelling quality. In the next chapter, a unified scene modelling system is proposed to address this problem. It is designed to model 3D scenes with dense object instances and complex support conditions.

Chapter 4

Unified Scene Understanding and Modelling

Dense indoor scene modelling from 2D images has been bottlenecked due to the absence of depth information and cluttered occlusions. This chapter presents an automatic indoor scene modelling approach using deep features from neural networks. Given a single RGB image, our method simultaneously recovers semantic contents, 3D geometry and object relationship by reasoning indoor environment context. Particularly, we design a unified pipeline on the basis of convolutional networks for semantic scene understanding and modelling. It involves multi-level convolutional networks to parse indoor semantics/geometry into non-relational and relational knowledge. Non-relational knowledge extracted from shallow-end networks (e.g. room layout, object geometry) is fed forward into deeper levels to parse relational semantics (e.g. support relationship). A Relation Network is proposed to infer the support relationship between objects. All the structured semantics and geometry above are assembled to guide a global optimisation for 3D scene modelling. Qualitative and quantitative analysis demonstrates the feasibility of our method in understanding and modelling semantics-enriched indoor scenes by evaluating the performance of reconstruction accuracy, computation performance and scene complexity.

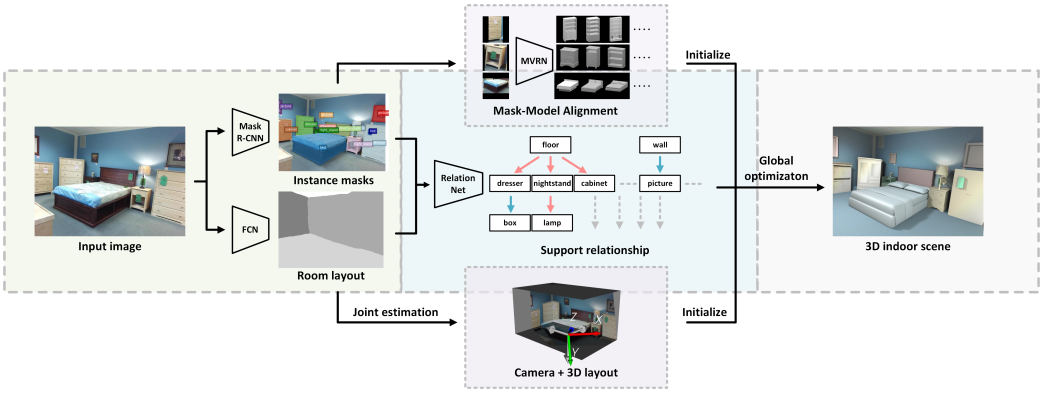


Figure 4.1: Pipeline of indoor scene modelling from a single image. The whole process is divided into three phases: 1. non-relational semantics parsing (e.g. room layout and object masks); 2. support relationship inference; 3. global scene optimization.

4.1 Method Overview

Our framework is built on the hypothesis that, features produced in each phase could be accumulated to feed into the consequent networks for deeper scene understanding. This process is divided into three phases, as illustrated in Figure 4.1. The first phase obtains non-relational semantics (i.e.room layout, object masks and labels) and retrieves a small set of 3D object candidates from a large model library (Section 4.2). This part takes advantage of a number of recent development in computer vision communities. We tailored a selection of methods to precondition the non-relational features for solving the scene modelling problem in later two phases.

In the second phase, we introduce a Relation Network to infer support relationships between objects (Section 4.3). This relational semantics offers physical constraints to organize those non-relational information into a reasonable contextual structure for 3D modelling.

The third phase assembles the geometric contents to model the 3D scene contextually consistent with these relational and non-relational semantics (Section 4.4). The 3D room layout and camera orientation are jointly estimated to ensure their consistence. It provides two coordinate systems (the room coordinate system and the camera coordinate system) for the global

optimisation in scene modelling and refinement.

4.2 Non-relational Semantics Parsing

2D Layout Estimation Layout estimation provides room boundary geometry (i.e. the location of the floor, ceiling and walls). Using CNNs to produce layout features, current works (Ren et al. 2016, Lee et al. 2017) generally ask for camera parameters to estimate vanishing points for layout proposal decision. We adopt the Fully Convolutional Network (FCN) from (Ren et al. 2016) to extract the layout edge map and label map. These feature maps present a coarse prediction of 2D layout (see Appendix B.4). An accurate 3D layout is jointly estimated along with camera parameters for further scene modelling (see Section 4.4.1).

Dense Scene Segmentation We segment images at the instance-level to obtain object category labels and corresponding 2D masks. Object masks present meaningful clues to initialize object sizes, 3D locations and orientations. Particularly, we introduce the Mask R-CNN (He et al. 2017) to capture object masks with instance segmentation. We customize the Mask R-CNN backboned by ResNet-101 (He et al. 2016), with the weights pre-trained on the MSCOCO dataset (Abdulla 2017). It is fine-tuned on the NYU v2 dataset (Silberman et al. 2012) which contains 1,449 densely labeled indoor images covering 37 common and 3 'other' categories. (The training strategy is detailed in Appendix B.1.1). Since object masks act significantly in the latter stages, we append Mask R-CNN with the Dense Conditional Random Field (DCRF) (Krähenbühl and Koltun 2011) to merge overlaps and improve mask edges. Besides, wall, floor and ceiling masks are removed as they are precisely decided in the 2D layout estimation. Segmentation samples are shown in Figure 4.2.

Image-based 3D Shape Retrieval This task is to retrieve CAD models with the most similar appearance to the segmented object images. A Multi-View Residual Network (MVRN) pretrained on ShapeNet Chang et al.

Figure 4.2: Instance segmentation samples

(2015) is introduced for shape retrieval. Similar with Izadinia et al. (2017), Huang et al. (2018b), we align and render each model from 32 viewpoints (two elevation angles at 15 and 30 degrees, and 16 uniform azimuth angles) for appearance-based matching. A Multi-View Convolutional Network Su et al. (2015) backboned with ResNet-50 is adopted as feature extractors to view CAD models from different viewpoints. This type of architecture is designed to mimic human eyes by observing an object from multiple viewpoints to recognize the object shape. Deep features from a single view is represented by a 2048-dimensional vector (i.e. the last layer size of ResNet-50). This compact descriptor enables us to match models efficiently in the vector space. The similarity between an image and a model can be measured by the cosine distance: $\max_{i\in[1,32]} \cos(\boldsymbol{f}, \boldsymbol{f}_i^{\mathrm{m}}), \boldsymbol{f}, \boldsymbol{f}_i^{\mathrm{m}} \in \mathcal{R}^{2048}$, where $\boldsymbol{f}$ and $\boldsymbol{f}_i^{\mathrm{m}}$ respectively denote the shape descriptor of the object image and a rendering of the model. The model set construction and training strategy are detailed in Appendix B.1.2. Furthermore, we fine-tune the orientation of

matched models with ResNet-34 (see Discussions). Figure 4.3 shows some matched samples on our model set. Top-5 candidates are selected for global scene optimisation in Section 4.4.

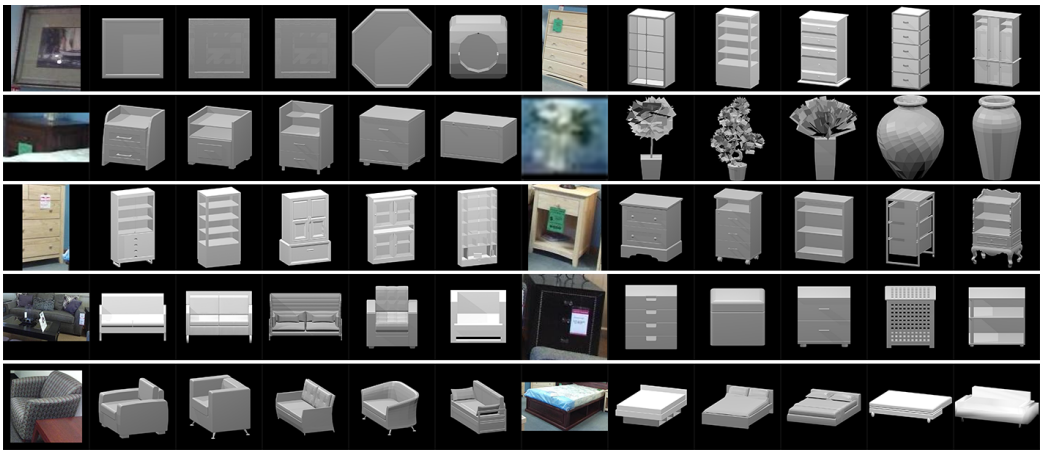


Figure 4.3: CAD model candidates. For each object image, we search our model dataset and output five similar candidates for scene modelling.

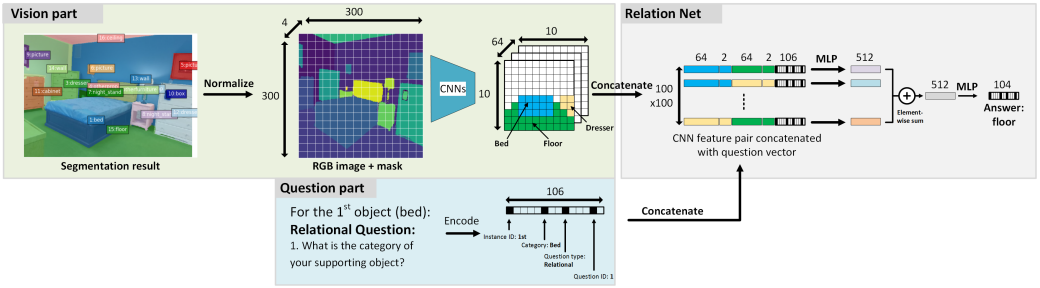


Figure 4.4: Relation Network for support inference. The whole architecture consists of three parts. The vision part and the question part are responsible for encoding object images and related questions separately, and the Relation Network answers these questions based on the image features.

4.3 Relational Support Reasoning

Section 4.2 dedicates to parsing indoor scenes into non-relational contents. We here aim to extract relational clues from these upstream outputs to con-

clude support relationships between objects. This relationship serves as physical constraints to guide scene modelling.

4.3.1 Relational Reasoning with Visual-Question Answering

As assumed in existing works (Silberman et al. 2012, Wong et al. 2015), two support types are considered in this thesis (i.e. support from behind, e.g. on a wall, or below, e.g. on a table). Every object except layout instances (i.e. wall, ceiling and floor) must be supported by another one. For objects which are supported by hidden instances, we treat them as being supported by layout instances.

Unlike non-relational semantics, relational context asks for not only the object property features, but also the contextual link between object pairs. Thus, a key is to combine the object feature pairs with specific task descriptions for support reasoning. It can be intuitively formulated as a Visual Question Answering (VQA) manner (Antol et al. 2015, Santoro et al. 2017): given the segmentation results, which instance is supporting object A? Is it supported from below or behind? With this insight, we configure a Relation Network to answer these support relationship questions by linking image features. Our network is designed as shown in Figure 4.4. The upstream of the Relation Network consists of two parts which encode visual images (with masks) and questions respectively.

4.3.2 Formulation for Support Inference

In the **Vision** part, the RGB image (colour intensities, 3-channel) is normalized to $[0, 1]$ and appended with its mask (instance labels, 1-channel), followed by a scale operation to a 300x300x4 matrix. We then generate 10x10x64 CNN feature vectors after convolutional operations. In the **Question** part, for each object instance, we customize our relational reasoning by answering two groups of questions: non-relational and relational; four questions for each. Taking the bed in Figure 4.4 as an example, the related questions and corresponding answers are encoded as shown in Figure 4.5. We

design the four relational questions for support inference, and the other four non-relational questions as regularization terms to make our network able to identify the target object we are querying. In our implementation, we train the network on NYU v2 (Silberman et al. 2012). In a single image, maximal 60 indoor instances with 40 categories are considered. Therefore, for the i-th object which belongs to the j-th category, we encode the k-th question from the m-th group to a 106-d (106=60+40+4+2) binary vector.

The outputs of the **Vision** and the **Question** parts are concatenated. We represent the 10x10x64 CNN features by 100 of 64-d feature vectors, and form all possible pairs of these feature vectors into 100x100 pairs. The 100x100 feature pairs are appended with their 2D coordinates (2-d) and exhaustively concatenated with the encoded question vector (106-d), then go through two multi-layer perceptrons to answer the questions (see network specifications in Appendix B.1.3). For each question, the Relation Network outputs a scalar between 0 and 103. We decode it into a human-language answer by indexing the lookup table as illustrated in Figure 4.5. The correct rate on the testing dataset of NYU v2 reaches 80.62% and 66.82 % on non-relational and relational questions respectively.

In our experiment, we observe that the numbering of instance masks is randomly given from the object segmentation, which undermines the network performance on the first relational question (see Figure 4.5). In our implementation, we use the last three relational questions to predict the category of the supporting object and the support type, and keep the first one as a regularization term. The exact supporting instance can be identified by maximizing the prior supporting probability between the target object and its neighbours:

$$\mathrm{O}_{j^*} = \underset{\mathrm{O}_j \in \mathcal{N}(\mathrm{O}_i)}{\operatorname{argmax}} \ \mathrm{P}(\mathcal{C}(\mathrm{O}_j)|\mathcal{C}(\mathrm{O}_i), \mathrm{T}_k),\ \mathcal{C}(\mathrm{O}_j) \in \mathcal{SC}(\mathrm{O}_i), \tag{4.1}$$

where O_i and $\mathcal{N}(\mathrm{O}_i)$ respectively denote the i-th object and its neighbouring instances (layout instances are neighbours to all objects). $\mathcal{C}(\mathrm{O}_j)$ represents the category label of object O_j. $\mathcal{SC}(\mathrm{O}_i)$ indicates the top-5 (in our experiment) category candidates of O_i's supporting object, and T_k denotes the

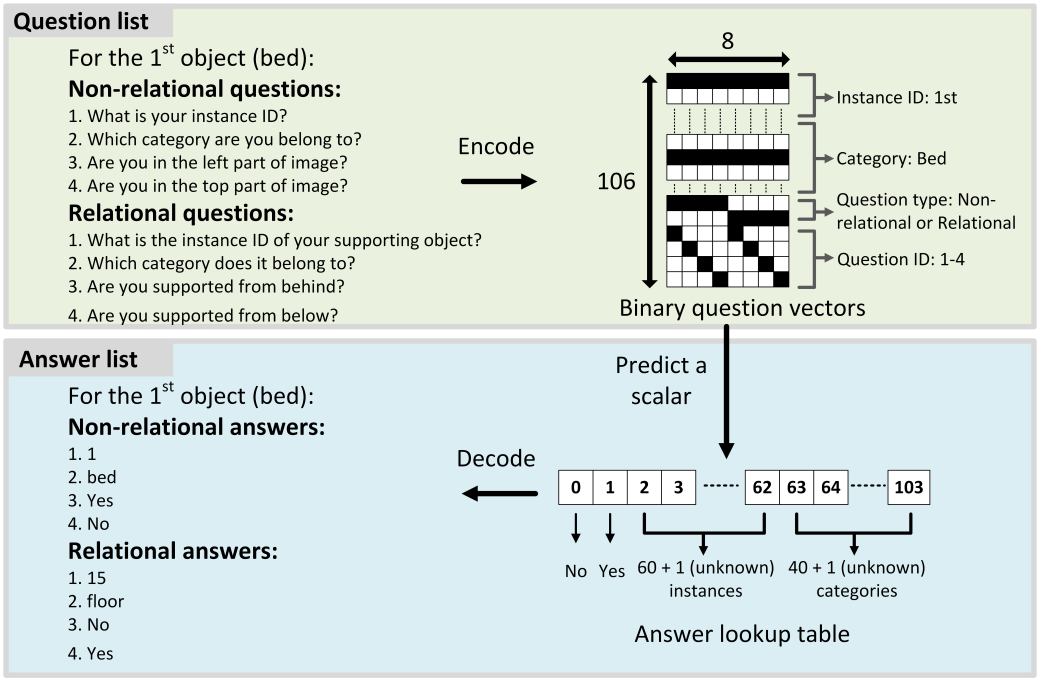


Figure 4.5: Questions and answers for training

support type, $k = 1, 2$. Hence $\mathrm{P}(\mathcal{C}(\mathrm{O}_j)|\mathcal{C}(\mathrm{O}_i), \mathrm{T}_k)$ means the probability of $\mathcal{C}(\mathrm{O}_j)$ supporting $\mathcal{C}(\mathrm{O}_i)$ by T_k. The prior probability P is obtained by counting from the training data (see Appendix B.2 for details). The supporting instance is represented by O_{j^*}. This process can improve the testing accuracy on the four relational questions by a large margin (from 66.82% to 82.74%).

4.4 Global Scene Optimization

The final process is composed of two steps: scene initialization and contextual refinement. The first step initializes camera, 3D layout and object properties. The second step involves an iterative refinement to pick correct object CAD models and fine-tune their sizes, locations and orientations with support relation constraints.

4.4.1 Scene Initialization

Camera-layout Joint Estimation The camera-layout estimation is illustrated in Figure 4.6. We jointly estimate camera parameters and a refined room layout by minimizing the angle deviations between the layout lines and vanishing lines in images (see Part I in Figure 4.6). We firstly detect line segments from both the original image and the layout label map using Line Segment Detection (LSD) (Von Gioi et al. 2012) and support vector machine (SVM) respectively. With the initialized camera parameters, orthogonal vanishing points are detected with the strategy proposed by Lu et al. (2017). The quality of vanishing points is scored by the count and length of the line segments they cross through. Longer line segments (like layout lines) would contribute more and guide the orthogonal vanishing lines in alignment with room orientation (see Part I in Figure 4.6). However, an improper camera initialization, particularly in cluttered environments, would often cause faulty estimation of 3D room layout (Huang et al. 2018b). We include iterations to improve the camera parameters from the detected line segments and produce a refined room layout simultaneously.

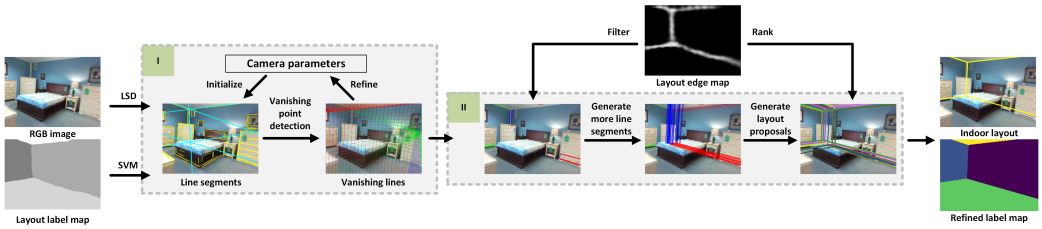


Figure 4.6: Camera-layout joint estimation. The camera parameters and vanishing points are jointly optimized in Part I, which leads to generate room layout proposals in Part II. The optimal layout is decided by the maximal probability score in layout edge map.

We denote the three orthogonal vanishing points by $\{\mathbf{vp}_i\}$, and the line segment set that (nearly) crosses through $\mathbf{vp}_i$ as $\mathcal{L}(\mathbf{vp}_i), i = 1, 2, 3$. Both of them are expressed by homogeneous coordinates. Similar to K-Means clustering, for the i-th cluster $\mathcal{L}(\mathbf{vp}_i)$, we re-estimate a new vanishing point $\mathbf{vp}_i^*$ by decreasing its distances to line segments in $\mathcal{L}(\mathbf{vp}_i)$. This problem can

be formulated as:

$$\begin{aligned} \mathbf{vp}_i^* &= \underset{\mathbf{vp}_i}{\operatorname{argmin}}\ \boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon}, \\ [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_{N_i}]^{\mathrm{T}}\mathbf{vp}_i &= \boldsymbol{\varepsilon}, i = 1, 2, 3, \end{aligned} \tag{4.2}$$

where $\boldsymbol{l}_k$ denotes the coordinates of a line segment in cluster $\mathcal{L}(\mathbf{vp}_i)$, $k = 1, 2, ..., N_i$. N_i is the capacity of $\mathcal{L}(\mathbf{vp}_i)$. We solve it with the eigen decomposition to obtain the eigen vector corresponding to the smallest eigen value of $[\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_{N_i}]^{\mathrm{T}}[\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_{N_i}]$ as the updated $\mathbf{vp}_i$. After that, camera parameters can be updated with the renewed vanishing points by Košecká and Zhang (2002). With this strategy, the vanishing points and camera parameters can be jointly optimized as each of them iteratively converges.

To obtain the optimal indoor layout (see Part II in Figure 4.6), the line segments that are not located in the layout edge map (high-intensity area) are removed, and we infer more line segments by connecting vanishing points with intersections of line segments from different clusters. More layout proposals can be generated by extensively combining these line segments (see this work Ren et al. (2016) for more details). We use the layout edge map to score each pixel in layout proposals and obtain the optimal one with the maximal sum. As the vanishing points provide the room orientation (Lu et al. 2017), we fit the indoor layout using a 3D cuboid, with the position of a room corner and layout sizes as optimisation variables (Hedau et al. 2009). Then the camera intrinsic and extrinsic parameters can be estimated. Samples of 3D room layout with calibrated cameras are shown in Figure 4.7.

Model Initialization Model retrieval (see Section 4.2) provides CAD models and orientations for indoor objects. In this part, we introduce single-view geometry combining with support relationship to estimate object sizes and positions with considering object occlusions. The room layout and vanishing points obtained in Section 4.4.1 are used to measure the height of each object. The whole process is illustrated in Figure 4.8.

Taking the nightstand and lamp in Figure 4.8 as examples, the object O_i (lamp) is supported by O_j (nightstand) from below. We denote the 2D mask of O_j by $\mathbf{M}_j$. $\mathbf{vp}_{\mathrm{v}} \in \mathcal{R}^2$ is the vertical vanishing point on the image

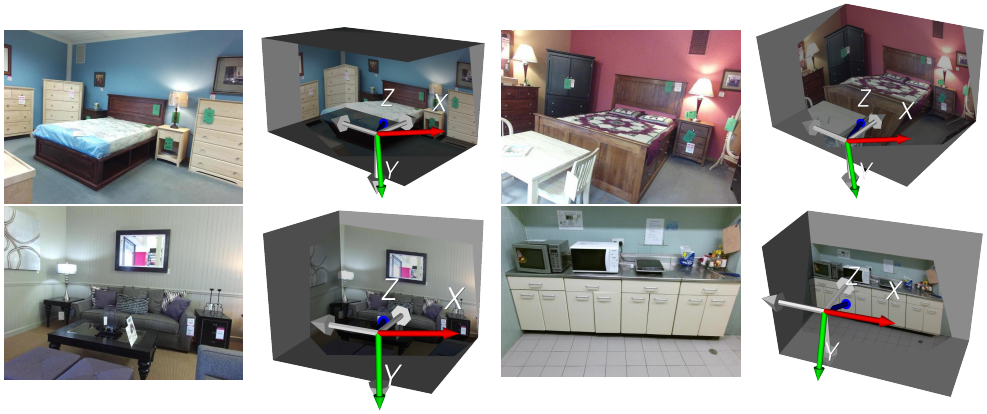


Figure 4.7: 3D room layout with camera orientation (left: original image, right: 3D layout). The coloured arrows represent the camera orientation. The gray arrows respectively point at the floor and walls, which indicates the room layout orientation.

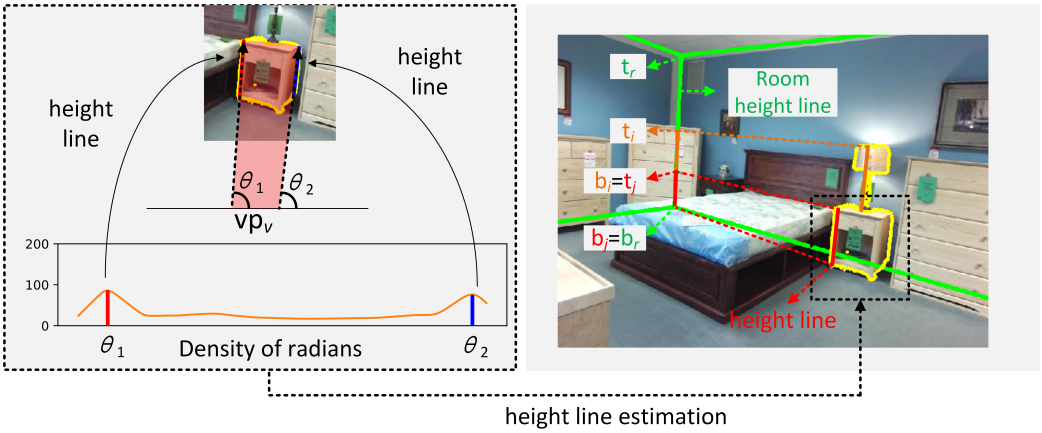


Figure 4.8: Single-view geometry for object height estimation

plane. For $\mathbf{M}_j$, we get its height line by scanning the mask boundary with rays originated from $\mathbf{vp}_\mathrm{v}$ (see Figure 4.8(left)). Each ray connects a pixel on the mask boundary with $\mathbf{vp}_\mathrm{v}$. We estimate the Gaussian kernel density of the radian of these rays, and extract the rays whose radian is a local maxima in density. The 'local maximal' ray that holds the longest intersection with the mask boundary is selected, and the longest intersection is taken as the optimal height line of O_j.

To estimate the real height of objects, we introduce single-view geometry for height measurement (see Figure 4.8(right)). Specifically, we take the room height line as the reference, and map object's height line onto the reference through the vanishing lines. For O_i (lamp), we denote its top and bottom of the mapped height line by t_i and b_i respectively. t_r and b_r respectively indicate the top and the bottom of the room height line. The height of O_i can be calculated by the cross ratio (Criminisi et al. 2000):

$$\begin{aligned} H_i &= A_i - A_j, \\ \frac{A_i}{H_r} &= \frac{\|t_i - b_r\|}{\|t_r - b_r\|} \cdot \frac{\|\mathbf{vp}_\mathrm{v} - t_r\|}{\|\mathbf{vp}_\mathrm{v} - t_i\|}, \end{aligned} \tag{4.3}$$

where A_i and A_j respectively denote the top altitude of O_i and O_j (i.e. the real height of $\overrightarrow{t_i b_r}$ and $\overrightarrow{t_j b_r}$). O_j is supporting O_i from below. Thus H_i is the real height of O_i. H_r is the real height of the room (i.e. the real height of $\overrightarrow{t_r b_r}$) and $\| * \|$ represents the Euclidean distance. We use this formula to recursively get the real height of O_i from the difference between the top altitude of O_i and its supporting object O_j. Rather than to address their real height individually, this recursive strategy asks for solving equations following the supporting order. It brings us benefits to verify the support type and solve occlusion problems. For example, the support type should be 'support from below' if H_i is larger than zero. Moreover, the bottom of an object (b_i) is usually invisible when it is occluded or not segmented out. While in practice, b_i is at the same altitude with t_j if O_j is supporting O_i from below. We replace b_i with t_j in calculations to estimate the real height of each object.

Unlike the 'support from below' scenarios where objects are stacked from the floor following the vertical direction, for objects that are supported from behind, the supporting surfaces are not guaranteed with a fixed normal direction. It would be much more complicated to get a closed-form solution. If O_i is supported by walls (like pictures), we can still get an accurate estimate by Equation 4.3 (i.e. height difference between $\overrightarrow{t_i b_r}$ and $\overrightarrow{b_i b_r}$). While for other cases (e.g. objects are supported by unknown surfaces), we still use this solution to get a rough estimate first. To ensure a reasonable height estimate, we

parse the ScanNet (Dai et al. 2017a) to generate a prior height distribution for each object category and replace those unreasonable estimates with the statistically average (see Appendix B.2 for details).

So far we have obtained the height estimate of each object and its altitude relative to the floor. With the room geometry and the camera parameters obtained in Section 4.4.1, the 3D location of objects can be estimated using the perspective relation between object masks and its spatial position, we refer readers to this work (Choi et al. 2015) for more details.

4.4.2 Contextual Refinement and Scene Modelling

When a room is full of clutter, there could still exist errors in scene initialization, and the aforementioned processes may not be sufficient to solve the scene modelling toward satisfaction. Therefore, a contextual refinement is adopted to fine-tune the CAD models and orientations from candidates (see Section 4.2). It refines their initial 3D size and position to make the reconstructed scene consistent in semantic and geometric meaning with the indoor context.

We formulate this into an optimisation problem:

$$
\begin{aligned}
&\max_{\theta_i, \boldsymbol{S}_i, \boldsymbol{O}_i, \boldsymbol{p}_i} \ \texttt{IoU}\{\texttt{Proj}[\boldsymbol{R}(\theta_i) \cdot \boldsymbol{S}_i \cdot \boldsymbol{O}_i + \boldsymbol{p}_i], \mathbf{M}_i\}, \\
&\boldsymbol{R}(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{S}_i = \begin{bmatrix} s_{i,1} & 0 & 0 \\ 0 & s_{i,2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot s_{i,3}, \\
&\boldsymbol{p}_i = \begin{bmatrix} p_{i,1}, & p_{i,2}, & p_{i,3} \end{bmatrix}^{\mathrm{T}}, \ i = 1, 2, ..., N.
\end{aligned}
\tag{4.4}
$$

$\boldsymbol{O}_i$ indicates 3D points in a model candidate of the i-th object. All CAD models are initially aligned and placed at the origin of the room coordinate system with the horizontal plane parallel to the floor. $\boldsymbol{S}_i$ is an anisotropic scaling matrix to control the 3D size of $\boldsymbol{O}_i$. $\boldsymbol{R}(\theta_i)$ and $\boldsymbol{p}_i$ are designed to adjust its orientation and position. $\texttt{Proj}[*]$ denotes the perspective projection to map coordinates from the room coordinate system to the image plane. $\texttt{IoU}[*]$ is the Intersection over Union operator. $\mathbf{M}_i$ represents the segmented mask of the i-th object. Therefore, the target of our contextual refinement is

to decide the CAD models $\{\boldsymbol{O}_i\}$ with orientations $\{\theta_i\}$, and adjust their size $\{\boldsymbol{S}_i\}$ and position $\{\boldsymbol{p}_i\}$ to make the 2D projections of those reconstructed objects approximate to our segmentation results. $i = 1, 2, ..., N$ and N indicates the count of segmented objects. We implement the scene refinement with a recursive strategy following the support relation constraints.

Support constraints from below For $\boldsymbol{O}_i$ that is supported by $\boldsymbol{O}_j$ from below, we ask for the geometric centre of $\boldsymbol{O}_i$ falling inside the supporting surface, and the bottom of $\boldsymbol{O}_i$ attached above the surface:

$$[\boldsymbol{R}(\theta_i) \cdot \boldsymbol{S}_i \cdot \boldsymbol{O}_i + \boldsymbol{p}_i]^{\mathrm{c}}_{x,y} \geqslant \min[\boldsymbol{O}_j]_{x,y}, \tag{4.5a}$$

$$[\boldsymbol{R}(\theta_i) \cdot \boldsymbol{S}_i \cdot \boldsymbol{O}_i + \boldsymbol{p}_i]^{\mathrm{c}}_{x,y} \leqslant \max[\boldsymbol{O}_j]_{x,y}, \tag{4.5b}$$

$$\min[\boldsymbol{R}(\theta_i) \cdot \boldsymbol{S}_i \cdot \boldsymbol{O}_i + \boldsymbol{p}_i]_{z|x,y} \geqslant \max[\boldsymbol{O}_j]_{z|x,y}, \tag{4.5c}$$

where $[*]^{c}_{x,y}$ indicates the horizontal coordinate (x, y) of the geometric centre, and $[*]_{z|x,y}$ is the altitude value at (x, y).

Support constraints from behind If $\boldsymbol{O}_i$ is supported by $\boldsymbol{O}_j$ from behind, we let $\boldsymbol{O}_i$ to be attached on a side surface of $\boldsymbol{O}_j$'s bounding box. Thus we do not ask for the orientation of $\boldsymbol{O}_i$ as it is consistent with the supporting surface. Considering there are four rectangular side surfaces, for each one, we build a local coordinate system $(\boldsymbol{o}_j^k, \boldsymbol{e}_j^{k,1}, \boldsymbol{e}_j^{k,2})$ on a vertex $\boldsymbol{o}_j^k$ and a pair of orthogonal edges $(\boldsymbol{e}_j^{k,1}, \boldsymbol{e}_j^{k,2})$ on these rectangles. $k \in [1, 2, 3, 4]$ indicates one of the four side surfaces, which is decided by solving the target function (4.4). Support constraints from behind can be written as:

$$0 \leqslant (\boldsymbol{c}_i - \boldsymbol{o}_j^k)^{\mathrm{T}} \cdot \boldsymbol{e}_j^{k,m} \leqslant \|\boldsymbol{e}_j^{k,m}\|^2, \; m = 1, 2, \tag{4.6a}$$

$$2(\boldsymbol{c}_i - \boldsymbol{o}_j^k)^{\mathrm{T}} \cdot \boldsymbol{n}_j^k = \mathtt{range}[(\boldsymbol{R}(\theta_i) \cdot \boldsymbol{S}_i \cdot \boldsymbol{O}_i)^{\mathrm{T}} \cdot \boldsymbol{n}_j^k], \tag{4.6b}$$

where

$$\boldsymbol{c}_i = [\boldsymbol{R}(\theta_i) \cdot \boldsymbol{S}_i \cdot \boldsymbol{O}_i + \boldsymbol{p}_i]^{\mathrm{c}}, \tag{4.6c}$$

$$\boldsymbol{n}_j^k = \boldsymbol{e}_j^{k,1} \times \boldsymbol{e}_j^{k,2} / \|\boldsymbol{e}_j^{k,1} \times \boldsymbol{e}_j^{k,2}\|. \tag{4.6d}$$

$\boldsymbol{c}_i$ is the geometric centre of the updated $\boldsymbol{O}_i$. $\boldsymbol{n}_j^k$ denotes the surface normal (see (4.6c) and (4.6d)). Hence, (4.6a) shows that the projection of $\boldsymbol{c}_i$ along $\boldsymbol{n}_j^k$ should fall inside the supporting surface. $\texttt{range}[x]$ means $x_{\max} - x_{\min}$. Therefore, (4.6b) implies that the distance between $\boldsymbol{c}_i$ and the surface should be a half of the object's size along the direction of $\boldsymbol{n}_j^k$. This is to secure the attachment of $\boldsymbol{O}_i$ onto the supporting surface. The only difference from constraint (4.5) is that the optimisation of object orientation turns to choosing a correct supporting surface.

To solve the target function (4.4), we adopt the exhaustive grid search to decide the exact $\{\boldsymbol{O}_i\}$ and $\{\theta_i\}$. For each grid, BOBYQA method (Powell 2009) is used to refine $\{\boldsymbol{S}_i\}$ and $\{\boldsymbol{p}_i\}$. We illustrate the convergence trajectory in Figure 4.9. The results demonstrate that the real height of every objects can be initially estimated before iterative refinement, even though there are heavy occlusions or objects that are not fully segmented. From the IoU curve, 30 iterations for model fine-tuning are enough to recover a whole scene.

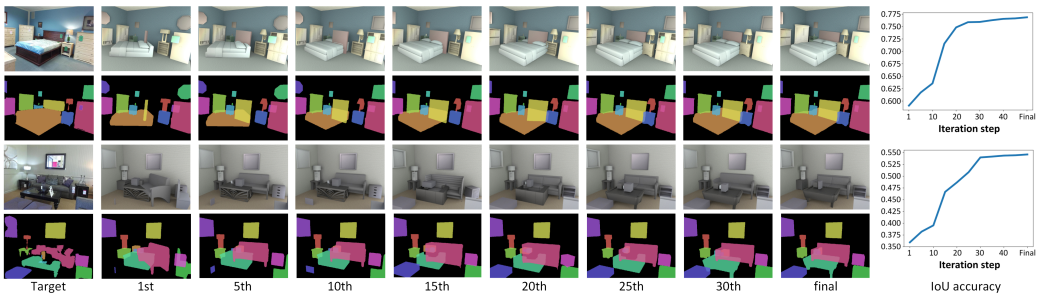


Figure 4.9: Scene modelling with contextual refinement. The leftmost column presents the original RGB images and the corresponding segmentation. The median part shows the scene modelling results by iterations. The rightmost column illustrates the iteration trajectory of `IoU` values correspondingly.

4.5 Experiments and Analysis

We present both qualitative and quantitative evaluation of our method with the NYU v2 (Silberman et al. 2012) and SUN RGB-D dataset (Song et al. 2015). All tests are implemented with Python 3.5 on a desktop PC with

one TITAN XP GPU and 8 Intel Xeon E5 CPUs. Parameters and network configurations are detailed in Appendix B.1.

4.5.1 Efficiency Analysis

We record the average time consumption of each phase for 654 testing samples of NYU v2 (see Table 4.1). The time cost in modelling a whole scene is related to its complexity. It is expected that modelling a cluttered room with more items costs more time. Object-specific tasks (segmentation, model retrieval) are processed in parallel. On average, it takes 2-3 minutes to process a indoor room of reasonable complexity containing up to 20 detected objects.

Table 4.1: Average time consumption (in seconds) of (1) 2D segmentation + DCRF refining, (2) model retrieval, (3) support inference, (4) camera-layout joint estimation, (5) model initialization and (6) scene modelling. 30 iterations are used in the contextual refinement, and the average number of detected objects is 16 over the 654 testing images.

Phase	(1)	(2)	(3)	(4)	(5)	(6)	Total
Time elapsed	9.87	9.72	2.08	25.53	0.95	69.68	117.84

4.5.2 Qualitative Evaluation

Figure 4.10 illustrates part of modelling results with different room types and various complexity (randomly picked from the SUN RGB-D dataset, see intermediate results and more samples in Appendix B.4). The results demonstrate that the detected objects are organized to make the overall presentation consistent with the original images (e.g., object orientation, position and support relationships). The same camera model as the one estimated from each input image is used in rendering, showing both the room layout and camera are reliably recovered with our joint estimation. Benefited from the robust support inference, objects that are heavily occluded or partly visible in the image are predicted with a plausible size.

We compare our outputs with the state-of-the-art works from (Izadinia et al. 2017, Huang et al. 2018b) (see Figure 4.11). For indoor cases with

Figure 4.10: Scene modelling samples on the SUN RGB-D dataset. Each sample consists of an original image (left), the reconstructed scene (raw mesh, middle) and the rendered scene with our estimated camera parameters (right).

few objects and occlusions (see Figure 4.11d, row (1), (2), (4) and (6)), our method extracts more small-size objects (like windows, books, pictures, pillows and lamps) in addition to the main furniture than both methods. This works well with the increasing of scene complexity. Objects that are of low-resolution, hidden or partly out of view can also be captured (see Figure 4.11a, row (1), (3), (6) and (7)). Both of the two works (Izadinia et al. 2017, Huang et al. 2018b) adopted detection-based methods to locate bounding boxes of objects in a 2D image, which would lose geometric details. Our 'instance segmentation + relational reasoning' approach not only provides more object shape details, but also preserves the relative size between objects. Our context refinement also aligns the recognized models in

a meaningful layout driven by the support-guided modelling.

Figure 4.11: Comparison with other methods. (a) and (d): The input images. (b) and (e): Reconstructed scenes from other works. The last row is provided by (Izadinia et al. 2017), and the remaining results are from (Huang et al. 2018b). (c) and (f): Our results. All the input images are from the SUN RGB-D dataset.

4.5.3 Quantitative Evaluation

We here quantitatively evaluate the 3D room layout prediction, support inference and 3D object placement. Dense modelling of indoor scenes requires the input image to be fully segmented at the instance level. Therefore, we adopt the NYU v2 dataset (795 images for training and 654 images for testing) to assess the tasks of support inference, and use its manually annotated

3D scenes (a subset of the SUN RGB-D annotation dataset) to evaluate the 3D layout prediction and object placement.

3D Room Layout The 3D room layout presents a reference for indoor object alignment and hence influences the object placement. Our method is validated by measuring the average 3D IoU of room bounding boxes between the prediction and the ground-truth (Song et al. 2015). Table 4.2 illustrates the performance of our method under two configurations: 1. with camera-layout joint estimation and 2. without joint estimation (to estimate camera parameters individually from vanishing points). The results from this ablation experiment show that the strategy of joint estimation consistently outperforms its counterpart in all room types. We also tested the average IoU for 'living rooms' and 'bedrooms' to compare with Izadinia et al. (2017). Our performance reaches 66.08% and Izadinia et al. (2017) achieves 62.6% on a subset of SUN RGB-D dataset.

Table 4.2: 3D room layout estimation. Our method is evaluated under two configurations in different room types.

Room type	bathroom	bedroom	classroom	computer lab	dining room	foyer
IoU (w/o joint)	30.71	39.36	47.60	20.47	46.28	54.30
IoU (w/ joint)	**34.90**	**62.86**	**68.23**	**83.21**	**60.41**	**65.59**
Room type	kitchen	living room	office	playroom	study room	mean IoU
IoU (w/o joint)	35.37	51.34	33.49	42.91	41.93	40.10
IoU (w/ joint)	**44.01**	**67.18**	**37.55**	**55.03**	**58.22**	**57.93**

Support Inference The testing dataset from NYU v2 contains 11,677 objects with known supporting instances and support types. Each object is queried with four relational questions. To make fair comparisons with existing methods, we use ground-truth segmentation to evaluate support relations (see Silberman et al. (2012)). The accuracy of our method is 72.99% at the object level, where a prediction is marked as correct only if all the four questions are correctly answered. This performance reaches the same plateau as existing methods using RGB-D inputs (74.5% by (Xue et al. 2015) and 72.6%

by (Silberman et al. 2012)) and largely outperforms the method using RGB inputs (48.2% by (Zhuo et al. 2017)). It demonstrates the feasibility of our Relation Network in parsing support relations from complicated occlusion scenarios without any depth clues.

3D Object Placement The accuracy of 3D object placement is tested using manually annotated 3D bounding boxes along with the evaluation benchmark provided by (Song et al. 2015), where the mean average precision (mAP) of the 3D IoU between the predicted bounding boxes and the ground-truth is calculated. We align the reconstructed and ground-truth scenes to the same size by unifying the camera altitude, and compare our result with the state-of-the-art (Huang et al. 2018b). Different from their work, our method is designed for modelling full scenes with considering all indoor objects, while they adopted a sparsely annotated dataset SUN-RGBD for evaluation with their 30 object categories. As the ground-truth bounding boxes of objects are not fully labelled, we remove those segmented masks that are not annotated to enable comparison under the same configuration. Table 4.3 shows our average precision scores on the NYU-37 classes (Silberman et al. 2012) (excluding 'wall', 'floor' and 'ceiling'; mAP is calculated with IoU threshold at 0.15). We obtain the mAP score at 11.49. From Huang et al. (2018b)'s work, they achieved 12.07 on 15 main furniture and 8.06 on all their 30 categories. It shows that our approach achieves better performance in 'smaller' objects, which is in line with the qualitative analysis. The reason could be twofold: 1. a well-trained segmentation network can capture more shape details of objects (e.g. object contour) than using 2D bounding box localization; 2. most human-made objects appear with clear line segments or contours (cabinet, nightstand, dresser, etc.) which benefits our camera-layout joint estimation and model initialization. However, for objects with a rather thin or irregular shape, or under incomplete segmentation (like chair, pillow and lamp et al.), the performance would drop by a small extent.

Ablation Analysis We implement the ablation analysis to discuss which module in our pipeline contributes most to the final 3D object placement.

Table 4.3: 3D object detection. We compare our method under three configurations: 1. without camera-layout joint estimation (w/o joint); 2. without Relation Network (w/o RN); 3. with joint estimation and Relation Network (all). The values show the average precision score on our shared object classes. The column 'others' contains the remaining NYU v2 categories (mAP is averaged by 34 categories, i.e. NYU-37 classes excluding 'wall', 'floor', 'ceiling').

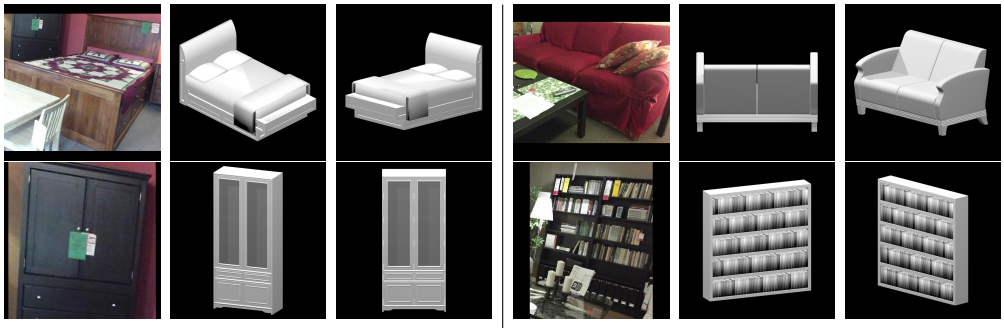
Method	bathtub	bed	bookshelf	cabinet	chair	desk	door	dresser	fridge	lamp
Huang et al. (2018b)	2.84	**58.29**	7.04	0.48	**13.56**	4.79	**1.56**	13.71	15.18	2.41
Ours (w/o joint)	30.83	22.62	5.83	1.82	1.12	4.31	0.68	28.53	25.25	3.12
Ours (w/o RN)	40.00	54.21	6.67	3.59	2.13	7.61	0.16	31.74	45.37	2.78
Ours (all)	**44.88**	55.53	**9.41**	**4.58**	6.49	**7.69**	0.18	**37.76**	**52.08**	**3.65**

Method	nightstand	person	pillow	shelves	sink	sofa	table	toilet	others	mAP
Huang et al. (2018b)	8.80	4.04	-	-	2.18	28.37	12.12	16.50	-	-
Ours (w/o joint)	8.35	5.00	0.58	1.02	0.00	24.26	7.65	13.13	0.00	5.41
Ours (w/o RN)	32.51	8.08	0.20	3.57	0.25	31.93	7.56	10.74	0.00	8.53
Ours (all)	**32.52**	**18.52**	**1.19**	**33.31**	**3.85**	**33.49**	**13.68**	**31.77**	0.00	**11.49**

Two ablated configurations are considered (see Table 4.3): 1. without camera-layout joint estimation (Izadinia et al. 2017), 2. without Relation Network (replaced with prior-based support inference (Nie et al. 2018)). The mAP scores of the first and second configurations are 5.41 and 8.53 respectively. Our final score is 11.49. It implies that both the camera-layout joint estimation and relational reasoning contribute to the final performance, and room layout has a higher impact to the object placement in single-view modelling. It is expected that, the orientation and placement of the room layout largely influence the object placement. We also observe that prior-based support inference is more sensitive to occlusions and segmentation quality (Nie et al. 2018, Xue et al. 2015). When indoor scenes are cluttered, occlusions generally make the supporting surfaces invisible and the segmentation under quality. Unlike the Relation Network, the prior-based method does not take spatial relationship into account and chooses a supporting instance only considering the prior probability, making it more error-prone to complicated scenes.

4.5.4 Discussions

Improving the Estimation of Object Orientation Although the view-based model matching provides an initial guess of object orientation (see



(a) (b) (c) (d) (e) (f)

Figure 4.12: Orientation correction. (a) and (d): The object images. (b) and (e): Matched models from MVRN. (c) and (f): Corrected orientations.

Section 4.2), those deep features are in some cases too abstract to decide sufficiently accurate orientation for trustworthy model initialization. For each object mask, we specifically append a ResNet-34 to predict the orientation angle relative to the camera. It is trained on our dataset considering eight uniformly sampled orientations (i.e. $\pi/4, \pi/2, ..., 2\pi$). However, there is a gap between the renderings (which we used for training) and the real-world images. Rather than conducting full-layer training, we fix the shallowest three layers with the weights pretrained on ImageNet to make our network sensitive to real images. The training data is augmented with coarse drop-out to mimic occlusion effects, and random perspective & affine transformations to mimic different camera poses. The top-1 precision on our testing dataset reaches 91.81% (22342 models for training, 2482 models for testing). Figure 4.12 illustrates samples from the testing dataset and their predicted orientations. In practice, orientation of some specific models is ambiguous (e.g. symmetric shapes). Top-3 orientation candidates are selected and transformed into the room coordinate system for global scene modelling.

Limitations Our method faces challenges when objects are segmented out with very few pixels (at the minimum of 24x21) which could be too raw for the MVRN to match their shape details. Our CAD model dataset currently contains 37 common categories of indoor objects. Its capacity is limited

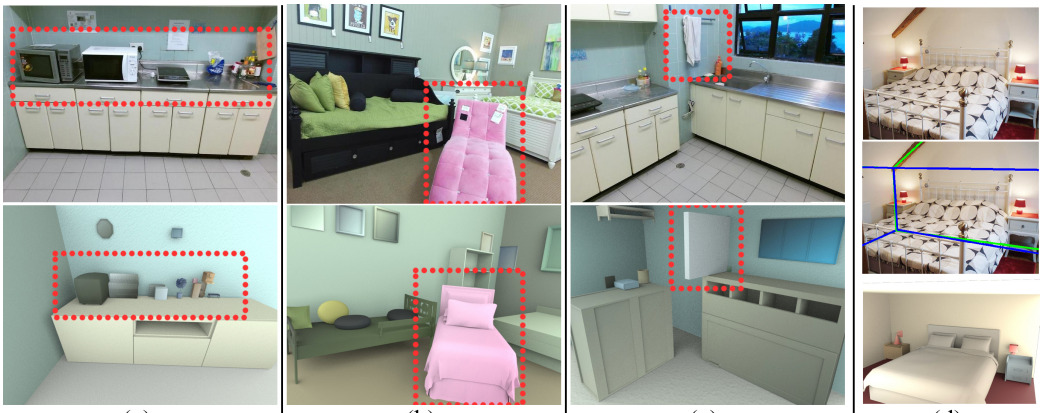


Figure 4.13: Limitation cases. Objects that are segmented with rather few pixels (a), out of our model repository (b) or from 'other category' (right) may not get a proper geometry estimate. For 'non-Manhattan' room layout (d), we fit it with a cuboid. The green and blue lines in (d) respectively represent the 2D room layout and the projection of the 3D layout.

relative to the diversity of real-world indoor environments. While for unknown objects (labelled as 'other category'), we currently use a cuboid to approximate their shape. Besides, our current method would fit any room layout with a box, which would fail when handling extremely irregular room shapes. Therefore, those reasons above would undermine the IoU accuracy in our contextual refinement, and we illustrate those cases in Figure 4.13.

4.6 Summary

We develop a unified scene modelling approach by fully leveraging convolutional features to reconstruct semantic-enriched indoor scenes from a single RGB image. A shallow-to-deep process parses relational and non-relational context into structured knowledge to guide the scene modelling. The experiments demonstrate the capability of our approach in (1) automatically inferring the support relationship of objects, (2) dense scene modelling to recover 3D indoor geometry, with enriched semantics and trustworthy modelling results. Our quantitative evaluations further demonstrate the functionality and effectiveness of each substep in producing semantically-consistent 3D scenes.

This work aims at 3D scene modelling through fully understanding scene context from images. There are high-level relational semantics among indoor objects that could be incorporated into the modelling-by-understanding approach, like other complex contact relations (e.g. a person sits on a chair and holds a mug). All these mixed semantics would help our system to better understand and represent the scene context in a meaningful way. It suggests our future work to provide an intelligent scene knowledge structure to configure and deploy them towards scene modelling. In the next chapter, a single end-to-end network is developed for semantic scene reconstruction. In contrast to scene modelling, no external CAD dataset and hand-crafted optimisation process are required. The 3D room layout, camera poses, object bounding boxes and meshes are predicted within a single architecture that produces the final 3D scenes in one go.

Chapter 5

Toward Total 3D Scene Understanding and Mesh Reconstruction

Semantic scene modelling usually requires a shape CAD dataset for model retrieval. In this chapter, we attempt to recover the scene geometry with an end-to-end semantic reconstruction manner. That is, given a single RGB image, we directly learn to predict semantic object instances with meshes as the output without relying on an external shape dataset.

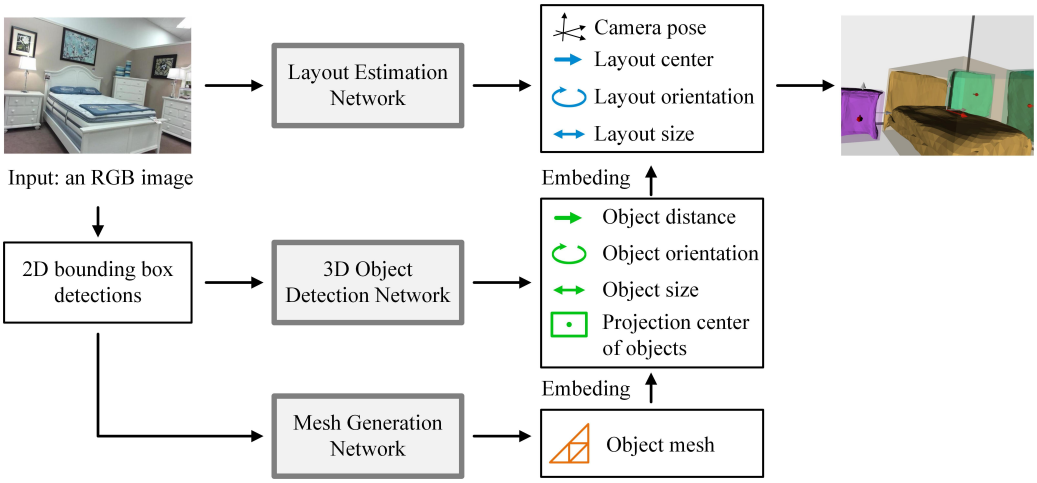
Semantic reconstruction of indoor scenes refers to both scene understanding and object reconstruction. Existing works either address one part of this problem or focus on independent objects. In this chapter, we bridge the gap between understanding and reconstruction, and propose an end-to-end solution to jointly reconstruct room layout, object bounding boxes and meshes from a single image. Instead of separately resolving scene understanding and object reconstruction, our method builds upon a holistic scene context and proposes a coarse-to-fine hierarchy with three components: 1. room layout with camera pose; 2. 3D object bounding boxes; 3. object meshes (see Figure 5.1). We argue that understanding the context of each component can assist the task of parsing the others, which enables joint understanding and reconstruction. The experiments on the SUN RGB-D and Pix3D datasets demonstrate that our method consistently outperforms existing methods in indoor layout estimation, 3D object detection and mesh reconstruction.

Figure 5.1: From a single image (left), we simultaneously predict the contextual knowledge including room layout, camera pose, and 3D object bounding boxes (middle) and reconstruct object meshes (right).

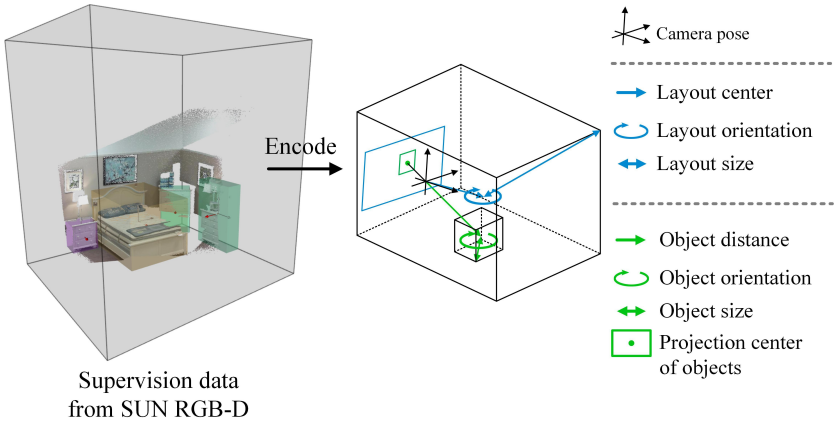
5.1 Method Overview

We illustrate our overview in Figure 5.2a. The network architecture follows a 'box-in-the-box' manner and consists of three modules: 1. Layout Estimation Network (LEN); 2. 3D Object Detection Network (ODN); 3. Mesh Generation Network (MGN). From a single image, we first predict 2D object bounding boxes with Faster R-CNN Ren et al. (2015). LEN takes the full image to produce the camera pose and the layout bounding box. Given the 2D object detections, ODN predicts their 3D bounding box in the camera system, while MGN generates the mesh geometry in their object-centric system. We reconstruct the full scene mesh by embedding the outputs of all networks together with joint training and inference, where object meshes from MGN are scaled and placed into their bounding boxes (by ODN) and transformed into the world system with the camera pose (by LEN).

To bridge the gap between scene understanding and object mesh reconstruction, we unify them together with joint learning, and simultaneously predict room layout, camera pose, 3D object bounding boxes and meshes (Figure 5.1). The insight is that object meshes in a scene manifest spatial occupancy that could help 3D object detection, and the 3D detection pro-



(a) Architecture of the scene reconstruction network



(b) Parameterisation of the learning targets

Figure 5.2: Overview of our approach. (a) The hierarchy of our method follows a 'box-in-the-box' manner using three modules: the Layout Estimation Network (LEN), 3D Object Detection Network (ODN) and Mesh Generation Network (MGN). A full scene mesh is reconstructed by embedding them together with joint inference. (b) The parameterisation of our learning targets in LEN and ODN (Huang et al. 2018a).

vides with object alignment that enables object-centric reconstruction at the instance-level. Unlike voxel grids, coordinates of reconstructed meshes are differentiable, thus enabling the joint training by comparing the output mesh with the scene point cloud (e.g. on SUN RGB-D (Song et al. 2015)). With the above settings, we observe that the performance on scene understanding

and mesh reconstruction can make further progress and reach the state-of-the-art on the SUN RGB-D (Song et al. 2015) and Pix3D (Sun et al. 2018) datasets.

In summary, we list our key components of this chapter as follows:

- We provide a solution to automatically reconstruct room layout, object bounding boxes, and meshes from a single image. To our best knowledge, it is the first work of end-to-end learning for comprehensive 3D scene understanding with mesh reconstruction at the instance level. This integrative approach shows the complementary role of each component and reaches the state-of-the-art on each task.

- We propose a novel density-aware topology modifier in object mesh generation. It prunes mesh edges based on local density to approximate the target shape by progressively modifying mesh topology. Our method directly tackles the major bottleneck of Pan et al. (2019a), which is in the requirement of a strict distance threshold to remove detached faces from the target shape. Compared with Pan et al. (2019a), our method is robust to diverse shapes of indoor objects under complex backgrounds.

- Our method takes into account the attention mechanism and multilateral relations between objects. In 3D object detection, the object pose has an implicit and multilateral relation with surroundings, especially in indoor rooms (e.g., bed, nightstand, and lamp). Our strategy extracts the latent features for better deciding object locations and poses, and improves 3D detection.

The details of each network are described below.

5.2 3D Object Detection and 3D Layout Estimation

To make the bounding box of layout and objects learnable, we parameterise a box as the prior work (Huang et al. 2018a) (Figure 5.2b). We set up

the world system located at the camera centre with its vertical (y-) axis perpendicular to the floor, and its forward (x-) axis toward the camera, such that the camera pose $\mathbf{R}(\beta, \gamma)$ can be decided by the pitch and roll angles (β, γ). In the world system, a box can be determined by a 3D centre $\boldsymbol{C} \in \mathbb{R}^3$, spatial size $\boldsymbol{s} \in \mathbb{R}^3$, orientation angle $\theta \in [-\pi, \pi)$. For indoor objects, the 3D centre $\boldsymbol{C}$ is represented by its 2D projection $\boldsymbol{c} \in \mathbb{R}^2$ on the image plane with its distance $d \in \mathbb{R}$ to the camera centre. Given the camera intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$, $\boldsymbol{C}$ can be formulated by:

$$\boldsymbol{C} = \mathbf{R}^{-1}(\beta, \gamma) \cdot d \cdot \frac{\mathbf{K}^{-1}[\boldsymbol{c}, 1]^T}{\|\mathbf{K}^{-1}[\boldsymbol{c}, 1]^T\|_2}. \tag{5.1}$$

The 2D projection centre $\boldsymbol{c}$ can be further decoupled by $\boldsymbol{c}^b + \boldsymbol{\delta}$. $\boldsymbol{c}^b$ is the 2D bounding box centre and $\boldsymbol{\delta} \in \mathbb{R}^2$ is the offset to be learned. From the 2D detection $\boldsymbol{I}$ to its 3D bounding box corners, the network can be represented as a function by $\mathbf{F}\left(\boldsymbol{I}|\boldsymbol{\delta}, d, \beta, \gamma, \boldsymbol{s}, \theta\right) \in \mathbb{R}^{3\times8}$. The ODN estimates the box property $(\boldsymbol{\delta}, d, \boldsymbol{s}, \theta)$ of each object, and the LEN decides the camera pose $\mathbf{R}(\beta, \gamma)$ with the layout box $\left(\boldsymbol{C}, \boldsymbol{s}^l, \theta^l\right)$.

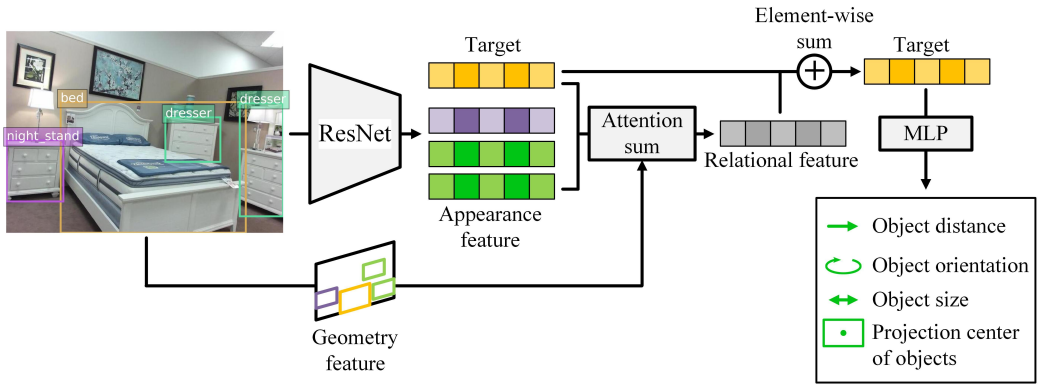


Figure 5.3: 3D Object Detection Network (ODN)

Object Detection Network (ODN) In indoor environments, object poses generally follow a set of interior design principles, making it a latent learnable pattern. Previous works either predict 3D boxes object-wisely (Huang et al. 2018a, Tulsiani et al. 2018) or only consider pair-wise relations (Kulkarni

et al. 2019). In our work, we assume each object has a *multi-lateral relation* between its surroundings, and take all in-room objects into account in predicting its bounding box. The network is illustrated in Figure 5.3. Our method is inspired by the consistent improvement of attention mechanism in 2D object detection (Hu et al. 2018). For 3D detection, we first object-wisely extract the appearance feature with ResNet-34 (He et al. 2016) from 2D detections, and encode the relative position and size between 2D object boxes into geometry feature with the method in Hu et al. (2018), Vaswani et al. (2017). For each target object, we calculate its **relational feature** to the others with the object relation module (Hu et al. 2018). It adopts a piece-wise feature summation weighted by the similarity in appearance and geometry from the target to the others, which we call '**attention sum**' in Figure 5.3. We then element-wisely add the relational feature to the target and regress each box parameter in $(\boldsymbol{\delta}, d, \boldsymbol{s}, \theta)$ with a two-layer MLP. For indoor reconstruction, the object relation module reflects the inherent significance in the physical world: objects generally have stronger relations with the others which are neighbouring or appearance-similar. We demonstrate its effectiveness in 3D object detection in our ablation analysis.

Layout Estimation Network (LEN) The LEN predicts the camera pose $\mathbf{R}(\beta, \gamma)$ and its 3D box $\left(\boldsymbol{C}, \boldsymbol{s}^l, \theta^l\right)$ in the world system. In this part, we employ the same architecture as ODN but remove the relational feature. $\left(\beta, \gamma, \boldsymbol{C}, \boldsymbol{s}^l, \theta^l\right)$ are regressed with two fully-connected layers for each target after the ResNet. Similar to Huang et al. (2018a), the 3D centre $\boldsymbol{C}$ is predicted by learning an offset to the average layout centre.

5.3 Density-aware 3D Mesh Generation

Our Mesh Generation Network directly tackles the major issue with one recent work, Topology Modification Network (TMN) (Pan et al. 2019a): TMN approximates object shapes by deforming and modifying the mesh topology, where a predefined distance threshold is required to remove detached faces from the target shape. However, it is non-trivial to give a general threshold

for different scales of object meshes (see Figure 5.5e). One possible reason is that indoor objects have a large shape variance among different categories. Another one is that complex backgrounds and occlusions often cause the failure of estimating a precise distance value.

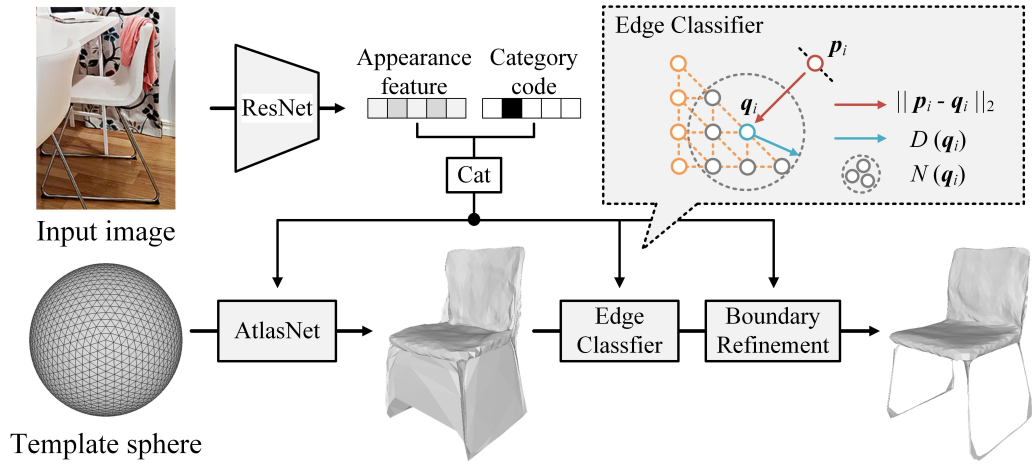


Figure 5.4: Mesh Generation Network (MGN). Our method takes as input a detected object which is vulnerable to occlusions, and outputs a plausible mesh.

5.3.1 Density Definition on Mesh Points

Density v.s. Distance Different from TMN where a strict distance threshold is used for topology modification, we argue that whether to reserve a face or not should be determined by its local geometry. In this part, we propose an adaptive manner that modifies meshes based on the local **density** of the ground-truth. We set $\boldsymbol{p}_i \in \mathbb{R}^3$ as a point on our reconstructed mesh, and $\boldsymbol{q}_i \in \mathbb{R}^3$ corresponds to its nearest neighbour on the ground-truth (see Figure 5.4). We design a binary classifier $f(*)$ to predict whether $\boldsymbol{p}_i$ is close to the ground-truth mesh in Equation 5.2:

$$
\begin{aligned}
f(\boldsymbol{p}_i) &= \begin{cases} \text{False} & \|\boldsymbol{p}_i - \boldsymbol{q}_i\|_2 > D(\boldsymbol{q}_i) \\ \text{True} & \text{otherwise} \end{cases}, \\
D(\boldsymbol{q}_i) &= \max_{\boldsymbol{q}_m, \boldsymbol{q}_n \in N(\boldsymbol{q}_i)} \min \|\boldsymbol{q}_m - \boldsymbol{q}_n\|_2, m \neq n
\end{aligned}
\tag{5.2}
$$

where $N(\boldsymbol{q}_i)$ are the neighbours of $\boldsymbol{q}_i$ on the ground-truth mesh, and $D(\boldsymbol{q}_i)$ is defined as its local density. This classifier is designed by our insight that: in shape approximation, a point should be reserved if it belongs to the neighbours $N(*)$ of the ground-truth. We also observe that this classifier shows better robustness with different mesh scales than using a distance threshold (see Figure 5.5).

Edges v.s. Faces Instead of removing faces, we choose to cut mesh edges for topology modification. We randomly sample points on mesh edges and use the classifier $f(*)$ to cut edges on which the average classification score is low. It is from the consideration that cutting false edges can reduce incorrect connections penalized by the edge loss (Wang et al. 2018a) and create compact mesh boundaries.

5.3.2 Mesh Generation Network

We illustrate our network architecture in Figure 5.4. It takes a 2D detection as input and uses ResNet-18 to produce image features. We encode the detected object category into a one-hot vector and concatenate it with the image feature. It is from our observation that the category code provides shape priors and helps to approximate the target shape faster. The augmented feature vector and a template sphere are fed into the decoder in AtlasNet (Groueix et al. 2018a) to predict deformation displacement on the sphere and output a plausible shape with unchanged topology. The edge classifier has the same architecture with the shape decoder, where the last layer is replaced with a fully connected layer for classification. It shares the image feature, takes the deformed mesh as input and predicts the $f(*)$ to remove redundant meshes. We then append our network with a boundary refinement module (Pan et al. 2019a) to refine the smoothness of boundary edges and output the final mesh.

5.4 End-to-end Learning for Total 3D Understanding

In this section, we conclude the learning targets with the corresponding loss functions, and describe our joint loss for end-to-end training.

Individual losses ODN predicts $(\boldsymbol{\delta}, d, \boldsymbol{s}, \theta)$ to recover the 3D object box in the camera system, and LEN produces $\left(\beta, \gamma, \boldsymbol{C}, \boldsymbol{s}^l, \theta^l\right)$ to represent the layout box, along with the camera pose to transform 3D objects into the world system. As directly regressing absolute angles or length with L2 loss is error-prone (Huang et al. 2018a, Qi et al. 2018). We keep inline with them by using the classification and regression loss $\mathcal{L}^{cls,reg} = \mathcal{L}^{cls} + \lambda_r \mathcal{L}^{reg}$ to optimize $\left(\theta, \theta^l, \beta, \gamma, d, \boldsymbol{s}, \boldsymbol{s}^l\right)$. We refer readers to Huang et al. (2018a) for details. As $\boldsymbol{C}$ and $\boldsymbol{\delta}$ are calculated by the offset from a pre-computed centre, we predict them with L2 loss. For MGN, we adopt the Chamfer loss $\mathcal{L}_c$, edge loss $\mathcal{L}_e$ and boundary loss $\mathcal{L}_b$ (as defined in Groueix et al. (2018a), Wang et al. (2018a), Pan et al. (2019a)) to supervise surface prediction, where Chamfer loss is to make sure the predicted surface points close to the ground-truth. The edge loss and boundary loss are to guarantee that the predicted surface points are uniformly distributed while keeping consistent boundary smoothness with the ground-truth. Besides, as mentioned in Section 5.3, we deploy a cross-entropy loss $\mathcal{L}_{ce}$ for classifying edges in mesh modification.

Joint losses We define the joint loss between ODN, LEN and MGN based on two insights: 1. the camera pose estimation should improve 3D object detection, and vice versa; 2. object meshes in a scene present spatial occupancy that should benefit the 3D detection, and vice versa. For the first, we adopt the cooperative loss $\mathcal{L}_{co}$ from Huang et al. (2018a) to ensure the consistency between the predicted world coordinates of layout & object boxes and the ground-truth. This cooperative loss is able to guarantee that each object box should be located into the room layout box, otherwise it will be penalized. For the second, we require the reconstructed meshes close to their point cloud in the scene. It exhibits global constraints by aligning mesh coordinates with

the ground-truth. We define the global loss as the partial Chamfer distance (Groueix et al. 2018a):

$$\mathcal{L}_g = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathbb{S}_i|}\sum_{\boldsymbol{q}\in\mathbb{S}_i}\min_{\boldsymbol{p}\in\mathbb{M}_i}\|\boldsymbol{p}-\boldsymbol{q}\|_2^2, \tag{5.3}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ respectively indicate a point on a reconstructed mesh $\mathbb{M}_i$ and the ground-truth surface $\mathbb{S}_i$ of i-th object in the world system. N is the number of objects and $|\mathbb{S}_i|$ denotes the point number on $\mathbb{S}_i$. Unlike single object meshes, real-scene point clouds are commonly coarse and partially covered (scanned with depth sensors), thus we do not use the Chamfer distance to define $\mathcal{L}_g$. All the loss functions in joint training can be concluded as:

$$\begin{aligned}\mathcal{L} = &\sum_{x\in\{\boldsymbol{\delta},d,\boldsymbol{s},\theta\}}\lambda_x\mathcal{L}_x + \sum_{y\in\{\beta,\gamma,\boldsymbol{C},\boldsymbol{s}^l,\theta^l\}}\lambda_y\mathcal{L}_y \\ &+ \sum_{z\in\{c,e,b,ce\}}\lambda_z\mathcal{L}_z + \lambda_{co}\mathcal{L}_{co} + \lambda_g\mathcal{L}_g,\end{aligned} \tag{5.4}$$

where the first three terms represent the individual loss in ODN, LEN and MGN, and the last two are the joint terms. $\{\lambda_*\}$ are the weights used to balance their importance.

5.5 Results and Evaluation

5.5.1 Experiment Setup

Datasets We use two datasets in our experiments: 1) **SUN RGB-D** dataset (Song et al. 2015) consists of 10,335 real indoor images with labelled 3D layout, object bounding boxes and coarse point cloud (depth map). We use the official train/test split and NYU-37 object labels (Silberman et al. 2012) for evaluation on layout, camera pose estimation and 3D object detection. 2) **Pix3D** dataset (Sun et al. 2018) contains 395 furniture models with 9 categories, which are aligned with 10,069 images. We use this for mesh generation and keep the train/test split inline with (Gkioxari et al. 2019). The object label mapping from NYU-37 to Pix3D for scene reconstruction is listed in Appendix C.4.

Metrics Our results are measured on both scene understanding and mesh reconstruction metrics. We evaluate layout estimation with average 3D Intersection over Union (IoU). The camera pose is evaluated by the mean absolute error. Object detection is tested with the average precision (AP) on all object categories. We test the single-object mesh generation with the Chamfer distance as previous works (Gkioxari et al. 2019, Pan et al. 2019a), and evaluate the scene mesh with Equation 5.3.

Implementation We train the 2D detector (Figure 5.2a) on the COCO dataset (Lin et al. 2014) first and fine-tune it on SUN RGB-D. In MGN, the template sphere has 2562 vertices with unit radius. We cut edges whose average classification score is lower than 0.2. Since SUN RGB-D does not provide full instance meshes for 3D supervision, and Pix3D is only labeled with one object per image without layout information. We first train ODN, LEN on SUN-RGBD, and train MGN on Pix3D individually. We then combine Pix3D into SUN RGB-D to provide mesh supervision and jointly train all networks with the loss $\mathcal{L}$ in Equation 5.4. Here we use one hierarchical batch (each batch contains one scene image with N object images) in joint training. We explain the full architecture, training strategies, time efficiency and parameter setting of our networks in Appendix C.2.

5.5.2 Qualitative Analysis and Comparison

In this section, we evaluate the qualitative performance of our method on both object and scene levels.

5.5.2.1 Object Reconstruction

We compare our MGN with the state-of-the-art mesh prediction methods (Gkioxari et al. 2019, Groueix et al. 2018a, Pan et al. 2019a) on Pix3D. Because our method is designed to accomplish scene reconstruction in real scenes, we train all methods inputted with object images but without masks. For AtlasNet (Groueix et al. 2018a) and Topology Modification Network (TMN) (Pan et al. 2019a), we also encode the object category into image

features enabling a fair comparison. Both TMN and our method are trained following a 'deformation+modification+refinement' process (see Pan et al. (2019a)). For Mesh R-CNN (Gkioxari et al. 2019), it involves an object recognition phase, and we directly compare with the results reported in their paper. The comparisons are illustrated in Figure 5.5, from which we observe that indoor furniture are often overlaid with miscellaneous backgrounds (such as books on the shelf). From the results of Mesh R-CNN (Figure 5.5b), it generates meshes from low-resolution voxel grids (24^3 voxels) and thus results in noticeable artifacts on mesh boundaries. TMN improves from AtlasNet and refines shape topology. However, its distance threshold τ does not show consistent adaptability for all shapes in indoor environments (e.g. the stool and the bookcase in Figure 5.5e). Our method relies on the edge classifier. It cuts edges depending on the local density, making the topology modification adaptive to different scales of shapes among various object categories (Figure 5.5f). The results also demonstrate that our method keeps better boundary smoothness and details.

5.5.2.2 Scene Reconstruction

As this is the first work, to our best knowledge, of combing scene understanding and mesh generation for full scene reconstruction, we illustrate our results on the testing set of SUN RGB-D in Figure 5.6 (see more samples in Appendix C.6). Note that SUN RGB-D does not contain ground-truth object meshes for training. We present the results under different scene types and diverse complexities to test the robustness of our method. The first row in Figure 5.6 shows the scenes with large repetitions and occlusions. We exhibit the cases with disordered object orientations in the second row. The third and the fourth rows present the results under various scene types, and the fifth row shows the performance in handling cluttered and 'out-of-view' objects. All the results manifest that, with different complexities, our method maintains visually appealing object meshes with reasonable object placement.

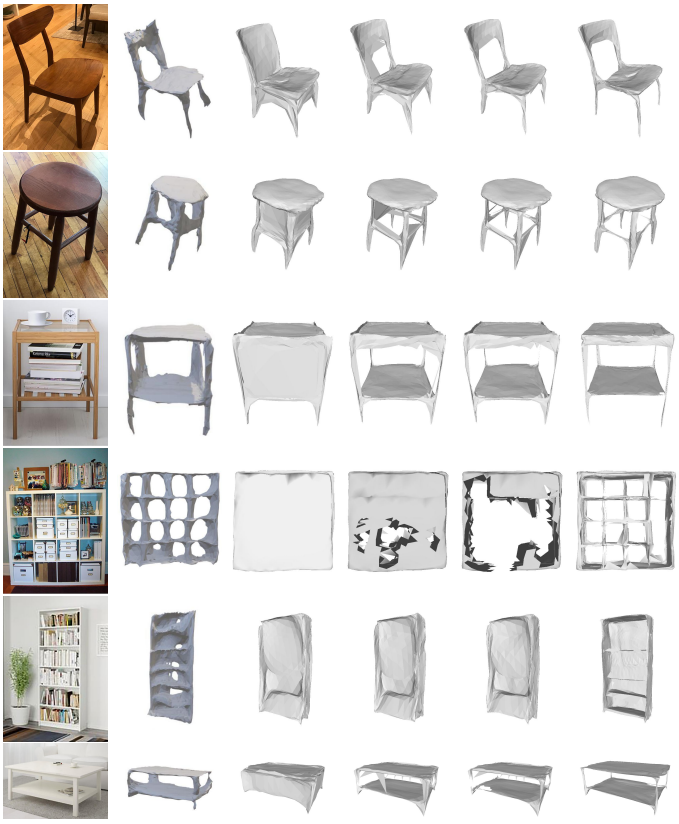


Figure 5.5: Mesh reconstruction for individual objects. From left to right: (a) Input images and results from (b) Mesh R-CNN Gkioxari et al. (2019), (c) AtlasNet-Sphere (Groueix et al. 2018a), (d, e) TMN with $\tau = 0.1$ and $\tau = 0.05$ (Pan et al. 2019a), (f) Ours.

5.5.3 Quantitative Analysis and Comparison

We compare the quantitative performance of our method with the state-of-the-arts on four aspects: 1. layout estimation; 2. camera pose prediction; 3. 3D object detection and 4. object and scene mesh reconstruction. The object mesh reconstruction is tested on Pix3D, and the others are evaluated on SUN RGB-D. We also ablate our method by removing joint training: each subnetwork is trained individually, to investigate the complementary benefits of combining scene understanding and object reconstruction.

Figure 5.6: Scene reconstruction on SUN RGB-D. Given a single image, our method end-to-end reconstructs the room layout, camera pose with object bounding boxes, poses and meshes.

Layout Estimation We compare our method with existing layout understanding works (Choi et al. 2013, Huang et al. 2018b a). As shown in Table 5.1, joint training with room layout, object bounding boxes and meshes helps to improve the layout estimation, providing a gain of 2 points than the state-of-the-arts.

Camera Pose Estimation Camera pose is defined by $\mathbf{R}(\beta, \gamma)$, hence we evaluate the pitch β and roll γ with the mean absolute error with the ground-truth. The results are show in Table 5.1, where we observe that joint learning also benefits the camera pose estimation.

3D Object Detection We investigate the object detection with the benchmark consistent with Huang et al. (2018a), where the mean average precision (mAP) is employed using 3D bounding box IoU. A detection is considered true positive if its IoU with the ground-truth is larger than 0.15. We compare our method with existing 3D detection works (Choi et al. 2013, Huang et al. 2018b a) on the shared object categories in Table 5.2. The full table on all

Table 5.1: Comparisons of 3D layout and camera pose estimation on SUN RGB-D. We report the average IoU to evaluate layout prediction (higher is better), and the mean absolute error of pitch and roll angles (in degree) to test camera pose (lower is better). Note that our camera axes are defined in a different order with (Huang et al. 2018a) (see Appendix C.1).

Method	3D Layout	Cam pitch	Cam roll
Choi et al. (2013)	19.2	-	-
Hedau et al. (2009)	-	33.85	3.45
Huang et al. (2018b)	54.9	7.60	3.12
Huang et al. (2018a)	56.9	3.28	2.19
Ours (w/o. joint)	57.6	3.68	2.59
Ours (joint)	**59.2**	**3.15**	**2.09**

object categories is listed in Appendix C.3. The comparisons show that our method significantly improves over the state-of-the-art methods, and consistently advances the ablated version. The reason could be two-fold. One is that the global loss $\mathcal{L}_g$ in joint learning involves geometry constraint which ensures the physical rationality, and the other is that multi-lateral relational features in ODN benefit the 3D detection in predicting spatial occupancy.

Table 5.2: Comparisons of 3D object detection. We compare the average precision of detected objects on SUN RGB-D (higher is better). Huang et al. (2018a)* shows the results from their paper, which are trained with fewer object categories. Huang et al. (2018a)** presents the model trained on the NYU-37 object labels for a fair comparison.

Method	bed	chair	sofa	table	desk	dresser	nightstand	sink	cabinet	lamp	mAP
Choi et al. (2013)	5.62	2.31	3.24	1.23	-	-	-	-	-	-	-
Huang et al. (2018b)	58.29	13.56	28.37	12.12	4.79	13.71	8.80	2.18	0.48	2.41	14.47
Huang et al. (2018a)*	**63.58**	17.12	41.22	26.21	9.55	4.28	6.34	5.34	2.63	1.75	17.80
Huang et al. (2018a)**	57.71	15.21	36.67	31.16	19.90	15.98	11.36	15.95	10.47	3.28	21.77
Ours (w/o. joint)	59.03	15.98	43.95	35.28	23.65	19.20	6.87	14.40	11.39	3.46	23.32
Ours (joint)	60.65	**17.55**	**44.90**	**36.48**	**27.93**	**21.19**	**17.01**	**18.50**	**14.51**	**5.04**	**26.38**

We also compare our work with Tulsiani et al. (2018) to evaluate object pose prediction. We keep consistent with them by training on the NYU v2 dataset (Silberman et al. 2012) with their six object categories and ground-truth 2D boxes. The results are reported in Table 5.3. Object poses are

tested with errors in object translation, rotation and scale. We refer readers to Tulsiani et al. (2018) for the definition of the metrics. The results further demonstrate that our method not only obtains reasonable spatial occupancy (mAP), but also retrieves faithful object poses.

Table 5.3: Comparisons of object pose prediction. The difference values of translation, rotation and scale between the predicted and the ground-truth bounding boxes on NYU v2 are reported, where the median and mean of the differences are listed in the first two columns (lower is better). The third column presents the correct rate within a threshold (higher is better).

Method	Translation (metres)			Rotation (degrees)			Scale		
	Median (lower is better)	Mean	(Err≤0.5m)% (higher is better)	Median (lower is better)	Mean	(Err≤30°)% (higher is better)	Median (lower is better)	Mean	(Err≤0.2)% (higher is better)
Tulsiani et al. (2018)	0.49	0.62	51.0	14.6	**42.6**	63.8	0.37	0.40	18.9
Ours (w/o. joint)	0.52	0.65	49.2	15.3	45.1	64.1	0.28	0.29	42.1
Ours (joint)	**0.48**	**0.61**	**51.8**	**14.4**	43.7	**66.5**	**0.22**	**0.26**	**43.7**

Mesh Reconstruction We evaluate mesh reconstruction on both the object and scene levels. For object reconstruction, we compare our MGN with the state-of-the-arts (Groueix et al. 2018a, Pan et al. 2019a) in Table 5.4. We ablate our topology modification method with two versions: 1. removing faces instead of edges (w/o. edge); 2. using distance threshold (Pan et al. 2019a) instead of our local density (w/o. dens) for topology modification. The results show that each module improves the mean accuracy, and combining them advances our method to the state-of-the-art. A possible reason is that using local density keeps small-scale topology, and cutting edges is more robust in avoiding incorrect mesh modification than removing faces. Mesh reconstruction of scenes is evaluated with $\mathcal{L}_g$ in Equation 5.3, where the loss is calculated with the average distance from the point cloud of each object to its nearest neighbour on the reconstructed mesh. Different from single object reconstruction, scene meshes are evaluated considering object alignment in the world system. In our test, $\mathcal{L}_g$ decreases from 1.89e-2 to 1.43e-2 with our joint learning.

Table 5.4: Comparisons of object reconstruction on Pix3D. The Chamfer distance is used in evaluation. 10K points are sampled from the predicted mesh after being aligned with the ground-truth using ICP. The values are in units of 10^{-3} (lower is better).

Category	bed	bookcase	chair	desk	sofa	table	tool	wardrobe	misc	mean
Groueix et al. (2018a)	9.03	6.91	8.37	8.59	6.24	19.46	6.95	4.78	40.05	12.26
Pan et al. (2019a)	7.78	**5.93**	6.86	7.08	4.25	17.42	4.13	4.09	**23.68**	9.03
Ours (w/o. edge)	8.19	6.81	6.26	5.97	4.12	15.09	3.93	4.01	25.19	8.84
Ours (w/o. dens)	8.16	6.70	6.38	**5.12**	4.07	16.16	3.63	4.32	24.22	8.75
Ours	**5.99**	6.56	**5.32**	5.93	**3.36**	**14.19**	**3.12**	**3.83**	26.93	**8.36**

5.6 Ablation Analysis and Discussion

To better understand the effect of each design on the final result, we ablate our method with five configurations:

C_0: without relational features (in ODN) and joint training (Baseline).

C_1: Baseline + relational features.

C_2: Baseline + (only) cooperative loss $\mathcal{L}_{co}$ in joint training.

C_3: Baseline + (only) global loss $\mathcal{L}_g$ in joint training.

C_4: Baseline + joint training ($\mathcal{L}_g$ + $\mathcal{L}_{co}$).

Full: Baseline + relational features + joint training.

Table 5.5: Ablation analysis in layout estimation, 3d object detection and scene mesh reconstruction on SUN RGB-D. The $\mathcal{L}_g$ values are in units of 10^{-2}. The mAP and $\mathcal{L}_g$ values are averaged on the object categories in Table 5.2.

Version	Layout (IoU) (higher is better)	3D Objects (mAP) (higher is better)	Scene mesh ($\mathcal{L}_g$) (lower is better)
C_0	57.63	20.19	2.10
C_1	57.63	23.32	1.89
C_2	58.21	21.77	1.73
C_3	57.92	24.59	1.64
C_4	58.87	25.62	1.52
Full	**59.25**	**26.38**	**1.43**

We test the layout estimation, 3D detection and scene mesh reconstruction with 3D IoU, mAP and $\mathcal{L}_g$. The results are reported in Table 5.5, from

which we observe that:
$\boldsymbol{C_0}$ v.s.$\boldsymbol{C_4}$ and $\boldsymbol{C_1}$ v.s. **Full**: Joint training consistently improves layout estimation, object detection and scene mesh reconstruction no matter using relational features or not.
$\boldsymbol{C_0}$ v.s.$\boldsymbol{C_1}$ and $\boldsymbol{C_4}$ v.s. **Full**: Relational features help to improve 3D object detection, which indirectly reduces the loss in scene mesh reconstruction.
$\boldsymbol{C_0}$ v.s.$\boldsymbol{C_2}$ and $\boldsymbol{C_0}$ v.s. $\boldsymbol{C_3}$: In joint loss, both $\mathcal{L}_{co}$ and $\mathcal{L}_g$ in joint training benefit the final outputs, and combing them further advances the accuracy.

We also observe that the global loss $\mathcal{L}_g$ shows the most effect on object detection and scene reconstruction, and the cooperative loss $\mathcal{L}_{co}$ provides more benefits than others on layout estimation. Besides, scene mesh loss decreases with the increasing of object detection performance. It is inline with the intuition that object alignment significantly affects mesh reconstruction. Fine-tuning MGN on SUN RGB-D can not improve single object reconstruction on Pix3D. It reflects that object reconstruction depends on clean mesh for supervision. All the facts above explain that the targets for full scene reconstruction actually are intertwined together, which makes joint reconstruction a feasible solution toward total scene understanding.

5.7 Summary

We develop an end-to-end indoor scene reconstruction approach from a single image. It embeds scene understanding and mesh reconstruction for joint training, and automatically generates the room layout, camera pose, object bounding boxes and meshes to fully recover the room and object geometry. Extensive experiments show that our joint learning approach significantly improves the performance on each subtask and advances the state-of-the-arts. It indicates that each individual scene parsing process has an implicit impact on the others, revealing the necessity of training them integratively toward total 3D reconstruction. One limitation of our method is the requirement for dense point cloud for learning object meshes, which is labour-consuming to obtain in real scenes. It is due to the depth ambiguity problem in single-view reconstruction. In the next chapter, we attempt to use different inputs

(i.e. depth images) for shape reconstruction to explore the future of scene reconstruction with hybrid input modalities.

Chapter 6

Skeleton-bridged Shape Completion

Previous chapters have discussed how to predict object shapes from single RGB images, i.e. with shape retrieval or reconstruction. However, the colour intensities from the input does not indicate geometric clues of object surfaces. In this chapter, we would like to investigate the reconstruction performance with different input modality, i.e. depth image. In contrary to RGB inputs, depth scans provide geometry constraints of object shapes, which enables better reconstruction on surface details. Given the partial, observable depth scan of a target object, how to predict the full object shape refers to the problem of shape completion. Existing works usually estimate the missing shape by decoding a latent feature encoded from the input points. However, real-world objects are usually with diverse topologies and surface details, which a latent feature may fail to represent to recover a clean and complete surface. To this end, we propose a skeleton-bridged point completion network (**SK-PCN**) for shape completion. Given a partial scan, our method first predicts its 3D skeleton to obtain the global structure, and completes the surface by learning displacements from skeletal points. We decouple the shape completion into structure estimation and surface reconstruction, which eases the learning difficulty and benefits our method to obtain on-surface details. Besides, considering the missing features during encoding input points, SK-PCN adopts a local adjustment strategy that merges the input point cloud to our predictions for surface refinement. Comparing with previous

Figure 6.1: Given a partial scan of an object (green points, backprojected from a depth image), SK-PCN estimates its meso-skeleton (grey points) to explicitly extract the global structure, and pairs the local-global features for displacement regression to recover the full surface points (blue points) with normals for mesh reconstruction (right).

methods, our skeleton-bridged manner better supports point normal estimation to obtain the full surface mesh beyond point clouds. The qualitative and quantitative experiments on both point cloud and mesh completion show that our approach outperforms the existing methods on various object categories.

6.1 Method Overview

To preserve the shape structure and complete surface details, we provide a new completion manner, namely **SK-PCN**, that maps the partial scan to the complete surface bridged via the *meso-skeleton* (Wu et al. 2015) (see Figure 6.1). Recovering the missing structure and details from an incomplete scan generally requires both global and local features. Instead of using encoders to extract a latent layer response as the global feature (Yu et al. 2018, Huang et al. 2020, Yuan et al. 2018, Wang et al. 2020), we explicitly learn the meso-skeleton as the global abstraction of objects, which is represented by 3D points located around the medial axis of a shape. Comparing with surface points, meso-skeleton holds a more smooth and compact shape domain, making networks easier to be trained. It also keeps the shape structure that helps predict topology-consistent meshes.

To further recover surface details, prior works usually expand the global feature with upsampling (Yu et al. 2018, Li et al. 2019b) or skip connections (Wang et al. 2020, Wen et al. 2020) by revisiting local features from previous layers. Our method completes shape details with an interpretable manner,

that is learning to grow surface points from the meso-skeleton. Specifically, SK-PCN dually extracts and pairs the local features from the partial scan and the meso-skeleton under multiple resolutions, to involve corresponding local features to skeletal points. As local structures are commonly with repetitive patterns (e.g., table legs are usually with the same geometry), we bridge these local-global feature pairs with a **Non-Local Attention** module to select and propagate those contributive local features from the global surface space onto skeletal points for missing shape completion. Moreover, to preserve the fidelity on observable regions, we devise a local refinement module and a patch discriminator to merge the original scan to the output. Unlike previous methods where point coordinates are directly regressed, we complete the surface by learning displacements from skeletal points. It is not only because learning residuals is easier for training (He et al. 2016). These displacement values show high relevance with surface normals (Wu et al. 2015), which better supports the point normal estimation for our mesh reconstruction.

Our contributions are three-fold. First, we provide a novel learning modality for point completion by mapping partial scans to complete surfaces bridged via meso-skeletons. This intermediate representation preserves better shape structure to recover a full mesh beyond point clouds. Second, we correspondingly design a completion network SK-PCN. It end-to-end aggregates the multi-resolution shape details from the partial scan to the shape skeleton, and automatically selects the contributive features in the global surface space for shape completion. Third, we fully leverage the original scan for local refinement, where a surface adjustment module is introduced to fine-tune our results for a high-fidelity completion. Extensive experiments on various categories demonstrate that our method outperforms previous methods and reaches the-state-of-the-art.

We illustrate the architecture of SK-PCN in Figure 6.2. Given a partial scan, we aim at completing the missing geometries while preserving fidelity on the observable region. To this end, our SK-PCN is designed with a generator for surface completion, and a patch discriminator to distinguish and refine our results with the ground-truth. The generator has two phases: skeleton

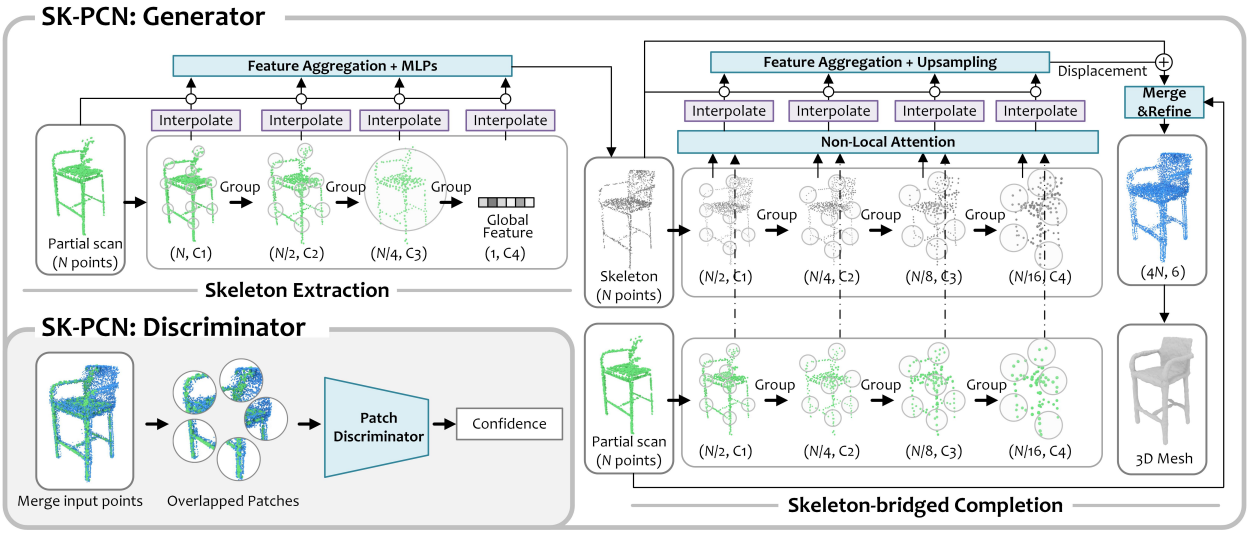


Figure 6.2: Network architecture of our method. SK-PCN consists of a shape generator and a patch discriminator. The shape generator produces a meso-skeleton first, and uses it to aggregate the multi-resolution local features on the global surface space for surface completion. The patch discriminator measures the fidelity score of our completion results on the overlapped area with the input scan. The layer specifications are detailed in Appendix D.1.

extraction and skeleton-bridged completion. The skeleton extraction module groups and parallelly aggregates the multi-resolution feature from the input to predict the skeletal points. The completion module shares the similar feature extraction process. It dually obtains multi-resolution features from both the skeleton and the input, and pairs them on each resolution scale (see Figure 6.2). For each pair, a Non-Local Attention module is designed to search the contributive local features from the partial scan to each skeletal point. These local features are then interpolated back to the skeletal points and aggregated to regress their displacements to the shape surface with the corresponding normal vectors on the surface. To preserve the shape information of the observable region, we merge the input to our shape followed with surface adjustment and produce the final mesh with Poisson Surface Reconstruction (Kazhdan and Hoppe 2013). The details of each submodule are elaborated as follows.

6.2 Learning Meso-Skeleton with Global Inference

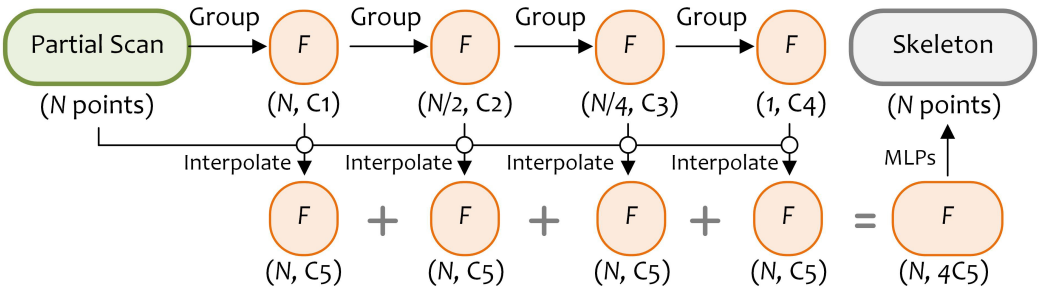
Meso-skeleton is an approximation of the shape medial axis. In our work, the ground-truth meso-skeletons are calculated with (Wu et al. 2015), and we represent them by 3D points for learning. As skeletons only keep shape structure, they do not preserve surface details. To this end, we devise a multi-scale feature aggregation to obtain point features under different resolutions (see Figure 6.3a). We adopt the set abstraction layers of (Qi et al. 2017b) to progressively group and downsample point clouds to the coarser scale and obtain the global feature. Afterwards, these multi-scale point features are interpolated back to the partial scan with the feature propagation (Qi et al. 2017b). Then we concatenate them together to regress the skeletal point coordinates with MLPs. It attaches global features from different resolutions to the partial scan and relies on the network to select the useful ones for skeletal point regression.

6.3 Skeleton-to-Surface Reconstruction with Non-Local Attention

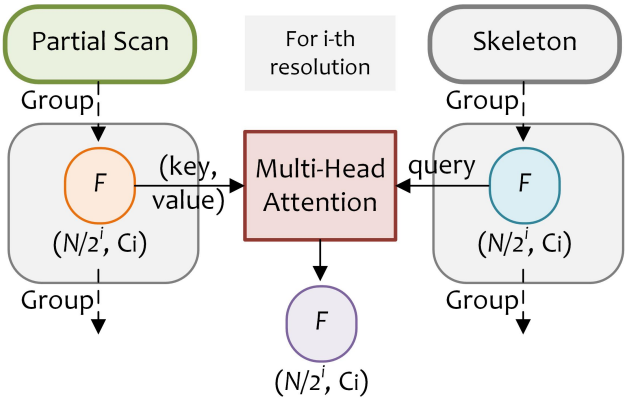
In Section 6.2, we have elaborated the details of how to extract shape skeleton from surface points. Different from learning skeletons where the global feature takes the primary role, surface completion from a shape skeleton focuses on keeping the observable region and completing missing details. In this section, we introduce our method in how to complete surface points from the shape skeleton.

6.3.1 Non-Local Attention

The insight of our method is that: shape skeletons indicate the shape structure, which could inform about the missing regions and guides the completion. On this point, by leveraging the observable information, SK-PCN



(a) Multi-Scale Feature Aggregation



(b) Non-Local Attention Module

Figure 6.3: Illustration of the multi-scale feature aggregation for our skeleton extraction (a) and the Non-Local Attention module to broadcast local details from the partial scan to skeletal points (b).

revisits the input scan to provide skeletal points with local details (see Figure 6.2). In our design, SK-PCN recovers the missing shape from different resolutions. Specifically, it dually extracts the multi-scale features from both the skeleton and input scan using the same down-sampling and feature aggregation module in Figure 6.3a (also see PointNet++ Qi et al. (2017b)). We do so to make skeletal points able to extract input details on different resolutions. Besides we observe that man-made objects are commonly with repetitive patterns (e.g., table legs are usually with the same structure). From this point, we hope our network can infer the unseen surfaces from the observable parts, if the unseen structure shares the similar shape information

with some substructure from the input. To this end, we design a Non-Local Attention module (see Figure 6.3b) to selectively and globally propagate local features from the observable input to the skeletal points. Specifically, on the i-th resolution, we denote the point features from the input and the skeleton by $\mathbf{P}_i, \mathbf{Q}_i \in \mathbb{R}^{\left(N/2^i, C_i\right)}$. N is the input point number, and C denotes its feature length. Here we adopt the attention strategy (Vaswani et al. 2017) to search the correlated local feature for each skeletal point in $\mathbf{Q}_i$ with:

$$\mathbf{Q}_i^* = \texttt{softmax}\left(\frac{\texttt{dot}(\mathbf{P}_i W_p, \mathbf{Q}_i W_q)}{\sqrt{d_i}}\right)\mathbf{P}_i, \tag{6.1}$$

where $W_p, W_q \in \mathbb{R}^{(C_i, d_i)}$ are the weights to be learned. $\texttt{dot}\,(*,*)$ measures the feature similarity between the skeleton and input points. Thus for skeletal points in $\mathbf{Q}_i$, it selects and combines those useful point features from the partial scan $\mathbf{P}_i$ as the updated skeleton feature $\mathbf{Q}_i^*$. In practice, we adopt the multi-head attention strategy (Vaswani et al. 2017) to consider different attention responses. In our ablative study, we demonstrate that this module brings significant benefits in searching local features. So far, the **Non-Local attention** module is able to extract the surface features from input scan for each skeletal point under different resolutions.

6.3.2 Learning Surface from Skeleton

After obtaining the multi-resolution local features for each skeletal point, we interpolate them back to the original skeleton and concatenate them together (same to Figure 6.3a). Thus each skeletal point is loaded with multi-level local features. After that, we upsample the N point features to four times denser with Yu et al. (2018) to recover more surface points. Specifically, the point features (N, C) are repeated with four copies, followed with grouped convolution (Zhang et al. 2018) to deform them individually and output a new feature matrix $(N, 4C)$. By reshaping it to $(4N, C)$, the upsampled points can be obtained with fully-connected layers. Rather than directly regressing point coordinates, we predict the displacements from skeletal points to the surface. It is because these displacement values show high relevance with surface normals (Wu et al. 2015), which better supports point normal estimation for

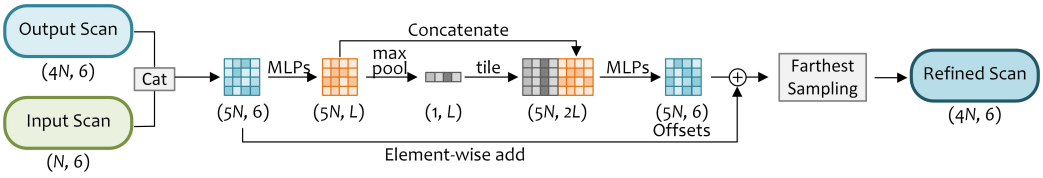


Figure 6.4: The pipeline of our surface adjustment module.

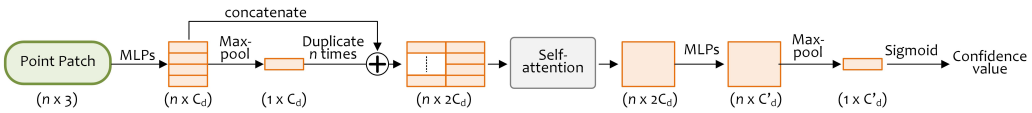


Figure 6.5: Point patch confidence prediction using Li et al. (2019b). Note that $C_d = 64$ and $C'_d = 256$.

our mesh reconstruction. Besides, learning residuals is beneficial to capture subtle details and also improves training efficiency.

6.3.3 Surface Adjustment with Local Guidance

As mentioned above, a completion method should preserve the geometry on the observable region. In this part, we design a generative adversarial module with a generator to merge input data to the output, and a discriminator to score the merging performance. The merging process is illustrated in Figure 6.4. We pre-calculate the normals of the input scan from coordinates $(N, 6)$ and concatenate them to our prediction $(4N, 6)$. The merged point cloud $(5N, 6)$ is followed with fully-connected layers and max-pooling to produce a shared feature. We tile and append this feature on each point to estimate offsets (inc. coordinates and normals) to the original output. However, merging input points results in denser distribution and defective boundaries on the overlapped area. For the first, we append the network with a farthest sampling layer to produce a uniformly-distributed point cloud with $4N$ points. For the second, to address the artifacts on boundaries, we adopt a patch discriminator to distinguish the merging result on the overlapped and boundary areas (see Figure 6.2). It randomly picks m seeds on the input scan. For each seed, it groups n points into a patch, which are located on the output scan within a radius r to these seeds ($m = 24, n = 128, r = 1/10 \times$object

size). For each patch, we utilize the basic architecture of Li et al. (2019b) with a sigmoid layer to score the confidence value (see Figure 6.5). It approximates 1 if the discriminator decides that a patch is similar to the ground-truth, and 0 if otherwise. Comparing with sampling patches over the whole surface, we observe that this method achieves better results in merging the input scan.

6.4 Loss Functions for End-to-end Training

In this section, we firstly define the point loss $\mathcal{L}_{\mathbf{P}}$ to compare the similarity between two point sets. Then a completion loss $\mathcal{L}_{\mathbf{C}}$ is provided to fulfill the surface completion. We denote the predicted/ground-truth skeletal points and surface points by $\mathbf{P}_k$ / $\mathbf{P}_k^*$ and $\mathbf{P}_s$ / $\mathbf{P}_s^*$ correspondingly.

Point Loss Since the outputs consist of unordered points, Chamfer Distance $\mathcal{L}_{\text{CD}}$ (Fan et al. 2017) is adopted to measure the permutation-invariant distance between two point sets. For normal estimation (only in surface completion), we use the cosine distance $\mathcal{L}_n$ (Park et al. 2019) to compare two normal vectors. Besides, we also adopt a repulsion loss $\mathcal{L}_r$ to obtain evenly distributed points (similar to Yu et al. (2018)). Thus for two point sets, we define the point loss $\mathcal{L}_{\mathbf{P}}$ between $\mathbf{P}$ and its ground-truth $\mathbf{P}^*$ by $\mathcal{L}_{\text{CD}} + \lambda_n \mathcal{L}_n + \lambda_r \mathcal{L}_r$, where

$$\mathcal{L}_{\text{CD}} = \sum_{x \in \mathbf{P}} \min_{y \in \mathbf{P}^*} \|x - y\|_2 / |\mathbf{P}| + \sum_{y \in \mathbf{P}^*} \min_{x \in \mathbf{P}} \|y - x\|_2 / |\mathbf{P}^*|, \tag{6.2}$$

$$\mathcal{L}_n = \sum_{x \in \mathbf{P}} \left(1 - \boldsymbol{n}_x \cdot \boldsymbol{n}_y\right) / |\mathbf{P}|, \quad y \in \mathbf{P}_*, \tag{6.3}$$

$$\mathcal{L}_r = \sum_{x \in \mathbf{P}} \sum_{x_p \in N(x)} \left(d - \|x_p - x\|_2\right) / |\mathbf{P}|. \tag{6.4}$$

$\mathcal{L}_{\text{CD}}$ in (6.2) presents the average nearest distance between $\mathbf{P}$ and $\mathbf{P}^*$. $|\mathbf{P}|$ denotes the point number in $\mathbf{P}$. In (6.3), $\boldsymbol{n}_y$ is the unit normal vector of the point in $\mathbf{P}^*$ that is the nearest neighbour to x. In (6.4), $\mathcal{L}_r$ requires the output points to be distant from each other and thus enforces a uniform distribution, where $N(x)$ are the neighbours of point x. d is the maximal distance threshold ($d = 3e^{-4}$).

Completion Loss Since SK-PCN predicts both skeleton and surface points, for each task, we adopt our point loss to measure their distance to the ground-truth. SK-PCN has three phases during shape completion: 1. skeleton estimation; 2. skeleton2surface; and 3. surface adjustment. Thus we define the completion loss in surface generation by

$$\mathcal{L}_C = \lambda_k \mathcal{L}_{\mathbf{P}_k} + \lambda_s \mathcal{L}_{\mathbf{P}_s} + \lambda_m \mathcal{L}_{\mathbf{P}_s^m}. \tag{6.5}$$

In $\mathcal{L}_{\mathbf{C}}$, the $(\mathcal{L}_{\mathbf{P}_k}, \mathcal{L}_{\mathbf{P}_s}, \mathcal{L}_{\mathbf{P}_s^m})$ respectively correspond to the point losses from the three phases, where $\mathbf{P}_s^m$ is the refined version of $\mathbf{P}_s$ (see section 6.3.3). $\mathcal{L}_{\mathbf{P}_k}$ is designed to minimize the distance of predicted and ground-truth skeletal points. $\mathcal{L}_{\mathbf{P}_k}$ explains that we hope the predicted shape surface points approximate the complete shape surface, and $\mathcal{L}_{\mathbf{P}_s^m}$ is to minimize the distance between the refined shape points and the ground-truth. $\{\lambda_*\}$ are the weights to balance their importance.

Adversarial Loss For the surface adjustment in section 6.3.3, we train our SK-PCN together with the patch discriminator using the least square loss (Mao et al. 2017, Li et al. 2019b) as the adversarial loss:

$$\mathcal{L}_{\mathcal{G}} = \left[D\left(\mathbf{P}_{patch}\right) - 1\right]^2 \tag{6.6}$$

$$\mathcal{L}_{\mathcal{D}} = D\left(\mathbf{P}_{patch}\right)^2 + \left[D\left(\mathbf{P}^*_{patch}\right) - 1\right]^2, \tag{6.7}$$

where D is the patch discriminator, $\mathbf{P}_{patch}$ and $\mathbf{P}^*_{patch}$ denote the estimated and ground-truth patch points on the overlapped area. A low $\mathcal{L}_{\mathcal{G}}$ means the discriminator scores our output with high confidence. We minimize the $\mathcal{L}_{\mathcal{D}}$ to make it able to distinguish our result with the ground-truth.

Overall, we train our SK-PCN end-to-end using the generator loss of $\mathcal{L}_C + \lambda_G \mathcal{L}_{\mathcal{G}}$ to implement shape completion, and the discriminator loss $\mathcal{L}_{\mathcal{D}}$ to preserve the fidelity on the observable region.

6.5 Experiment Setups

6.5.1 Datasets

Two datasets are used for our training. 1) ShapeNet-Skeleton (Tang et al. 2019) for skeleton extraction, and 2) ShapeNetCore (Chang et al. 2015) for surface completion. We adopt the train/validation/test split from (Yi et al. 2016) with five categories (i.e., chair, airplane, rifle, lamp, table) and 15,338 models in total. For each object model, we align and scale them within a unit cube and obtain 8 partial scans by back-projecting the rendered depth maps from different viewpoints (see Appendix D.2 for data and split preparation). The full scan and its corresponding shape skeleton are used as supervisions.

Metrics. In our evaluation, we adopt the Chamfer Distance-L_2 (CD) (Huang et al. 2020) and Earth-Mover's Distance (EMD) (Yuan et al. 2018) to evaluate completion results on surface points, and use CD together with normal consistency defined in Mescheder et al. (2019) to test the quality of the estimated point coordinates and normals for our mesh reconstruction (see evaluations and comparisons on more metrics in Appendix D.4).

6.5.2 Implementation

From skeleton estimation, skeleton2surface to surface adjustment, we first train each subnet of SK-PCN separately with fixing the former modules using our point loss. Then we train the whole network end-to-end with the generator and discriminator loss. We adopt the batch size at 16 and the learning rate at 1e-3, which decreases by the scale of 0.5 if there is no loss drop within five epochs. 200 epochs are used in total. The weights used in the loss functions are: $\lambda_k, \lambda_s = 1, \lambda_m = 0.1, \lambda_n = 0.001, \lambda_r = 0.1, \lambda_G = 0.01$. We present the full list of module and layer parameters, and inference efficiency in Appendix D.1.

6.5.3 Running Time

We train our network with two TITAN-Xp GPUs and test it on a single GPU. The average time cost on point completion is 1.628 seconds per instance. The

mesh reconstruction relies on an external Poisson Surface Reconstruction Library (Molero 2020), which takes 1.412 seconds per instance on average.

6.5.4 Benchmark Configuration

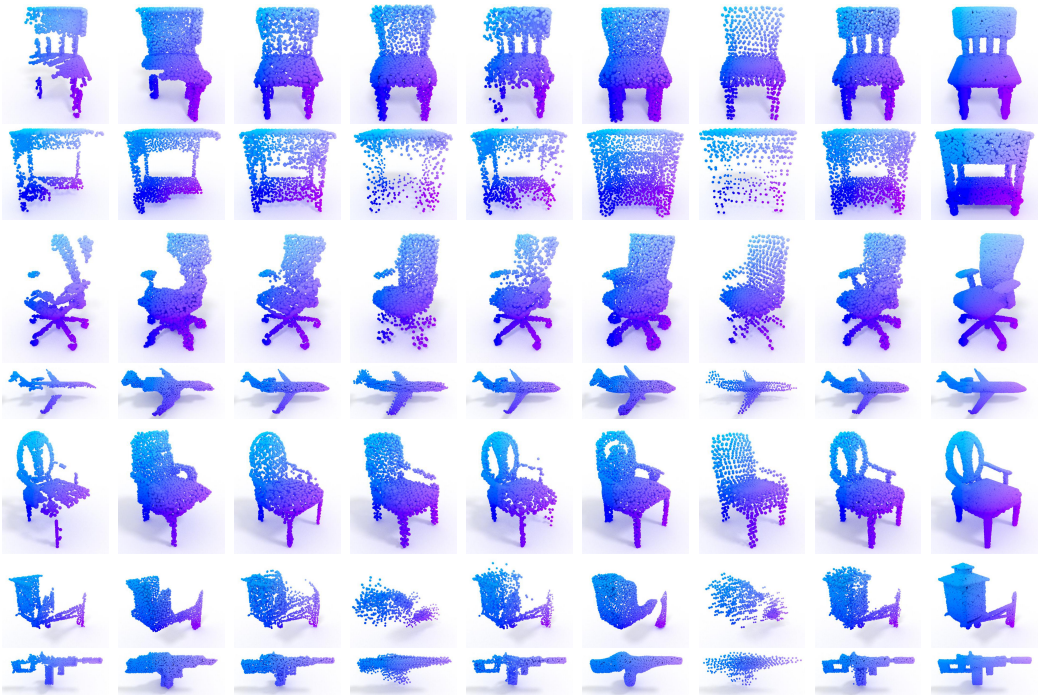
To investigate the performance of our method, we comprehensively compare our SK-PCN with state-of-the-art methods including MSN (Liu et al. 2019), PF-Net(Huang et al. 2020), PCN (Yuan et al. 2018), P2P-Net (Yin et al. 2018), DMC (Liao et al. 2018a), ONet (Mescheder et al. 2019) and IF-Net (Chibane et al. 2020) on point cloud/mesh completion. We train all the models on the same dataset for a fair comparison. Inline with PF-Net(Huang et al. 2020), we benchmark the input scale with 2048 points, and the ground-truth with 10k points in evaluation.

6.5.5 Comparisons with Point Completion Methods

We compare our method on point cloud completion with the baseline approaches including DMC (Liao et al. 2018a), MSN (Liu et al. 2019), PF-NetHuang et al. (2020), P2PNet (Yin et al. 2018), ONet Mescheder et al. (2019) and PCN (Yuan et al. 2018). For all methods, the number of output points is set to 2,048 for a fair comparison (i.e., the upsampling rate is set to 1). We present the qualitative and quantitative results on the test set in Figure 6.6 and Table 6.1 respectively. From the results, we observe that the traditional encoder-decoder methods show inadequacy in preserving small structures (row 1 & 2) and original topology (row 3 & 5) when completing missing shapes. Using skeletons as guidance, our method better preserves the topology of shapes, where thin structures (row 1) and holes (row 5) are well recovered. Moreover, by merging the input information, our results achieve higher fidelity on the observable region (row 6 & 7). The quantitative results in Table 6.1 further demonstrate that we obtain superior scores in both coordinates approximation (CD values) and distribution similarity (EMD values).

Table 6.1: Quantitative comparisons on point cloud completion.

Category	Chamfer Distance-L_2 (×1000) ↓ / Earth Mover's Distance (×100) ↓						
	DMC	MSN	PF-Net	P2P-Net	ONet	PCN	Ours
Airplane	0.392 / 0.411	0.111 / 0.194	0.280 / 0.685	0.127 / 1.323	0.355 / 0.300	0.287 / 3.960	**0.104** / **0.197**
Rifle	0.337 / 0.631	0.086 / 0.107	0.213 / 0.913	0.045 / 0.850	0.281 / 0.294	0.190 / 3.927	**0.033** / **0.082**
Chair	0.383 / 1.057	0.322 / 0.541	0.581 / 2.090	0.294 / 3.125	1.426 / 1.552	0.530 / 3.228	**0.255** / **0.486**
Lamp	0.521 / 1.633	0.630 / 1.473	1.283 / 2.273	0.302 / 3.271	1.480 / 1.937	2.278 / 4.542	**0.141** / **1.135**
Table	0.442 / 1.083	0.498 / 0.639	0.933 / 3.165	0.374 / 3.005	1.439 / 1.230	0.700 / 3.098	**0.343** / **0.594**
Average	0.415 / 0.963	0.329 / 0.591	0.658 / 1.825	0.228 / 2.315	0.996 / 1.063	0.797 / 3.751	**0.175** / **0.499**



(a) Input (b) DMC (c) MSN (d) PF-Net (e) P2P-Net (f) ONet (g) PCN (h) Ours (i) GT

Figure 6.6: Comparisons on point cloud completion. From left to right respectively are: a) input partial scan; b) DMC (Liao et al. 2018a); c) MSN (Liu et al. 2019); d) PF-Net (Huang et al. 2020); e) P2P-Net (Yin et al. 2018); f) ONet (Mescheder et al. 2019); g) PCN (Yuan et al. 2018); h) ours; i) ground-truth scan.

6.5.6 Comparisons with Mesh Reconstruction Methods

As SK-PCN estimates point normals along with coordinates, we further evaluate our reconstructed meshes using Poisson Surface Reconstruction by com-

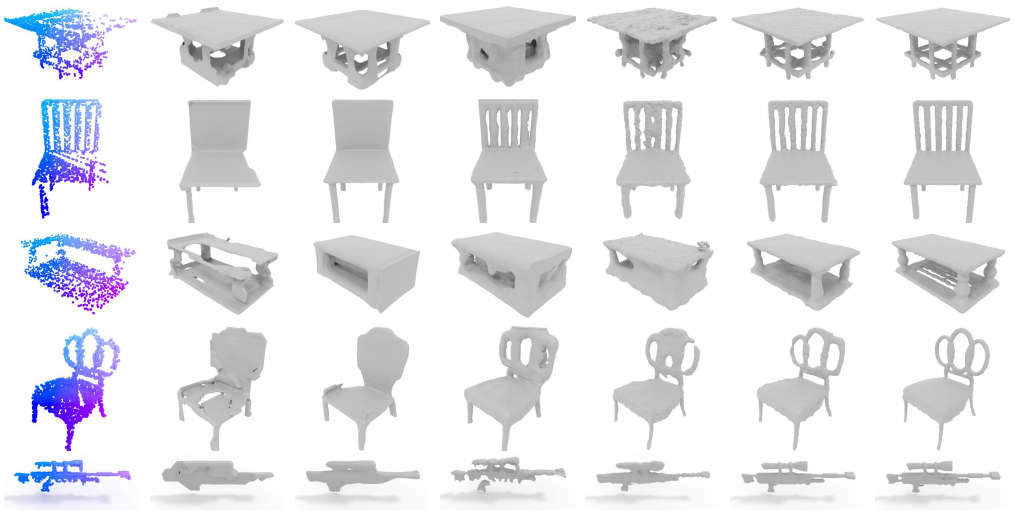
Table 6.2: Quantitative comparisons on mesh reconstruction.

	Chamfer Distance-L_2 ($\times$1000) $\downarrow$					Normal Consistency $\uparrow$				
Category	DMC	ONet	IF-Net	P2P-Net*	Ours	DMC	ONet	IF-Net	P2P-Net*	Ours
Airplane	0.361	0.337	0.447	0.102	**0.072**	0.810	0.835	0.813	0.828	**0.851**
Rifle	0.326	0.272	0.297	0.035	**0.022**	0.682	0.747	0.857	0.831	**0.925**
Chair	0.328	1.400	0.745	0.258	**0.159**	0.781	0.770	0.824	0.801	**0.863**
Lamp	0.472	1.451	0.875	0.392	**0.261**	0.793	0.818	0.830	0.791	**0.842**
Table	0.280	1.405	0.910	0.321	**0.246**	0.838	0.826	0.846	0.810	**0.881**
Average	0.353	0.973	0.655	0.222	**0.152**	0.781	0.799	0.834	0.812	**0.872**

paring with the existing mesh completion methods including IF-Net (Chibane et al. 2020), ONet (Mescheder et al. 2019) and DMC (Liao et al. 2018a). In this part, 8,192 points are uniformly sampled from each output to calculate the CD and normal consistency with the ground-truth (10k points with normals). Furthermore, we augment the P2P-Net (Yin et al. 2018) with normal prediction to investigate the completion performance without skeleton guidance (named by P2P-Net*). Specifically, we use the deformation module in P2P-Net to estimate the displacements from the input scan to the shape surface with point normals using extra channels (same to ours), and append the output layer with our upsampling module to keep a consistent number of points. We present the comparisons in Table 6.2 and Figure 6.7 (see more samples in Appendix D.3). The results demonstrate that shape completion by decoding a latent feature (as in DMC (Liao et al. 2018a), ONet (Mescheder et al. 2019) and IF-Net Chibane et al. (2020)) can produce an approximate and smooth shape but fail to represent small-scale structures. Besides, from Figure 6.7e, 6.7f and Table 6.2, we observe that using skeletal points as an intermediate representation significantly improves the normal estimation and produces local consistent normals (e.g., row 1, 2 & 4 in Figure 6.7f).

6.5.7 Ablation Analysis

To understand the effect of each module, we ablate our method with three configurations: $\boldsymbol{C_1}$**:** w.o. Non-Local Attention & w.o. Surface Adjustment (Baseline); $\boldsymbol{C_2}$**:** Baseline + Surface Adjustment; $\boldsymbol{C_3}$**:** Baseline + Non-Local Attention; **Full:** Baseline + Non-Local Attention + Surface Adjustment.



(a) Input (b) DMC (c) ONet (d) IF-Net (e) P2P* (f) Ours (g) GT

Figure 6.7: Comparisons on mesh completion. From left to right respectively are: a) input partial scan; b) DMC; c) ONet; d) IF-Net; e) P2P-Net*; f) ours; g) ground-truth mesh.

Table 6.3: Comparisons between ablated versions.

Metric	C_1	C_2	C_3	**Full**
CD ↓	0.340	0.293	0.205	**0.175**
CD_{comp} ↓	0.353	0.338	0.197	**0.184**
CD_{acc} ↓	0.326	0.248	0.212	**0.166**
EMD ↓	2.261	1.013	0.725	**0.499**
Normal Con. ↑	0.796	0.828	0.842	**0.853**

Note that the baseline method predicts the full scan by deforming via skeletal points without extra modules. It completes surface points only using predicted skeletons. We devise this baseline to instigate how much the other modules leverage the input to improve the results.

Here we output 2048 points for evaluation, and use the CD, normal consistency, EMD, completeness metric (CD_{comp}) and accuracy metric (CD_{acc}) to investigate the effects of each module. CD_{comp} is defined with the average distance from each ground-truth point to its nearest predicted point, and CD_{acc} is defined in the opposite direction. Their mean value is the Chamfer distance. We list the evaluations in Table 6.3 and visual results in Figure 6.8.

The results indicate that the Non-Local Attention module manifests the most significant improvement of the overall performance ($\boldsymbol{C_3}$ v.s. $\boldsymbol{C_1}$). Merging input scan brings more gains in improving the CD_{acc} values ($\boldsymbol{C_2}$ v.s. $\boldsymbol{C_1}$). It implies that merging a partial scan helps to extract more local information from the observable region, and combing the two modules achieves the best performance for shape completion.

We also investigate the significance of using 'skeleton' as the bridge for shape completion by replacing the skeletal points with 2,048 coarse surface points (see section 6.2). We implement this ablation on 'chair' category which presents sophisticated topology (see Figure 6.9). The CD↓ and Normal Consistency ↑ values are 2.96e-4 and 0.81 respectively compared to our 1.59e-4 and 0.86. We think the reason could be that, coarse point cloud is still a type of surface points. Differently, skeletal points keep compact topology of the shape without surface details. Using it as a bridge makes our method easier to recover complex structures.

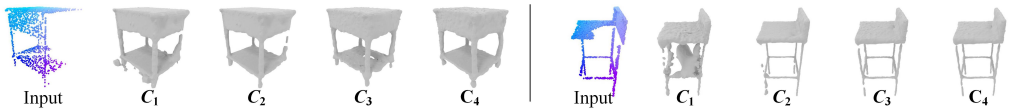


Figure 6.8: Mesh reconstruction with the configuration $\boldsymbol{C_1}$, $\boldsymbol{C_2}$, $\boldsymbol{C_3}$ and **Full**.

Figure 6.9: Skeleton v.s. Coarse points in shape completion. From left to right for each sample: input scan, results bridged with coarse points, and ours (bridged with skeletal points).

6.5.8 Discussions

In this section, we mainly discuss the performance on skeleton extraction with its impacts on the final results, and demonstrate the qualitative tests on real scans.

Skeleton Extraction Since skeleton extraction performs significant role in our pipeline, we illustrate some quantitative samples of skeleton extraction in Figure 6.10 and the average CD value on all shape categories is 2.98e-4. Besides, we also find that the skeleton quality as a structure abstraction has an intuitive impact on the final results. For example, for a skeleton failed to represent some local structure (e.g. with unclear skeletal points), our Skeleton2Surface module will struggle to grow the counterpart surface, and we conclude these scenarios as our limitations (see Figure 6.11).

Figure 6.10: Skeleton extraction results. From left to right for each sample: input scan; predicted shape skeleton (2,048 points); and the ground-truth.

Figure 6.11: Limitation cases. From left to right for each sample: input partial scan, predicted skeleton, points and mesh, ground-truth mesh.

Tests on Real Scans We also test our network (trained with ShapeNet) on real scans to investigate its robustness to real-world noises (see Figure 6.12). The input partial point clouds are back-projected from a single depth map and aligned to a canonical system (Choi et al. 2016). From the results, we can observe that our method can achieve plausible results under different levels of incompleteness and noise.

The Effects of Different Losses. In this section, we mainly discuss the effects of the normal loss, adversarial loss and repulsive loss to the results of point completion. We use the P2P-Net as the baseline method and output 2048 points evaluated with CD and EMD metrics on 'chair' category to make the comparison. We augment P2PNet with extra dimension (see Section 6.5.6) to estimate point normals (i.e. P2P-Net+normal loss), and extend

it with our adversarial module to (P2PNet+normal&adversarial loss). Repulsive loss is added to all the methods. The (CD×e4↓, EMD×e2↓) values of the original P2P-Net are (2.94, 3.13), and the others achieve (2.98, 3.19) and (2.76, 1.70) respectively, while ours are (2.55, 0.49). The results indicate that normal loss is for point normal estimation, but unlike the adversarial loss, it can not help the point estimation.

Figure 6.12: Tests on real scans (Choi et al. 2016). From left to right: image of the target object; input partial scan; predicted point cloud; predicted 3D mesh.

6.6 Summary

In this chapter, we present a novel learning modality for point cloud completion, namely SK-PCN. It end-to-end completes missing geometries from a partial point cloud by bridging the input to the complete surface via the shape skeleton. Our method decouples the shape completion into skeleton learning and surface recovery, where full surface points with normal vectors are predicted by growing from skeletal points. We introduce a Non-Local Attention module into point completion. It propagates multi-resolution shape details from the input scan to skeletal points, and automatically selects the contributive local features on the global shape surface for shape completion. Moreover, we provide a surface adjustment module to fully leverage input information and obtain high completion fidelity. Extensive experiments on both point cloud and mesh completion tasks demonstrate that our skeleton-bridged method presents high fidelity in preserving the shape topology and local details, and significantly outperforms the existing encoder-decoder-based methods.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The target of this thesis is to build a 3D vision system that is able to perceive, analyse and predict the 3D semantic and geometric contents in indoor scenes from single images. To achieve this target, we proposed 3D scene understanding solutions with different strategies from a unified pipeline to an end-to-end deep neural network.

In Chapter 3, we introduced a streamline of how to preprocess the input image and understand indoor semantics (i.e. object categories, segmentation masks, layout maps) and geometry (i.e. depth maps and 3D room layout) using deep learning techniques. With these intermediate outputs, a traversal mask-model matching algorithm is proposed to search the object CAD shapes for each object image in the scene. We built this system for indoor scene modelling based entirely on fully convolutional networks. Besides, we also provided a data-driven support inference approach to deduce the support relationships between neighbouring objects. It composes the indoor semantics into a hierarchical structure with support relations. In our experiments, the qualitative evaluation also demonstrated that involving support relations shows great effectiveness in modelling occluded objects from only a single image.

The method in Chapter 3 adopts a traversal strategy for shape retrieval, which is not time-efficient, and the pure prior-based support inference is also

error-prone for object pairs that are uncommon in a dataset. In Chapter 4, we focused on a unified vision system for holistic 3D scene modelling. it is fully backboned by convolutional neural networks (CNN). It involves multi-level convolutional networks to parse indoor semantics/geometry into non-relational and relational knowledge. Non-relational knowledge (i.e., room layout, object segmentations and geometry) extracted from shallow-end networks is fed forward into deeper levels to parse relational semantics (i.e., support relationship). A Relation Network is proposed to infer the support relationship between objects. All the structured semantics and geometry above are assembled to guide a global optimisation for 3D indoor scene synthesis. This synthesis incorporates the outputs from former networks and iteratively optimise 3D scenes to make them contextually consistent with the scene context. From both the qualitative and quantitative comparisons, it performs effectively in inferring the shape of severe occluded objects and presents better modelling performance than the prior art Huang et al. (2018b) on 3D object detection and room layout estimation.

Although we have proposed a unified 3D scene understanding system in Chapter 3 and 4, semantic scene modelling has an inherent problem that it usually requires a shape CAD dataset for model retrieval instead of end-to-end inference. In Chapter 5, we focused on end-to-end semantic scene reconstruction. That is, given a single RGB image, we directly predicted semantic object instances with meshes as the output without relying on an external shape dataset. Different with the prior works, we provided a solution to automatically reconstruct room layout, object bounding boxes, and meshes from a single image. To our best knowledge, it is the first work of end-to-end learning for comprehensive 3D scene understanding with mesh reconstruction at the instance level. Instead of using model retrieval to recover object geometry, we proposed a novel density-aware topology modifier to predict object meshes from single object images. It generates object meshes from a mesh template and modifies mesh topologies to approximate the target shape. Extensive experiments on the SUN RGB-D (Song et al. 2015) and Pix3D (Sun et al. 2018) datasets demonstrate that our method consis-

tently outperforms existing methods on indoor layout estimation, 3D object detection, camera pose estimation and mesh reconstruction.

Single-view reconstruction presents the inherent singularity in depth direction. In Chapter 6, we extended our work to explore the possibility of object shape reconstruction from depth scans. Previous works often predict the missing shape by decoding a latent feature encoded from the input points. However, real-world objects are usually with diverse topologies and surface details, where using a latent feature may fail to represent a clean and complete surface. On this top, we provided a new learning modality for point completion by mapping partial scans to complete surfaces bridged via meso-skeletons. This intermediate representation preserves better shape structure, and it enables to predict point normals and recover a full mesh beyond point clouds. Motivated by this inspiration, we correspondingly designed a completion network named by **SK-PCN**. It end-to-end aggregates the multi-resolution shape details from the partial scan to the shape skeleton, and automatically selects the contributive features in the global surface space for shape completion. Besides, since the input scan preserves the fidelity on observable object surfaces, we fully leveraged the original scan for local refinement, where a surface adjustment module is introduced to fine-tune our results for a high-fidelity completion. Extensive experiments demonstrate that our method outperforms previous methods on the metrics of Chamfer distance, normal consistency and Earth Mover's distance.

7.2 Future Work

The research in this thesis laid a foundation for the future work in single-view 3D scene understanding, modelling and reconstruction, and opened up several new directions:

Weak/Self/Un-supervised Instance Mesh Reconstruction in 3D scenes The end-to-end scene reconstruction method in Chapter 5 requires full supervisions for learning, which includes 2D & 3D object bounding boxes, camera poses, aligned object meshes. However, some ground-truth data are very hard

or impossible to be obtained, for example, the ground-truth object meshes of real-life objects that requires to scan an object under different viewpoints without occlusions. Therefore, how to design an end-to-end instance mesh reconstruction network not relying on paired mesh supervisions could be a significant direction in the future.

High-quality Instance Mesh Reconstruction with Multi-view Images Chapter 4 and 5 have demonstrated the possibility of reconstructing instance meshes from a single image. However, this method does not take into account multi-frame scenarios. Image information from other views could well address the problem in object occlusion and indoor clutter. How to involve multi-view images or a video sequence to jointly enhance the object detection and reconstruction performance on each single frame could be a potential topic in the following project.

3D Human-Scene Interaction and Context Understanding Apart from 3D objects and room layout on a static image, human behaviours are also a important part for robots to understand indoor scenes. The behaviour analysis of a human in scenes can be concluded into the human-scene interaction problem. Some works have been done at human motion prediction (Cao et al. 2020) and human body reconstruction in indoor scenes (Zhang et al. 2020c). However, there are few works on how to holistically analyse, reconstruct and predict the human-object interaction at the instance level, which could be an interesting topic in the future.

Efficient Multi-modality Scene Understanding and Reconstruction As discussed in Chapter 6, using depth scan cans well produce the object topology and surface geometry. However, consumer-level scanners usually presents undesirable depth maps with missing or noisy depth values. While colour images are generally with high resolution, rich appearance features like texture, and without much missing information in capturing. Therefore, how to end-to-end blend image and depth features during scanning for indoor

scene understanding and reconstruction could be an important direction in the future.

Bibliography

Abdulla, W., 2017. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. `https://github.com/matterport/Mask_RCNN`.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C. and Parikh, D., 2015. Vqa: Visual question answering. *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A. X. and Nießner, M., 2019a. Scan2cad: Learning cad model alignment in rgb-d scans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2614–2623.

Avetisyan, A., Dai, A. and Nießner, M., 2019b. End-to-end cad model retrieval and 9dof alignment in 3d scans. *Proceedings of the IEEE International Conference on Computer Vision*, 2551–2560.

Avetisyan, A., Khanova, T., Choy, C., Dash, D., Dai, A. and Nießner, M., 2020. Scenecad: Predicting object alignments and layouts in rgb-d scans. *arXiv preprint arXiv:2003.12622*.

Baek, S., In Kim, K. and Kim, T.-K., 2018. Augmented skeleton space transfer for depth-based hand pose estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8330–8339.

Brock, A., Lim, T., Ritchie, J. M. and Weston, N., 2016. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*.

Bu, S., Han, P., Liu, Z. and Han, J., 2016. Scene parsing using inference embedded deep networks. *Pattern Recognition*, 59, 188–198.

Cao, J., Tagliasacchi, A., Olson, M., Zhang, H. and Su, Z., 2010. Point cloud skeletons via laplacian based contraction. *2010 Shape Modeling International Conference*, IEEE, 187–197.

Cao, Y., Wu, Z. and Shen, C., 2016. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *arXiv preprint arXiv:1605.02305*.

Cao, Z., Gao, H., Mangalam, K., Cai, Q., Vo, M. and Malik, J., 2020. Long-term human motion prediction with scene context.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S., 2020. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H. et al., 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, K., Lai, Y.-K. and Hu, S.-M., 2015. 3d indoor scene modeling from rgb-d data: a survey. *Computational Visual Media*, 1 (4), 267–278.

Chen, K., Lai, Y.-K., Wu, Y.-X., Martin, R. and Hu, S.-M., 2014. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Transactions on Graphics*, 33 (6).

Chen, Z. and Zhang, H., 2019. Learning implicit fields for generative shape modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5939–5948.

Chibane, J., Alldieck, T. and Pons-Moll, G., 2020. Implicit functions in feature space for 3d shape reconstruction and completion. *arXiv preprint arXiv:2003.01456*.

Choi, S., Zhou, Q.-Y., Miller, S. and Koltun, V., 2016. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*.

Choi, W., Chao, Y.-W., Pantofaru, C. and Savarese, S., 2013. Understanding indoor scenes using 3d geometric phrases. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 33–40.

Choi, W., Chao, Y.-W., Pantofaru, C. and Savarese, S., 2015. Indoor scene understanding with geometric and semantic contexts. *International Journal of Computer Vision*, 112 (2), 204–220.

Choy, C. B., Xu, D., Gwak, J., Chen, K. and Savarese, S., 2016a. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *European conference on computer vision*, Springer, 628–644.

Choy, C. B., Xu, D., Gwak, J., Chen, K. and Savarese, S., 2016b. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *Proceedings of the European Conference on Computer Vision (ECCV)*, 628–644. URL `https://doi.org/10.1007/978-3-319-46484-8_38`.

Coughlan, J. M. and Yuille, A. L., 1999. Manhattan world: Compass direction from a single image by bayesian inference. *Proceedings of the seventh IEEE international conference on computer vision*, IEEE, volume 2, 941–947.

Coughlan, J. M. and Yuille, A. L., 2001. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. *Advances in Neural Information Processing Systems*, 845–851.

Criminisi, A., Reid, I. and Zisserman, A., 2000. Single view metrology. *International Journal of Computer Vision*, 40 (2), 123–148.

Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T. and Nießner, M., 2017a. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5828–5839.

Dai, A., Ruizhongtai Qi, C. and Nießner, M., 2017b. Shape completion using 3d-encoder-predictor cnns and shape synthesis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5868–5877.

Dai, J., Li, Y., He, K. and Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 379–387.

Dasgupta, S., Fang, K., Chen, K. and Savarese, S., 2016. Delay: Robust spatial layout estimation for cluttered indoor scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 616–624.

Deng, Z. and Latecki, L. J., 2017. Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images. *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2.

Deprelle, T., Groueix, T., Fisher, M., Kim, V. G., Russell, B. C. and Aubry, M., 2019. Learning elementary structures for 3d shape generation and matching. *arXiv preprint arXiv:1908.04725*.

Eigen, D. and Fergus, R., 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of the IEEE International Conference on Computer Vision*, 2650–2658.

Eigen, D., Puhrsch, C. and Fergus, R., 2014. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 2366–2374.

Fan, H., Su, H. and Guibas, L. J., 2017. A point set generation network for 3d object reconstruction from a single image. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 605–613.

Firman, M., Mac Aodha, O., Julier, S. and Brostow, G. J., 2016. Structured prediction of unobserved voxels from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5431–5440.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V. and Garcia-Rodriguez, J., 2017. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*.

Garg, R., BG, V. K., Carneiro, G. and Reid, I., 2016. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *European Conference on Computer Vision*, Springer, 740–756.

Geng, Q., Zhou, Z. and Cao, X., 2018. Survey of recent progress in semantic image segmentation with cnns. *Science China Information Sciences*, 61 (5), 051101.

Girshick, R., 2015. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Computer Vision and Pattern Recognition*.

Gkioxari, G., Malik, J. and Johnson, J., 2019. Mesh r-cnn. *Proceedings of the IEEE International Conference on Computer Vision*, 9785–9795.

Godard, C., Mac Aodha, O. and Brostow, G. J., 2017. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, volume 2, 7.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. and Aubry, M., 2018a. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Groueix, T., Fisher, M., Kim, V. G., Russell, B. C. and Aubry, M., 2018b. A papier-mâché approach to learning 3d surface generation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 216–224.

Guo, R. and Hoiem, D., 2013. Support surface prediction in indoor scenes. *Proceedings of the IEEE International Conference on Computer Vision*, 2144–2151.

Gupta, A., Hebert, M., Kanade, T. and Blei, D., 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *Advances in neural information processing systems*, 23, 1288–1296.

Gupta, S., Girshick, R., Arbeláez, P. and Malik, J., 2014. Learning rich features from rgb-d images for object detection and segmentation. *European Conference on Computer Vision*, Springer, 345–360.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y. et al., 2020. A survey on visual transformer. *arXiv preprint arXiv:2012.12556*.

Han, X., Li, Z., Huang, H., Kalogerakis, E. and Yu, Y., 2017. High-resolution shape completion using deep neural networks for global structure and local geometry inference. *Proceedings of the IEEE International Conference on Computer Vision*, 85–93.

Hariharan, B., Arbeláez, P., Girshick, R. and Malik, J., 2014. Simultaneous detection and segmentation. *European Conference on Computer Vision*, Springer, 297–312.

He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. *Computer Vision (ICCV), 2017 IEEE International Conference on*, IEEE, 2980–2988.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hedau, V., Hoiem, D. and Forsyth, D., 2009. Recovering the spatial layout of cluttered rooms. *Computer vision, 2009 IEEE 12th international conference on*, IEEE, 1849–1856.

Hedau, V., Hoiem, D. and Forsyth, D., 2010. Thinking inside the box: Using appearance models and context based on room geometry. *European Conference on Computer Vision*, Springer, 224–237.

Hoiem, D., Efros, A. A. and Hebert, M., 2007. Recovering surface layout from an image. *International Journal of Computer Vision*, 75 (1), 151–172.

Hou, J., Dai, A. and Nießner, M., 2020. Revealnet: Seeing behind objects in rgb-d scans. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2098–2107.

Hu, H., Gu, J., Zhang, Z., Dai, J. and Wei, Y., 2018. Relation networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3588–3597.

Hua, B.-S., Pham, Q.-H., Nguyen, D. T., Tran, M.-K., Yu, L.-F. and Yeung, S.-K., 2016. Scenenn: A scene meshes dataset with annotations. *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 92–101.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. et al., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. *IEEE CVPR*, volume 4.

Huang, S., Chen, Y., Yuan, T., Qi, S., Zhu, Y. and Zhu, S.-C., 2019. Perspectivenet: 3d object detection from a single rgb image via perspective points. *Advances in Neural Information Processing Systems*, 8905–8917.

Huang, S., Qi, S., Xiao, Y., Zhu, Y., Wu, Y. N. and Zhu, S.-C., 2018a. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. *Advances in Neural Information Processing Systems*, 207–218.

Huang, S., Qi, S., Zhu, Y., Xiao, Y., Xu, Y. and Zhu, S.-C., 2018b. Holistic 3d scene parsing and reconstruction from a single rgb image. *European Conference on Computer Vision*, Springer, 194–211.

Huang, S., Qi, S., Zhu, Y., Xiao, Y., Xu, Y. and Zhu, S.-C., 2018c. Holistic 3d scene parsing and reconstruction from a single rgb image. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Huang, Z., Yu, Y., Xu, J., Ni, F. and Le, X., 2020. Pf-net: Point fractal network for 3d point cloud completion. *arXiv preprint arXiv:2003.00410*.

Hueting, M., Reddy, P., Kim, V., Yumer, E., Carr, N. and Mitra, N., 2017. Seethrough: finding chairs in heavily occluded indoor scene images. *arXiv preprint arXiv:1710.10473*.

Hueting, M., Reddy, P., Yumer, E., Kim, V. G., Carr, N. and Mitra, N. J., 2018. Seethrough: Finding objects in heavily occluded indoor scene images. *Proceedings of International Conference on 3DVision (3DV)*. Selected for oral presentation.

Iizuka, S., Simo-Serra, E. and Ishikawa, H., 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36 (4), 1–14.

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Ishimtsev, V., Bokhovkin, A., Artemov, A., Ignatyev, S., Niessner, M., Zorin, D. and Burnaev, E., 2020. Cad-deform: Deformable fitting of cad models to 3d scans. *arXiv preprint arXiv:2007.11965*.

Izadinia, H., Shan, Q. and Seitz, S. M., 2017. Im2cad. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5134–5143.

Jia, Z., Gallagher, A., Saxena, A. and Chen, T., 2013. 3d-based reasoning with blocks, support, and stability. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Jiang, H., Cai, J. and Zheng, J., 2019. Skeleton-aware 3d human shape reconstruction from point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 5431–5441.

Jones, D. R., Perttunen, C. D. and Stuckman, B. E., 1993. Lipschitzian optimization without the lipschitz constant. *Journal of optimization Theory and Applications*, 79 (1), 157–181.

Kato, H., Ushiku, Y. and Harada, T., 2018. Neural 3d mesh renderer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3907–3916.

Kazhdan, M. and Hoppe, H., 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32 (3), 1–13.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S. and Shah, M., 2021. Transformers in vision: A survey.

Kim, Y. M., Mitra, N. J., Yan, D.-M. and Guibas, L., 2012. Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31 (6), 1–11.

Konolige, K. and Mihelich, P., 2011. Technical description of kinect calibration. *Tech. Rep., Willow Garage*.

Košecká, J. and Zhang, W., 2002. Video compass. *European conference on computer vision*, Springer, 476–490.

Krähenbühl, P. and Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 109–117.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

Kulkarni, N., Misra, I., Tulsiani, S. and Gupta, A., 2019. 3d-relnet: Joint object and relational network for 3d prediction. *Proceedings of the IEEE International Conference on Computer Vision*, 2212–2221.

Kurenkov, A., Ji, J., Garg, A., Mehta, V., Gwak, J., Choy, C. and Savarese, S., 2018. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 858–866.

Kuznietsov, Y., Stückler, J. and Leibe, B., 2017. Semi-supervised deep learning for monocular depth map prediction. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 6647–6655.

Ladicky, L., Shi, J. and Pollefeys, M., 2014. Pulling things out of perspective. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 89–96.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F. and Navab, N., 2016. Deeper depth prediction with fully convolutional residual networks. *3D Vision (3DV), 2016 Fourth International Conference on*, IEEE, 239–248.

Lee, C.-Y., Badrinarayanan, V., Malisiewicz, T. and Rabinovich, A., 2017. Roomnet: End-to-end room layout estimation. *Proceedings of the IEEE International Conference on Computer Vision*, 4865–4874.

Lee, D. C., Hebert, M. and Kanade, T., 2009. Geometric reasoning for single image structure recovery. *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2136–2143.

Li, L., Khan, S. and Barnes, N., 2019a. Silhouette-assisted 3d object instance reconstruction from a cluttered scene. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.

Li, R., Li, X., Fu, C.-W., Cohen-Or, D. and Heng, P.-A., 2019b. Pu-gan: a point cloud upsampling adversarial network. *Proceedings of the IEEE International Conference on Computer Vision*, 7203–7212.

Li, Y., Qi, H., Dai, J., Ji, X. and Wei, Y., 2017a. Fully convolutional instance-aware semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2359–2367.

Li, Y., Qi, H., Dai, J., Ji, X. and Wei, Y., 2017b. Fully convolutional instance-aware semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2359–2367.

Liao, Y., Donne, S. and Geiger, A., 2018a. Deep marching cubes: Learning explicit surface representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2916–2925.

Liao, Y., Donne, S. and Geiger, A., 2018b. Deep marching cubes: Learning explicit surface representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2916–2925.

Lim, J. J., Khosla, A. and Torralba, A., 2014. Fpm: Fine pose parts-based model with 3d cad models. *European conference on computer vision*, Springer, 478–493.

Lin, G., Shen, C., Van Den Hengel, A. and Reid, I., 2018a. Exploring context with deep structured models for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40 (6), 1352–1366.

Lin, H. J., Huang, S.-W., Lai, S.-H. and Chiang, C.-K., 2018b. Indoor scene layout estimation from a single image. *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 842–847.

Lin, T.-Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B. and Belongie, S. J., 2017. Feature pyramid networks for object detection. *CVPR*, volume 1, 4.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014. Microsoft coco: Common objects in context. *European conference on computer vision*, Springer, 740–755.

Liu, F., Shen, C. and Lin, G., 2015. Deep convolutional neural fields for depth estimation from a single image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5162–5170.

Liu, M., Guo, Y. and Wang, J., 2017. Indoor scene modeling from a single image using normal inference and edge features. *The Visual Computer*, 33 (10), 1227–1240.

Liu, M., Sheng, L., Yang, S., Shao, J. and Hu, S.-M., 2019. Morphing and sampling network for dense point cloud completion. *arXiv preprint arXiv:1912.00280*.

Liu, M., Zhang, K., Zhu, J., Wang, J., Guo, J. and Guo, Y., 2018. Data-driven indoor scene modeling from a single color image with iterative object segmentation and model retrieval. *IEEE transactions on visualization and computer graphics*.

Liu, T., Chaudhuri, S., Kim, V. G., Huang, Q., Mitra, N. J. and Funkhouser, T., 2014. Creating consistent scene graphs using a probabilistic grammar. *ACM Transactions on Graphics (TOG)*, 33 (6), 211.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C., 2016. Ssd: Single shot multibox detector. *European conference on computer vision*, Springer, 21–37.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Lu, X., Yaoy, J., Li, H. and Liu, Y., 2017. 2-line exhaustive searching for real-time vanishing point estimation in manhattan world. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 345–353.

Mallya, A. and Lazebnik, S., 2015. Learning informative edge maps for indoor scene layout prediction. *Proceedings of the IEEE international conference on computer vision*, 936–944.

Mandikal, P., KL, N. and Venkatesh Babu, R., 2018. 3d-psrnet: Part segmented 3d point cloud reconstruction from a single image. *Proceedings of the European Conference on Computer Vision (ECCV)*, 0–0.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z. and Paul Smolley, S., 2017. Least squares generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2794–2802.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. and Geiger, A., 2019. Occupancy networks: Learning 3d reconstruction in function space. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4460–4470.

Michalkiewicz, M., Pontes, J. K., Jack, D., Baktashmotlagh, M. and Eriksson, A., 2019. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*.

Molero, M., 2020. pypoisson. `https://github.com/mmolero/pypoisson`.

Nan, L., Xie, K. and Sharf, A., 2012. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31 (6), 1–10.

Navaneet, K., Mandikal, P., Agarwal, M. and Babu, R. V., 2019. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8819–8826.

Nazeri, K., Ng, E., Joseph, T., Qureshi, F. and Ebrahimi, M., 2019. Edgeconnect: Structure guided image inpainting using edge prediction. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 0–0.

Nie, Y., Chang, J., Chaudhry, E., Guo, S., Smart, A. and Zhang, J. J., 2018. Semantic modeling of indoor scenes with support inference from a single photograph. *Computer Animation and Virtual Worlds*, 29 (3-4), e1825.

Nie, Y., Guo, S., Chang, J., Han, X., Huang, J., Hu, S.-M. and Zhang, J. J., 2020a. Shallow2deep: Indoor scene modeling by single image understanding. *Pattern Recognition*, 103, 107271.

Nie, Y., Han, X., Guo, S., Zheng, Y., Chang, J. and Zhang, J. J., 2020b. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 55–64.

Nie, Y., Lin, Y., Han, X., Guo, S., Chang, J., Cui, S., Zhang, J. et al., 2020c. Skeleton-bridged point completion: From global inference to local adjustment. *Advances in Neural Information Processing Systems*, 33.

Pan, J., Han, X., Chen, W., Tang, J. and Jia, K., 2019a. Deep mesh reconstruction from single rgb images via topology modification networks. *Proceedings of the IEEE International Conference on Computer Vision*, 9964–9973.

Pan, J., Han, X., Chen, W., Tang, J. and Jia, K., 2019b. Deep mesh reconstruction from single rgb images via topology modification networks. *Proceedings of the IEEE International Conference on Computer Vision*, 9964–9973.

Park, J. J., Florence, P., Straub, J., Newcombe, R. and Lovegrove, S., 2019. Deepsdf: Learning continuous signed distance functions for shape representation. *arXiv preprint arXiv:1901.05103*.

Paschalidou, D., Ulusoy, A. O. and Geiger, A., 2019. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10344–10353.

Poirson, P., Ammirato, P., Fu, C.-Y., Liu, W., Kosecka, J. and Berg, A. C., 2016. Fast single shot detection and pose estimation. *3D Vision (3DV), 2016 Fourth International Conference on*, IEEE, 676–684.

Powell, M. J., 2009. The bobyqa algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 26–46.

Qi, C. R., Liu, W., Wu, C., Su, H. and Guibas, L. J., 2018. Frustum pointnets for 3d object detection from rgb-d data. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 918–927.

Qi, C. R., Su, H., Mo, K. and Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H. and Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 5099–5108.

Ramalingam, S., Pillai, J. K., Jain, A. and Taguchi, Y., 2013. Manhattan junction catalogue for spatial reasoning of indoor scenes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3065–3072.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 91–99.

Ren, T., Lin, L., Guo, S., Lin, J., Liao, M., Deng, S., Xu, P. and Nie, Y., 2020. Salient object segmentation for image composition: A case study of group dinner photo. *Neurocomputing*.

Ren, Y., Li, S., Chen, C. and Kuo, C.-C. J., 2016. A coarse-to-fine indoor layout estimation (cfile) method. *Asian Conference on Computer Vision*, Springer, 36–51.

Riegler, G., Osman Ulusoy, A. and Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3577–3586.

Roberts, L. G., 1963. *Machine perception of three-dimensional solids*. Ph.D. thesis, Massachusetts Institute of Technology.

Rother, C., Kolmogorov, V. and Blake, A., 2004. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23 (3), 309–314.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P. and Lillicrap, T., 2017. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 4967–4976.

Saxena, A., Chung, S. H. and Ng, A. Y., 2006. Learning depth from single monocular images. *Advances in neural information processing systems*, 1161–1168.

Saxena, A., Sun, M. and Ng, A. Y., 2009. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31 (5), 824–840.

Schonberger, J. L. and Frahm, J.-M., 2016. Structure-from-motion revisited. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4104–4113.

Schwing, A. G., Fidler, S., Pollefeys, M. and Urtasun, R., 2013. Box in the box: Joint 3d layout and object reasoning from single images. *Proceedings of the IEEE International Conference on Computer Vision*, 353–360.

Schwing, A. G. and Urtasun, R., 2012. Efficient exact inference for 3d indoor scene understanding. *European Conference on Computer Vision*, Springer, 299–313.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Shao, T., Xu, W., Zhou, K., Wang, J., Li, D. and Guo, B., 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31 (6), 1–11.

Shin, D., Ren, Z., Sudderth, E. B. and Fowlkes, C. C., 2019. 3d scene reconstruction with multi-layer depth and epipolar transformers. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Silberman, N., Hoiem, D., Kohli, P. and Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. *European conference on computer vision*, Springer, 746–760.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sinha, A., Unmesh, A., Huang, Q. and Ramani, K., 2017. Surfnet: Generating 3d shape surfaces using deep residual networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6040–6049.

Song, S., Lichtenberg, S. P. and Xiao, J., 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 567–576.

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M. and Funkhouser, T., 2017. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*.

Stekovic, S., Hampali, S., Rad, M., Sarkar, S. D., Fraundorfer, F. and Lepetit, V., 2020. General 3d room layout from a single view by render-and-compare. *European Conference on Computer Vision*, Springer, 187–203.

Stutz, D. and Geiger, A., 2018. Learning 3d shape completion from laser scan data with weak supervision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1955–1964.

Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. *Proceedings of the IEEE international conference on computer vision*, 945–953.

Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B. and Freeman, W. T., 2018. Pix3d: Dataset and methods for single-image 3d shape modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2974–2983.

Szegedy, C., Reed, S., Erhan, D., Anguelov, D. and Ioffe, S., 2014. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tagliasacchi, A., Zhang, H. and Cohen-Or, D., 2009. Curve skeleton extraction from incomplete point cloud. *ACM SIGGRAPH 2009 papers*, 1–9.

Tang, J., Han, X., Pan, J., Jia, K. and Tong, X., 2019. A skeleton-bridged deep learning approach for generating meshes of complex topologies from single rgb images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4541–4550.

Tatarchenko, M., Dosovitskiy, A. and Brox, T., 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. *Proceedings of the IEEE International Conference on Computer Vision*, 2088–2096.

Tchapmi, L. P., Kosaraju, V., Rezatofighi, H., Reid, I. and Savarese, S., 2019. Topnet: Structural point cloud decoder. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 383–392.

Thoma, M., 2016. A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*.

Tian, Y., Luo, A., Sun, X., Ellis, K., Freeman, W. T., Tenenbaum, J. B. and Wu, J., 2019. Learning to infer and execute 3d shape programs. *arXiv preprint arXiv:1901.02875*.

Tulsiani, S., Gupta, S., Fouhey, D. F., Efros, A. A. and Malik, J., 2018. Factoring shape, pose, and layout from the 2d image of a 3d scene. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 302–310.

Tulsiani, S., Su, H., Guibas, L. J., Efros, A. A. and Malik, J., 2017. Learning shape abstractions by assembling volumetric primitives. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2635–2643.

Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A. and Brox, T., 2017. Demon: Depth and motion network for learning monocular stereo. *IEEE Conference on computer vision and pattern recognition (CVPR)*, volume 5, 6.

Urtasun, R., Pollefeys, M., Hazan, T. and Schwing, A., 2012. Efficient structured prediction for 3d indoor scene understanding. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2815–2822.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 5998–6008.

Von Gioi, R. G., Jakubowicz, J., Morel, J.-M. and Randall, G., 2012. Lsd: a line segment detector. *Image Processing On Line*, 2, 35–55.

Wallace, B. and Hariharan, B., 2019. Few-shot generalization for single-image 3d reconstruction via priors. *Proceedings of the IEEE International Conference on Computer Vision*, 3818–3827.

Wang, H., Gould, S. and Roller, D., 2013. Discriminative learning with latent variables for cluttered indoor scene understanding. *Communications of the ACM*, 56 (4), 92–99.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W. and Jiang, Y.-G., 2018a. Pixel2mesh: Generating 3d mesh models from single rgb images. *Proceedings of the European Conference on Computer Vision (ECCV)*, 52–67.

Wang, P.-S., Sun, C.-Y., Liu, Y. and Tong, X., 2018b. Adaptive o-cnn: a patch-based deep representation of 3d shapes. *SIGGRAPH Asia 2018 Technical Papers*, ACM, 217.

Wang, W., Huang, Q., You, S., Yang, C. and Neumann, U., 2017. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2298–2306.

Wang, X., Ang Jr, M. H. and Lee, G. H., 2020. Cascaded refinement network for point cloud completion. *arXiv preprint arXiv:2004.03327*.

Wang, Y.-F., 2011. A comparison study of five 3d modeling systems based on the sfm principles. Technical report, Technical Report, Visualsize Inc. TR 2011-01, Sept 8: 1-30.

Wei, H. and Wang, L., 2018. Understanding of indoor scenes based on projection of spatial rectangles. *Pattern Recognition*, 81, 497–514.

Wen, X., Li, T., Han, Z. and Liu, Y.-S., 2020. Point cloud completion by skip-attention network with hierarchical folding. *arXiv preprint arXiv:2005.03871*.

Williams, F., 2020. point-cloud-utils. `https://github.com/fwilliams/point-cloud-utils`.

Wong, Y.-S., Chu, H.-K. and Mitra, N. J., 2015. Smartannotator an interactive tool for annotating indoor rgbd images. *Computer Graphics Forum*, Wiley Online Library, volume 34, 447–457.

Wu, J., Xue, T., Lim, J. J., Tian, Y., Tenenbaum, J. B., Torralba, A. and Freeman, W. T., 2016. Single image 3d interpreter network. *European Conference on Computer Vision*, Springer, 365–382.

Wu, S., Huang, H., Gong, M., Zwicker, M. and Cohen-Or, D., 2015. Deep points consolidation. *ACM Transactions on Graphics (ToG)*, 34 (6), 1–13.

Xiao, J., Russell, B. and Torralba, A., 2012. Localizing 3d cuboids in single-view images. *Advances in neural information processing systems*, 746–754.

Xu, Q., Wang, W., Ceylan, D., Mech, R. and Neumann, U., 2019. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*.

Xue, F., Xu, S., He, C., Wang, M. and Hong, R., 2015. Towards efficient support relation extraction from rgbd images. *Information Sciences*, 320, 320–332.

Yang, M. Y., Liao, W., Ackermann, H. and Rosenhahn, B., 2017. On support relations and semantic scene graphs. *ISPRS journal of photogrammetry and remote sensing*, 131, 15–25.

Yi, L., Kim, V. G., Ceylan, D., Shen, I.-C., Yan, M., Su, H., Lu, C., Huang, Q., Sheffer, A. and Guibas, L., 2016. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35 (6), 1–12.

Yin, K., Huang, H., Cohen-Or, D. and Zhang, H., 2018. P2p-net: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics (TOG)*, 37 (4), 1–13.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D. and Heng, P.-A., 2018. Pu-net: Point cloud upsampling network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2790–2799.

Yu, Z., Jin, L. and Gao, S., 2020. P2net: Patch-match and plane-regularization for unsupervised indoor depth estimation. *arXiv preprint arXiv:2007.07696*.

Yuan, W., Khot, T., Held, D., Mertz, C. and Hebert, M., 2018. Pcn: Point completion network. *2018 International Conference on 3D Vision (3DV)*, IEEE, 728–737.

Zaitoun, N. M. and Aqel, M. J., 2015. Survey on image segmentation techniques. *Procedia Computer Science*, 65, 797–806.

Zhan, H., Garg, R., Weerasekera, C. S., Li, K., Agarwal, H. and Reid, I., 2018. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 340–349.

Zhang, J., Nie, Y., Lyu, Y., Li, H., Chang, J., Yang, X. and Zhang, J. J., 2020a. Symmetric dilated convolution for surgical gesture recognition. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 409–418.

Zhang, W., Zhang, W. and Gu, J., 2019. Edge-semantic learning strategy for layout estimation in indoor environment. *IEEE transactions on cybernetics*, 50 (6), 2730–2739.

Zhang, W., Zhang, W. and Zhang, Y., 2020b. Geolayout: Geometry driven room layout estimation based on depth maps of planes. *European Conference on Computer Vision*, Springer, 632–648.

Zhang, X., Zhou, X., Lin, M. and Sun, J., 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6848–6856.

Zhang, Y., Hassan, M., Neumann, H., Black, M. J. and Tang, S., 2020c. Generating 3d people in scenes without people. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6194–6204.

Zhang, Y., Liu, Z., Miao, Z., Wu, W., Liu, K. and Sun, Z., 2015. Single image-based data-driven indoor scene modeling. *Computers & Graphics*, 53, 210–223.

Zhang, Y., Song, S., Tan, P. and Xiao, J., 2014. Panocontext: A whole-room 3d context model for panoramic scene understanding. *European conference on computer vision*, Springer, 668–686.

Zheng, B., Zhao, Y., Yu, J., Ikeuchi, K. and Zhu, S.-C., 2015. Scene understanding by reasoning stability and safety. *International Journal of Computer Vision*, 112 (2), 221–238.

Zheng, C., Cham, T.-J. and Cai, J., 2019. Pluralistic image completion. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1438–1447.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A. and Torralba, A., 2018. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40 (6), 1452–1464.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X. and Dai, J., 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

Zhuo, W., Salzmann, M., He, X. and Liu, M., 2017. Indoor scene parsing with instance segmentation, semantic labeling and support relationship inference. *30Th Ieee Conference On Computer Vision And Pattern Recognition (Cvpr 2017)*, Ieee, CONF.

Appendix A

Supplementary Material for Chapter 3

A.1 Parameter setting

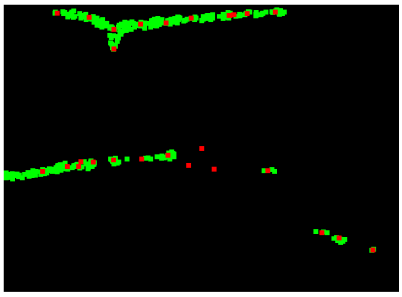
In image segmentation, we keep training configurations following the suite of Li et al. (2017b). In the room corner searching step (see Algorithm 1 in Appendix (B)), we set the maximal iteration number as 1000, the goal of inlier number as $0.7 *$ number of edge pixels, distance threshold as 10.

In object modeling, we set $\mathbf{d}_1 = [0.5, 0.5, 0.5]^{\mathbf{T}}$ (in meters, the same below) for normal objects. For those supported by a wall, $\mathbf{d}_1$ is set as $[0.2, \infty, \infty]^{\mathbf{T}}$ or $[\infty, 0.2, \infty]^{\mathbf{T}}$ depending on the orientation of the wall. $\mathbf{d}_2$ is set as $[1.0, 1.0, 0.5]^{\mathrm{T}}$ as the point cloud is noisier in horizontal plane than in the vertical direction (see Figure 3.8). For model scales, we set ${\rho_1}^{\mathrm{L}} = {\rho_2}^{\mathrm{L}} = {\rho_3}^{\mathrm{L}} = 0.8$, ${\rho_1}^{\mathrm{U}} = {\rho_2}^{\mathrm{U}} = 1.2$, and ${\rho_3}^{\mathrm{U}} = 1.0$. While for objects whose top part is occluded (see the rightmost chair in the fourth case in Figure 3.10a), the point cloud could underestimate the model height size. We hence change the lower bounds to ${\rho_1}^{\mathrm{L}} = {\rho_2}^{\mathrm{L}} = {\rho_3}^{\mathrm{L}} = 1.0$, and the upper bounds to ${\rho_1}^{\mathrm{U}} = {\rho_2}^{\mathrm{U}} = {\rho_3}^{\mathrm{U}} = 2.0$ or more. In the global searching step, the maximal iterations number is limited to 50, while in the local matching, generally we do not set the maximal iteration number to ensure convergence, the only stopping criteria is set as when the absolute tolerance reaches 10^{-3}.

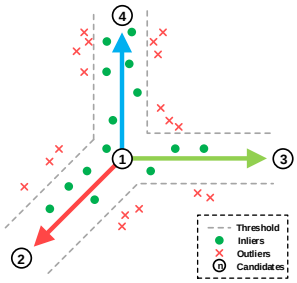
A.2 Room corner searching method

Based on the edge map (see Figure 3.4a), we reserve pixels with a probability score higher than the median value (see green dots in Figure A.1a). Then the Harris corner detector is adopted to find all possible corners on the map (see red dots on Figure A.1a). A relative smaller block size (3x3) is used in the Harris detector to produce extra candidates for searching four points to find the room corner. We utilize the RANSAC method to get the optimal four-point set among the corner candidates (see Figure 3.4b). With the point cloud, the rotation matrix and the translation vector of the room corner can be estimated as the camera's extrinsic parameters. The pseudo codes of our designed RANSAC algorithm is described in Algorithm 1.

In Algorithm 1, based on our observations, we claim four points forming a room corner when one of them is covered by the convex hull of the others (e.g. in Figure A.1b, point 1 is covered by the triangular comprised by point 2, 3 and 4). Here we down-sample the edge map by the sampling interval as 30 pixels to improve efficiency. The whole algorithm only costs several seconds.



(a) Candidates for corner searching



(b) Inliers and outliers in RANSAC searching

Figure A.1: Searching the optimal corner on an edge map of the room layout

Algorithm 1 RANSAC algorithm for searching the room corner

1: **Data**: Point candidates (red dots), Edge pixels (green dots) (see Figure 3.4b);
2: **Result**: The optimal four points set $\mathbf{S}^*$ which forms a corner;

3: **Initialization**: Set the maximal number of iterations as max_iter, current step as i, goal of inlier number as $inliers^{\mathrm{g}}$, current inlier number as $inliers$, current best inlier number as $inliers^{\mathrm{b}} = 0$, distance threshold as $dist\_thresh$, current best points set as $\mathbf{S}^{\mathrm{b}} = \{\}$;
4: **while** $i \leq max\_iter$ & $inliers^{\mathrm{b}} \leq inliers^{\mathrm{g}}$ **do**
5: Randomly pick out a four points set $\mathbf{S} = \{\mathbf{O}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$ from the point candidates without replacement;
6: **if S** forms a corner **then**
7: Calculate the $inliers$ as the number of pixels on the edge map whose the shortest distance from $\{\overrightarrow{\mathbf{OX}}, \overrightarrow{\mathbf{OY}}, \overrightarrow{\mathbf{OZ}}\}$ is smaller than $dist\_thresh$ (see Figure A.1b);
8: $i = i + 1$;
9: **if** $inliers > inliers^{\mathrm{b}}$ **then**
10: $inliers^{\mathrm{b}} = inliers$;
11: $\mathbf{S}^{\mathrm{b}} = \mathbf{S}$;
12: **end if**
13: **end if**
14: **end while**
15: $\mathbf{S}^{*} = \mathbf{S}^{\mathrm{b}}$;

Appendix B

Supplementary Material for Chapter 4

B.1 Technical illustrations

The network configurations and parameter decisions involved in our scene modelling are detailed in this part.

B.1.1 Indoor scene segmentation

As Mask R-CNN (He et al. 2017) is designed for general instance segmentation, to make it robust in learning from a small indoor dataset (795 images in our case), we augment the training data with a horizontal flip, and train the network by stages. Specifically, the whole training is divided into three phases, we firstly train the Region Proposal Network, Feature Pyramid Network and mask prediction layers with other parts frozen (60 epochs with learning rate at 1e-3), and fine-tune the ResNet by freezing the shallowest four layers (120 epochs with learning rate at 1e-3) followed by an all-layer training (160 epochs with learning rate at 1e-4). In the inference phase, the searched region proposals go through Non-Maximal Suppression to remove overlaps and keep objects with higher classification scores.

B.1.2 Model Retrieval

To build the CAD model dataset, we collect 26,695 indoor models covering 37 categories from ShapeNet (Chang et al. 2015) and SUNCG (Song et al. 2017), along with a 'cuboid' category for objects that are labeled as 'other' in NYU v2. We align and render each model from 32 viewpoints for appearance-based matching, with two elevation angles (15 and 30 degrees) and 16 uniform azimuth angles. The Multi-View Convolutional Network (Su et al. 2015) is customized with 32 parallel branches of ResNet-50 (He et al. 2016) as feature extractors (with the last layer removed). All those ResNets share the same weights. The deep features outputted from those branches are max-pooled and fully connected for recognition. The full network is pretrained on ShapeNet for shape recognition task. In our scene modeling, the major color texture from object masks is mapped to CAD models for rendering 3D scenes.

B.1.3 Relation Network

The whole architecture consists of three parts (see Figure 4.4): the Vision part, the Question part, and the Relation reasoning part. The Vision part is designed to encode the image and its segmentation by a set of abstract CNN features. The Question part is to rephrase each question into an encoded vector to ensure our system able to understand human language. The Relational reasoning part is responsible to analyze the image features and answer the corresponding questions. In the Vision part, we adopt five layers of convolutional kernels (3x3x64 for each layer with the stride and padding size at 2 and 1 respectively). Each convolution is followed by a ReLU and a Batch Normalization layer. The input end is a 300x300x4 matrix (the resized image appended with its mask), and it outputs a 10x10x64 feature map which can be seen as 10x10 of 64-dimensional feature vectors. In the Relational reasoning part, we get exhaustive pair combinations of those 10x10 feature vectors. Each pair of combination is concatenated with their 2D image coordinates correspondingly and the question vector. Thus the image features and the question vector are concatenated into 100x100 visual question vectors. All

those vectors separately go through four fully-connected layers, and it generates 100x100 512-dimensional vectors. We take element-wise summation of them and output a (104 dimensional) answer vector after walking-through three fully-connected layers. All the three fully-connected layers above consist of 512 hidden neurons, and each layer is followed by a ReLU unit except the final prediction layer. The initial learning rate is at 0.001 with the wight decay rate at 0.5 in every 10 steps. 60 epoches in total are used for training.

B.1.4 Global scene optimisation

In Section 6, we set the room height at three meters, and the height of every objects are calculated relatively. To ensure that each height estimate is in a reasonable interval, we parse the ScanNet dataset (Dai et al. 2017a) to conclude a prior height distribution for each object category (see Figure B.2 - B.5). Each sample in this normal distribution represents a height ratio of a real scanned object relative to the room. A height estimate is regarded as outliers if it is outside the 3σ interval, and should be replaced by the mean value.

The object sizes and positions are fine-tuned with our contextual refinement. In the optimisation problem (see Equation (4)), there are six continuous variables (in $\boldsymbol{S}_i$ and $\boldsymbol{p}_i$) we can control in the optimisation process with BOBYAQ method. The constraints (5) and (6) have guaranteed that all objects are attached on their supporting surface. Practically, we further constrain the boundary of $\boldsymbol{S}_i$ to make its size only adjustable in a given interval. we use $s_{i,3}$ in $\boldsymbol{S}_i$ to control the aspect ratio of a CAD model, and $s_{i,1}$ and $s_{i,2}$ to decide its horizontal ratio relative to its height. For common objects (labelled as known NYU v2 categories), we opt to set $s_{i,3} \in [0.9, 1.1]$, and $s_{i,1}, s_{i,2} \in [0.8, 1.2]$. For other objects (labelled as 'other furniture' or 'other structure'), 3D cuboid is used for model retrieval. In this case, we set the boundary of the horizontal ratio more flexible as $s_{i,1}, s_{i,2} \in [0.1, 10]$.

B.2 Priors for support inference and height estimation

We parse the ScanNet (Dai et al. 2017a) dataset to get the priors about support relationships and object heights. It contains 1,513 real scene scans with 37,831 indoor objects, and those objects are categorized by the same label set with our experiments. We estimate the bounding box of each object and get the height distribution as the Figure B.2 - B.5 shows. Each sample in these distributions is a ratio number of the object height to the room height. If a height estimate is beyond $[\mu - 3\sigma, \mu + 3\sigma]$ (μ is the mean value and σ is the standard deviation of the corresponding distribution), we replace the estimate with μ to initialize the object height.

Moreover, we extract the point cloud of objects to obtain support relationships within all of the scans and get one-to-one support relationship priors (with the method in Wong et al. (2015)) as the Figure B.6 shows. Each block in the two matrices denotes the number of cases that an object (in row) is supported by another object (in column) from below (Figure B.6(a)) or behind (Figure B.6(b)). Floating objects are removed, and each object must be supported by another object. When multiple support relationships exist, only the primary one is kept (see Wong et al. (2015)).

B.3 2D object segmentation comparisons with existing works

2D segmentation is designed to provide the object locations in the image. Detection loss in 2D images directly results in their 3D counterparts missing in the final CAD scenes. Besides that, whether an object is segmented with a fine-grained mask would also affect the geometry estimation. With this concern, we measure the Pixel Accuracy (PA), Mean Accuracy (MA) and Intersection over Union (IoU) (Garcia-Garcia et al. 2017) between the predicted and ground-truth masks to assess our performance on 40 categories in NYU v2 dataset. In testing, we select object masks with detection score greater than 0.5 from Mask R-CNN and layout masks from FCN to fully

segment images. Table B.1 illustrates the comparison with state-of-the-art methods. The results demonstrate that we achieve higher performance in terms of PA and IoU scores. It is worth noting that we are mostly concerned about the IoU score which is the optimisation target of our contextual refinement.

Table B.1: Semantic segmentation on NYU v2 (40 classes). IoU* score is the metric we are concerned in the step of contextual refinement.

Method	Data type	PA	MA	IoU*
Gupta et al. (2014)	RGB-D	60.3	-	28.6
FCN-32s (Long et al. 2015)	RGB	60.0	42.2	29.2
FCN-HHA (Long et al. 2015)	RGB-D	65.4	46.1	34.0
Lin et al. (2018a)	RGB	70.0	**53.6**	40.6
Our work	RGB	**70.3**	49.0	**41.6**

The 2D IoU from Mask R-CNN (He et al. 2017) only reaches 41.6% though it have reached the state-of-the-art. The accuracy of 3D object placement (i.e. 3D IoU) generally should be much lower for the depth ambiguity. Its indeed a bottleneck for all kinds of single image based scene reconstruction methods (Huang et al. 2018b, Izadinia et al. 2017). However, different from 2D segmentation, the physical plausibility in 3D scene modelling (i.e. relative orientations, sizes, and support relations between objects) could affect the visual performance greater, comparing with the impact from object placement accuracy (i.e. 3D IoU).

In our work, there basically are two factors we most concern: plausibility and placement accuracy. On this basis, we found that obtaining trustworthy physical constraints shows better plausibility and takes more semantic meanings (e.g. support relations) than only chasing placement accuracy. We present an example in Figure B.1. In indoor scenes, there are 40 object categories (NYU-40 (Silberman et al. 2012)). Except big-size categories like beds, sofas, tables, etc., most objects are thin or small and occupy little spatial volume (see the pictures and windows in Figure B.1). In our experiment, we observed that the 3D IoUs between them and the ground-truth are close to zero, because of their 'skinny' size making the IoU metric vulnerable to

placement disparities. However, they are still reconstructed with plausible visual performance because their orientations, sizes and support relations are reasonable. That means, a small 3D offset from the ground-truth will largely lower the accuracy of 3D IoU, but would not affect the visual plausibility given reasonable physical constraints (support relations, orientations and object sizes).

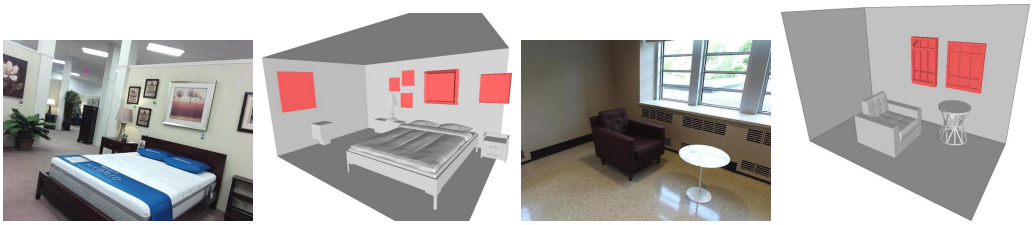


Figure B.1: Reconstruction of 'thin' structures.

B.4 Intermediate results in scene modelling

We randomly pick around 50 indoor images with different complexity from SUN-RGBD dataset (Song et al. 2015). The modelling results with intermediate outputs are illustrated in Figure B.7. The first column shows the input image. The layout edge map and label map are placed in the second and the third column respectively. The fourth column presents the jointly estimated room layout. We illustrate the scene segmentation and the support inference results in the fifth column. Note that the support relationship is represented with an arrow. For example, the red arrow from A to B denotes A supports B from below, and the blue arrow denotes A supports B from behind. We put the modelled scenes in the sixth column (raw scene meshes without texture-mapping and rendering)

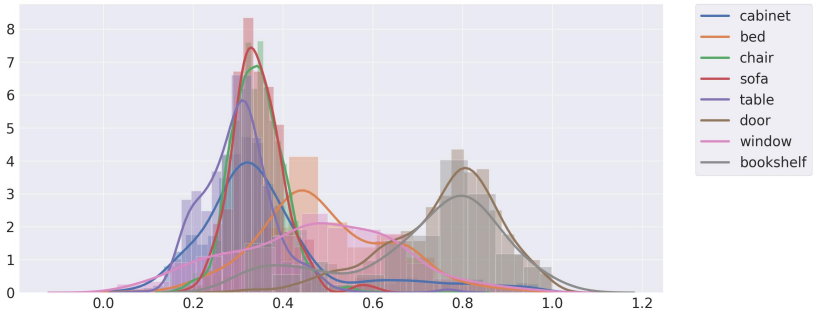


Figure B.2: Height distribution for each object category. (1-8 categories)

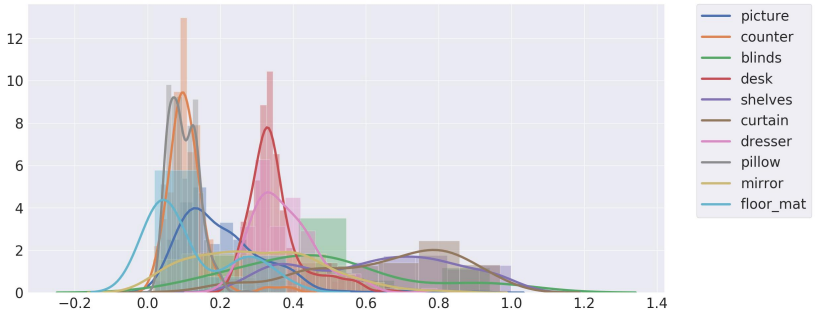


Figure B.3: Height distribution for each object category. (9-18 categories)

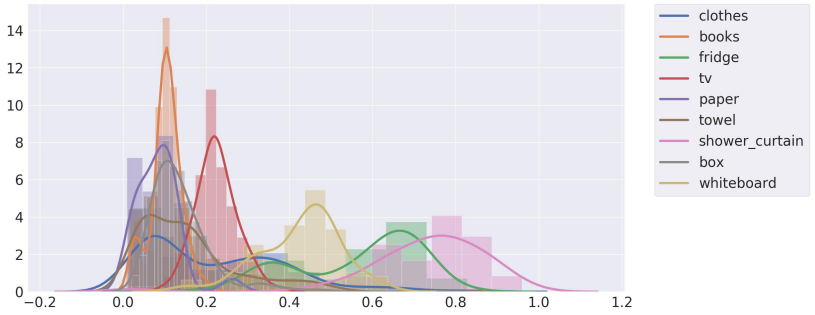


Figure B.4: Height distribution for each object category. (19-27 categories)

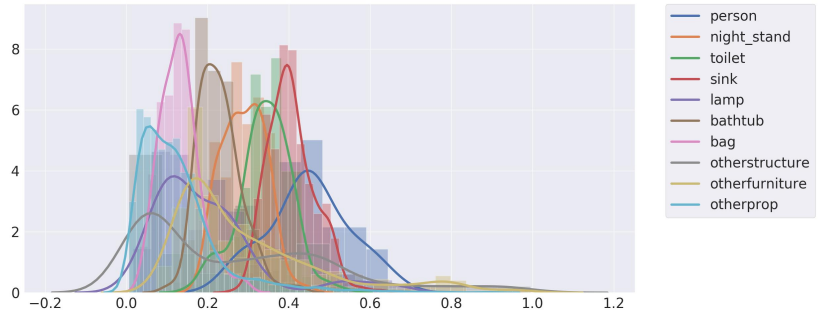
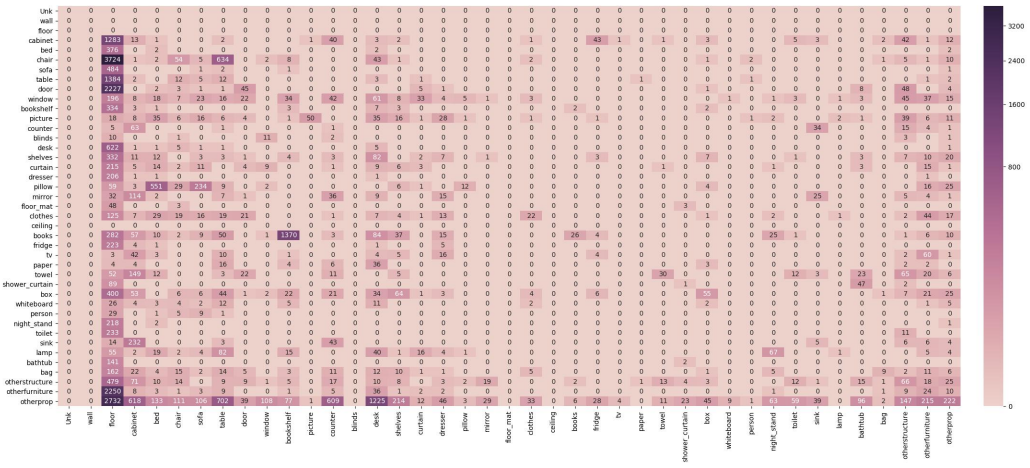
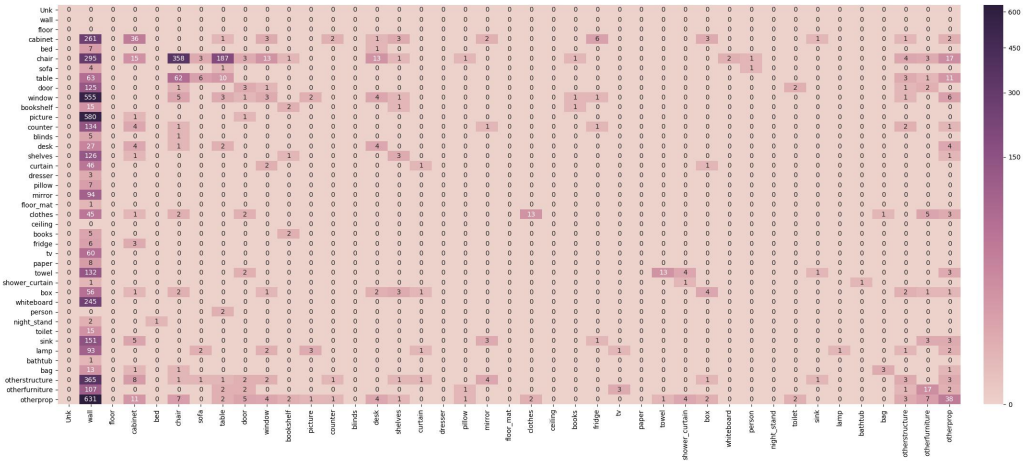


Figure B.5: Height distribution for each object category. (28-37 categories)

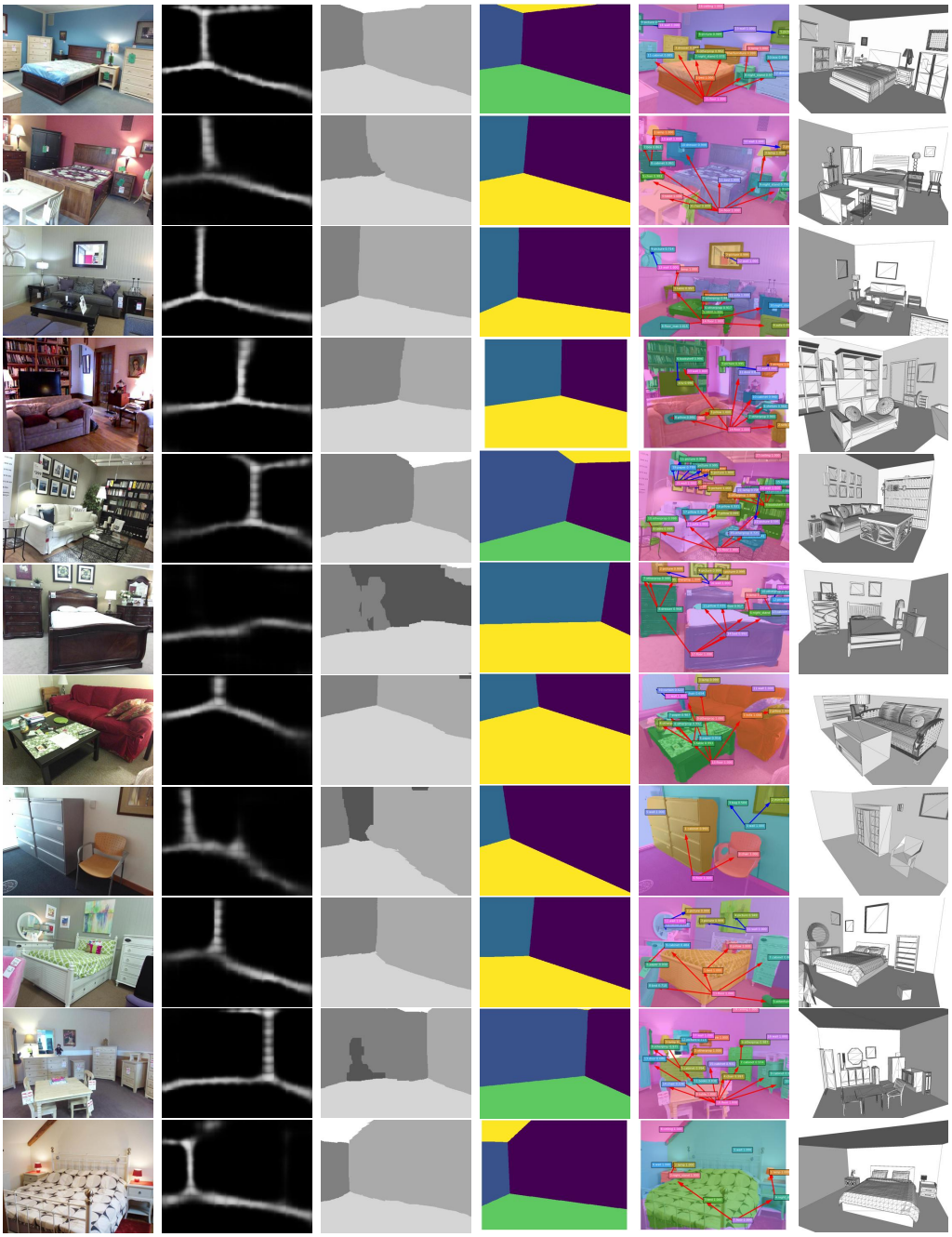


(a) Support from below



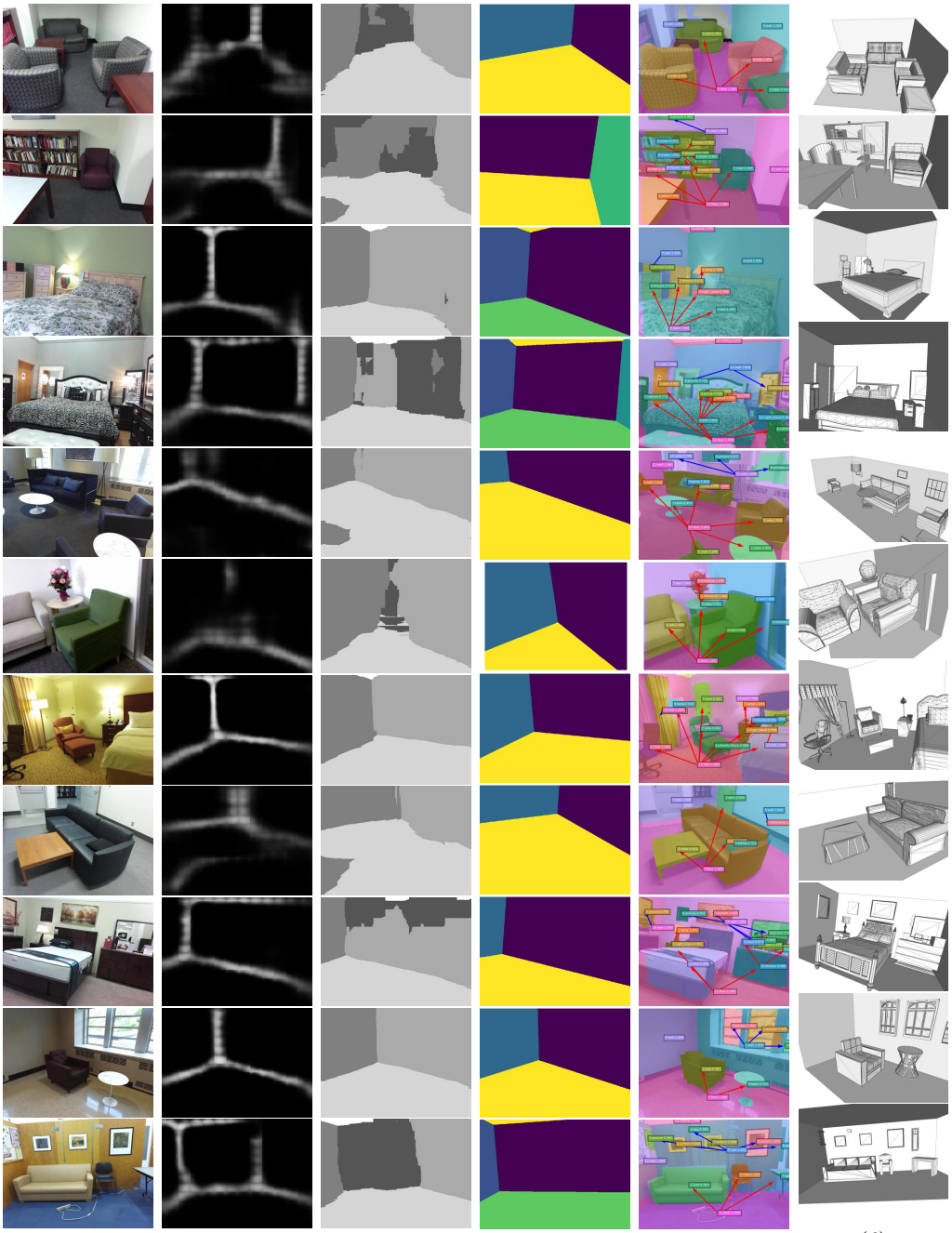
(b) Support from behind

Figure B.6: Support relationship priors



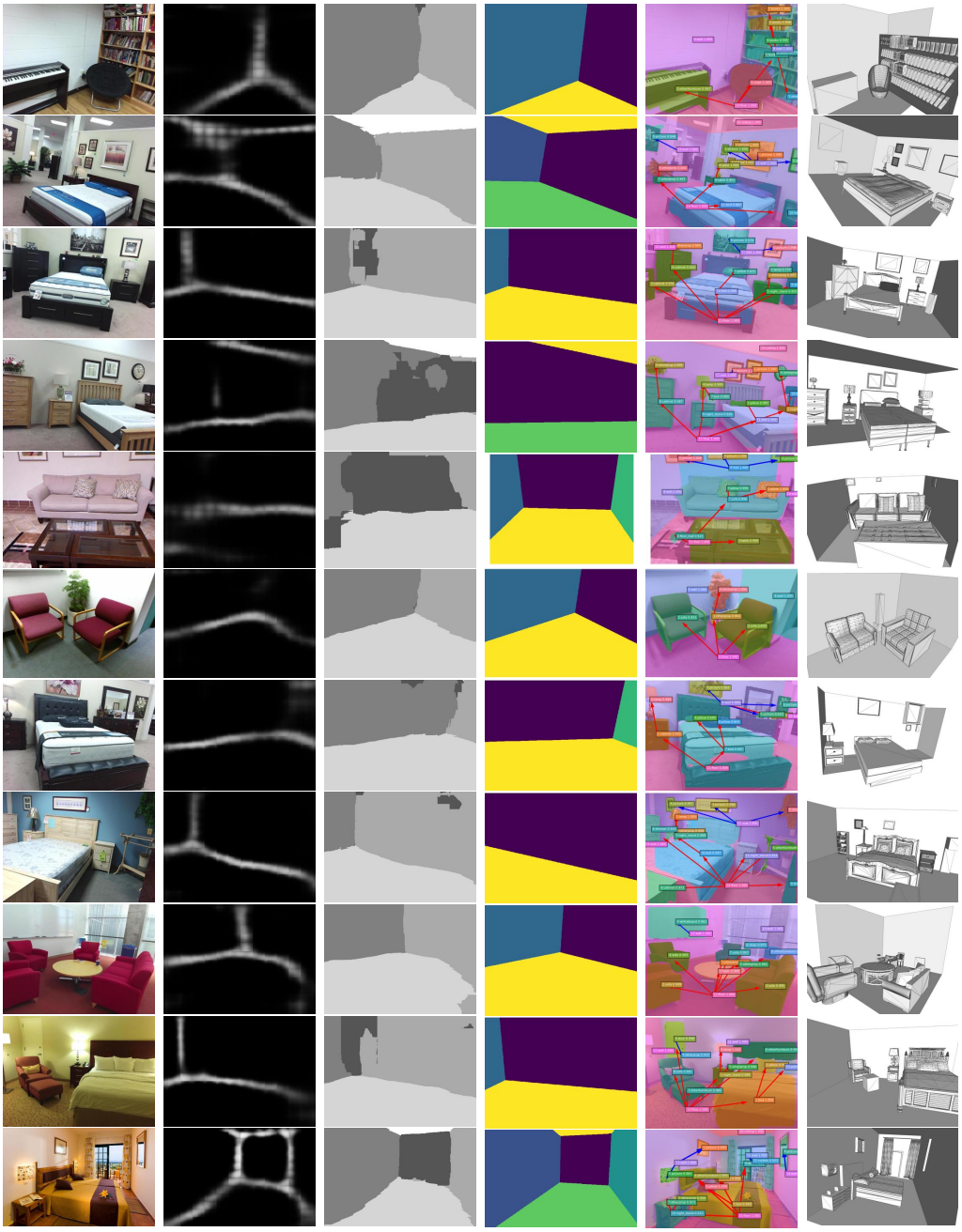
(a) (b) (c) (d) (e) (f)

Continue to the next page.



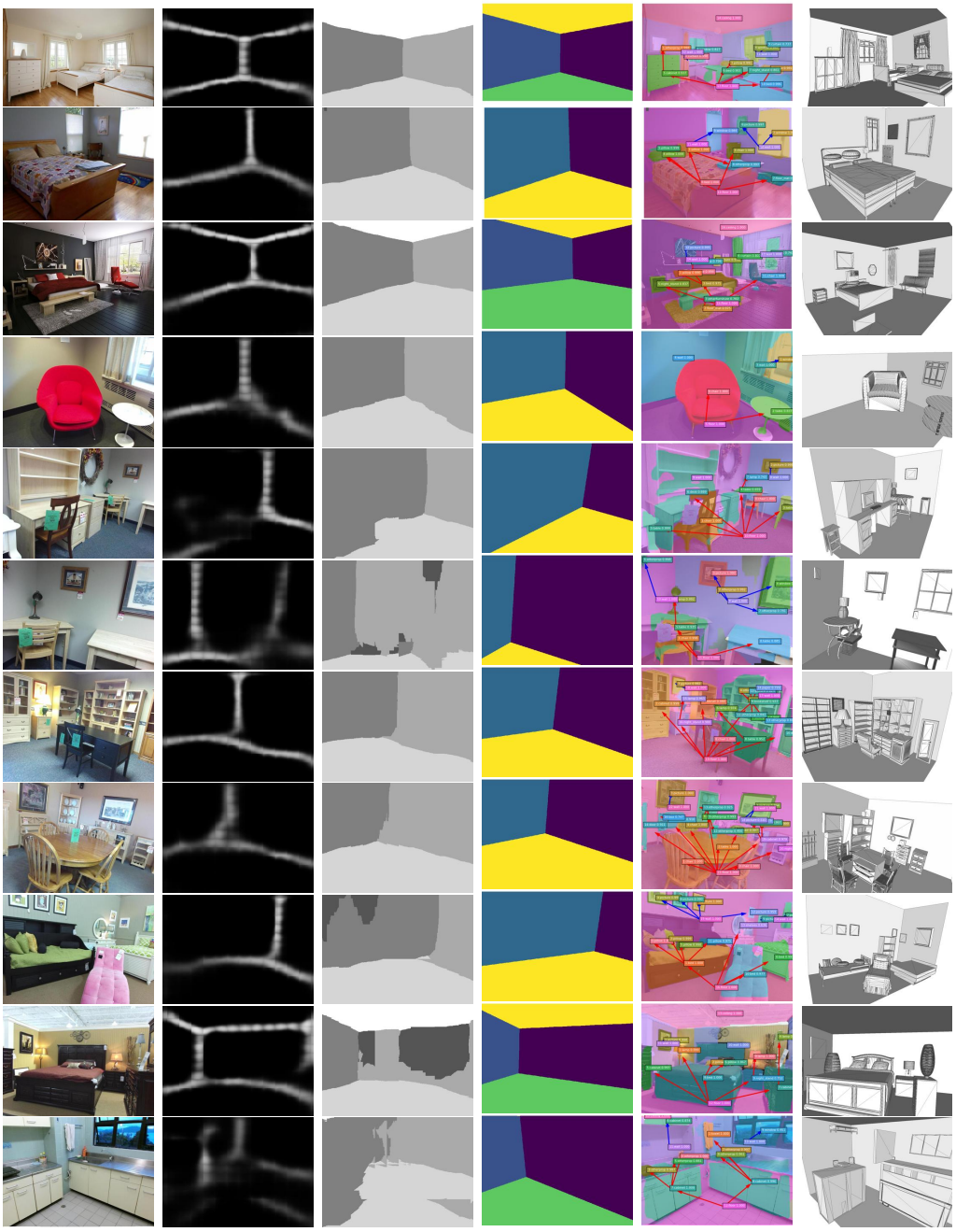
(a) (b) (c) (d) (e) (f)

Continue to the next page.



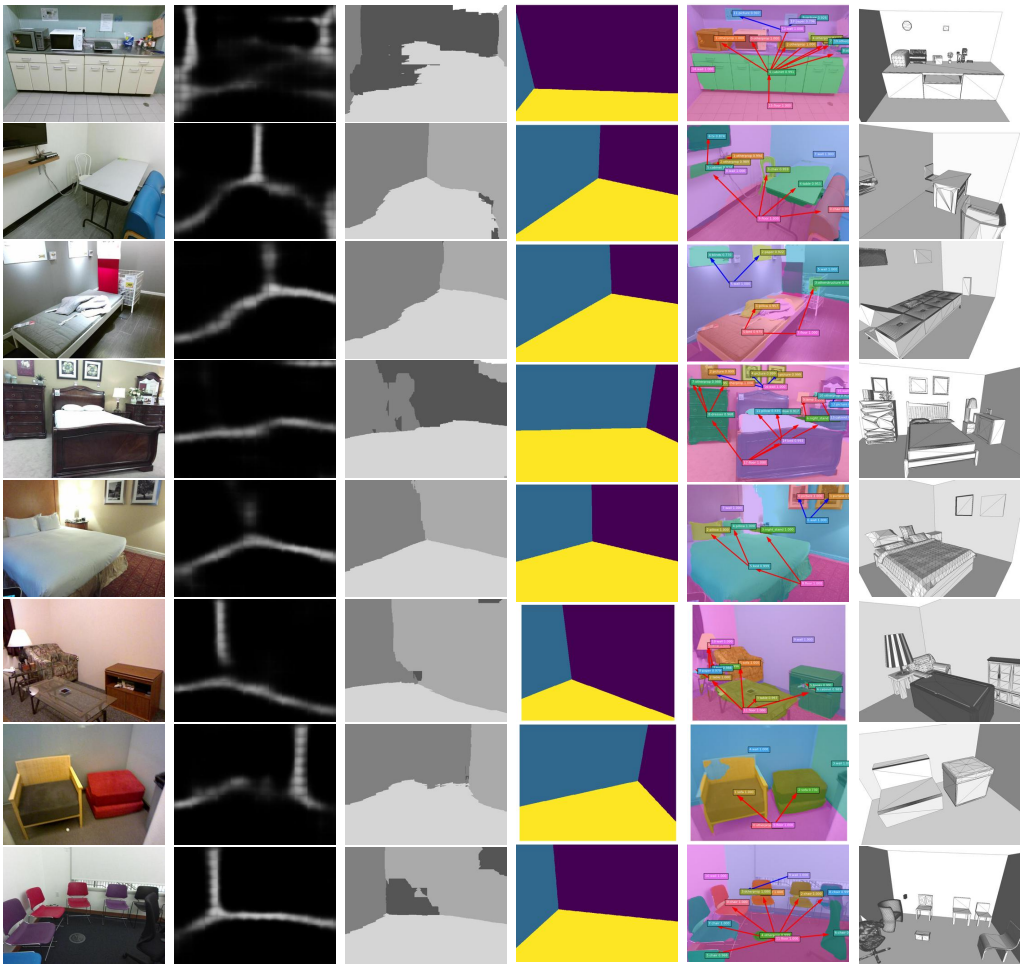
(a) (b) (c) (d) (e) (f)

Continue to the next page.



(a) (b) (c) (d) (e) (f)

Continue to the next page.



(a) (b) (c) (d) (e) (f)

Figure B.7: Intermediate results in scene modelling.

Appendix C

Supplementary Material for Chapter 5

C.1 Camera and World System Setting

We build the world and the camera systems in this chapter as Figure C.1 shows. The two systems share the same centre. The y-axis indicates the vertical direction perpendicular to the floor. We rotate the world system around its y-axis to align the x-axis toward the forward direction of the camera, such that the camera's yaw angle can be removed. Then the camera pose relative to the world system can be expressed by the angles of pitch β and roll γ:

$$\mathbf{R}(\beta, \gamma) = \begin{bmatrix} \cos(\beta) & -\cos(\gamma)\sin(\beta) & \sin(\beta)\sin(\gamma) \\ \sin(\beta) & \cos(\beta)\cos(\gamma) & -\cos(\beta)\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix}.$$

C.2 Network Architecture

Architecture. We present the architecture of our Object Detection Network (ODN), Layout Estimation Network (LEN) and Mesh Generation Network (MGN) in Table C.1-C.3.

Training strategy. Our training involves two phases. We first train the three networks individually. ODN and LEN are trained on SUN RGB-D

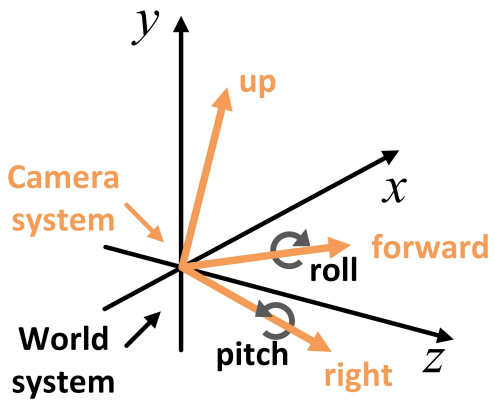


Figure C.1: Camera and world systems

(Song et al. 2015), while MGN is trained on Pix3D (Sun et al. 2018) with their specific loss ($\sum \lambda_x \mathcal{L}_x$, $\sum \lambda_y \mathcal{L}_y$ and $\sum \lambda_z \mathcal{L}_z$ respectively) (see Line 455, Page 5). All of them are with the batch size of 32 and learning rate at 1e-3 (scaled by 0.5 for every 20 epochs, 100 epochs in total). The MGN is trained with a progressive manner following Pan et al. (2019a). Afterwards, we fine-tune them with the joint losses $\lambda_{co}\mathcal{L}_{co}$ and $\lambda_g \mathcal{L}_g$ (see Equation 4) together on SUN RGB-D. Specifically, in the joint training, we randomly blend a few Pix3D samples into each batch of SUN RGB-D data to supervise the mesh generation network (i.e. to optimize the mesh loss $\sum \lambda_z \mathcal{L}_z$). We do so to regularize the mesh generation network because not like Pix3D, SUN RGB-D provides only a partial point-cloud scan of objects, which is not sufficient to supervise full mesh generation. For joint training, we input the full network with a hierarchical batch, where the scene image (from SUN RGB-D) is inputted to LEN, and the object images (from SUN RGB-D and Pix3D) are fed into ODN and MGN for object detection and mesh prediction. We set the hierarchical batch size at 1, and the learning rate at 1e-4 (scaled by 0.5 for every 5 epochs, 20 epochs in total). All the training tasks are implemented on 6x Nvidia 2080Ti GPUs. During testing, our network requires 1.2 seconds on average to predict a scene mesh on a single GPU.

Parameters. We set the threshold in our MGN at 0.2. Edges with the classification score below it are removed. In joint training (Section 3.3), we

let $\lambda_r = 10$, $\lambda_x = 1, \forall x \in \{\boldsymbol{\delta}, d, \boldsymbol{s}, \theta\}$, $\lambda_y = 1, \forall y \in \{\beta, \gamma, \boldsymbol{C}, \boldsymbol{s}^l, \theta^l\}$, $\lambda_c = 100$, $\lambda_e = 10$, $\lambda_b = 50$, $\lambda_{ce} = 0.01$, $\lambda_{co} = 10$, $\lambda_g = 100$.

Table C.1: Architecture of Object Detection Network. It takes all object detections in a scene as input and outputs their projection offset $\boldsymbol{\delta}$, distance d, orientation θ and size $\boldsymbol{s}$. N is the number of objects in a scene.

Index	Inputs	Operation	Output shape
(1)	Input	Object images in a scene	Nx3x256x256
(2)	Input	Geometry features (Hu et al. 2018, Vaswani et al. 2017)	N x N x 64
(3)	(1)	ResNet-34 (He et al. 2016)	Nx2048
(4)	(2), (3)	Relation Module (Hu et al. 2018)	Nx2048
(5)	(3), (4)	Element-wise sum	Nx2048
(6)	(5)	FC(128-d)+ReLU+Dropout+FC	$\boldsymbol{\delta}$
(7)	(5)	FC(128-d)+ReLU+Dropout+FC	d
(8)	(5)	FC(128-d)+ReLU+Dropout+FC	θ
(9)	(5)	FC(128-d)+ReLU+Dropout+FC	$\boldsymbol{s}$

Table C.2: Architecture of Layout Estimation Network. LEN takes the full scene image as input and produces the camera pitch β and roll γ angles, the 3D layout centre $\boldsymbol{C}$, size $\boldsymbol{s}$ and orientation θ in the world system.

Index	Inputs	Operation	Output shape
(1)	Input	Scene image	3x256x256
(2)	(1)	ResNet-34 (He et al. 2016)	2048
(3)	(2)	FC(1024-d)+ReLU+Dropout+FC	β
(4)	(2)	FC(1024-d)+ReLU+Dropout+FC	γ
(5)	(2)	FC+ReLU+Dropout	2048
(6)	(5)	FC(1024-d)+ReLU+Dropout+FC	$\boldsymbol{C}$
(7)	(5)	FC(1024-d)+ReLU+Dropout+FC	$\boldsymbol{s}^l$
(8)	(5)	FC(1024-d)+ReLU+Dropout+FC	θ^l

C.3 3D Detection on SUN RGB-D

We report the full results of 3D object detection on SUN RGB-D in Table C.5.

Table C.3: Architecture of Mesh Generation Network. Note that d_c denotes the number of object categories, and N_e represents the number of points sampled on edges. The edge classifier has the same architecture with AtlasNet decoder, where the last layer is replaced with a fully connected layer for classification.

Index	Inputs	Operation	Output shape
(1)	Input	Object image	3x256x256
(2)	Input	Object class code	d_c
(3)	Input	Template Sphere	3x2562
(4)	(1)	ResNet-18 (He et al. 2016)	1024
(5)	(2),(4)	Concatenate	1024+d_c
(6)	(5)	Repeat	(1024+d_c)x2562
(7)	(3),(6)	Concatenate	(1024+d_c+3)x2562
(8)	(7)	AtlasNet decoder (Groueix et al. 2018a)	3x2562
(9)	(3),(8)	Element-wise sum	3x2562
(10)	(9)	Sample points	3xN_e
(11)	(5)	Repeat	(1024+d_c)xN_e
(12)	(10),(11)	Concatenate	(1024+d_c+3)xN_e
(13)	(12)	Edge classifier	1xN_e
(14)	(13)	Threshold	1xN_e (Mesh topology)
(15)	(6),(9)	Concatenate	(1024+d_c+3)x2562
(16)	(15)	AtlasNet decoder (Groueix et al. 2018a)	3x2562
(17)	(9),(16)	Element-wise sum	3x2562 (Mesh points)

Table C.4: Object class mapping from NYU-37 to Pix3D

cabinet	bed	chair	sofa	table	door	window
8	1	3	5	6	8	9
bookshelf	picture	counter	blinds	desk	shelves	curtain
2	9	9	9	4	2	9
dresser	pillow	mirror	floor mat	clothes	books	fridge
8	9	9	9	9	9	8
tv	paper	towel	shower curtain	box	whiteboard	person
8	9	9	9	8	8	9
nightstand	toilet	sink	lamp	bathtub	bag	wall
8	9	9	9	9	8	-
floor	ceiling	-	-	-	-	-
-	-	-	-	-	-	-

Table C.5: Comparison of 3D object detection. We compare the average precision (AP) of detected objects on SUN RGB-D (higher is better). CooP** (Huang et al. 2018a) presents the model trained on the NYU-37 object labels for a fair comparison.

Method	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter
CooP**	10.47	57.71	15.21	36.67	31.16	0.14	0.00	3.81	0.00	27.67
Ours (w/o. joint)	11.39	59.03	15.98	43.95	35.28	0.36	0.16	**5.26**	0.24	**33.51**
Ours (joint)	**14.51**	**60.65**	**17.55**	**44.90**	**36.48**	**0.69**	**0.62**	4.93	**0.37**	32.08
Method	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat	clothes	books
CooP**	**2.27**	19.90	2.96	1.35	15.98	2.53	**0.47**	-	0.00	**3.19**
Ours (w/o. joint)	0.00	23.65	**4.96**	2.68	19.20	2.99	0.19	-	0.00	1.30
Ours (joint)	0.00	**27.93**	3.70	**3.04**	**21.19**	**4.46**	0.29	-	0.00	2.02
Method	fridge	tv	paper	towel	shower curtain	box	whiteboard	person	nightstand	toilet
CooP**	21.50	5.20	0.20	2.14	20.00	**2.59**	0.16	20.96	11.36	42.53
Ours (w/o. joint)	20.68	4.44	0.41	**2.20**	20.00	2.25	0.43	23.36	6.87	**48.37**
Ours (joint)	**24.42**	**5.60**	**0.97**	2.07	20.00	2.46	**0.61**	**31.29**	**17.01**	44.24
Method	sink	lamp	bathtub	bag	wall	floor	ceiling			
CooP**	15.95	3.28	24.71	1.53	-	-	-			
Ours (w/o. joint)	14.40	3.46	**27.85**	2.27	-	-	-			
Ours (joint)	**18.50**	**5.04**	21.15	**2.47**	-	-	-			

C.4 Object Class Mapping

Pix3D has nine object categories for mesh reconstruction, which contains: 1. bed, 2. bookcase, 3. chair, 4. desk, 5. sofa, 6. table, 7. tool, 8. wardrobe, 9. miscellaneous. In 3D object detection, we obtain object bounding boxes with NYU-37 labels in SUN RGB-D. As our MGN is pretrained on Pix3D, and the object class code is required as an input for mesh deformation, we manually map the NYU-37 labels to Pix3D labels based on topology similarity for scene reconstruction (see Table C.4).

C.5 More Comparisons of Object Mesh Reconstruction on Pix3D

More qualitative comparisons with Topology Modification Network (TMN) (Pan et al. 2019a) are shown in Figure C.2. The threshold τ in TMN is set at 0.1 to be consistent with their paper.

C.6 More Samples of Scene Reconstruction on SUN RGB-D

We list more reconstruction samples from the testing set of SUN RGB-D in Figure C.3.

Figure C.2: Qualitative comparisons between the proposed method and TMN (Pan et al. 2019a) on object mesh reconstruction. From left to right: input images, results from TMN, and our results.

Figure C.3: Reconstruction results of test samples on SUN RGB-D

Appendix D

Supplementary Material for Chapter 6

D.1 Network Architecture and Parameters

We provide the details of our network architecture and layer specifications in this section. In this chapter, we adopt the same notations as Qi et al. (2017b). The set abstraction layer is denoted by $SA\left(K, r, [l_1, l_2, ..., l_d]\right)$, and the feature propagation layer is represented by $FP\left([l_1, l_2, ..., l_d]\right)$. K is the number of patches that are grouped from the input points. r is the radius of the bounding ball for each patch (see Figure 6.2 in Chapter 6). $[l_1, l_2, ..., l_d]$ represent the fully-connected layers inside the set abstraction and the feature propagation, where l_i denotes the number of neurons in the i-th layer. Similarly, the fully-connected layers are represented by $MLP\left([l_1, l_2, ..., l_d]\right)$.

D.1.1 Learning Meso-Skeleton with Global Inference

We input our network with N points and normals calculated from point coordinates, i.e., $\left(x, y, z, n_x, n_y, n_z\right)$. The parameter setting of our skeleton estimation network (see Section 6.2 of this chapter) is illustrated in Figure D.1, where N denotes the number of input points. R represents the scale of the 3D shape ($2R$ equals to the side length of the bounding box of the shape). $N \times (3 + d)$ means two outputs: the 3-dimensional point coordinates with corresponding d-dimensional point features. $N = 2048$ and $R = 0.5$.

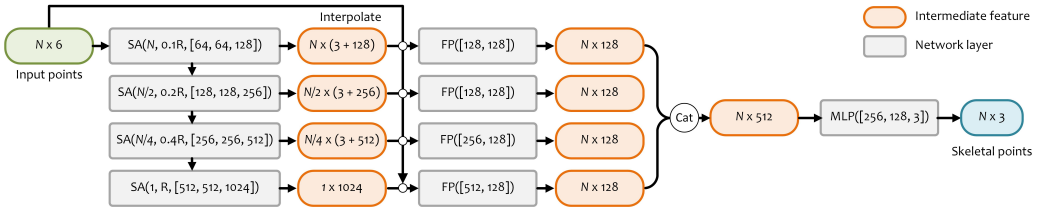


Figure D.1: Skeleton Estimation Network.

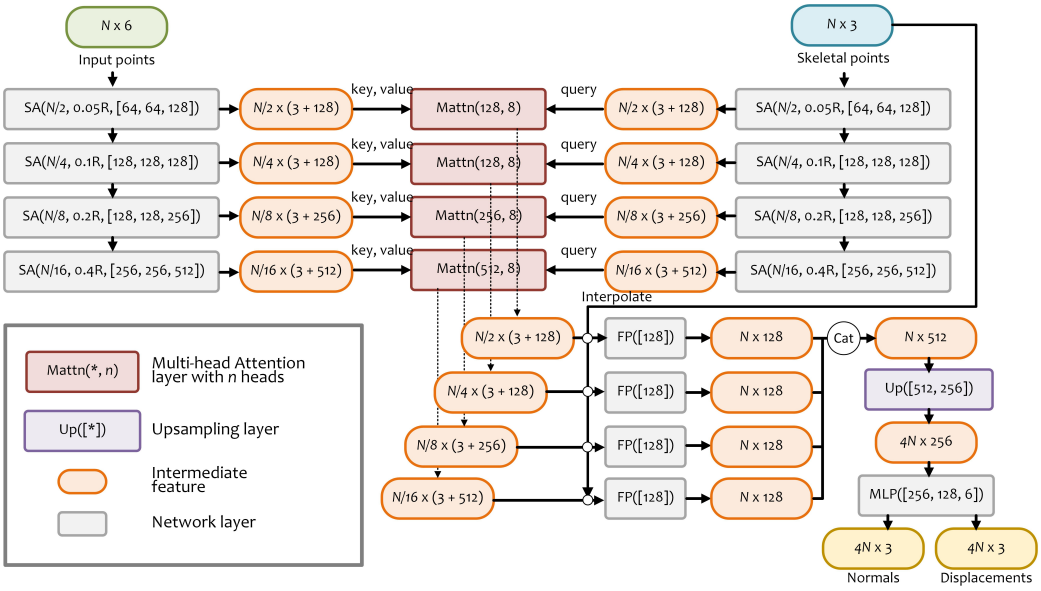


Figure D.2: Network Architecture of Skeleton2Surface.

D.1.2 Skeleton2Surface with Non-local Attention

With the estimated shape skeleton, we propagate the surface features from the input scan to each skeletal point with our Non-Local Attention module (see Section 6.3.1 in Chapter 6). Then the skeletal point features are aggregated to regress the displacements to the shape surface and the corresponding normal vectors on the surface. The network architecture is illustrated in Figure D.2, wherein the upsampling layer is explained in Figure D.3. The four parallel fully-connected layers in Figure D.3 can be implemented with the efficient group convolutions (Zhang et al. 2018).

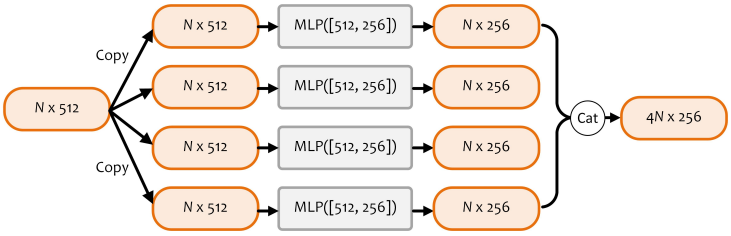


Figure D.3: Upsampling layer in Skeleton2Surface.

D.1.3 Surface Adjustment with Local Guidance

With the above layers, we can preliminarily obtain the surface points with normals. In Section 6.3.3 of Chapter 6, we involve a surface adjustment to merge the input scan to improve the fidelity on observable regions. We present the surface adjustment network in Figure D.4. For the discriminator, we utilize the basic architecture of Li et al. (2019b) with a sigmoid layer to score the confidence value of each patch on our predicted surface. It approximates 1 if the discriminator decides that a patch is similar to the ground-truth, and 0 if otherwise.

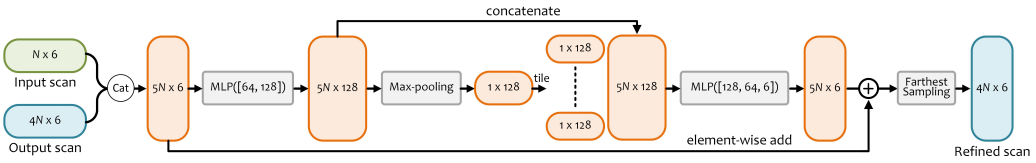


Figure D.4: Surface adjustment.

D.2 Data Preparation

Full scan data. In Chapter 6, we adopt the ShapeNet (Chang et al. 2015) dataset in our experiments. We observe that the man-made objects in ShapeNet are usually with non-manifold meshes and inner structures. To obtain watertight surface points and meshes of an object, we set up eight virtual cameras around the ShapeNet model to capture depth maps and reconstruct the surface mesh (see Figure D.5). Specifically, we align and scale

each model into a unit cube, and render the depth maps from eight viewpoints (centered at the eight corners of a cube with the side length at 2). We back-project the depth maps to 3D and obtain the ground-truth surface points. For points from each depth map, we also calculate their normal vectors with Williams (2020). The direction of normal vectors are flipped outside the shape surface. The ground-truth surface mesh are reconstructed with Poisson Surface Reconstruction (PSR) (Kazhdan and Hoppe 2013). The surface points and normals in the full scan data are used to supervise our surface point completion.

Skeleton data. We utilize the shape skeletons from ShapeNet-Skeleton dataset (Tang et al. 2019) to supervise our skeleton estimation network, where skeletal points are correspondingly aligned and scaled to the same scope with ShapeNet models.

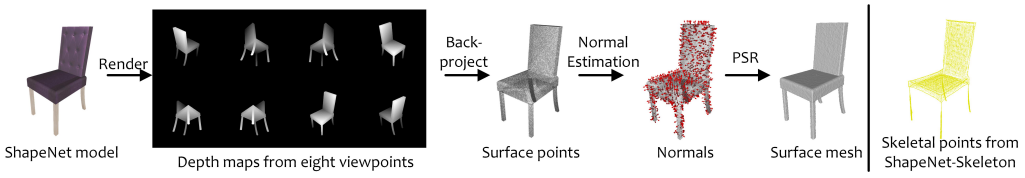


Figure D.5: Ground-truth data preparation

Partial scan data. In our training, we randomly select one partial scan from the eight viewpoints as the input data (see Figure D.6), and use the full scan in Figure D.5 as the ground-truth.

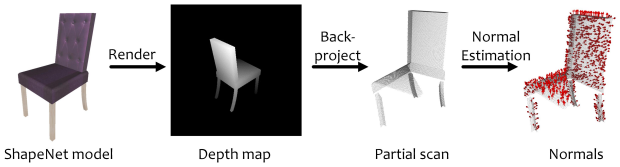


Figure D.6: Input data preparation

We adopt the train/validation/test split from Yi et al. (2016), which contains limited ShapeNet categories (containing airplane, chair, table, lamp, car in our experiments). For extra categories (inc. rifle, bench and watercraft), we adopt the split ratio of 6/2/2 in experiments.

D.3 More Qualitative Comparisons

We list more qualitative comparisons with previous methods, i.e., MSN (Liu et al. 2019), PF-Net(Huang et al. 2020), PCN (Yuan et al. 2018), P2P-Net Yin et al. (2018), DMC (Liao et al. 2018a), ONet (Mescheder et al. 2019), IF-Net (Chibane et al. 2020) and P2P-Net* (augment P2P-Net with normal estimation channels, see Section 6.5.6 in Chapter 6) in Figure D.7, where both the point and mesh completion results are compared on seven categories (inc. chair, lamp, rifle, table, airplane, bench and watercraft).

D.4 More Quantitative Comparisons

D.4.1 Comparisons on Extra Categories and Metrics

We compare our method on extra two categories (bench and watercraft) on both point and mesh completion in Table D.1 and Table D.2. In point completion evaluation, the number of output points is set to 2,048 for a fair comparison. In Table D.2, 8,192 points are uniformly sampled to evaluate the mesh reconstruction performance. We benchmark the input scale with 2,048 points, and the ground-truth with 10k points in both point and mesh evaluations.

Table D.1: Quantitative comparisons on point cloud completion.

	Chamfer Distance-L_2 ($\times$1000) $\downarrow$					Earth Mover's Distance ($\times$100) $\downarrow$				
Category	MSN	PF-Net	P2P-Net	PCN	Ours	MSN	PF-Net	P2P-Net	PCN	Ours
Bench	0.267	0.538	0.237	0.489	**0.204**	**0.234**	0.775	2.782	5.534	0.464
Watercraft	0.258	0.452	0.179	0.429	**0.153**	0.248	0.900	1.871	3.405	**0.232**

Table D.2: Quantitative comparisons on mesh reconstruction.

	Chamfer Distance-L_2 ($\times$1000) $\downarrow$					Normal Consistency $\uparrow$				
Category	DMC	ONet	IF-Net	P2P-Net*	Ours	DMC	ONet	IF-Net	P2P-Net*	Ours
Bench	0.312	0.857	0.428	0.180	**0.125**	0.772	0.743	0.791	0.780	**0.839**
Watercraft	0.363	1.152	0.619	0.148	**0.092**	0.794	0.766	0.815	0.835	**0.898**

Besides the Chamfer distance and normal consistency used in mesh evaluation (see Section 6.5.6 in Chapter 6), we also list the 3D IoU scores (Mescheder et al. 2019) in Table D.3.

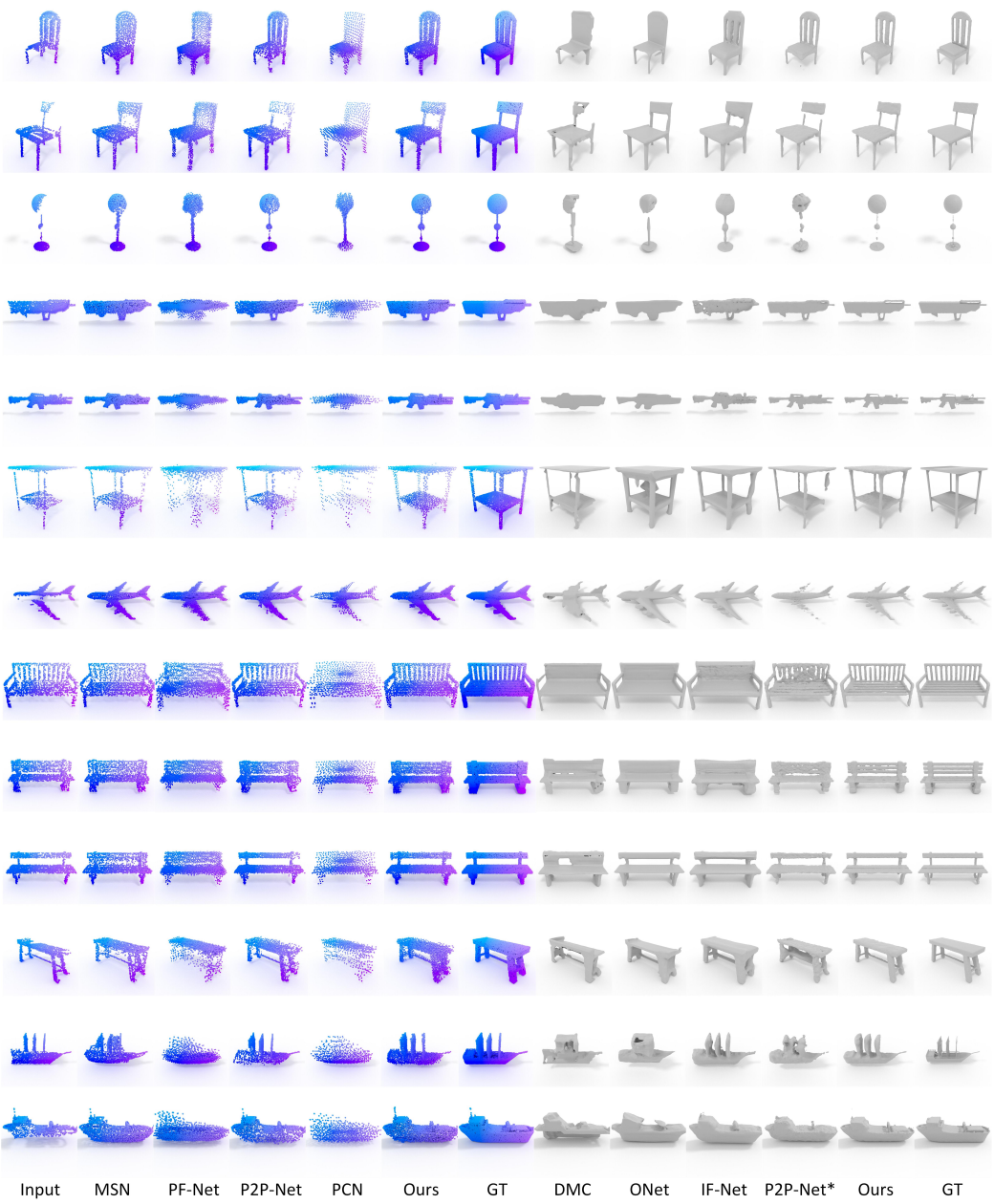


Figure D.7: More qualitative comparisons on the testing set.

D.4.2 Discussions on Normal Estimation

In this part, we mainly discuss the effects of using skeletal points impacted on point normal estimation. To this end, we design two configurations of

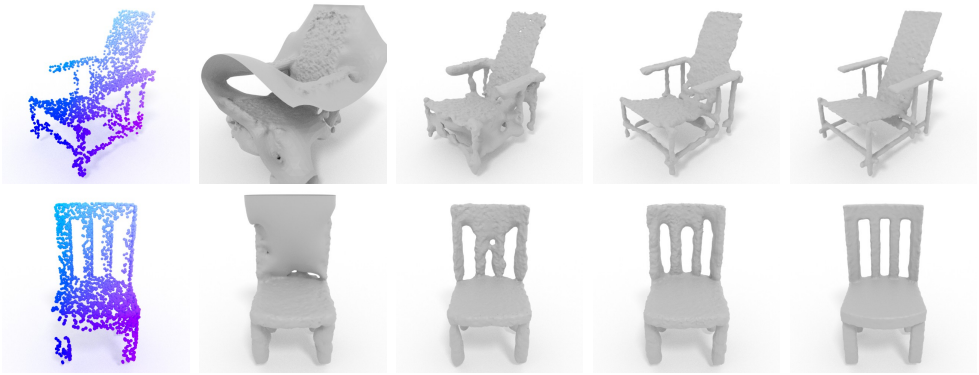
Table D.3: Average 3D IoU on object categories (%)↑

Category	DMC	ONet	IF-Net	P2P-Net*	Ours
Airplane	33.45	55.25	**72.95**	68.15	69.07
Rifle	29.33	51.37	30.77	60.81	**66.38**
Chair	24.75	39.45	59.90	57.12	**66.31**
Lamp	22.28	43.59	60.25	56.15	**73.27**
Table	24.35	35.93	**69.80**	56.38	57.92
Bench	26.89	51.65	**68.27**	45.22	48.85
Watercraft	26.78	48.90	73.34	62.45	**74.52**
Average	26.83	46.59	62.18	58.04	**65.19**

networks. Since P2P-Net (Yin et al. 2018) shows promising results on both point and mesh completion, we adopt P2P-Net as the baseline method. As the original P2P-Net does not take account the normal estimation, we augment P2P-Net with extra channels for normal regression (see Section 6.5.6 in Chapter 6). The second configuration is our SK-PCN without surface adjustment (Ours*), which is to investigate the effects of using shape skeletons. All networks produces 8,192 points for each object (inline with Section 6.5.6 in Chapter 6). We present the CD and Normal Consistency scores on the chair category in Table D.4, and the visualizations in Figure D.8. Figure D.8b shows the PSR results using normals directly calculated from our point clouds with Williams (2020). The results in Table D.4 indicate that skeletal points benefit both the point approximation and normal estimation.

Table D.4: Ablative comparisons on mesh reconstruction.

	Chamfer Distance-L_2 ($\times$1000) ↓		Normal Consistency ↑	
	P2P-Net*	Ours*	P2P-Net*	Ours*
Score	0.258	**0.177**	0.801	**0.847**



(a) Input (b) PSR (c) P2P-Net* (d) Ours* (e) GT

Figure D.8: Reconstruction results with different configurations.