Volume xx (200y), Number z, pp. 1–29

Advances in Data-Driven Analysis and Synthesis of 3D Indoor Scenes

Akshay Gadi Patil[1] Supriya Gadi Patil[1] Manyi Li[2†] Matthew Fisher[3] Manolis Savva[1] Hao Zhang[1]

[1]Simon Fraser University [2] Shandong University [3] Adobe Research

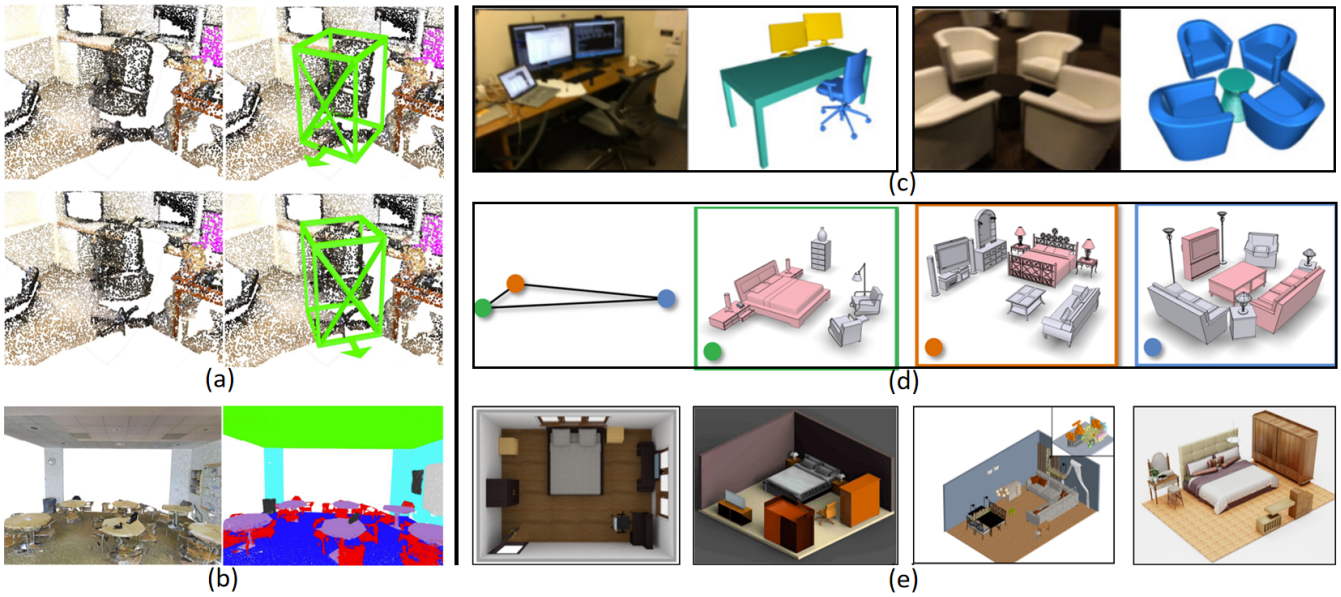

Figure 1: *A sampler of representative results for different indoor scene modeling tasks surveyed in this report – (a) 3D object detection [YWY22]* (Section 5.1)*, (b) 3D scene segmentation [ZJJ*21]* (Section 5.2)*, (c) scene reconstruction as image-CAD model alignment [GDN22]* (Section 5.3)*, (d) 3D scene similarity [XMZ*14]* (Section 5.4)*, and (e) 3D scene synthesis [WSCR18, LPX*19, YGZT21, PKS*21]* (Section 6)*. We survey advances in these indoor scene modeling tasks mainly in the realm of 3D geometry.*

Abstract

This report surveys advances in deep learning-based modeling techniques that address four different 3D indoor scene analysis tasks, as well as synthesis of 3D indoor scenes. We describe different kinds of representations for indoor scenes, various indoor scene datasets available for research in the aforementioned areas, and discuss notable works employing machine learning models for such scene modeling tasks based on these representations. Specifically, we focus on the analysis *and* synthesis *of 3D indoor scenes. With respect to analysis, we focus on four basic scene understanding tasks – 3D object detection, 3D scene segmentation, 3D scene reconstruction and 3D scene similarity. And for synthesis, we mainly discuss neural scene synthesis works, though also highlighting model-driven methods that allow for human-centric, progressive scene synthesis. We identify the challenges involved in modeling scenes for these tasks and the kind of machinery that needs to be developed to adapt to the data representation, and the task setting in general. For each of these tasks, we provide a comprehensive summary of the state-of-the-art works across different axes such as the choice of data representation, backbone, evaluation metric, input, output etc., providing an organized review of the literature. Towards the end, we discuss some interesting research directions that have the potential to make a direct impact on the way users interact and engage with these virtual scene models, making them an integral part of the metaverse.*

CCS Concepts

• ***Computing methodologies*** → ***3D indoor scenes***, ***scene analysis and synthesis***, ***neural models*** ;

1. Introduction

A central goal in computer graphics (CG) is to develop tools for generating real as well as imagined artifacts and environments, such as 3D objects and scenes. The pursuit of this goal has been revived in the past decade with a remarkable development in computing technology, including but not limited to, hardware, compute and machine learning algorithms. Specifically, the dawn of the big-data era in visual computing coupled with the fast assimilation of deep learning technology has pushed the frontiers of CG research, especially in the realm of content creation and understanding. In this report, we narrow down the focus of the word “content", to simply refer to 3D indoor scenes.

In real life, an indoor scene is physically realized by a sequential arrangement of objects in a region-bounded indoor space. The ubiquity of 3D indoor scenes in real life, has placed an increasing demand for simulations in a wide variety of applications, ranging from AR/VR, video games and indoor navigation, to creating virtual runs for AI agents that live and interact in those environments. These indoor scenes are characterized by their constituent elements – 3D object models laid out in a spatially constrained manner. These objects need to be *held together* in some form for functional reasoning and/or contextual interpretation based on the intended human activity.

To vividly simulate such indoor environments, one needs access to a repository of 3D object models, and posses familiarity with not-so-easy 3D modeling tools. A proxy to this would be to collect large quantities of real-world scenes through acquisition devices (stored as sequences of RGB-D image frames, which can then be converted to 3D point cloud) and perform object reconstruction that adhere to the captured scene layout. This alternative has its own unavoidable limitations – captured scenes have inherent errors arising out of sensor limitations that need to be processed before deploying for downstream scene modeling tasks, and 3D reconstructions at both the object level and arrangement level are poor. This premise, then, opens up a multitude of research possibilities in 3D indoor scene modeling, with a *analysis-for-synthesis* theme, keeping in mind the overall goal of content creation.

The first step in reconstructing an acquired 3D scene is to understand its composition, which reduces to localizing constituent 3D objects. Given a large collection of such real-world scans, algorithms can be developed that can learn occurrence and placement patterns of prominent/all objects, leveraging the localization module. These priors can be used to generate more of such scene layouts, tackling the content creation bottleneck at the arrangement level. Though object-level reconstruction from images/scans is a challenging task, existing approaches could be borrowed to roughly visualize the underlying objects. In addition, semantic scene segmentation can complement 3D object localization (and vice-versa) in heavily occluded scenes at the object level, leading to better scene reconstruction. The knowledge gained during these analysis tasks can help in generating diverse scenes. We, therefore, focus on modeling scenes in the context of both *analysis* and *synthesis* tasks.

†Corresponding Author: manyili@sdu.edu.cn

1.1. Related Surveys

Our focus is on data-driven modeling of indoor scenes, that includes both *analysis* and *synthesis* of scenes, irrespective of their representation. In the past, [PMG*20] focused on structured reconstruction of 3D indoor scenes, and [CRW*20] focused on generative models for 3D structures, which partly covers 3D indoor scenes as applications to presented approaches in different papers surveyed. Both reports focus on structural methods, one on reconstruction and the latter on generation, respectively. Our report differs from the two in the sense that it is not confined to structured modalities, and includes different aspects of scene analysis, going beyond reconstruction. As well, for scene generation, we focus mainly on neural generation, though also highlighting model-driven methods that allow for human-centric, progressive scene synthesis. With a mix of historical and contemporary works, we provide a comprehensive survey on fundamental scene modeling tasks.

2. Scope of the report

This report deals with 3D indoor scenes, which has a rich literature on different aspects of analysis techniques, and a relatively smaller literature on synthesis techniques. As such, it is hardly possible to exhaustively survey all such publications. This report is focused on providing technical insights into some of the prominent works in scene analysis and synthesis tasks, with an emphasis on how different scene representations necessitate the development of deep learning models that cater to such representation while addressing the scene modeling task at hand.

Individual scene modeling tasks presented in this report deserve a survey of their own. Our aim is to provide directional pointers, with fundamental technical insights, on some of the seminal, popular and recent works in these areas. To the best of our knowledge, this is the first such attempt to bring all scene modeling tasks in a single report, with a focus on neural network based modeling (prominent model-driven approaches have also been touched upon where the context so necessitates).

Organization We first present different types of indoor scene representations (Section 3), followed by various indoor scene datasets publicly available for use (Section 4) for different analysis and synthesis techniques.

In general, analysis of layouts, both 2D and 3D, spans a wide range of goals, from low-level understanding tasks such as primitive detection (corners, line-segments) and semantic segmentation, to high-level layout understanding tasks such as saliency detection, layout reflowing and layout retrieval to name a few.
In the context of 3D indoor scenes, analysis refers to understanding of the object layout withing a confined space, which can be categorized into two fundamental tasks – 3D object detection (Section 5.1) and 3D scene segmentation (Section 5.2). A more high-level but challenging task in scene analysis we cover in this report is that of scene reconstruction, either from single image or posed images (Section 5.3). Finally, we discuss relevant literature

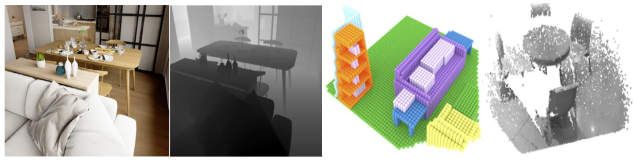

Figure 2: *Different forms of visual representation for indoor scenes as discussed in Section 3.1: (First two) RGB image depicting an indoor scene and its corresponding depth map [AW18], (third) voxelized representation of an indoor scene [SYZ*17] and (right) point cloud representing a subscene [QCLG20].*

in 3D scene similarity (Section 5.4), thus concluding our coverage of scene analysis tasks. For synthesis techniques (Section 6), we mainly look at recent progress towards this goal, which by default, has been skewed towards neural models. A more detailed discussion of model-driven techniques for scene generation can be found in [CRW*20].

Audience This report is written keeping in mind new graduate students in computer science (and allied disciplines). Readers should have a basic understanding of linear algebra, probability and statistics, machine learning and standard deep learning machinery (ex: CNN, GCN, GAN, VAE). This report, is by no means, an exhaustive collection of works dealing in analysis and/or synthesis of indoor scenes. It is more of a directional digest for research in this area, exposing the main problems and challenges involved, the observed gains due to a paradigm shift from model-driven approaches to data-driven ones where applicable, the interplay between scene representation and choice of computing machinery (neural network), and the evolving trends that could inspire novel problems in this area. Finally, we discuss open problems in this domain that have wider industrial applicability.

3. Scene Representations

Representation of a scene should convey information about the *combination* of atleast two things – (1) the composition of its layout, either as a single entity or as a set of constituent objects (semantics) and, (2) the arrangement of objects (and perhaps their relations) in a given space. Such a combination could be either explicitly encoded or may need to be inferred separately. Representations which necessitate additional processing to infer information about the object semantics and their placement are oblivious to the underlying structure of the layout, and are said to be purely visual in nature (ex:raster images in 2D, point clouds and voxel grids in 3D). On the other hand, if the semantics of the constituent objects and their placement (and even relations) is explicitly encoded, such representations are said to be structural in nature (ex: multi-channel segmented images, graphs). Thus, we broadly classify scene representation into two categories: (a) visual, and (b) structural.

3.1. Visual Representations

Visual representations, as the name suggests, mainly represent a scene as a single entity. Common examples of this kind of representation include 2D images (monochrome, RGB and RGB-D) and 3D point cloud. Figure 2 shows these common visual representations used for scenes, and layout data, in general.

2D images and 3D voxels Two-dimensional rectangular grids of picture elements in the form of images are the common form of visual representation for scenes. Such data could be obtained in different ways such as using digital cameras or scanning devices (RGB-D cameras) such as Microsoft Kinect. There also exists a 3D counterpart to 2D pixel grids called 3D voxel grids that approximate a 3D surface. Such volumetric data representation are memory intensive and have been found to be intractable for modeling 3D shapes, let alone 3D scenes.

3D point cloud Real world 3D indoor scenes are also digitized using commercial 3D scanners, where the acquired data is stored as a point cloud representing the surface of objects in the 3D environment, as shown in Figure 2. Point clouds do not encode topological information of the underlying 3D content and simply depict the 3D data in the simplest visual form possible.

This kind of 2D grid representation in the form of images and 3D point cloud respectively, depict an indoor scene as a single entity, i.e., the semantics of the constituent objects, their geometric arrangement and their relationships are not accounted for by the representation, and will have to be inferred separately. Examples of such representations include monochrome images for 2D indoor scenes in the form of a floorplan [KYH*19], RGB (+D) images [NSF12] and point clouds [DCS*17]. Figure 2 illustrates different visual representations of indoor scenes.

The above representations convey information about indoor scenes in a structure-agnostic manner – that is, only an abstraction of the scene layout is available. Its composition based on constituent objects will need to be inferred separately.

3.2. Structural representations

Structure refers to the atomic composition of an entity/matter. In the case of indoor scenes, it has to do with the type, arrangement and/or relationship of different objects forming the scene layout. There are many ways of representing structured data, which have been surveyed [CRW*20] for 3D structures in general. While most of it directly applies to 3D scenes, the choice of such structural representations for different works chosen for this report needs to be discussed. We fill this piece of information below by briefly describing structure representations and the associated works in the context of 3D indoor scenes.

Segmented Scenes Structure, in its simplest and weakest form, can be thought of as assigning semantic labels to visual representations of 2D, 3D scenes (images, voxels, point cloud etc.), where each pixel/voxel/point represents the type of room/object entity present at that location. For example, assigning labels to (a) rooms in a floorplan image as in the RPLAN dataset [WFT*19, PBEPAE20] and (b) objects (bed, night stand, table lamp, cabinet etc.) in a top-down scene image [WSCR18] can be considered as collections of simplest scene structures. Figure 3 shows a few examples of this form of structural representation. This representation is built upon standard visual representations. Furthermore, machine learning models used for processing (analysing and synthesizing) visual representations can be directly employed for these

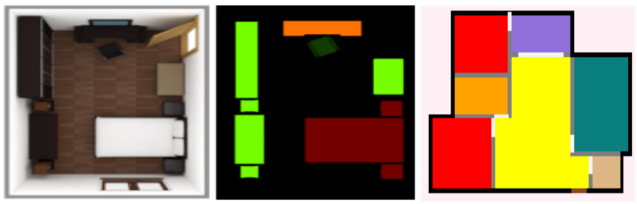

Figure 3: *Figure illustrating semantic segmentation of indoor scenes: a top-down scene image on the left [WSCR18] and a 2D floorplan on right [WFT*19, PLF*21]. Such segmented semantic entities form the simplest form of structural representation as discussed in 3.2.*

segmented layouts, making them easier to work with. The disadvantage is that they do not (and can not) understand the underlying relationships among different elements, and therefore, not a strong fit for geometric analysis and synthesis tasks.

Component/Entity Sets A set of indoor scene elements, i.e., objects, with information about their semantics and oriented bounding box, explicitly accounts for "atoms" in the structural representation. Such a set is called an entity set, which essentially is a set of freely-floating elements in space, with no relationship information encoded between any pair of elements. This kind of scene representation is used in conjunction with a wide range of neural networks that process just the elements (their box coordinates and semantics), as evidenced in sequence-to-sequence analysis techniques [AGSK20] which makes use of a Recurrent Neural Network, or, for synthesis tasks such as in [WYN20, PKS*21] which make use of a Transformer.

Graphs Adding relationships between pairs of elements (nodes of a graph) in the entity set, in the form of edges, reveals the full structure of a scene layout [PLF*21, ZCZ*21]. These edges connecting nodes in a graph usually encode spatial relationships, such as adjacency, proximity [LYJ*20, HHT*20] and physical support [FSH11], but can also be simply connected between any two pairs of objects/elements, regardless of their spatial relationships [MRC20, PLF*21]. Figure 4 shows one such example of a semantic relationship graph for an indoor 3D scene.

The advantage of using graphs is that they are a more general and flexible form of structured representation. However, structured scene modeling is constrained by advances in graph modeling techniques, which is an active area of research in the broader machine learning community. As such, development of sophisticated architectures for analysis tasks [LGD*19] and generative models of arbitrary graphs is still in nascent stages.

Any flat graph can be encoded as a hierarchy by repeatedly contracting its edges. Trees or hierarchies, are therefore, less dense. The key difference is that hierarchies consist of internal nodes that represent groups of objects, while all the nodes of a graph represent the objects. Hierarchies are a class of restricted graphs and can be used to represent much of the naturally occurring structure in the real world. For example, a 3D scene can be thought of as a hierarchy of objects [LCK*14, LPX*19], where objects are grouped based on their spatial positions, which in turn, is based on the functionality of individual objects in a group, see Figure 4. This representation was also extended to 2D documents in [PBEPAE20] where individual document entities were merged along a tree using spatial relationships. A major bottleneck of representing 3D scenes using hierarchies is that there is no unique way of doing so, and as such, task-specific models that consume hierarchies inherit design limitations as a result of hard-coded heuristics used to construct such hierarchies.

4. Indoor 3D scene datasets

There exist indoor scene datasets that either capture real world scenes using acquisition devices or are professionally designed using curated 3D CAD models of furniture assets. Table 1 summarizes various such indoor 3D scene datasets, which have been discussed in literature at various points such as in [PMG*20, LYS*21, FCG*21]. We aim to provide a comprehensive list of such up-to-date datasets along with the potential applications they could serve, each of which are briefly discussed below.

SUN 3D [XOT13] offers a dataset of large-scale RGB-D video frames with semantic object segmentations and camera pose. The dataset contains 415 videos captured for 254 different indoor spaces, in 41 different buildings. Geographically, the places scanned are mainly distributed across North America, Europe and Asia. The dataset can be used to obtain (a) a point cloud of the scene; (b) 3D object models obtained from segmentation; (c) all viewpoints of an object, and corresponding camera poses relative to that object; (d) a map of a room, showing all of the objects and their semantic labels from a bird's-eye view.

UZH 3D dataset [UZH] contains 40 laser-scanned models of office environments and apartments. Some scenes have arbitrarily oriented walls which pose a challenge to many techniques in reconstructing floor plans. The point cloud models in the dataset are provided in ASCII PTX format with color information.

ETH 3D dataset [ETH] consists of 898 RGB images of both indoor and outdoor spaces. The dataset also provides ground truth point cloud and depth maps which can be used to benchmark multi-view stereo algorithms.

Matterport3D [CDF*17] provides a large-scale RGB-D dataset of 90 building-scale scenes. The dataset contains 10,800 panoramas and 194,400 RGB-D images. It is also provided with reconstructed textured 3D mesh with object level semantic annotations and camera pose. The dataset can be used for various tasks such as room-type classification, semantic segmentation, surface normal estimation, keypoint matching and view overlap prediction.

ScanNet [DCS*17] is a RGB-D video dataset of 1513 indoor scenes. The 3D reconstructed mesh has texture information and is labeled with object-level semantic segmentations. Moreover, the dataset also provides aligned 3D CAD models for a subset of scans. The dataset can be used for many 3D scene understanding tasks including 3D object classification, semantic voxel labeling and CAD model alignment and retrieval.

CRS4/ViC dataset [CRS] contains equirectangular RGB images covering 360x180 degrees of multi-room residential as well

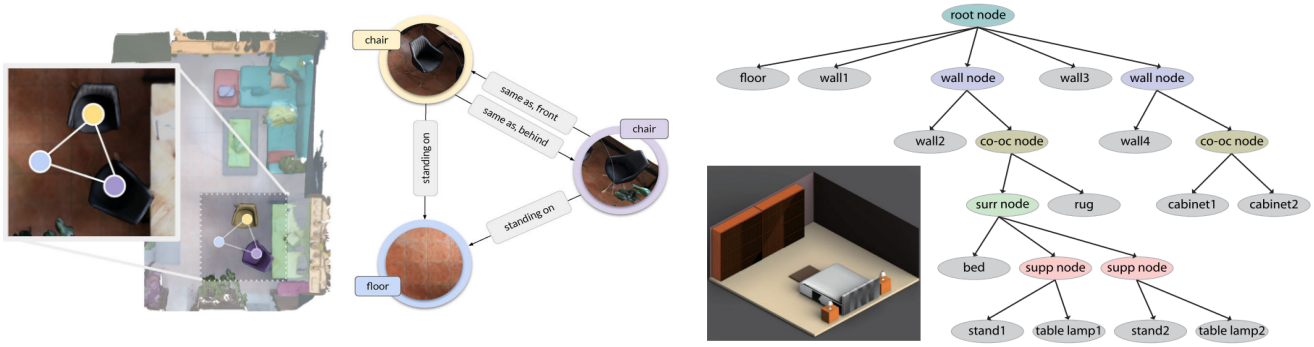

Figure 4: *Representing 3D indoor scenes via strong structural representations (Section 3.2): on the left is an indoor scene represented as a semantic-relational graph [WDNT20], and on the right is a bedroom scene represented as a hierarchy [LPX*19].*

Name	Data	Coverage	Capture	#scenes	#CAD models	Model Textures	3D Annotation
Real scans							
SUN 3D [XOT13]	Registered RGB-D	Perspective	Hand-held video	254	-	No texture	Raw PCD
SUN RGB-D [SX14]	Registered RGB-D	Perspective	Hand-held video	10779	-	No texture	Raw PCD
Matterport3D [CDF*17]	Registered RGB-D	Panoramic	Tripod	2,056	-	Rec. from scans	Raw Mesh
ScanNet [DCS*17]	Registered RGB-D	Perspective	Hand-held video	1,506	296	Rec. from scans	Raw Mesh
Scan2CAD [ADD*19]	Mesh	All	Manual modeling	1506	3049	No texture	Mesh
OpenRoom [LYS*21]	RGB-D	Perspective	Hand-held decide	1287	44	UV mapping	Mesh
UZH 3D [UZH]	Registered PCD	Scan	Tripod	53	-	No texture	PCD
3DSSG [WDNT20]	PCD	All	Hand-held device	1482	-	Rec from scans	Mesh + scene graph
SceneNN [HPN*16]	RGB-D	Perspective	Hand-held device	100	-	Rec. from scans	Mesh
Synthetic							
Replica [SWM*19]	CAD model	All	Manual modeling	18	-	RGB texture camera	Mesh
Structured3D [ZZL*20]	CAD model	All	Manual modeling	3500	-	No texture	3D structure
SceneNet [HPB*16]	Mesh	All	Manual modeling	57	3699	No texture	Mesh
InteriorNet [LSM*18]	RGB	Perspective	Manual modeling	-	-	No texture	-
Hypersim [RRR*21]	RGB	Perspective	Manual modeling	461	-	Per pixel color	RGB-D
3D-FRONT [FCG*21]	Mesh	All	Manual modeling	18968	13151	Professional	Mesh
Real scene images							
ETH 3D [ETH]	Registered RGB	Perspective	Tripod	898	-	No Texture	PCD
CRS4/ViC [CRS]	Registered RGB	Panoramic	Tripod	191	-	No texture	-
NYU Depth v2 [SHKF12]	Registered RGB-D	Perspective	Hand-held video	1449	-	No texture	RGB-D
TUM [SEE*12]	Registered RGB-D	Perspective	Hand held video	39	-	No texture	RGB-D

Table 1: *A summary of publicly available 3D indoor scenes datasets, grouped based on acquisition source, along different axes that include high-level details such as the physical mode of capture to low-level ones such as the kinds of annotations on the scene and the number of CAD models/scenes. #scenes indicates number of rooms/scenes populated with 3D furniture objects, PCD="Point cloud".*

as commercial environments. It also contains images of rooms with double sloped ceiling and cluttered with many objects, making it a challenging dataset to use for reconstructing a 3D floor plan.

Replica dataset [SWM*19] provides 3D indoor scene reconstructions of rooms and buildings with a rich semantic variety of environments and their scale. The dataset contains high-dynamic-range (HDR) textures and per-primitive semantic class and instance information. Due to high level of realism of renderings from Replica dataset, the creators believe that deep learning models trained on this dataset can adapt well to real-world images and videos of indoor scenes.

Structured3D dataset [ZZL*20] contains rich ground truth 3D structure annotations of 21,835 rooms in 3,500 houses, and more than 196k photo-realistic 2D renderings of the rooms. The scenes are represented in the format of "primitive + relationship". Usefulness of the dataset is demonstrated on room layout estimation task.

NYU Depth v2 dataset [SHKF12] consists of 1449 RGB-D images of commercial and residential buildings comprising of 464 indoor scenes. The dataset provides dense per-pixel labeling, where each object in the image is labeled with class label and instance annotations. The dataset also includes *support* annotations between two objects in the image. This dataset can be used for tasks such as object recognition, segmentation and inference of physical support relationships.

SUN RGB-D dataset [SX14] consists of 10779 RGB-D images of real indoor scenes captured using four different sensors. The en-

tire dataset is densely annotated with room category, 2D and 3D oriented bounding boxes for objects, and camera pose. Specifically, it includes 146,617 2D polygons and 58,657 3D bounding boxes with accurate object orientations, as well as a 3D room layout and category for scenes. This dataset can be used for scene-understanding tasks and evaluate such models meaningful 3D metrics.

TUM dataset [SEE*12] contains 39 image sequences capturing office environments and industrial halls. Each sequence contains color and depth images and also ground truth trajectory. The dataset is aimed at evaluating visual odometry and visual SLAM systems.

3DSSG dataset [WDNT20] provides 3D semantic scene graphs for 1482 scenes from 3RScan [JW19] dataset. 3DSSG dataset contains scene graphs with 40 different types of object relationships, 93 different attributes for objects from 534 different class labels represented in class hierarchies. Such semantically rich scene graphs can be used for many applications such as semantic scene graph prediction and cross-domain scene retrieval task.

SceneNN [HPN*16] is a dataset of RGB-D scans of 100 indoor spaces. The dataset provides information about camera pose, reconstructed mesh, color and texture information, axis-aligned and oriented bounding boxes, as well as object pose. This dataset can be used for shape completion, scene relighting, creating synthetic scenes using CAD models by using object distribution statistics of real scenes from SceneNN dataset and novel view synthesis task.

Scan2CAD [ADD*19] is a large-scale dataset of 1506 ScanNet [DCS*17] scene objects aligned to 14225 (3049 unique) CAD models of ShapeNet dataset [CFG*15]. It contains 97607 pairwise keypoint correspondences between scene objects and CAD models. The dataset also contains oriented bounding boxes for objects in the scenes. This information can be used in various applications such as correspondence prediction between unseen 3D scenes and CAD models, and their pose estimation task.

OpenRoom [LYS*21] dataset is aimed at creating photo-realistic indoor scenes by adding high-quality material and lighting information. The dataset uses 1287 ScanNet [DCS*17] scenes to create such photo-realistic indoor scenes. The dataset is annotated with ground truth scene layout, high quality material, spatially-varying BRDF lighting, including direct and indirect illumination, light sources, per-pixel environment maps and visibility. This dataset is useful in inverse rendering, scene understanding and robotics applications. The dataset can also be used for shape, material and lighting estimation which are crucial in augmented reality (AR) and virtual reality (VR) applications.

SceneNet [HPB*16] is a dataset of synthetically generated 3D indoor scenes. It contains 57 scenes of five categories: bedroom, office, kitchen, living room and bathroom. Each scene has 15-250 objects. The RGB-D renderings of these scenes can be used for per-pixel semantic segmentation task. The dataset also provides a tool which can be used to generate unlimited labeled 3d indoor scenes programmatically, which is helpful in training data-driven machine learning models.

InteriorNet [LSM*18] is a synthetic 3D indoor scene dataset created using 1M furniture CAD models and 22M interior layouts. The dataset has 15K sequences of 10K randomly selected layouts and 5M images rendered from 1.7M layouts. The dataset can be used to train and evaluate SLAM systems.

Hypersim [RRR*21] is photo-realistic synthetic 3D indoor scene dataset. It contains 77400 images rendered from 461 indoor scenes with per-pixel label, ground truth scene geometry, material and lighting information, semantic segmentation label. The dataset was evaluated on two scene understanding tasks: semantic segmentation and 3D shape prediction.

3D-FRONT [FCG*21] dataset contains professionally designed 3D indoor scenes of 31 scene categories. It has 6813 CAD houses with 18968 rooms furnished with high-quality textured 3D models from 3D-FUTURE [FJG*21] dataset. The usefulness of the dataset was demonstrated on scene understanding task such as 3D indoor scene synthesis and object texturing in scene context.

The explosion of NeRF [MST*21] has brought a variety of scene images into focus, most of which are single-object images and are not catered to indoor scenes. Here we briefly touch upon a few datasets used in novel view synthesis.

RealEstate10K [ZTF*18] is a dataset of camera poses on 10K real estate YouTube videos that contain indoor and outdoor scenes of houses. These videos are divided into clips of 1-10 seconds, and for each clip, the dataset provides information such as camera position, orientation and field of view per frame. This dataset finds its usefulness in tasks related to view synthesis.

ACID [LTJ*21] Another commonly used dataset for view synthesis is the Aerial Coastline Imagery Dataset (ACID) [LTJ*21]. It is a dataset of outdoor nature videos (891 videos) annotated with camera pose information.

Common Objects in 3D (Co3D) [RSH*21] is another dataset consisting of 18,619 videos of objects from 50 MS-COCO categories. Compared to RealEstate10K and ACID, Co3D dataset is simpler as the videos are focused on single objects with no occlusion.

Ego4D [GWB*22] provides a bit different dataset with videos capturing everyday activities from first-person perspective. It consists of 3025 hours of videos shot in indoor and outdoor scenarios. In addition to videos, it also provides other information such as 3D scans, audio, gaze, stereo, multiple synchronized wearable cameras, and textual narrations. This dataset finds usefulness not only in view synthesis but also in other challenging tasks such as analyzing hand-object interaction, audio-visual conversation and forecasting activities.

5. 3D scene analysis

The first step in computational scene synthesis, i.e., teaching computers to generate indoor environments, is scene analysis, i.e., teaching computers to understand their composition – what characterizes scenes of a particular category (say, bedrooms), what kind of furniture goes in there, how to identify different furniture objects, how to detect different instances of the same object in the scene, and how to reason about their placements in the context of global scene plausibility. In other words, computational analysis of scenes, a.k.a scene understanding, is a prelude to scene synthesis.

Related Work	Learning framework	Scene rep	Backbone	Input	Output	Dataset	Evaluation Metric(s)
[SX14]	Supervised	RGB-D image	SVM	TSDF + 3D Normal + Point Density + Shape features	*Pr*(C) for a 3D OBB	SUN RGBD	AP; 2D,3D IoU
[SX16]	Supervised	RGB-D image	3D CNN	TSDF scene and RGB image	*Pr*(C) and 3D OBB	SUN RGBD	AP, AR, 3D IoU
[RS16]	Supervised	RGB-D image	SVM	Point density and normal features	3D OBB	SUN RGBD	AP, AR
[QLHG19]	Supervised	RGB-D image	PointNet++	3D point cloud	3D OBB	SUN RGBD, ScanNet	3D IoU, AR, mAP
[QCLG20]	Supervised	RGB-D image	PointNet++, 2D CNN	RGB image and 3D point cloud	3D OBB	SUN RGBD	3D IoU, AP
[XLW*20]	Supervised	RGB-D image	PointNet++	3D point cloud	3D OBB	SUN RGBD, ScanNet	3D IoU, mAP
[ZSYH20]	Supervised	RGB-D image	PointNet	3D point cloud	3D OBB	SUN RGBD, ScanNet	3D IoU, mAP
[LZC*21]	Supervised	RGB-D image	PointNet, Transformer	3D point cloud	3D OBB	SUN RGBD, ScanNet	3D IoU, mAP
[YWY22]	Supervised	RGB-D image	PointNet and Equivariant Point Network	3D point cloud	3D OBB	Sun RGBD, ScanNet	3D IoU, AP, mAP

Table 2: *Table summarizing prominent **3D object detection works** in indoor scenes. We provide details on the scene representation, the input for and output of the model, the central algorithm that makes the task possible, dataset used and the metrics employed to evaluate results from proposed methods. AP - Average Precision, mAP - mean of Average precision, AR - Average Recall, IoU - Intersection over Union.*

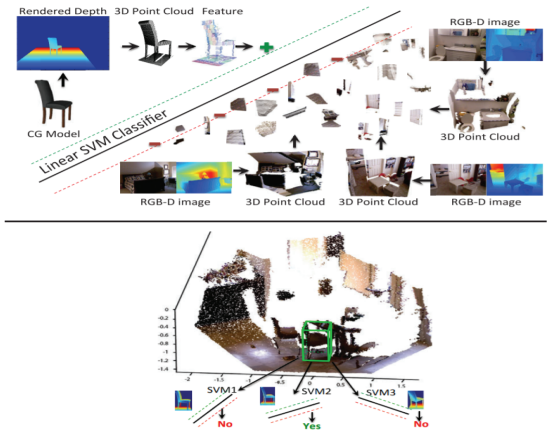

Figure 5: *Illustration of training (top) and test (bottom) phases for 3D object detection using* Sliding Shapes *[SX14]. Refer to the text in Section 5.1 for details.*

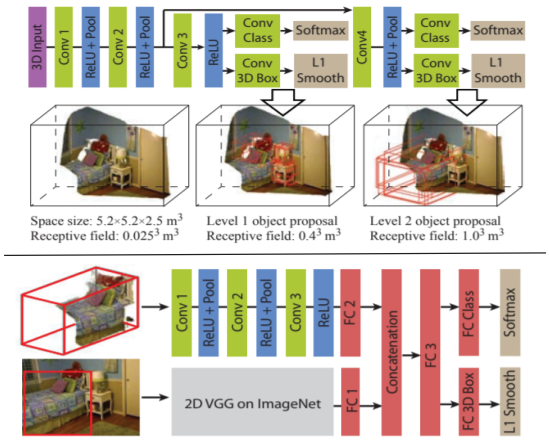

Figure 6: *Figure illustrating the technique behind* Deep Sliding Shapes *[SX16]. Top: using a 3D CNN to propose regions of interest in 3D. Bottom: training pipeline based on the 3D and 2D CNNs to localize bounding boxes in 3D space.*

Understanding comes from observations, which are relayed by indoor scene datasets, which have been discussed in Section 4. Each of these datasets uses a different form of representation for indoor scenes, as categorized in Section 3. In literature, different scene analysis tasks use different kinds of representation, which could be motivated by different factors such as easy availability of a dataset with one form of representation, friendliness toward off-the-shelf networks used as a part of the proposed approach, or the need for developing novel architectures due to the choice of a certain representation. In the upcoming sections, we provide a summary of different works along similar axes.

The main analysis tasks we cover in this report include 3D object detection (Section 5.1), semantic scene segmentation (Section 5.2), 3D scene reconstruction (Section 5.3, and 3D scene similarity (Section 5.4).

5.1. 3D object detection

Recognizing objects in a scene, i.e., identifying their semantic category and localizing their spatial position via 2D/3D bounding boxes has been a fundamental and long-standing goal of computer vision. This report does *not* cover works on 2D object detection in RGB images. Rather, we focus on notable works on 3D object detection in indoor scenes, a task that is challenging mainly due to variations in shape (both inter and intra-class), texture, illumination, viewpoint and the presence of clutter and occlusions. Table 2 provides a comprehensive overview of notable methods on 3D object detection in indoor scenes. Broadly, these works can be categorized into three types: sliding window techniques [SX14, SX16], grouping techniques [QLHG19,QCLG20,XLW*20] and group-free techniques [LZC*21], as discussed below.

Sliding window techniques. Song et al. [SX14] introduce Slid-

ing Shapes, a supervised machine learning-based approach for 3D object detection. The work makes use of depth maps for designing a 3D object detector, in addition to a collection of 3D CAD models, where each CAD model is rendered from many viewpoints, obtaining synthetic depth maps for every viewpoint. For each depth rendering of a CAD model, features based on truncated sign distance fields (TSDF) values, 3D normals, point density, and voxel occupancy are extracted (collectively known as point features) and an exemplar support vector machine (Exemplar-SVM) classifier is trained on point features of the sensor-acquired scenes. During test-time, a sliding window is moved through the 3D scene space to detect an object; see Fig 5.

The successor to Sliding Shapes, termed, Deep Sliding Shapes, was presented in [SX16]. It is a supervised deep learning-based framework which makes use of a 3D ConvNet that takes a 3D volumetric scene from an RGB-D image as input and outputs 3D object bounding boxes. A 3D Region Proposal Network (RPN) is trained at two different scales to learn object-ness from geometric shapes. An Object Recognition Network (ORN) is *jointly* trained with RPN to extract geometric features in 3D and color features in 2D, to eventually output a category label and 3D box coordinates. Figure 6 illustrates these two steps during training. At test time, a sliding window is again moved through the space of a 3D scene to detect the presence of an object.

One main limitation of both the above approaches is that they do not explicitly encode object orientation, which can hurt the performance of a 3D object detection system. Ren et al. [RS16] overcome this limitation by designing a new set of features, called, cloud-of-oriented-gradient (COG), that robustly link 3D object pose to 2D image boundaries. COG features are nothing but the gradients of 2D projections of oriented cuboid points falling inside the object voxel. COG features, in addition to the point cloud density features and 3D normal histogram features form the point features, are used to train an SVM for 3D object detection similar in spirit to Sliding Shapes. Sedaghat et al. [SZAB17] also address the limitation of Deep Sliding Shapes by adding orientation classification as an auxiliary task, and demonstrate that speed and accuracy of 3D detection using a sliding window increases when the 3D CNN is jointly trained on object labels, location and pose.

Voting techniques. In recent years, voting concepts, specifically, Hough voting, have made a comeback in the space of 3D object detection. VoteNet [QLHG19] and ImVoteNet [QCLG20] are two such works that are built on voting strategies. VoteNet demonstrates two advantages of using the voting strategy for 3D object detection – first, it does not make use of any 2D object detectors which used to be the de-facto step in 3D object detection, and second, the Hough voting technique used in the work is now differentiably learned in an end-to-end supervised manner.

As shown in Figure 7, the input to the system is a colorless point cloud of a scene, which is processed by PointNet++ [QYSG17], to produce features for every point. A voting net, which is nothing but a MLP on these point features, produces virtual points, called votes, for centers of 3D bounding boxes. The votes are clustered in the 3D space using farthest point sampling and L_2 distance, from where the extent of the 3D bounding boxes and their centroids are regressed using another MLP. All of this training is done in a supervised manner, where the votes on the training data are available since it is supervised. ImVoteNet [QCLG20] incorporates all the steps from VoteNet, but in addition, makes use of a 2D object detector, where votes are obtained in the image space, which is lifted to the 3D space (along with the 2D object center) by ray-casting.

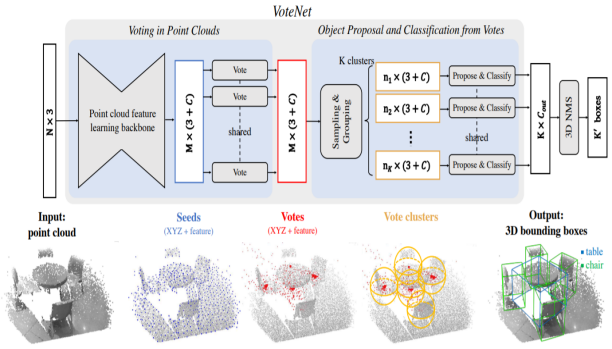

Figure 7: *An end-to-end learning pipeline for 3D object detection using voting technique (*VoteNet*) [QLHG19] that directly operates on 3D data without the need for any 2D image priors, such as 2D object detectors.*

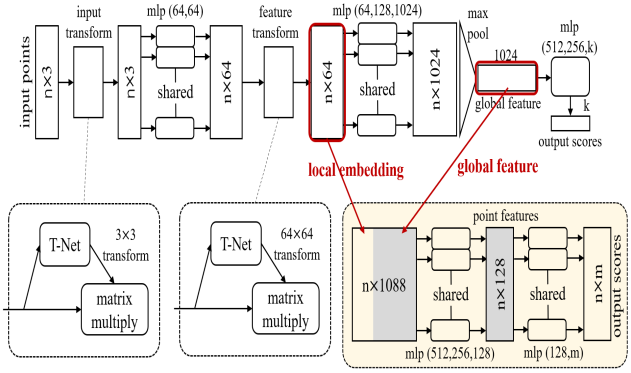

Figure 8: PointNet *architecture [QSMG17] that was originally proposed for the task of shape classification and segmentation at the object part level, as well as for indoor scenes. This work has been the basis for many 3D object detection works, especially for developing task-specific feature descriptors.*

Hybrid technique. A hybrid model for 3D object detection on colorless scene point clouds was proposed by Zhang et al. [ZSYH20], where bounding boxes are represented using three geometric primitives – bounding box centers, face centers and edge centers. These hybrid geometric primitives represent an overcomplete set of constraints that are predicted using a neural network. The predicted geometric primitives are converted into object proposals by defining a distance function between an object and the geometric primitives. The main purpose served by the distance function is that it helps in continuous optimization of object proposals. A final matching and refinement module is proposed to classify object proposals into detected objects.

Context-aware techniques. Despite a large inter- and intra-class shape variation and illumination effects in 3D indoor scenes, the

context relation of object arrangements provides useful cues for object detection. Some prior works seek holistic scene understanding to improve object detection with other auxiliary tasks. For example, Lin et al [LFU13] first estimate the candidate 3D cuboids of the objects, and then use a conditional random field (CRF) model to *jointly* solve for the scene classification and 3D object recognition. This holistic approach takes the scene context into account while also accounting for relations between different objects for the task of object classification.

Zhang et al [ZBK*] integrates the context relations into neural networks using automatically constructed scene templates. They first select a template and align it with the input scene by a transformation network, and then compute the global and local features of the input scene based on the aligned template. What makes this approach holistic is that it uses global feature to classify scenes, while using both the global and local features to predict the existence of objects and the box offset for better alignment. Huang et al [HQX*18] refer to their approach of jointly recovering the object bounding boxes, room layout, and camera pose as a holistic one. They estimate all this information from a given RGB image, then use the predicted camera pose to project the inferred 3D bounding boxes back to the 2D plane, in order to obtain a more consistent prediction.

Recent investigations have looked into utilizing context information to complement 3D object detection. Feng et al [FGW*21] take PointNet++ [QSMG17] as the backbone to generate candidate 3D object bounding boxes, and then use the object-object relation graph to reduce uncertainty during 3D bounding box regression. Duan et al [DZL*22] argue that 3D object detection can be improved by adopting relations between representative proposals, which is more efficient than those between all the predicted proposals. They accomplish this by proposing what they call a DisARM module, which first samples relation anchors with rich information and then estimates the weight of each proposal w.r.t the anchors based on spatial- and feature-aware displacements. The weighted proposal-anchor features provide contextual information to complement anchor proposal feature. In addition, Sun et al [SFZ*22] propose an online data augmentation pipeline based on the functional relation between objects, called Correlation Field, that helps boost the performance of object detection.

Transformer-based techniques. A more recent supervised learning approach by Liu et al. [LZC*21] to 3D object detection on scene point clouds makes use of the transformer model [VSP*17], which is essentially an attention-based feature aggregation module on the input. Unlike voting-based object detection works such as VoteNet [QLHG19] and ImVoteNet [QCLG20] where the points are assigned to an object candidate via a heuristic point-grouping stage, [LZC*21] uses a grouping-free approach for detecting objects in a scene point cloud. Instead of obtaining a candidate object feature from a heuristically-grouped set of points (also called "votes"), candidate features are computed by neurally estimating the contribution of each point to the object candidate using attention mechanism.

Note that all the above works have been discussed in the context of indoor scenes, which is the focus of this report. There exist impactful works that have been developed in the context of outdoor scenes, specifically on the KITTI dataset [GLSU13], such as PointRCNN [SWL19], PointPillars [LVC*19], CenterPoint [YZK21] that operate on the 3D point cloud of the scene and are able to predict a 3D bounding box for scene objects. In theory, these architectures could very well be extended to 3D indoor scenes, but supporting experiments have not been presented in the above methods.

Discussion With an anticipated shift from feature-engineered methods [SX14, RS16, SZAB17] to deeply learned ones [SX16, QLHG19, QCLG20, ZSYH20, LZC*21], 3D object detection in indoor scenes has drawn significant research interest over the years, leading to improved performances of proposed approaches. While some of these methods employ RGB image for 2D object detection [QCLG20] as an intermediate step for localizing object centroids in 3D, other do not make use of any scene image, and operate directly on the 3D scene point cloud.

A strongly desired property in 3D deep learning is rotation equivariance. Accounting for equivariance to object rotations in 3D scenes, not at the global input level, but rather at the object level, is an interesting future direction – attempts have been made to take into account object rotations when developing a 3D object detector [RS16, SZAB17], but has not been explored by deeply learned methods. Since we are discussing about indoor scenes, the rotations are invariably around the gravity axis. Recently, [YWY22] propose a rotationally equivariant 3D object detector, which is able to detect bounding box that are equivariant to the object pose. But even this work accounts for object rotations along the gravity axis alone. Theoretically, it may be easy to extend the framework in [YWY22] to SO(3) rotations, but many fundamental issues may need to be solved in practice. Developing a 3D object detector that is equivariant to SO(3) rotations invariably begs the question:“Are there robust shape descriptors that are rotation-equivariant?".

5.2. 3D semantic scene segmentation

A more in-depth understanding of indoor scenes, beyond 3D object detection, involves segmenting objects based on their semantics. An even comprehensive 3D indoor scene understanding pushes segmentation further, going into *instance* segmentation (not simply semantic). Indoor scenes contain a variable number of objects that occur in different positions and orientations. Moreover, some subscenes may contain identical sets of similar/identical objects, which can simply be represented by a single model. Determining the spatial extent of objects in an indoor scene (i.e., segmentation) in the presence of different categories with varying number of model instances, geometries and rotation distributions is quite challenging. This problem also is related to object detection, since for scene segmentation, the underlying models will need to implicitly reason about objects in a scene, allowing for fine-grained localization that result in segmenting the overall object.

Most works in the area of 3D indoor scene understanding restrict themselves to semantic segmentation, although there are some works that tackle the instance segmentation problem. Such works primarily build on the intuitions of semantic scene segmentation networks. As such, we cover notable works that propose methods

Related Work	Learning Framework	Scene rep	Backbone	Input	Output	Dataset	Evaluation Metric(s)
[QSMG17]	Supervised	RGB-D image	MLP	Scene point cloud	Per-point scores	S3DIS	Acc, mIoU
[QYSG17]	Supervised	RGB-D image	PointNet	Scene point cloud	Per-point scores	ScanNet	Acc, mIoU
[DN18]	Supervised	RGB-D image	2D, 3D CNN	Multi-view RGB images and voxel grid	Per-voxel scores	ScanNet	Acc
[LBS*18]	Supervised	RGB-D image	CNN	Scene point cloud	Per-point scores	ScanNet, S3DIS	Acc, mIoU
[TQD*19]	Supervised	RGB-D image	PointNet (MLP)	Scene point cloud	Per-point scores	ScanNet, S3DIS	Acc, mIoU
[WSL*19]	Supervised	RGB-D image	Message Passing MLP	Scene point cloud, point graph	Per-point scores	S3DIS	Acc, mIoU
[LMTG19]	Supervised	RGB-D image	MLP (GCN)	Scene point cloud, point graph	Per-point scores	S3DIS	Acc, mIoU
[ZJFJ19]	Supervised	RGB-D image	MLP (GCN)	Scene point cloud	Per-point scores	ScanNet, S3DIS	Acc, mIoU
[ZFF*21]	Supervised	RGB-D image	MLP (GCN)	Scene point cloud	Per-point scores	S3DIS	Acc, mIoU
[ZJJ*21]	Supervised	RGB-D image	MLP, Transformer	Scene point cloud	Per-point scores	S3DIS	Acc, mIoU, mean classwise acc

Table 3: *For the task of **3D indoor semantic scene segmentation** (3D-SSG), we summarize state-of-the-art methods in the table above–* Input *and* Output *refer to the input consumed by the* Backbone *and its output, respectively. Acc - Accuracy, AP - Average Precision, mAP - mean of Average precision, AR - Average Recall, IoU - Intersection over Union, RMSE - Root of Mean Squared Error.*

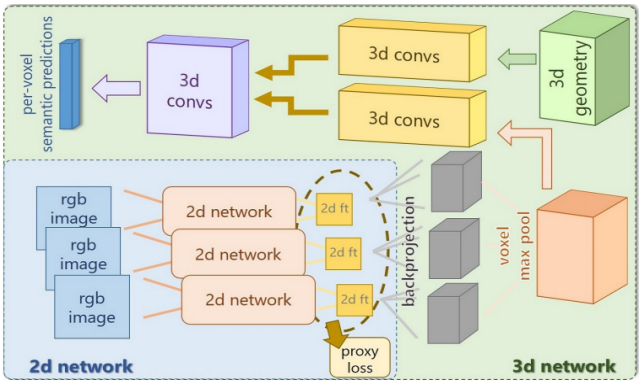

Figure 9: *Network architecture for semantic scene segmentation as proposed in [DN18]. It is composed of a 2D network and a 3D one. A 2D CNN extracts features from aligned images of a scene for which a geometric reconstruction is also performed from RGB-D scans. These 2D CNN features are mapped to 3D space using a differentiable back-projection layer. Features from multiple views are max-pooled on a per-voxel basis and fed into a stream of 3D convolutions, along with the reconstructed 3D geometry. Finally, both 3D streams are joined and the 3D per-voxel labels are predicted. The whole network is trained in an end-to-end fashion.*

for semantic segmentation, all of which use point clouds as the choice of scene representation. Table 3 lists notable works on 3D semantic segmentation in indoor scenes.

Point cloud-based techniques. Qi et al. [QSMG17] propose the very first point-cloud processing network, PointNet, that can be used for 3D semantic segmentation on shapes and scenes. It has a unified framework (supervised) for object classification and segmentation. By design, the semantic segmentation network of PointNet is an extension of the classification network which takes n points as input, applies transformations on the input as well as intermediate feature, and then aggregates point features by max pooling; see Figure 8 for an overview. Experiments were performed on the Stanford 3D semantic parsing data which contains 3D scans of indoor environment with semantic annotations per point. During training, random crops with 4096 points coming from a room in the dataset is passed through the network which learns to assign a semantic label to each point (segmentation is essentially a classification task on each point). During test time, all points forming a room are input to the system and semantic labels are obtained.

PointNet++ proposed by Qi et al. [QYSG17] builds on top of PointNet to further improve semantic scene segmentation. They propose a hierarchical feature learning architecture that uses PointNet as a feature processing block. While PointNet uses a single max pooling operation to aggregate the whole point set, PointNet++ builds a hierarchical grouping of points and progressively abstract larger and larger local regions along the hierarchy. This leads to better semantic scene segmentation. However, the max pooling operation used to aggregate features in local neighborhood regions often causes loss of information. PointCNN [LBS*18] proposed an χ-conv operator to adapt the convolutional networks for point clouds. Specifically, to aggregate the local region feature of each point spanned by its K nearest neighbors, the network predicts an χ-transformation matrix to weight and permute the per-point features, which is then processed using element-wise product and sum operations present in a typical convolution operation. In continuation with point convolutions, Thomas et al. [TQD*19] introduce *KPConv*, a convolution operation that operates on point clouds, taking radius neighborhoods as inputs and processing them with weights spatially located by a small set of kernel points. A deformable version of this convolution operator is also proposed that learns local shifts, effectively deforming the convolution kernels to make them fit the point cloud geometry. They demonstrated an improved performance on semantic segmentation which could be attributed to the flexibility offered by deformable KPConv.

Wang et al. [WSL*19] propose a supervised dynamic graph convolutional neural network (DGCNN) that uses PointNet as the backbone network, and demonstrate the application of their proposed method for semantic scene segmentation. The key idea is to *compute* point graphs at every layer and applying *edge convolutions* that are invariant to neighbor ordering. Point graphs are computed using k-nearest neighbor (based on L_2 distance) between points. The dynamic nature of graph computation at every layer of the graph convolution network enables them to capture better local and global features, leading to improved semantic scene segmentation results. Li et al. [LMTG19] propose a dilated version of graph

convolution neural networks, which enables them to capture global features better, which is demonstrated by the results of semantic scene segmentation.

In contrast to these sparse graphs connecting the center point and its neighbors, Zhao et al propose PointWeb [ZJFJ19], which builds dense, fully connected graphs in local regions and processes them using the Adaptive Feature Adjustment (AFA) module. This module predicts the impact of neighborhood points by adaptively aggregating contextual information on graph edges. Further, Zhou et al [ZFF*21] propose an adaptive graph convolution (AdaptConv) that generates adaptive kernels for points within a local region, rather than weighting the features based on fixed/isotropic kernels. On the other hand, Cheng et al [CHXY21] propose SSPC-Net, a semi-supervised method for 3D point cloud segmentation, which partitions the point clouds into super-point graphs, then dynamically propagates information from the super-point labels and uses the coupled attention mechanism to enhance the super-point features for more accurate segmentation.

In 3D-SIS [HDN19], instance segmentation on 3D scans is performed by learning from both color and geometry input obtained from real RGB-D scans. Specifically, the proposed learning framework has two branches – one that uses color images corresponding to reconstructed scan geometry, and the second one uses 3D point cloud reconstruction, either chunks of an indoor scene or a full one, from many different frames of RGB-D scan. The backbone for the first branch is a 2D CNN that extracts meaningful color features, which are brought to 3D using a differentiable back projection layer. The second branch uses a 3D CNN to obtain geometry features. Both the color and geometry features are joined in 3D.
In order to obtain object masks, they need to be localized. To this end, the work proposes a region proposal network to predict object bounding boxes. From these box predictions, class labels are predicted using a classification head, which are both used for informing instance mask predictions. Note that this is a strongly supervised approach. During inference, instances can be inferred on a full test scene in a single forward pass.

More recently, Zhao et al. [ZJJ*21] present a transformer-based architecture that serves as the backbone for many recognition and segmentation tasks on 3D point clouds, including semantic scene segmentation. The key insight driving this work is that the self-attention operator at the core of a transformer is essentially a set operator, and point clouds are essentially sets embedded in metric space. Much like [VSP*17], their method, called Point Transformer, makes use of encoder and decoder branches which are stacked layers of what they call, a point transformer layer, which is roughly, a piece-wise summation and aggregation of outputs from two different linear layers and an MLP, all of which take points from the point cloud as input; see figure 10 and 11. Although a supervised learning framework like all other works discussed till now, the gain comes from self-attention mechanism that learns correlation between points in the input point cloud, outperforming state-of-the-art designs including graph-based models, sparse convolutional networks, and continuous convolutional networks.

Voxel-based techniques. Dai et al. [DN18] propose a supervised multi-view prediction approach for semantic scene segmentation.

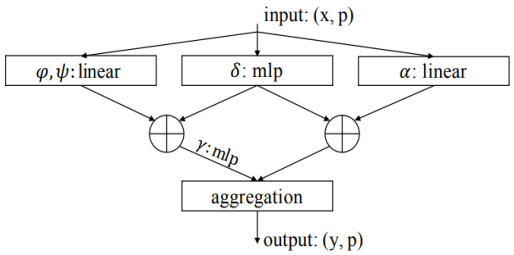

Figure 10: Point transformer layer *from [ZJJ*21]. This module is essentially responsible for extracting meaningful features, which are obtained from a combination of three neural sub-modules as depicted above.*

The goal of their method is to infer semantic class labels on per-voxel level of the grid of a 3D reconstruction. To achieve this, they propose a 2D-3D neural network that leverages both RGB and geometric information obtained from 3D scans. Their method takes as input a reconstruction of an RGB-D scan along with its color images, and predicts a 3D semantic segmentation in the form of per-voxel labels; see Figure 9 for an overview. To allow for 2D features to influence the per-voxel semantic predictions, they combine the 2D features (2D convolution on multi-view RGB images centered around a voxel location in *xy* plane) with the 3D ones (3D convolutions on volumetric chunks of a scene centered around a point in the *xy* plane) using a differentiable back-projection layer. This joint 2D-3D network is trained in an end-to-end fashion to predict per-voxel classes, resulting in a semantically segmented scene.

Discussion With the introduction of PointNet [QSMG17], there has been an explosion of related works that attempt to solve standard shape+scene understanding tasks – object classification and part/semantic scene segmentation. It is observed that architectures that are built for the task of semantic scene segmentation are all based on features meant to be used for classification purposes. In recent years, sophisticated architectures have made inroads that are based on point transformer layers [ZJJ*21]. All in all, backbone architectures for semantic scene segmentation seem to be pretty matured, with recent efforts being invested in engineering these networks for slightly better performance.

A more exciting research direction, one that is directly an extension of semantic segmentation, is *instance segmentation* in 3D scenes. This area allows for fine-grained understanding of scenes since we often encounter sets of identical objects in a scene. For example, a set of chairs surrounding a dining table or a set of place settings on the table. With a wide disparity in the number of instances observed, the distribution over observed rotations, and the geometric variations among instances per model within a category, the challenges are galore. Developing advanced techniques in this direction provides a deeper insight for scene understanding tasks.

5.3. 3D scene reconstruction

Single-view reconstruction is a severely ill-posed problem, mainly due to the lack of sufficient priors for obtaining a faithful reconstruction. Inferring a 3D structure, either for an object or a scene,

Related Work	Learning framework	Scene rep	Backbone	Input	Output	Dataset	Evaluation Metrics
[ISS17]	Unsupervised	RGB image	2D CNN	2D image and 3D CAD models	CAD models with placements, 3D Room layout	ShapeNet, LSUN, SUN RGBD	Classification Acc, mAP, voxel IoU
[HQX*18]	Supervised	RGB image	2D CNN	2D image	3D OBB for object, room layout	SUN RGB-D	mAP and 2D IoU
[NHG*20]	Supervised	RGB image	2D CNN with attention	2D image	Room layout OBB, 3D objects with OBBs	SUN RGBD, Pix3D	3D IoU
[ZCZ*21]	Supervised	RGB image	2D CNN with attention, GCN	2D image	Room layout OBB, 3D objects with OBBs	SUN RGBD, Pix3D	3D IoU
[MvAB*20]	Supervised	RGB-D image	2D and 3D CNN	Video/2D image frames	3D mesh	ScanNet	AP, AR, F-score, RMSE
[AKC*20]	Supervised	RGB-D image	2D CNN, GNN	Scene point cloud, CAD models	Room layout box, CAD models with placements	SUNCG, ScanNet	3D IoU, F1 score
[GRJ22]	Unsupervised	RGB image	2D CNN	RGB image	3D objects and spatial placement	Scene-ShapeNet, HyperSim, ScanNet	2D Box and Mask IoU

Table 4: *Notable works on **3D indoor scene reconstruction** –* Input *and* Output *refer to the input consumed by the* Backbone *and its output, respectively. We focus on object layout reconstruction over room layouts alone and not on room layout reconstruction. Abbreviations used for evaluation metrics: Acc - Accuracy, AP - Average Precision, mAP - mean of Average precision, AR - Average Recall, IoU - Intersection over Union, RMSE - Root of Mean Squared Error.*

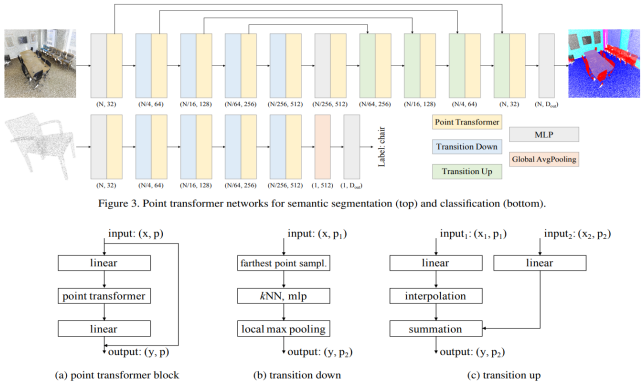

Figure 3. Point transformer networks for semantic segmentation (top) and classification (bottom).

(a) point transformer block (b) transition down (c) transition up

Figure 11: *Using the point transformer layer from 10 as the backbone, [ZJJ*21] propose an architecture for point cloud processing tasks such as shape classification, part segmentation and semantic scene segmentation.*

from an input image is a complex process that combines low-level images cues to learn the structural arrangement of parts/objects and the high-level semantic object/scene information. Variation in the views and shading, along with variation in textures make conventional reconstruction algorithms fail. For indoor 3D scenes, the focus is more on reconstructing object arrangements than the objects themselves. The challenge here is the to reason about 3D positions from a single image, while leveraging contextual information about object arrangements reflected in the input image.

In general, indoor 3D scene reconstruction can be categorized into two parts: room layout reconstruction and object layout reconstruction. Room layout reconstruction deals with recovering the spatial layout of walls of a room, whereas object layout reconstruction involves recovering the spatial arrangement of 3D objects. We limit the scope to reconstructing object arrangements in this section (there is a rich literature on reconstructing the room layouts alone). The input representations for 3D scene reconstruction tasks span the entire spectrum of visual representations, the most prominent ones being RGB (D) images and point clouds. We cover 3D object layout reconstruction works from a single RGB image [ISS17, NHG*20, ZCZ*21, HQX*18, HQZ*18, DT20], or a set of posed RGB images [MvAB*20], or a point cloud scan of an indoor scene [SYZ*17], as summarized below and in Table 4.

Image-based techniques. Izadania et al. [ISS17] propose a method for reconstructing a 3D scene of an RGB image (see Figure 12). They make use of a pre-trained 2D object detector (Faster-RCNN [RHGS15]) to detect objects in the input RGB image, and compare the box features of these 2D detections with that obtained from multi-view renderings of a database of CAD models (ShapeNet). This comparison enables them to retrieve an approximately aligned CAD model for a detected object in the input image. This process is done for all the object categories in consideration (eight, to be precise). In parallel, a fully convolutional network (FCN) [LSD15] is trained for room layout estimation that estimates per-pixel surface labels for ceiling, floor, and walls. The FCN network is trained on annotated scenes from the LSUN database [YSZ*15].

In order to find the object location and scale in the x and y directions (i.e., parallel to ground plane) in the scene, a ray is cast from a camera center through the input image pixels corresponding to the bottom four corners of an aligned CAD model cube (note that all CAD models in ShapeNet dataset are confined within a unit cube). To compute the object scale along the z axis, they compute the ratio between the length of the four vertical edges of the projected cube and the length of those edges from the ground plane to the intersection of those lines with the horizontal vanishing line. After estimating the 3D room geometry and the initial placement of the objects in the scene, object placements are refined by optimizing the visual similarity of the rendered scene with that of the input image. To this end, they solve an optimization problem where the variables are the 3D object configurations in the scene and the objective function is the minimization of the cosine distance between the convolutional features obtained from the camera view rendered scene and the input image.

Huang et al. [HQX*18] use a strongly supervised approach for

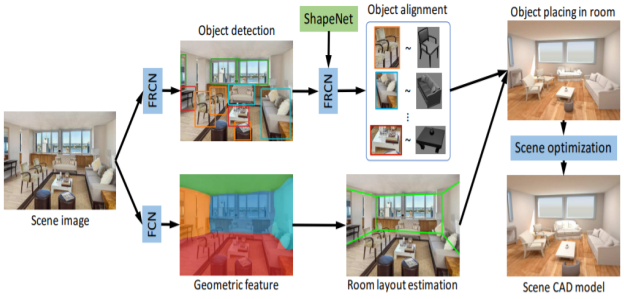

Figure 12: IM2CAD *[ISS17] proposes a system to reconstruct a CAD modeled scene based on a single input image. The main idea is to first render CAD models from different viewpoints and match their CNN features to that of detected objects in the input image. Once CAD models are retrieved corresponding to objects in the input image, an optimization algorithm modifies places them in a scene to reflect object arrangement in the input image. Color and lighting are injected using an additional module.*

reconstructing a 3D scene for a given RGB image, in what they call, a "cooperative" manner. Instead of developing independent modules for reconstructing parts of the scene, they propose to cooperatively estimate 3D object bounding boxes, layout bounding box and the camera pose, and project the resulting 3D layout to image plane forcing consistency between the input image and the projected image, see Figure 14. Specifically, the following three cooperative losses are used: 3D box loss (*directly* optimize final estimation of the 3D boxes), 2D projection loss (maintain the *consistency* between 2D image and estimated 3D boxes) and physical loss (penalize the *physical* violations between 3D objects and 3D room layout). This kind of cooperation is shown to improve the estimation accuracy of 3D bounding boxes, and the physical plausibility of the overall scene. Supervisory signals are obtained from the SUNRGB-D dataset [SLX15].

Similar to [HQX*18], Nie et al. [NHG*20] propose a supervised method to jointly reconstruct room layout, object poses and meshes in an indoor scene from a single RGB image as the input, termed as *Total3DUnderstanding* (T3DU). Their approach consists of three parts: room layout estimation (in world coordinate system), 3D object detection (in camera coordinate system), and mesh generation (in object canonical system). The output of these three modules are embedded together in the reverse order (see Figure 13), to establish an end-to-end joint training mechanism. The 3D object detection module makes use of an attention mechanism to obtain contextual object features (called "Relational features") for a detected 2D object in the input image. The relational features are combined with the detected 2D object features (obtained using a ResNet), and the resulting features are regressed through an MLP to get 3D bounding boxes. Room layout estimation is done similarly, where the room bounding box parameters are regressed using the 3D object detector module. Finally, a mesh generation module based on AtlasNet [GFK*18] is employed to reconstruct 3D object meshes.

During inference, the generated meshes in the canonical system are transformed to the camera system for viewing object bounding boxes, which are in turn converted into world coordinate system for combined interpretation with the estimated room layout bounding box. This method of training in an end-to-end fashion produced improvements over [HQX*18] on room layout estimation and 3D object detection, and this gain can be attributed to the incorporation of relational features in 3D object detection and room layout estimation, that take scene context into account via the attention mechanism.

Building on T3DU, Zhang et al. [ZCZ*21] make improvements in the quality of generated object meshes, and reduce cases of object intersections observed in T3DU. In terms of network architecture, the method is split into two stages: initial estimation stage, and a refinement stage; see Figure 15. The initial estimation stage is the same as T3DU (Figure 13(a)), with the only difference being the employment of a network outputting a shape code based on local implicit fields [GCS*20], instead of using AtlasNet [GFK*18] for mesh generation. In the refinement stage, the scene is modeled as a graph to capture the scene context, where the node features are concatenation of features obtained from the three modules in the initial estimation step. The node features are updated using message passing, which are later decoded into residuals to refine the initial estimation. The refined poses are then incorporated with the object shapes decoded from shape code with LDIF [GCS*20] to get the final reconstruction of the whole scene. The results show an improvement over existing relevant works which can be attributed to: (a) an improved mesh generation network, (b) the incorporation of a loss term that penalises physical interaction of objects (in addition to other losses) and (c) modeling scenes as graphs that helps to better capture scene context and in turn, can effectively refine the initial estimates.

Different from the aforementioned approaches, Murez et al. [MvAB*20] present *Atlas*, an end-to-end 3D scene reconstruction approach from *posed images*. The method takes a calibrated monocular video as an input. Image features from each frame are extracted using a 2D convolutional neural network. These features are back-projected along rays into a 3D voxel volume using known camera intrinsics and extrinsics. Feature volumes are accumulated using a simple running average. After accumulation, a 3D convolutional neural network refines the features and regresses a truncated signed distance function (TSDF). The overall approach is shown in Figure 16. Finally, a mesh is extracted from the TSDF volume using marching cubes. Additionally, semantic labels can be predicted by adding a classification head to the 3D CNN. This approach demonstrated superior quality of reconstructions on challenging long-frame temporal sequences with unobserved geometry, despite not making use of any depth information.

A more recent work called USL [GRJ22] presents an approach to scene reconstruction without any layout supervision, albeit from multi-view images of a scene during training. Their proposed system models object shapes and scene layout to roughly mimic an input image during test time. During training, given two views of a scene, one being the input to the system and the other being the target, a 3D scene is produced for the input view based on the prior work of MeshRCNN [GMJ19]. This 3D scene is now rendered from a the view-point of the target image, and an optimization algo-

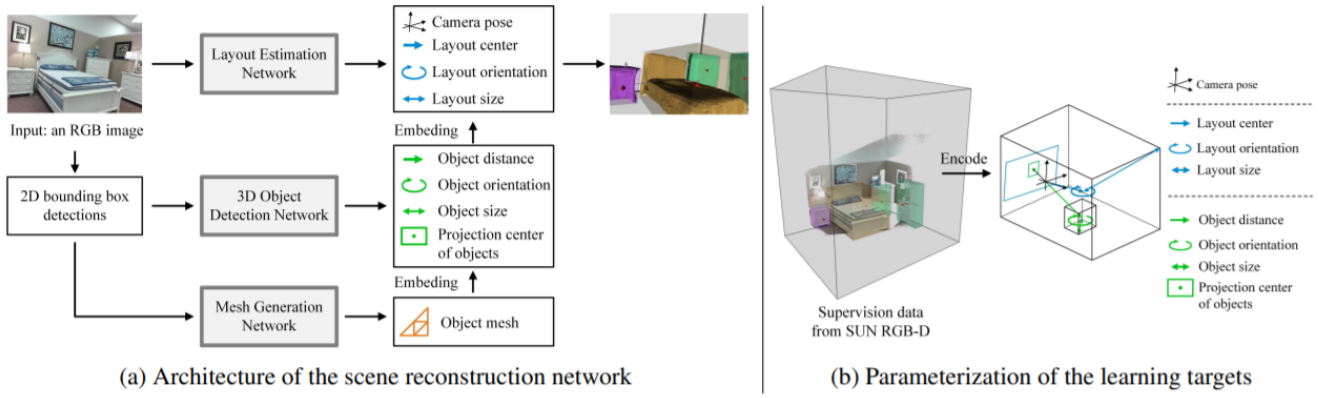

Figure 13: *Figure illustrates the pipeline for reconstructing a 3D scene from a single image in a supervised setting. The work is called* Total3DUnderstanding *[NHG*20], or* T3DU *in short. Relationships between detected objects in the input image are captured using attention mechanism. Both object and room layout are recovered and represented in the form of a cuboid box. Objects are reconstrcuted using a mesh generation network from [GFK*18].*

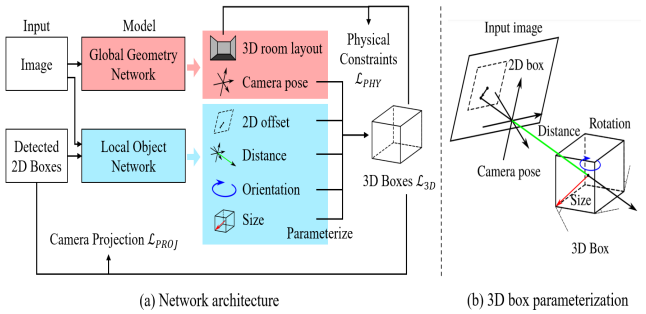

Figure 14: *Single-view scene reconstruction pipeline from [HQX*18]. The method proposes to cooperatively estimate 3D object bounding boxes, layout bounding box and the camera pose, and project the resulting 3D layout to image plane forcing consistency between the input image and the projected image.*

rithm compares object masks in the two. This helps in optimizing object arrangements thereby improving the scene layout.

Scan-based methods. In SceneCAD [AKC*20], a solution to scene reconstruction from RGB-D scans is proposed. The problem is broken down to that of aligning CAD models to objects in RGB-D scans. The prelude to SceneCAD is a prior work called Scan2CAD [ADD*19] that aims at estimating object arrangements from 3D scans by learning to align CAD models to RGB-D scans. The emphasis here is on the type of input – RGB-D scans, which tend to be very noisy and incomplete, with no semantic information. SceneCAD solves a joint problem of estimating both object and room layout information, by capturing relationships among objects and between objects and room elements, such as walls. The end result is a light-weight digitized representation for the input RGB-D scan. Note that this is also a supervised learning approach, with supervision on class labels during object mask predictions and edge labels between graph nodes. Many recent approaches to single-view scene reconstruction [KALD20, KALD21, GDN22] propose similar proxy solutions based on shape recovery of detected 2D objects in the input image via CAD model alignment.

Discussion 3D supervision is hard to obtain, which necessitates exploration of challenging tasks to scene reconstruction from one or more images. In order to truly reconstruct an indoor 3D scene, both room layout (walls and their arrangement), and object layout (spatial placements of objects in a room) should be recovered, which are non-trivial to solve. In 3D vision, a lot of effort has been put in recovering room layouts from a single image, starting from [HHF] to more recent works like [DFCS16, LBMR17, ZCSH18, HQZ*18, YWP*19]. Model-based room layout reconstruction algorithms such as the ones proposed by [HHF, ML15] are limited by hand-crafted features based on one/few properties of physical existence. Learning from data, however, can uncover the full potential of realizable results, allowing to explore further in that direction.

For object layout reconstruction from single/multiple images or 3D scans, two major approaches exist – (1) learning contextual placements of objects, or (2) aligning CAD models to the input. The first approach is mainly a supervised setting, based on attention mechanism [NHG*20] or message passing [ZCZ*21] since they need a notion of what plausible placements look like. In addition to recovering object layouts, these works also perform reconstruction at the object level, which may not be ideal for visualizing results since single-view reconstruction for objects is a research area in itself. A more recent work [GRJ22] does not make use of layout supervision, but gains additional training information by making use of *multi-view* images of the same scene, which compensates for the lack of layout-level supervision.

The second approach is an ad-hoc setting, moving towards unsupervision at the layout level, where the goal is to simply align CAD models from a database to detected objects in the input image [ISS17] or input 3D scans [AKC*20, ADN19, ADD*19, KALD20, KALD21, GDN22, MPNF22]. These approaches are more closer to works that estimate object poses from a single input image, with an added component of optimizing for object placements that reflect

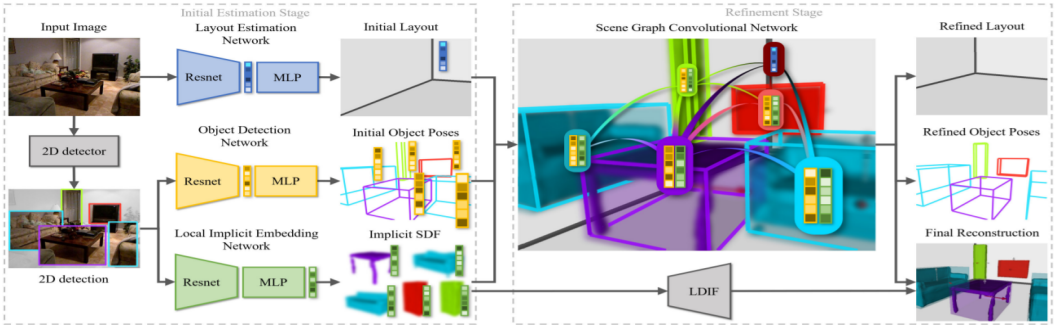

Figure 15: *Figure illustrates various modules present in reconstructing a 3D scene from a single image as proposed in [ZCZ*21]. It is a supervised learning framework. The work mainly builds on Total3DU [NHG*20], with the major difference being the use of implicit representations for objects in a scene. In addition, a graph neural network is employed on the scene objects, allowing to capture better contextual information. Room and object layout are represented using box cuboids, and objects are reconstructed using LDIF [GCS*20].*

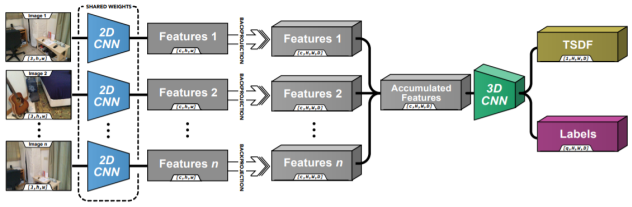

Figure 16: *End-to-end training pipeline for 3D scene reconstruction from posed images [MvAB*20]. The method takes a monocular video as input and extracts 2D CNN features from each frame, which are back-projected to 3D voxel grids using known camera information. A 3D CNN refines the features and regresses a TSDF function, from which a scene mesh is extracted using the marching cubes algorithm.*

the input image/scan. Overall, an unsupervised learning framework that could be trained from a single input image is missing.

5.4. 3D scene similarity

In visual computing, a metric is used to compare different data representations such as two images, meshes, voxels etc., and provide a measure of closeness or similarity between samples in consideration. As a result, they find applications in database retrieval, data clustering, and evaluating the diversity of generative models.

Similarity metrics for 3D shapes (Chamfer Distance, IoU, Light Field Distance etc.) and 2D images (L2 distance, PSNR) make an underlying assumption that the data being compared can be globally aligned. However, the concept of global alignment between two 3D scenes, even if they are of the same type, is rather weak since there is no "correct" sequence of populating 3D objects in a space to compose a plausible scene, where same objects can be placed quite differently in two scenes. In addition, scene comparison is complicated when the two scenes (of the same type) have different 3D objects, both semantically and geometrically. As such, developing a metric for comparing 3D scenes is quite challenging, but interesting at the same time due to many degrees of design freedom.

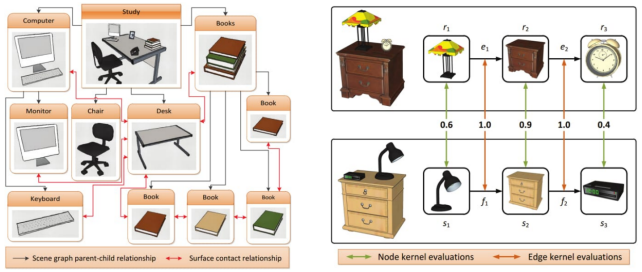

Figure 17: Graph Kernels *in [FSH11] is a method for characterizing the structural relationships between two subscenes. In the figure above, a scene is represented as a relational graph shown on the left. Two types of relationships are considered, as indicated by the arrows. The figure on the right shows an example of the process involved in finding similarity between two subscenes using graph walks. Both walks in each scene are rooted at the lamp node. The two walks are compared by taking the product of kernel evaluations for their constituent nodes and edges.*

Graph Kernels. Fisher et al. [FSH11] was the first work to develop a method to measure 3D scene similarity, called *Graph Kernels*. Specifically, they characterize 3D scenes using graphs (see Figure 17 left), where the edges of a graph encode physical proximity relationships (such as support, contact, enclosure) between objects (nodes) in a scene and the nodes correspond to objects in the scene, which encode the geometric information of the objects. With such graph-based representation of scenes, a kernel is defined for comparison of two relationship graphs in a way that similarities between the graph nodes and edges are computed and accumulated to produce an overall similarity of two graphs (see Figure 17 right).

Interaction descriptors. Zhao et al. [ZWK14] propose a scene relationship descriptor, called Interaction Bisector Surface (IBS), to characterize complex relationships in a scene, or rather, a sub-scene. IBS describes topological (wrapped in, linked to or tangled with) as well as spatially proximal relationships between objects (see Figure 18). IBS is defined as the set of points that are equidistant from two objects, which form an approximation

Figure 18: Interaction Bisector Surface (IBS) *[ZWK14] is a rich representation between objects in a scene that describes topological and geometric relationships between objects in a scene. IBS is the set of points equidistant from two sets of points sampled on different objects, shown as the blue colored surface above.*

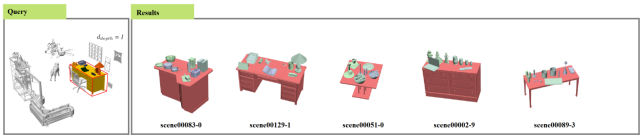

Figure 19: *Retrievals results using IBS features [ZWK14] – In the query scene on the left, a desk, overlayed using its bounding box, is the query object for IBS algorithm. On the right are the scenes ordered based on their similarity, where the red object is the retrieved object with a similar context to the query desk.*

of the Voronoi diagram for objects in the scene. IBS is used to define a similarity metric between objects, which is then used to group similar objects in a bottom-up manner to automatically construct hierarchies. Unlike [FSH11], IBS does not make use of object labels, and instead, focuses on modeling interaction between objects where spatial relationships between objects are characterized by topological and geometric features. Thus, IBS enables content-based relationship retrieval based on interaction similarity. Figure 19 provides an example of content-based relationship retrieval based on IBS.

Object-centric descriptor. Xu et al. [XMZ*14] present a method to organize a heterogeneous collection of scenes by what they call *focal points*. The key insight in this work is that analyzing complex and heterogeneous scenes in a collection is difficult without references to certain points of attention or focus. *Focal points* provide such points of attention against which two or more complex scenes could be compared. Specifically, *focal points* are defined as representative substructures in a scene collection, using which similarity distances between scenes could be computed, see Figure 20 for an illustration. Identifying focal points from a collection of scenes is a problem that is coupled with clustering scenes based on a common point of reference. To solve the coupled problems of focal point extraction and scene clustering, a co-analysis algorithm which interleaves frequent pattern mining and subspace clustering is presented to extract a set of contextual focal points that guide scene clustering from within the collection. This is shown in Figure 21. This co-analysis-based method of focal point extraction extends itself towards scene comparison (retrieval), and exploration of a heterogeneous collection of scenes.

Learning on scene graphs. Recently, [WDNT20] propose a neural network that infers a semantic scene graph from an instance-segmentation of a 3D scene, represented as point cloud. Leveraging these *learned* semantic scene graphs, a 3D scene retrieval is performed where the objects in the scene represent the nodes in the learned graph, and edges represent generic connection as well as semantic relations (such as: next to, lying on, close by) between scene objects. This is achieved by graph matching, not using any neural network, but using a deterministic similarity function based on two types of metric – Jaccard coefficent and Szymkiewicz-Simpson (SS) coefficient. When matching two graphs G and $G^{'}$, they combine the similarity metric of the object semantics, generic node edges E as well as semantic relationships R.

Since the semantic graphs are rich with relational semantics between objects, and the similarity function based on either Jaccard coefficient or Szymkiewicz-Simpson coefficient can provide meaningful similarity scores, especially SS coefficient when the two scenes A and B have very different sizes, one can use this retrieval method to find rooms that fullfill certain requirements such as the availability of objects e.g. meeting room with a TV, whiteboard.

Discussion Efficient retrieval of 3D scenes is helpful in visualizing interior design possibilities. The query to such systems can be in different forms – 2D image, text input, a sketch, a scene graph, a sequence of attributes etc., each of which poses unique challenges. The central goal to this problem is to define a similarity function that captures the space of scenes based on either pre-defined properties (such as focal-centric themes) or incorporates as much attributes of a scene as possible directly from the data.

As discussed above, graph neural networks (GNN) appear to sit comfortably in becoming the preferred tool to tackle 3D scene retrieval problems. In recent years, *neural* graph matching networks [PLF*21] have been shown to effectively capture similarity on 2D layouts, albeit with inherent sluggishness due to dependent graph embeddings (embedding of one graph is equally governed by the other graph in a pair). This could be extended to 3D scenes, where the focus should be on efficiently matching scene graphs in a large database, using hashing techniques.

6. 3D scene synthesis

Real world 3D scenes are realized from sequential placement and adjustment of objects carried out in a region-bounded space. Such object placements follow certain interior design rules based on room functionality and layout, which provide useful priors for developing algorithms for modeling indoor 3D scenes.

Before deep learning made inroads in this field [WSCR18, LPX*19], many scene synthesis works [YYT*11, MSL*11, FRS*12, FSL*15, MLZ*16, MPF*18, SCH*16, YYT15] were model-driven and learned from a few hundred 3D scenes. They were progressive in nature (vs. *auto-regressive* terminology used with neural-based works), i.e., the placement of a new object in the scene is conditioned on either one or a set of already existing objects in the scene thus far. These methods are example-based approaches, i.e., the underlying method for scene modeling depends on a set of scene/sub-scene examples to learn priors from.

We cover notable works on scene synthesis that incorporate different forms of representation discussed in Sec 3. We borrow some pointers from the survey on generative models for structured scenes [CRW*20] for our report, albeit it is less-up-to-date and

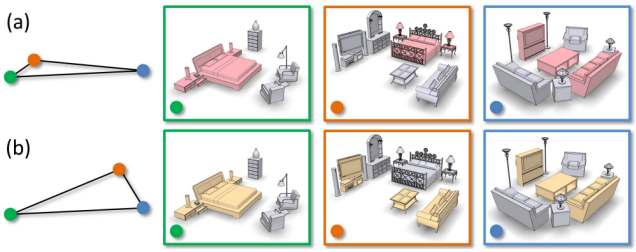

Figure 20: *[XMZ*14] introduce* focal points*, shown in yellow and pink color in scene renderings above, to analyze and organize 3D indoor scenes. Focal points are essentially sub-scenes. The triangles on the left provide a visual illustration for similarity distances between scenes based on focal points.*

focuses only on structural representation.

Probabilistic synthesis Relying on probabilistic reasoning over scene exemplars forms the core of example-based scene synthesis approaches [MSL*11, FRS*12, JLS12, SCH*16, ZHG*16]. Notably, Fisher et al. [FRS*12] develop a Bayesian network for object co-occurrences and model object placements using a Gaussian mixture model (GMM). To synthesize new scenes from an example scene, object contextual placements are sampled from the learned GMM model. Most of the probabilistic scene synthesis algorithms follow this paradigm, but with variations on both heuristics and models capturing object co-occurrences and their relationships.

Progressive synthesis Models that synthesize a scene by sequential placements of an object or a set of objects (called subscene) are said to be progressive in nature. This is a reflection of how scenes evolve in the real world – based on human actions, which in turn, depend on the functionality of objects present in the scene. This forms the basis of most human-centric scene synthesis works such as [MLZ*16, SCH*16], discussed later below. In other words, progressive scene synthesis involves user input in some form, be it activity-driven or language-driven [MPF*18]. Such methods can even be made interactive offering more controllability, where the overall system is localized at every synthesis step.

First instance of such progressive synthesis via interactive modeling was demonstrated by Merrell et al. [MSL*11]. They developed a modeling tool for furniture layout arrangement based on interior design guidelines. The design guidelines are encoded as terms in a probability density function and the suggested layouts are generated by conditional sampling of this function. Another related work called *Make It Home* [YYT*11] offers an interactive modeling tool to synthesize furniture layouts by optimizing a layout function which encodes spatial relationships between furniture objects. *ClutterPalette* [YYT15] presents another interactive modeling tool that progressively populates a scene by suggesting a set of possible objects, the priors of which are learned from the data, when a user clicks on a particular region of the scene.

More recently, Ma et al. [MPF*18] use language commands to drive scene synthesis. Language commands are parsed into scene

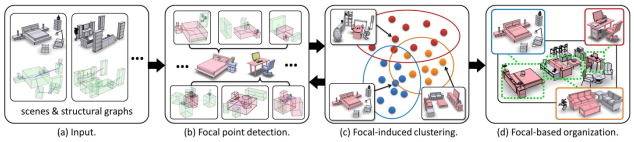

Figure 21: *In* focal-cenric graph kernel (FCGK) *[XMZ*14], the input to the system is a non-uniform collection of 3D scenes, where each scene is represented by a structural graph (left). The proposed method performs a co-analysis on the collection of 3D scenes to obtain a set of contextual focal points and is an iterative process (middle two). Once focals are obtained, the entire scene collection can be organized with reference to these focals, serving as the interlinks between scenes from various clusters (right).*

graphs, which are used to retrieve subscenes from a scene database. The key idea leveraged by this method is that semantic scene graphs act as a bridge between language commands and scene arrangements, and as such, aligning the scene graph from one domain will allow to retrieve corresponding scenes. To account for the lack of an exact match, the system also allows to augment retrieved subscenes with additional objects based on the language context. At each step, a 3D scene is synthesized by coalescing the retrieved sub-scene, with augmented objects, into the current scene.

Human-centric scene synthesis Fisher et al. [FSL*15] present a method that can produce multiple plausible 3D scenes from an input RGBD scan. here, plausibility means that the synthesized 3D scenes allow for the same functional activities as the captured environment. A scene template is estimated based on the input scan that captures likely human activities (as a probabilistic map) over the scene space. The core model, called the activity model, encodes object distribution with respect to human activities, and would guide the synthesis process based on predicted activities in the scene template. In a slightly different setting, Savva et al. [SCH*16] capture human poses with object arrangements in the scene based on human activity. The underlying modeling is a probabilistic model. The functional relationships between humans and objects in the form of physical contacts and visual-attention linkages are represented using what they call Prototypical Interaction Graphs (in short, piGraphs). Joint probability distributions over human pose and object geometries are encoded in the PiGraphs and learned from data and in turn, the learned PiGraphs serve to guide the generation of interaction snapshots.

A concurrent work from Ma et al. [MLZ*16] guides the scene generation process from human activity. In contrast to the above two methods, observations of human-object interactions in this work come from 2D images. That is, the action models are learned from annotated photographs in the Microsoft COCO dataset, which makes the problem challenging since such 2D images do not contain object/human-pose designation. The key idea of the work is to formulate transition probabilities to account for a transition in a human activity. An action graph is constructed whose nodes correspond to actions and edges encode transition probabilities. Synthesizing a new scene would correspond to sampling from the model capturing action graph priors.

Deep Generative Models Wang et al. [WSCR18] present a deep

Related Work	Learning framework	Scene rep	Backbone	Input	Output	Dataset	Evaluation Metrics
[LPX*19]	Self-supervised	Hierarchy	RvNN-VAE (MLP)	Scene hierarchy with object labels, 3D bounding box, relative position between two siblings	Scene hierarchy	SUNCG	Perceptual studies, Object co-occurrence map
[WSCR18]	Self-supervised	2D image	2D CNN, MLP	C+6 channel image	$Pr^{(x,y,z,\theta)}(C)$	SUNCG	Perceptual studies
[WLW*19]	Self-supervised	2D image, Graph	2D CNN, GNN	Scene graph, C+6 channel image	$Pr^{(x,y,z,\theta)}(C)$	SUNCG	Perceptual studies, Real/synthetic classifi -cation accuracy
[ZYM*20]	Supervised	Top-view scene image, object matrix	2D CNN, linear layers	RGB image, $n\times(k+9)$ scene matrix	$n\times(k+9)$ scene matrix	SUNCG	Perceptual studies, Object co-occurrence map
[WYN20]	Self-supervised	2D floor image, *ordered* object set	2D CNN, Transformer	1-channel image + $O^{i}(c_i,s_i,r_i,t_i)$	$O^{j}(c_i,s_i,r_i,t_i)$	SUNCG	Perceptual studies, Next-object prediction accuracy
[YGZT21]	Supervised	RGB-D images	CNN-GAN	Collection of segmented depth images	Volumetric scene (voxels)	Structure3D, Matterport3D, ShapeNet	Perceptual studies, Object co-occurrence map
[PKS*21]	Self-supervised	2D floor image, *unordered* object set	2D CNN, Transformer	1-channel image + $O^{i}(c_i,s_i,r_i,t_i)$	$O^{j}(c_i,s_i,r_i,t_i)$	3D-FRONT	FID score, Category KL divergence, Real/synthetic scene classi -fication accuracy

Table 5: *A summary of **neural scene synthesis works** informing about the kind of learning framework employed (supervised* vs. *unsupervised), the representation of indoor scenes, the kind of machinery employed to process the incorporated scene representation, the input to and output of the neural network, the dataset used and metrics employed to evaluate the generative models.*

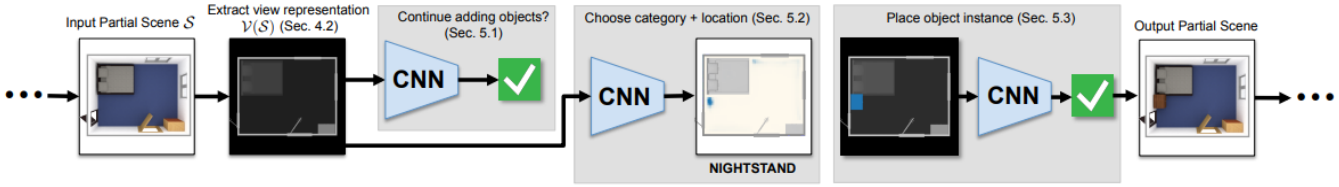

Figure 22: *Scene synthesis pipeline in deep convolutional priors [WSCR18]: given an input scene, its top-down view image is obtained that contains multiple features per pixels. Such feature-rich image is analyzed using a 2D CNN to determine if an object should be added to the current scene, and if so, what type of object category and at which location. Once the category and location are determined, an instance of that category is retrieved from a model database and added to the scene at an appropriate orientation.*

convolutional, autoregressive approach for 3D scene synthesis. A scene is represented as a multichannel top-view image where each channel encodes the mask of an object in the scene, in addition to depth information. An autoregressive neural network is then trained with such images (corresponding to 3D scenes) to output object placement priors as a 2D distribution map, see Figure 22. To synthesize a new scene, objects are sequentially placed based on the learned placement priors.

Moving towards a stronger structural representation, Wang et al. [WLW*19] present an autoregressive graph generative model called *PlanIT*, for 3D scenes based on Graph Neural networks employing message passing convolutions. They represent a 3D scene as a graph with scene objects as nodes and their spatial or semantic relationships by edges of a graph. During training, the network learns relationship priors between different kinds of objects in a scene type (ex: bedrooms). To generate a new scene from an empty or a partially complete scene (or scene graph), the learned autoregressive model is used to obtain a scene graph, which is instantiated via an image-based reasoning module to generate a 3D scene corresponding to that scene graph; see Figure 24 for an overview of this method.

Different from the above two methods, Li et al [LPX*19] present *GRAINS*, a generative neural network for 3D scenes that can efficiently generate a large quantity and variety of indoor scenes. Their key observation is that indoor scenes are inherently hierarchical (so they represent 3D scenes as hierarchies), and use a recursive neural network (RvNN) architecture coupled with a VAE to model the space of scenes following the pipeline shown in Figure 23. Using a dataset of annotated scene hierarchies, they train an RvNN-VAE, which performs scene object grouping during its encoding phase and scene generation during decoding.
Specifically, a set of encoders is recursively applied to group 3D objects (represented as semantically oriented bounding boxes) in a scene, bottom up, and encodes information about the objects and their relations, where the resulting fixed-length codes roughly follow a Gaussian distribution. To generate a new scene, a random vector is sampled from the learned Gaussian and branched down through the RvNN decoder to obtain the scene hierarchy. Shape models are retrieved from a shape database based on the semantics and dimensions of leaf nodes in the generated hierarchy.

Zhang et al. [ZYM*20] present a generative model for indoor

scenes based on a GAN, which learns to map a normal distribution to the distribution of primary objects in indoor scenes. In this work, a 3D scene is represented as a matrix that encodes all the information about every object in a scene. A scene is encoded into a latent vector by a set of interleaved sparse and fully connected layers. The decoder, which mirrors the encoder, generates scene matrices. A discriminator is trained to classify whether the input to it is a real scene or not. In addition, an image-based discriminator is also used to differentiate between the top-view renderings of 3D scenes; see Figure 25 for this method's overview.

Very recently, [WYN20, PKS*21] developed *conditional* generative models for 3D scenes by making use of attention-based Transformer models [VSP*17]. The advantage of using Transformer models is that they alleviate the need for hand-crafting spatial relationships between objects, and instead, implicitly learn object relations through attention mechanism. Specifically, Wang et al. [WYN20] condition the generation process on two kinds of inputs – room layout (including the position of doors and windows), and text descriptions. They represent indoor scenes as a sequence of object properties, converting the scene generation task to a sequence generation one. During training, an empty room (represented by the floor dimensions) or a text description (encoded using one of GloVe [PSM14], ELMo [PNI*18] or BERT [DCLT18] techniques) is input to their model along with a sequential ordering of object categories; see Figure 26 for an overall pipeline of their approach. The transformer model learns to sequentially generate the properties of the next object in the predefined ordered set. During inference, given the type of user input (empty room or text description), the trained model sequentially outputs an ordered set of objects and inserts them into the existing scene.

On the other hand, Paschalidou et al. [PKS*21] reduce the problem of scene generation to that of generating an unordered set of objects, where meaningful object arrangements are obtained by sequentially placing objects in a permutation-invariant fashion. They represent a scene as an unordered set of objects where each object is encoded using its cateory, size, orientation (relative to the floor normal) and location. During training, given a training scene with M objects, they randomly permute them and keep the first T objects (here T=3). The network is tasked to predict the next object to be added in the scene given the subset of kept objects, and the floor layout feature. During inference, they start with an empty context embedding $\mathbf{C}$ and the floor representation of the room to be populated. From here, they autoregressively sample attribute values from the predicted distributions; see Figure 27 for their method overview. Once a new object is generated, it is appended to the context $\mathbf{C}$ to be used in the next step of the generation process until the end symbol is generated. To transform the predicted labeled bounding boxes to 3D models, object retrieval from the dataset based on Euclidean distance of the bounding box dimensions is performed.

Another recent work from Yang et al. [YGZT21] developed a conditional volumetric generative model of indoor scenes using a GAN framework. They represent scenes as voxels, and take the room size as a conditional input to a GAN that is trained to map the distribution of indoor scene to a normal distribution. The discriminator is trained on depth and semantic images of the volumetric scenes. To this end, they employ a differentiable renderer to render depth and semantic maps of generated volumetric scenes, which are used with the depth, semantic maps of scenes from the training database for learning the GAN discriminator. At generation time, given a room size ϕ and a latent vector $\mathbf{zs}$ randomly sampled from the latent space, the trained volumetric GAN can generate a semantic scene volume that stores both layout and rough shapes of the objects instances in the room. To obtain the final 3D scene, they extract object instances from the semantic scene volume and replace them with the CAD models retrieved (based on Chamfer Distance) from a 3D object database.

Discussion With the availability of large synthetic 3D scene datasets such as 3D-FRONT [FCG*21], and the impressive advancements made in developing generative neural networks for 3D scenes, a basic question naturally arises – do we need more such generative models, and what purpose would more of such scenes serve anyway?

While it is surely worth having access to large quantities of synthetic and generated 3D scenes, they are not of practical use unless they are functional. That is, human activity should be adequately supported by these scenes. For example, if the area for in-and-out movement for a family of four in a generated living room is insufficient, then such a generated scene, although appearing plausible, does not find real-world applicability. This issue is systemic, in the sense that the typical way to scene synthesis has been to treat humans and scenes separately. We need to model them together, allowing us to generate functionally plausible environments, and in turn, using such scenes to improve human pose within that space and optimize activity. Scenes and humans complement each other.

In recent years, neural scene rendering [MST*21], diffusion models [SDWMG15, HJA20] and CLIP-based models [RKH*21] have gained a lot of traction for generating novel data samples. They have been predominantly employed in the 2D domain, with only recent exploration for 2D-to-3D generation [GSW*22, PJBM22] on single-object images. Leveraging these models for 3D indoor scenes is an under-explored direction mainly due to the complexity of the scene structures. Below, we present recent works that use these newer techniques for synthesizing indoor scenes.

Using Radiance Fields The basic idea in neural rendering is to first sample spatio-temporal coordinates with respect to a given 3D scene, feed it through a neural network to recover radiance/signed-distance fields and employ a differentiable forward map (such as sphere tracing or volume rendering) that outputs a novel view RGB image (see [XTS*22] for a detailed report). Using these radiance fields to hallucinate scene geometry or to manipulate object arrangements allows their employability for indoor scene modeling applications.

To this end, [DBS*21, YZD*21] present some of the early works on using radiance fields for scene hallucination/completion and scene editing, respectively. Yang et al. [YZD*21] develop a neural rendering system that enables editing on real-world scenes by learning an object-compositional neural radiance field. The main idea is to use two separate branches to encode the scene, one to process the scene background and the other to process the constituent

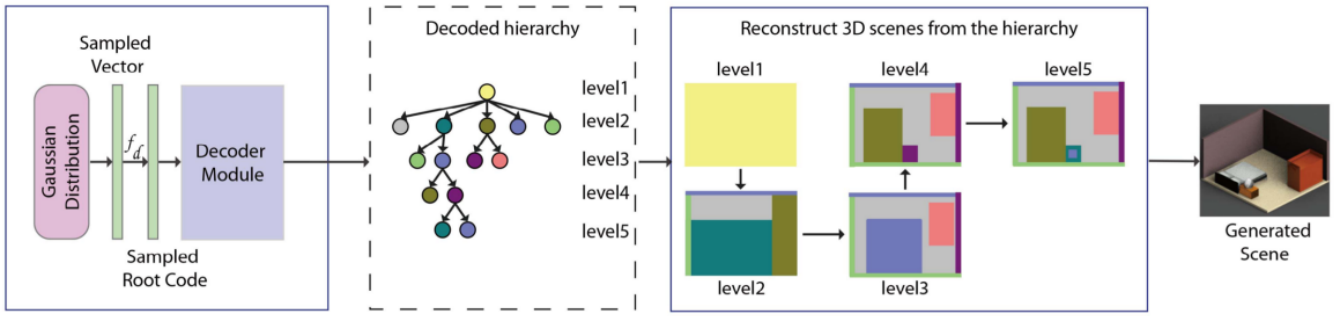

Figure 23: GRAINS *[LPX*19] represents a scene as a hierarchy based on commonly occurring object relations. The learning pipeline is based on recursive neural networks (RvNN) coupled with a variational autoencoder. At inference time, a random vector is sampled from the learned latent space (which is approximated to a Gaussian distribution), and passed through the trained RvNN decoder to obtain a scene hierarchy. A 3D scene is recovered from the decoded hierarchy in a top-down fashion until all the leaf nodes of the hierarchy are traced. 3D objects are placed by retrieving from a collection of CAD models present in the scene database based on their attributes generated.*

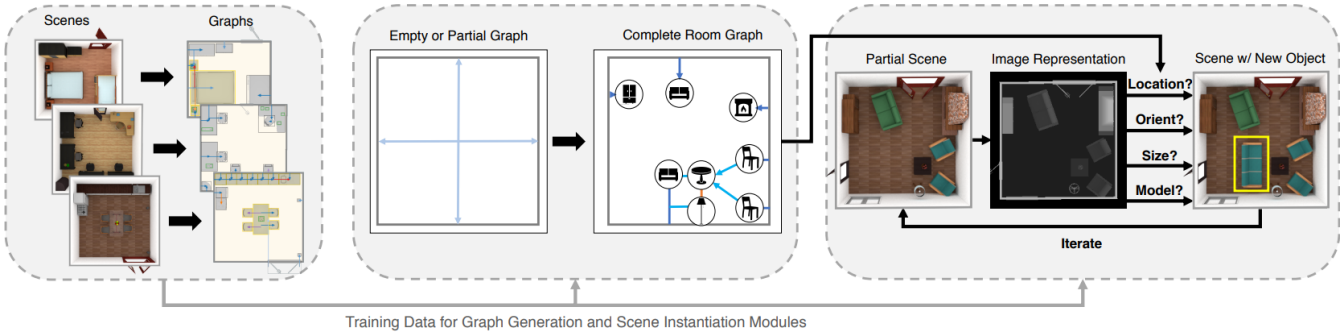

Figure 24: *Scene synthesis pipeline in* PlanIT *[WLW*19]: Given a database of 3D scenes, relational graphs are automatically extracted from them (left), which are fed to a graph neural network (middle). Eventually, to generate a 3D scene, a graph is instantiated using image-based reasoning (right) and 3D models corresponding to each node are inserted.*

objects, both undergoing a neural rendering pipeline of their own. In doing so, the system runs a neural radiance field pipeline on the object branch that takes as inputs object voxel features and an object activation code, allowing scene edits at the object level based on the object activation code.

DeVries et al. [DBS*21] make use of the radiance fields to learn scene priors from which novel scenes can be sampled. Specifically, they train a GAN in which the generator learns to decompose scenes into a collection of many *local* radiance fields that can be rendered from a free moving camera. The generator of this proposed GAN tries to learn a distribution of novel view images (using a NeRF model) that is similar to the prior distribution. The discriminator is tasked to classify the images at the output of generator to be fake. Once trained on many diverse scenes and view points, a novel scene sample can be generated using a random vector which can hallucinate parts of a scene captured in the training images.

[WLJ*22] propose NeuralRooms to reconstruct indoor scenes (represented by meshes) from unposed multi-view 2D images based on neural radiance fields. The main motivation of this work is that shape-radiance ambiguity and the presence of texture-less regions in indoor scenes make it difficult to faithfully reconstruct them using multi-view stereo (MVS) algorithms or using NeRF models. To address this issue, they propose a two-part learning framework wherein the first part makes use of a MVS algorithm [SF16] (ensuring accuracy of texture-rich and edge areas) and a normal estimation network [BBC21] (ensuring completeness of texture-less regions) to acquire geometry prior. The input RGB images and the geometry prior are used in a neural rendering-based surface reconstruction pipeline. Finally, ray-tracing on the reconstructed scene is done with TSDF fusion to obtain a 3D mesh for the reconstructed indoor scene.

Using Diffusion and CLIP-based models Unlike in the 2D domain, the deployment of diffusion models for modeling 3D indoor scenes has not seen much activity. To the best of our knowledge, the recent works of Lego-Net [WDP*23] and [LTJ22] are the only work devoted to this task.

Lego-Net [WDP*23] develops a denoising diffusion model to learn rearrangement of objects in a messy indoor scene, where, similar to ATISS, a scene is represented by an unordered set of objects and their transformations. The underlying model that is used is a transformer. Given an input messy scene, the transformer iteratively computes the denoising gradient towards the clean manifold, until a so-called regular state is reached; see Figure 28. Note that the denoising process is not driven by any end goal state, and the

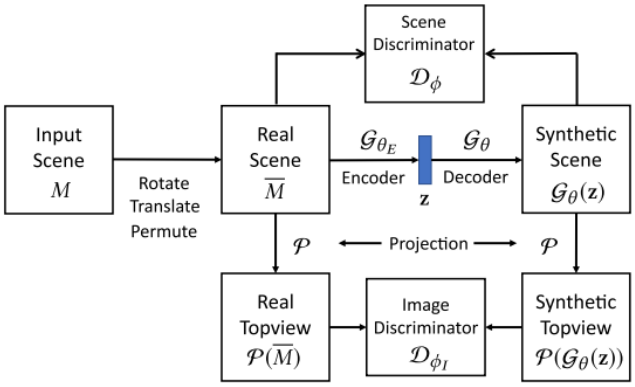

Figure 25: *Schematic pipeline for scene synthesis using hybrid scene representations [ZYM*20], where a 3D scene is represented by its top-view rendering as well as using a matrix of object properties based on their occurrences in the scene. Above, an encoder G_{θ_E} encodes a scene into a latent space, which the decoder G_θ uses to produce a scene data matrix. A scene discriminator D_ϕ then determines if the generated scenes are real. A projection layer P projects 3D scenes to top-view images and an image discriminator D_{ϕ_I} classifies if its top-view images are real are not.*

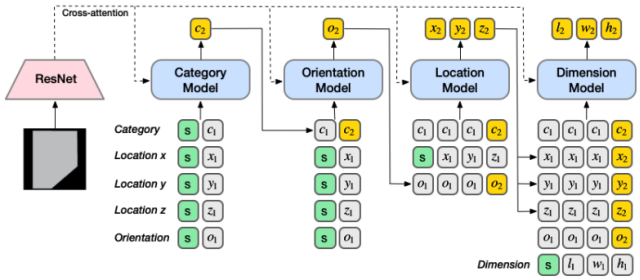

Figure 26: SceneFormer *[WYN20] uses a transformer to generate new scenes in an autoregressive manner, where a scene is represented by an ordered set of objects. In addition, it consumes a floor image, both during train and test time.*

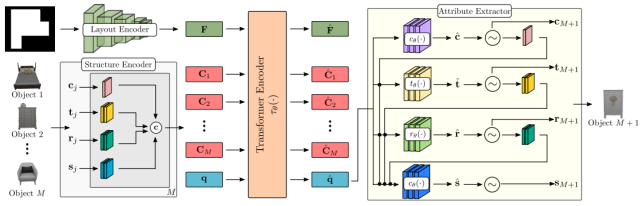

Figure 27: ATISS *[PKS*21] poses the scene synthesis task as one of unordered set generation. Given a room type and its shape (in the form of a top-down floor image), it generates plausible furniture arrangements in an autoregressive, permutation-invariant fashion using a transformer.*

concept of an end goal state is not made use of in this work. At each step of the denoising process, the transformer takes scene attributes of the current state and outputs 2D transformations of each object that would make the scene "cleaner". Training this model takes place in a reverse fashion where clean scenes are perturbed to make them messy.

Lei et al. [LTJ22], on the other hand, propose to synthesize an indoor scene via incremental inpainting. They represent indoor scenes as RGBD images. Given a sparse set of multi-view RGBD images, the goal is to generate a coherent 3D scene mesh by predicting RGBD frames along a *novel* camera trajectory. To this end, they firstly fuse the input view images into an initial mesh, which is then rendered to get an incomplete, hole-present image of the scene. This incomplete scene is then inpainted using a RGBD diffusion model which is back-projected to the 3D domain to get a 3D mesh. This mesh is integrated with the very initial mesh to get a new mesh. This process takes place in an iterative fashion, eventually generating a novel indoor scene.

CLIP [RKH*21] is a popular text-to-image generative model capable of producing novel 2D images from text prompts. In the 3D domain, CLIP has been used to generate 3D shapes as shown in [SCL*22, SFL*22]. Its extension to generating 3D indoor scenes in not straight forward owing to the presence of multiple objects. CLIP-Layout [LXJ*23] presents the first work that leverages a CLIP model to synthesize 3D indoor scenes.

Similar to ATISS [PKS*21], CLIP-Layout [LXJ*23] adopts an auto-regressive approach for indoor 3D scene synthesis, but the synthesis is now additionally constrained on a text input that acts as a style description prompt. This work also encodes a 3D scene as an unordered set of objects and their transformations, and the underlying machinery is based on a transformer model. The text prompt is encoded using a CLIP encoder [RKH*21], which takes eight view-renderings of a 3D scene as input and outputs a 512-dimensional vector, which is then concatenated with all other object-plus-floor features extracted using the transformer as done in ATISS. Note that a pre-trained text-to-image CLIP encoder is used during the training phase. This allows in representing fine visual details of each furniture instance while remaining agnostic to object encoding format. Sample results are shown in Figure 29.

7. Conclusions and open problems

In this report, we have surveyed historical and state-of-the-art works in data-driven analysis as well as synthesis of 3D indoor scenes. Starting from different possible representations (both visual and structural) and the available datasets, we discussed fundamental scene analysis tasks such as 3D object detection, 3D scene segmentation, 3D scene reconstruction and 3D scene retrieval. For synthesis techniques, we have mainly documented recent progress in that direction, which by default, has been skewed towards neural models. During the course of these discussions, we have identified the suitability of specific neural architecture for the chosen scene representations.

Overall, indoor scene modeling has made impressive strides in recent years, pushing the boundaries of computer graphics research. Yet, there exist many interesting avenues at a higher level to be pursued. We conclude this report by offering thoughts on what we regard as some important and interesting research directions.

Modeling rotation equivariance for 3D object detection: A

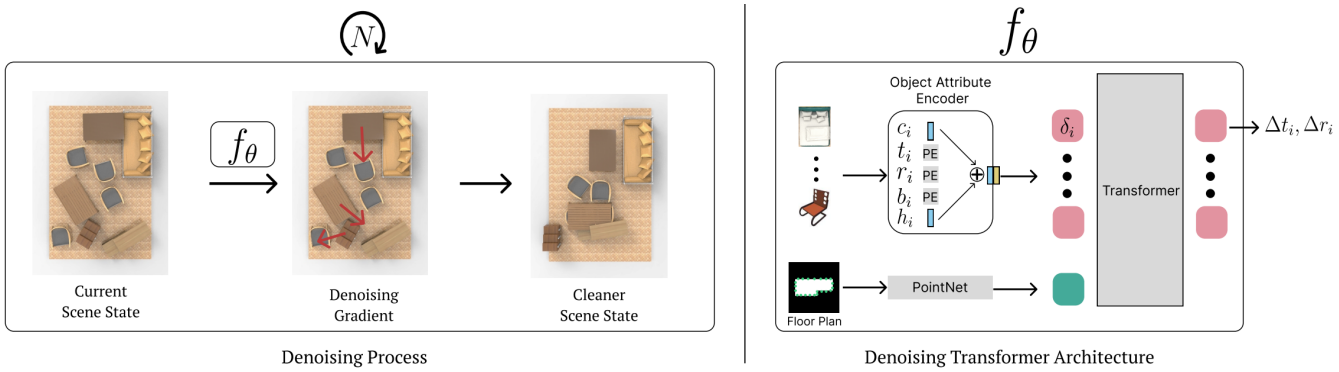

Figure 28: *Denoising process of LEGO-Net [WDP*23](left) and the backbone transformer (right) performing the denoising process. LEGO-Net takes current scene state and iteratively modifies the scene state by computing denoising gradient towards the clean manifold. The transformer on the right transformer takes the scene attributes of the current state and outputs 2D transformations of each object.*

Figure 29: *CLIP-Layout [LXJ*23] takes a floorplan and a text prompt as conditions for style descriptions, and generates plausible, diverse 3D indoor scenes that are stylistically consistent, adhering to the input text prompt.*

strongly desired property in 3D deep learning is rotation equivariance. In the context of object detection in 3D indoor scenes, it is desired that the detected bounding box be *equivariant* to the object pose. This means achieving equivariance not at the global input scene level, but at the object level. The first step towards this goal is to extract equivariant features at the object level, and finding a way to inject them into recent 3D object detectors. Since objects in a scene are rotated along the gravity axis, it is natural to limit development of techniques to 1D rotations. A recent work of [YWY22] has laid some groundwork in this regard. However, it is the equivariance in SO(3) space that is challenging and underexplored, finding wide industrial applicability such as in simulating flight aerodynamics, building constructions and in assembly lines in manufacturing industries.

Instance detection and segmentation in scenes: In the real world, sets of *identical* objects are observed in different orientations. In synthetic scenes, such identical objects are typically represented with instances of the *same* 3D model appropriately transformed into various positions. The ability to segment scenes, not simply semantically, but also at the instance-level, provides greater machine understanding of indoor environments. The solution to this problem raises a more fundamental question that is related to modeling object rotations in the SO(3) space – can existing shape descriptors distinguish between rotated instances of the same object, and if not, how can we go about doing that? In addition, if an agent can identify instances of same objects in a scene (through instance segmentation), motion articulations on one such model can be easily applied to all other instances of the model in the scene. These are all connected problems, but can only be attempted if the fundamental question posed above has been firmly answered.

Unsupervised scene reconstruction from a single image: Object reconstruction from single-view images is a difficult problem in itself. At the scene layout level, different challenges exist, particularly of cluttered scene recovery. The development of such models can facilitate visualizing possibilities for architects, furniture retailers and interior design firms. Recent supervised works [NHG*20,ZCZ*21] have achieved decent results on this task while also performing reconstruction at the object level. When no layout supervision is provided, the problem of recovering the scene layout from a single-view image becomes extremely challenging. A potential solution to this problem is via self-supervision, where the model needs to reason about placement priors from many homogeneous scene images. A lack of existing works in this direction indicate the degree of difficulty involved in the task, something to actively research about.

Neural 3D scene similarity: Alignment of visual data provides a reference point to study data similarity. For 3D scenes, this is notoriously hard, since there is no standard reference point – it all depends on the scene object in focus. Developing algorithms for 3D scene similarity has continued to attract interest, starting from graph kernels [FSH11] to later works on focal-centric graph kernels [XMZ*14] and IBS [ZWK14]. A recent work [PLF*21] uses graph neural network (GNN) to learn structural similarity for 2D layouts. To explore a deployment of GNNs with intuitive focal-centric approaches that combine not just the structural layout properties, but also object appearance traits, for 3D scene similarity/retrieval remains an interesting direction. Such a metric could also be used to qualitatively evaluate the plausibility of 3D scenes, which is a missing piece in the literature, especially in the realm of neural generative models.

Modeling geometry and object textures for scene synthesis: Access to a rich 3D scene database that contains diverse 3D objects is crucial for building neural models tasked for scene generation. In all such works, objects are placed in the generated scene layout by retrieving from an object database based on generated object attributes, which typically are nothing but bounding box dimensions, category and scene centroid. Although geometric infor-

mation is generated, appearance properties are not accounted for. This needs to be addressed since existing structural and geometric attributes provide strong cues to material and texture appearance, something that has been under-explored at the object level [JTRS12, LAK*18, CXY*15]. Learning to model a coupling of this with the scene layouts is an interesting approach to bypass the typical object retrieval step and directly generate objects with novel appearances.

Neural text-to-3D scene: With DALL-E [RDN*22] and IMAGEN [SCS*22], great advances have been made this year for language-driven image synthesis that produce impressive, high-quality results. However, these models do not offer control over the generated results. That is, the underlying theme is of a one-shot text-conditioned generation, something that is rarely desired when dealing with 3D content. Rather, the ability to progressively generate 3D content in controlled manner is prioritized in the graphics community. In the realm of indoor 3D scenes, a few attempts [CSM14, CFG*15, MPF*18] have been made in the past to generate scenes from text input. All these works are model-driven, and are limited by incorporated heuristics. Admitting neural networks for this task that learn directly from the data can make up for the "lost ground", and open up further avenues to improve such scene generation systems.

Modeling scene style: Scene style is a high-level concept, generally referring to a setting where the furniture designs (ornate antique sofa vs. a flat IKEA sofa) are in agreement with, and even complement, the ambient decorations in the scene (ex; ceiling arches, wall carvings and other beautification), and themselves exhibit geometric uniformity. In addition, room color, ambience lighting, furniture color, texture and material, all factor into what we call as scene style. To be able to generate scenes of different styles implies modeling a function of object styles and room decorations. A recent work [SHS*22] gives a teaser on scene style based on a person's mood. The question of how to define this function and how to factorize object/room style in terms of material, color and texture remains an open research problem.

In addition to such fine details, scene "style" can also refer to its tidiness. Real world scenes are often cluttered with objects and are far from the impeccable renderings depicted in the literature. A significant energy is devoted in producing near-perfect scenes, ignoring investigations into producing messy (sub)scenes that mimic environments encountered in daily life. Learning to model such scenes, although challenging, can be beneficial in providing realism to virtual scenes and remains an open research direction.

Modeling scene interaction: For an immersive metaverse experience, users should be able to interact with objects in a scene. This can occur in two ways – (1) using objects and furniture in their intended purposes (ex: sitting on a chair, adjusting the sofa to appropriately face the tv, using the tv remote, picking up a pen), and (2) playing with furniture models (ex: facilitating articulations on the drawers of a file cabinet, where the user can touch a drawer and it pops open; allowing part mobility such as adjusting the height of a chair seat; manipulating object functionalities in novel ways, perhaps making them non-functional). There have been some attempts in modeling object functionalities and part mobility [SHL*14, LHAZ, HZvK*15, HvKW*16, HLVK*17, HYZ*20]. However, modeling interactions at the scene level poses novel challenges and is an interesting research direction that has been under-explored.

Move planning, scene rearrangement and teleportation: The ability to move freely and efficiently in indoor environments influences productivity and work culture, either for shared workplaces such as in office, restaurant kitchens, warehouses, airports or for personal spaces such as living room and open kitchen. More often than not, such planning takes ample time and layout considerations, especially with a large object inventory. Suggesting a plausible arrangement of objects leading to an optimized workplan, as well as space design is extremely useful in industrial applications. In a recent work [ZHPY21], such workspaces and workplans are automatically designed given the input space and workspace equipment, in addition to staff properties as inputs. Such designs may benefit from data-driven modeling, for which, a diverse, rich, large-scale database of indoor environments spanning different industries is the first necessary step.

This move planning concept can be used to inject teleportation feature in metaverse, where a system can automatically suggest possible teleportation locations within an already seen environment based on navigability, and the user can hop between such spots virtually. One way to do this is to sample a set of desirable teleport positions, assessing ease of navigation (and properties such as coverage and connectivity from a subscene or a focal point). Such an application has recently been explored in [LHLY21] which synthesizes scene-aware teleportation graphs.

A slight deviation from the above, but with the potential to serve teleportation application, albeit slower, would be to allow the user to select teleportation spots apriori in an already configure environment, and developing a system that can re-arrange objects in the current physical state of the environment to a new state so as to optimize for the desired tele-movement. Different versions of this task exist have been described in [BCC*20] where the target environment state can be described by object poses, images, language description or by letting an agent experience the target state environment, if possible. A recent work [WLY20] makes use of reinforcement learning for automatic move planning of 3D objects from an initial 3D layout to a target layout. This application serves well in practice, and requires further exploration.

8. Author Bios

Akshay Gadi Patil is a senior Ph.D Candidate in the GrUVi lab at Simon Fraser University, Canada. His research focuses on understanding and modeling 3D shapes and scenes using machine learning techniques, which is strongly related to the theme of this report. With his collaborators, he developed the first hierarchical deep learning framework for scene synthesis, named GRAINS, and also authored a SIGGRAPH Asia paper on synthesizing 3D scenes using compact natural language. His work on 2D layouts, named READ, won the best paper award at this CVPR 2020 workshop. Before joining the PhD program, he earned a M.Tech degree in Electrical Engineering from the Indian Institute of Technology Gandhinagar. He has co-organized the Learning to Generate 3D Shapes and Scenes workshop at ECCV 2022.

Supriya Gadi Patil is a graduate student working in the GrUVi lab at Simon Fraser University, Canada. She focuses on incorporating deep learning techniques for geometric modeling of 3D shapes and scenes, mainly in the realm of reconstruction. Her expertise includes domain knowledge of various 3D indoor scene datasets and scene reconstruction methods, which nicely complements the content delivery of this survey report. She obtained a M.Tech degree in Computer Science from the Indian Institute of Technology Hyderabad working in the area of Graph Neural Networks, and was a student researcher at the Max Plank Institute for Software Systems (MPI-SWS), Germany. In the industry, she worked on integrating machine learning models into commercial products as a full-time employee at Adobe India.

Manyi Li is an Associate Researcher in the School of Software at Shandong University. She received B.Sc. and Ph.D. degrees from Shandong University in 2013 and 2018 respectively and was a Post-Doctoral research scholar in the GrUVi Lab at Simon Fraser University, from 2019–2021. Her main interests are in 3D content creation and understanding, with a special focus on 3D man-made objects and indoor scenes, which is highly correlated with this STAR report. With her collaborators, she authored the first hierarchical deep learning framework, named GRAINS, catered towards fast and diverse synthesis of 3D indoor scenes. She has also worked on developing unsupervised learning techniques for object understanding and reconstruction, which are relevant to object insertion in 3D indoor scenes. She was a co-organizer of the Learning to Generate 3D Shapes and Scenes workshop at CVPR 2021.

Matthew Fisher Matthew Fisher is a Principal Scientist at Adobe Research. He obtained his Ph.D. in Computer Science from Stanford University and a B.S. in CS from the California Institute of Technology. Matt's research focuses on combining computer graphics, vision, and machine learning to make it faster and more fun to complete creative tasks. He has published over 50 papers in graphics and vision, including many foundational papers in scene understanding and scene synthesis such as 3D Graph Kernel, SceneSynth, FunctionalSceneSynth and his PhD thesis was on data-driven tools for scene modeling. His recent work looks into applying deep generative models to help artists more quickly design and model 3D objects and scenes. At Adobe Research, he works extensively with creative professionals to build and deliver new tools that accelerate the artistic process across painting, animation, and video editing applications.

Manolis Savva is an Assistant Professor at Simon Fraser University, and a Canada Research Chair in Computer Graphics. He completed his PhD at the Stanford graphics lab, advised by Pat Hanrahan. His research focuses on human-centric 3D scene analysis, 3D scene generation, and simulation for scene understanding. He has also worked in data visualization, grounding of natural language to 3D content, and in creating large-scale datasets for 3D deep learning. He has published a number of papers in related areas including Plan2Scene, Habitat, and PiGraphs. He organized workshops on relevant topics such as the Embodied AI workshop (CVPR 2020, CVPR 2021) and Learning to See from 3D Data (ICCV 17). He has also taught a related course Learning 3D Functionality Representations at SIGGRAPH Asia 2020, and co-authored and presented a STAR on Functionality Representations and Applications for Shape Analysis at EuroGraphics 2018.

Hao (Richard) Zhang is a distinguished professor in the School of Computing Science at Simon Fraser University, Canada, and an Amazon Scholar. He obtained his Ph.D. from the Dynamic Graphics Project (DGP), University of Toronto, and M.Math. and B.Math degrees from the University of Waterloo, all in computer science. Richard's research is in computer graphics with special interests in geometric modeling, analysis and synthesis of 3D contents (e.g., shapes and indoor scenes), geometric deep learning, as well as computational design and fabrication. He has published more than 170 papers on these topics. Most relevant to this survey topic, Richard was one of the co-authors of the Eurographics STARs on Structure-Aware Shape Processing, Learning Generative Models of 3D Structures, and taught SIGGRAPH courses on closely related topics such as "modeling and remodeling 3D worlds." With his collaborators, he has made original and impactful contributions to structural analysis and synthesis of 3D shapes and environments including co-analysis, hierarchical modeling, semi-supervised learning, topology-varying shape correspondence and modeling, and deep generative models.

References

[ADD*19] AVETISYAN A., DAHNERT M., DAI A., SAVVA M., CHANG A. X., NIESSNER M.: Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition* (2019), pp. 2614–2623. 5, 6, 14

[ADN19] AVETISYAN A., DAI A., NIESSNER M.: End-to-end cad model retrieval and 9dof alignment in 3d scans. In *Proceedings of the IEEE/CVF International Conference on computer vision* (2019), pp. 2551–2560. 14

[AGSK20] AGGARWAL M., GUPTA H., SARKAR M., KRISHNAMURTHY B.: Form2seq: A framework for higher-order form structure extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2020), pp. 3830–3840. 4

[AKC*20] AVETISYAN A., KHANOVA T., CHOY C., DASH D., DAI A., NIESSNER M.: Scenecad: Predicting object alignments and layouts in rgb-d scans. In *European Conference on Computer Vision* (2020), Springer, pp. 596–612. 12, 14

[AW18] ALHASHIM I., WONKA P.: High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941* (2018). 3

[BBC21] BAE G., BUDVYTIS I., CIPOLLA R.: Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 13137–13146. 20

[BCC*20] BATRA D., CHANG A. X., CHERNOVA S., DAVISON A. J., DENG J., KOLTUN V., LEVINE S., MALIK J., MORDATCH I., MOTTAGHI R., ET AL.: Rearrangement: A challenge for embodied ai. *arXiv preprint arXiv:2011.01975* (2020). 23

[CDF*17] CHANG A., DAI A., FUNKHOUSER T., HALBER M., NIESSNER M., SAVVA M., SONG S., ZENG A., ZHANG Y.: Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158* (2017). 4, 5

[CFG*15] CHANG A. X., FUNKHOUSER T., GUIBAS L., HANRAHAN P., HUANG Q., LI Z., SAVARESE S., SAVVA M., SONG S., SU H., ET AL.: Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015). 6, 23

[CHXY21] CHENG M., HUI L., XIE J., YANG J.: Sspc-net: Semi-supervised semantic 3d point cloud segmentation network. In *Thirty-Fifth AAAI Conference on Artificial Intelligence* (2021), pp. 1140–1147. 11

[CRS] Crs4 visual computing dataset. http://vic.crs4.it/download/datasets/. Accessed: 23-June-2022. 4, 5

[CRW*20] CHAUDHURI S., RITCHIE D., WU J., XU K., ZHANG H.: Learning generative models of 3d structures. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 643–666. 2, 3, 16

[CSM14] CHANG A., SAVVA M., MANNING C. D.: Learning spatial knowledge for text to 3d scene generation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 2028–2038. 23

[CXY*15] CHEN K., XU K., YU Y., WANG T.-Y., HU S.-M.: Magic decorator: automatic material suggestion for indoor digital scenes. *ACM Transactions on graphics (TOG) 34*, 6 (2015), 1–11. 23

[DBS*21] DEVRIES T., BAUTISTA M. A., SRIVASTAVA N., TAYLOR G. W., SUSSKIND J. M.: Unconstrained scene generation with locally conditioned radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14304–14313. 19, 20

[DCLT18] DEVLIN J., CHANG M.-W., LEE K., TOUTANOVA K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018). 19

[DCS*17] DAI A., CHANG A. X., SAVVA M., HALBER M., FUNKHOUSER T., NIESSNER M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 5828–5839. 3, 4, 5, 6

[DFCS16] DASGUPTA S., FANG K., CHEN K., SAVARESE S.: Delay: Robust spatial layout estimation for cluttered indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 616–624. 14

[DN18] DAI A., NIESSNER M.: 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), pp. 452–468. 10, 11

[DT20] DENNINGER M., TRIEBEL R.: 3d scene reconstruction from a single viewport. In *European Conference on Computer Vision* (2020), Springer, pp. 51–67. 12

[DZL*22] DUAN Y., ZHU C., LAN Y., YI R., LIU X., XU K.: Disarm: Displacement aware relation module for 3d detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), IEEE, pp. 16959–16968. 9

[ETH] Eth zurich 3d dataset. https://www.eth3d.net/datasets. Accessed: 22-June-2022. 4, 5

[FCG*21] FU H., CAI B., GAO L., ZHANG L.-X., WANG J., LI C., ZENG Q., SUN C., JIA R., ZHAO B., ET AL.: 3D-FRONT: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10933–10942. 4, 5, 6, 19

[FGW*21] FENG M., GILANI S. Z., WANG Y., ZHANG L., MIAN A.: Relation graph network for 3d object detection in point clouds. *IEEE Trans. Image Process. 30* (2021), 92–107. 9

[FJG*21] FU H., JIA R., GAO L., GONG M., ZHAO B., MAYBANK S., TAO D.: 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision 129*, 12 (2021), 3313–3337. 6

[FRS*12] FISHER M., RITCHIE D., SAVVA M., FUNKHOUSER T., HANRAHAN P.: Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 1–11. 16, 17

[FSH11] FISHER M., SAVVA M., HANRAHAN P.: Characterizing structural relationships in scenes using graph kernels. In *ACM SIGGRAPH 2011 papers*. 2011, pp. 1–12. 4, 15, 16, 22

[FSL*15] FISHER M., SAVVA M., LI Y., HANRAHAN P., NIESSNER M.: Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG) 34*, 6 (2015), 1–13. 16, 17

[GCS*20] GENOVA K., COLE F., SUD A., SARNA A., FUNKHOUSER T.: Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 4857–4866. 13, 15

[GDN22] GÜMELI C., DAI A., NIESSNER M.: Roca: Robust cad model retrieval and alignment from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 4022–4031. 1, 14

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 216–224. 13, 14

[GLSU13] GEIGER A., LENZ P., STILLER C., URTASUN R.: Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research 32*, 11 (2013), 1231–1237. 9

[GMJ19] GKIOXARI G., MALIK J., JOHNSON J.: Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 9785–9795. 13

[GRJ22] GKIOXARI G., RAVI N., JOHNSON J.: Learning 3d object shape and layout without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2022), pp. 1695–1704. 12, 13, 14

[GSW*22] GAO J., SHEN T., WANG Z., CHEN W., YIN K., LI D., LITANY O., GOJCIC Z., FIDLER S.: Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems* (2022). 19

[GWB*22] GRAUMAN K., WESTBURY A., BYRNE E., CHAVIS Z., FURNARI A., GIRDHAR R., HAMBURGER J., JIANG H., LIU M., LIU X., ET AL.: Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18995–19012. 6

[HDN19] HOU J., DAI A., NIESSNER M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 4421–4430. 11

[HHF] HEDAU V., HOIEM D., FORSYTH D.: Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th international conference on computer vision*, IEEE, pp. 1849–1856. 14

[HHT*20] HU R., HUANG Z., TANG Y., VAN KAICK O., ZHANG H., HUANG H.: Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 118–1. 4

[HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33* (2020), 6840–6851. 19

[HLVK*17] HU R., LI W., VAN KAICK O., SHAMIR A., ZHANG H., HUANG H.: Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 1–13. 23

[HPB*16] HANDA A., PATRAUCEAN V., BADRINARAYANAN V., STENT S., CIPOLLA R.: Understanding real world indoor scenes with synthetic data. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4077–4085. 5, 6

[HPN*16] HUA B.-S., PHAM Q.-H., NGUYEN D. T., TRAN M.-K., YU L.-F., YEUNG S.-K.: Scenenn: A scene meshes dataset with annotations. In *2016 fourth international conference on 3D vision (3DV)* (2016), Ieee, pp. 92–101. http://hkust-vgd.ust.hk/scenenn/home/, Accessed: 24-June-2022. 5, 6

[HQX*18] HUANG S., QI S., XIAO Y., ZHU Y., WU Y. N., ZHU S.-C.: Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. In *NeurIPS* (2018). 9, 12, 13, 14

[HQZ*18] HUANG S., QI S., ZHU Y., XIAO Y., XU Y., ZHU S.-C.: Holistic 3d scene parsing and reconstruction from a single rgb image. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 187–203. 12, 14

[HvKW*16] HU R., VAN KAICK O., WU B., HUANG H., SHAMIR A., ZHANG H.: Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 1–13. 23

[HYZ*20] HU R., YAN Z., ZHANG J., VAN KAICK O., SHAMIR A., ZHANG H., HUANG H.: Predictive and generative neural networks for object functionality. *arXiv preprint arXiv:2006.15520* (2020). 23

[HZvK*15] HU R., ZHU C., VAN KAICK O., LIU L., SHAMIR A., ZHANG H.: Interaction context (icon) towards a geometric functionality descriptor. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–12. 23

[ISS17] IZADINIA H., SHAN Q., SEITZ S. M.: Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 5134–5143. 12, 13, 14

[JLS12] JIANG Y., LIM M., SAXENA A.: Learning object arrangements in 3d scenes using human context. *arXiv preprint arXiv:1206.6462* (2012). 17

[JTRS12] JAIN A., THORMÄHLEN T., RITSCHEL T., SEIDEL H.-P.: Material memex: Automatic material suggestions for 3d objects. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 1–8. 23

[JW19] JOHANNA WALD ARMEN AVETISYAN N. N. F. T. M. N.: Rio: 3d object instance re-localization in changing indoor environments. 6

[KALD20] KUO W., ANGELOVA A., LIN T.-Y., DAI A.: Mask2cad: 3d shape prediction by learning to segment and retrieve. In *European Conference on Computer Vision* (2020), Springer, pp. 260–277. 14

[KALD21] KUO W., ANGELOVA A., LIN T.-Y., DAI A.: Patch2cad: Patchwise embedding learning for in-the-wild shape retrieval from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12589–12599. 14

[KYH*19] KALERVO A., YLIOINAS J., HÄIKIÖ M., KARHU A., KANNALA J.: Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis. In *Scandinavian Conference on Image Analysis* (2019), Springer, pp. 28–40. 3

[LAK*18] LIN H., AVERKIOU M., KALOGERAKIS E., KOVACS B., RANADE S., KIM V., CHAUDHURI S., BALA K.: Learning material-aware local descriptors for 3d shapes. In *2018 International Conference on 3D Vision (3DV)* (2018), IEEE, pp. 150–159. 23

[LBMR17] LEE C.-Y., BADRINARAYANAN V., MALISIEWICZ T., RABINOVICH A.: Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 4865–4874. 14

[LBS*18] LI Y., BU R., SUN M., WU W., DI X., CHEN B.: Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems* (2018), pp. 828–838. 10

[LCK*14] LIU T., CHAUDHURI S., KIM V. G., HUANG Q., MITRA N. J., FUNKHOUSER T.: Creating consistent scene graphs using a probabilistic grammar. *ACM Transactions on Graphics (TOG) 33*, 6 (2014), 1–12. 4

[LFU13] LIN D., FIDLER S., URTASUN R.: Holistic scene understanding for 3d object detection with RGBD cameras. In *IEEE International Conference on Computer Vision* (2013), pp. 1417–1424. 9

[LGD*19] LI Y., GU C., DULLIEN T., VINYALS O., KOHLI P.: Graph matching networks for learning the similarity of graph structured objects. In *International conference on machine learning* (2019), PMLR, pp. 3835–3845. 4

[LHAZ] LI H., HU R., ALHASHIM I., ZHANG H.: Foldabilizing furniture. 23

[LHLY21] LI C., HUANG H., LIEN J.-M., YU L.-F.: Synthesizing scene-aware virtual reality teleport graphs. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–15. 23

[LMTG19] LI G., MULLER M., THABET A., GHANEM B.: Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 9267–9276. 10

[LPX*19] LI M., PATIL A. G., XU K., CHAUDHURI S., KHAN O., SHAMIR A., TU C., CHEN B., COHEN-OR D., ZHANG H.: Grains: Generative recursive autoencoders for indoor scenes. *ACM Transactions on Graphics (TOG) 38*, 2 (2019), 1–16. 1, 4, 5, 16, 18, 20

[LSD15] LONG J., SHELHAMER E., DARRELL T.: Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440. 12

[LSM*18] LI W., SAEEDI S., MCCORMAC J., CLARK R., TZOUMANIKAS D., YE Q., HUANG Y., TANG R., LEUTENEGGER S.: Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716* (2018). 5, 6

[LTJ*21] LIU A., TUCKER R., JAMPANI V., MAKADIA A., SNAVELY N., KANAZAWA A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 14458–14467. 6

[LTJ22] LEI J., TANG J., JIA K.: Generative scene synthesis via incremental view inpainting using rgbd diffusion models. *arXiv preprint arXiv:2212.05993* (2022). 20, 21

[LVC*19] LANG A. H., VORA S., CAESAR H., ZHOU L., YANG J., BEIJBOM O.: Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 12697–12705. 9

[LXJ*23] LIU J., XIONG W., JONES I., NIE Y., GUPTA A., OĞUZ B.: Clip-layout: Style-consistent indoor scene synthesis with semantic furniture embedding. *arXiv preprint arXiv:2303.03565* (2023). 21, 22

[LYJ*20] LEE H.-Y., YANG W., JIANG L., LE M., ESSA I., GONG H., YANG M.-H.: Neural design network: Graphic layout generation with constraints. *ECCV. Springer, Heidelberg* (2020). 4

[LYS*21] LI Z., YU T.-W., SANG S., WANG S., SONG M., LIU Y., YEH Y.-Y., ZHU R., GUNDAVARAPU N., SHI J., ET AL.: Openrooms: An open framework for photorealistic indoor scene datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 7190–7199. 4, 5, 6

[LZC*21] LIU Z., ZHANG Z., CAO Y., HU H., TONG X.: Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021). 7, 9

[ML15] MALLYA A., LAZEBNIK S.: Learning informative edge maps for indoor scene layout prediction. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 936–944. 14

[MLZ*16] MA R., LI H., ZOU C., LIAO Z., TONG X., ZHANG H.: Action-driven 3d indoor scene evolution. *ACM Trans. Graph. 35*, 6 (2016), 173–1. 16, 17

[MPF*18] MA R., PATIL A. G., FISHER M., LI M., PIRK S., HUA B.-S., YEUNG S.-K., TONG X., GUIBAS L., ZHANG H.: Language-driven synthesis of 3d scenes from scene databases. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–16. 16, 17, 23

[MPNF22] MANINIS K.-K., POPOV S., NIESSER M., FERRARI V.: Vid2cad: Cad model alignment using multi-view constraints from videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022). 14

[MRC20] MANANDHAR D., RUTA D., COLLOMOSSE J.: Learning structural similarity of user interface layouts using graph networks. In *European Conference on Computer Vision* (2020), Springer, pp. 730–746. 4

[MSL*11] MERRELL P., SCHKUFZA E., LI Z., AGRAWALA M., KOLTUN V.: Interactive furniture layout using interior design guidelines. *ACM transactions on graphics (TOG) 30*, 4 (2011), 1–10. 16, 17

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural

radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 6, 19

[MvAB*20] MUREZ Z., VAN AS T., BARTOLOZZI J., SINHA A., BADRINARAYANAN V., RABINOVICH A.: Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16* (2020), Springer, pp. 414–431. 12, 13, 15

[NHG*20] NIE Y., HAN X., GUO S., ZHENG Y., CHANG J., ZHANG J. J.: 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* (2020), pp. 55–64. 12, 13, 14, 15, 22

[NSF12] NATHAN SILBERMAN DEREK HOIEM P. K., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *ECCV* (2012). 3

[PBEPAE20] PATIL A. G., BEN-ELIEZER O., PEREL O., AVERBUCH-ELOR H.: READ: Recursive autoencoders for document layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (2020), pp. 544–545. 3, 4

[PJBM22] POOLE B., JAIN A., BARRON J. T., MILDENHALL B.: Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022). 19

[PKS*21] PASCHALIDOU D., KAR A., SHUGRINA M., KREIS K., GEIGER A., FIDLER S.: Atiss: Autoregressive transformers for indoor scene synthesis. In *Thirty-Fifth Conference on Neural Information Processing Systems* (2021). 1, 4, 18, 19, 21

[PLF*21] PATIL A. G., LI M., FISHER M., SAVVA M., ZHANG H.: LayoutGMN: Neural graph matching for structural layout similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 11048–11057. 4, 16, 22

[PMG*20] PINTORE G., MURA C., GANOVELLI F., FUENTES-PEREZ L., PAJAROLA R., GOBBETTI E.: State-of-the-art in automatic 3d reconstruction of structured indoor environments. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 667–699. 2, 4

[PNI*18] PETERS M. E., NEUMANN M., IYYER M., GARDNER M., CLARK C., LEE K., ZETTLEMOYER L.: Deep contextualized word representations. In *Proceedings of NAACL-HLT* (2018), pp. 2227–2237. 19

[PSM14] PENNINGTON J., SOCHER R., MANNING C. D.: Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543. 19

[QCLG20] QI C. R., CHEN X., LITANY O., GUIBAS L. J.: Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 4404–4413. 3, 7, 8, 9

[QLHG19] QI C. R., LITANY O., HE K., GUIBAS L. J.: Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 9277–9286. 7, 8, 9

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 652–660. 8, 9, 10, 11

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413* (2017). 8, 10

[RDN*22] RAMESH A., DHARIWAL P., NICHOL A., CHU C., CHEN M.: Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022). 23

[RHGS15] REN S., HE K., GIRSHICK R., SUN J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems 28* (2015), 91–99. 12

[RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763. 19, 21

[RRR*21] ROBERTS M., RAMAPURAM J., RANJAN A., KUMAR A., BAUTISTA M. A., PACZAN N., WEBB R., SUSSKIND J. M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 10912–10922. 5, 6

[RS16] REN Z., SUDDERTH E. B.: Three-dimensional object detection and layout prediction using clouds of oriented gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 1525–1533. 7, 8, 9

[RSH*21] REIZENSTEIN J., SHAPOVALOV R., HENZLER P., SBORDONE L., LABATUT P., NOVOTNY D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision* (2021). 6

[SCH*16] SAVVA M., CHANG A. X., HANRAHAN P., FISHER M., NIESSNER M.: Pigraphs: learning interaction snapshots from observations. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 1–12. 16, 17

[SCL*22] SANGHI A., CHU H., LAMBOURNE J. G., WANG Y., CHENG C.-Y., FUMERO M., MALEKSHAN K. R.: Clip-forge: Towards zero-shot text-to-shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 18603–18613. 21

[SCS*22] SAHARIA C., CHAN W., SAXENA S., LI L., WHANG J., DENTON E., GHASEMIPOUR S. K. S., AYAN B. K., MAHDAVI S. S., LOPES R. G., ET AL.: Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487* (2022). 23

[SDWMG15] SOHL-DICKSTEIN J., WEISS E., MAHESWARANATHAN N., GANGULI S.: Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (2015), PMLR, pp. 2256–2265. 19

[SEE*12] STURM J., ENGELHARD N., ENDRES F., BURGARD W., CREMERS D.: A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (2012), IEEE, pp. 573–580. 5, 6

[SF16] SCHONBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4104–4113. 20

[SFL*22] SANGHI A., FU R., LIU V., WILLIS K., SHAYANI H., KHASAHMADI A. H., SRIDHAR S., RITCHIE D.: Textcraft: Zero-shot generation of high-fidelity and diverse shapes from text. *arXiv preprint arXiv:2211.01427* (2022). 21

[SFZ*22] SUN J., FANG H., ZHU X., LI J., LU C.: Correlation field for boosting 3d object detection in structured scenes. In *Thirty-Sixth AAAI Conference on Artificial Intelligence* (2022), pp. 2298–2306. 9

[SHKF12] SILBERMAN N., HOIEM D., KOHLI P., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *European conference on computer vision* (2012), Springer, pp. 746–760. 5

[SHL*14] SHARF A., HUANG H., LIANG C., ZHANG J., CHEN B., GONG M.: Mobility-trees for indoor scenes manipulation. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 2–14. 23

[SHS*22] SOLAH M., HUANG H., SHENG J., FENG T., POMPLUN M., YU L.-F.: Mood-driven colorization of virtual indoor scenes. *IEEE Transactions on Visualization and Computer Graphics 28*, 5 (2022), 2058–2068. 23

[SLX15] SONG S., LICHTENBERG S. P., XIAO J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 567–576. 13

[SWL19] SHI S., WANG X., LI H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 770–779. 9

[SWM*19] STRAUB J., WHELAN T., MA L., CHEN Y., WIJMANS E., GREEN S., ENGEL J. J., MUR-ARTAL R., REN C., VERMA S., CLARKSON A., YAN M., BUDGE B., YAN Y., PAN X., YON J., ZOU Y., LEON K., CARTER N., BRIALES J., GILLINGHAM T., MUEGGLER E., PESQUEIRA L., SAVVA M., BATRA D., STRASDAT H. M., NARDI R. D., GOESELE M., LOVEGROVE S., NEWCOMBE R.: The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019). 5

[SX14] SONG S., XIAO J.: Sliding shapes for 3d object detection in depth images. In *European conference on computer vision* (2014), Springer, pp. 634–651. 5, 7, 9

[SX16] SONG S., XIAO J.: Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 808–816. 7, 8, 9

[SYZ*17] SONG S., YU F., ZENG A., CHANG A. X., SAVVA M., FUNKHOUSER T.: Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 1746–1754. 3, 12

[SZAB17] SEDAGHAT N., ZOLFAGHARI M., AMIRI E., BROX T.: Orientation-boosted voxel nets for 3d object recognition. In *Proceedings of the British Machine Vision Conference* (2017). 8, 9

[TQD*19] THOMAS H., QI C. R., DESCHAUD J.-E., MARCOTEGUI B., GOULETTE F., GUIBAS L. J.: Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 6411–6420. 10

[UZH] University of zurich dataset. https://www.ifi.uzh.ch/en/vmml/research/datasets.html. Accessed: 22-June-2022. 4, 5

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008. 9, 11, 19

[WDNT20] WALD J., DHAMO H., NAVAB N., TOMBARI F.: Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 3961–3970. 5, 6, 16

[WDP*23] WEI Q. A., DING S., PARK J. J., SAJNANI R., POULENARD A., SRIDHAR S., GUIBAS L.: Lego-net: Learning regular rearrangements of objects in rooms. *arXiv preprint arXiv:2301.09629* (2023). 20, 22

[WFT*19] WU W., FU X.-M., TANG R., WANG Y., QI Y.-H., LIU L.: Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–12. 3, 4

[WLJ*22] WANG Y., LI Z., JIANG Y., ZHOU K., CAO T., FU Y., XIAO C.: Neuralroom: Geometry-constrained neural implicit surfaces for indoor scene reconstruction. *arXiv preprint arXiv:2210.06853* (2022). 20

[WLW*19] WANG K., LIN Y.-A., WEISSMANN B., SAVVA M., CHANG A. X., RITCHIE D.: Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–15. 18, 20

[WLY20] WANG H., LIANG W., YU L.-F.: Scene mover: automatic move planning for scene arrangement by deep reinforcement learning. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–15. 23

[WSCR18] WANG K., SAVVA M., CHANG A. X., RITCHIE D.: Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–14. 1, 3, 4, 16, 17, 18

[WSL*19] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog) 38*, 5 (2019), 1–12. 10

[WYN20] WANG X., YESHWANTH C., NIESSNER M.: Sceneformer: Indoor scene generation with transformers. *arXiv preprint arXiv:2012.09793* (2020). 4, 18, 19, 21

[XLW*20] XIE Q., LAI Y.-K., WU J., WANG Z., ZHANG Y., XU K., WANG J.: Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 10447–10456. 7

[XMZ*14] XU K., MA R., ZHANG H., ZHU C., SHAMIR A., COHEN-OR D., HUANG H.: Organizing heterogeneous scene collections through contextual focal points. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 1–12. 1, 16, 17, 22

[XOT13] XIAO J., OWENS A., TORRALBA A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision* (2013), pp. 1625–1632. 4, 5

[XTS*22] XIE Y., TAKIKAWA T., SAITO S., LITANY O., YAN S., KHAN N., TOMBARI F., TOMPKIN J., SITZMANN V., SRIDHAR S.: Neural fields in visual computing and beyond. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 641–676. 19

[YGZT21] YANG M.-J., GUO Y.-X., ZHOU B., TONG X.: Indoor scene generation from a collection of semantic-segmented depth images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 15203–15212. 1, 18, 19

[YSZ*15] YU F., SEFF A., ZHANG Y., SONG S., FUNKHOUSER T., XIAO J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365* (2015). 12

[YWP*19] YANG S.-T., WANG F.-E., PENG C.-H., WONKA P., SUN M., CHU H.-K.: Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 3363–3372. 14

[YWY22] YU H.-X., WU J., YI L.: Rotationally equivariant 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 1456–1464. 1, 7, 9, 22

[YYT*11] YU L. F., YEUNG S. K., TANG C. K., TERZOPOULOS D., CHAN T. F., OSHER S. J.: Make it home: automatic optimization of furniture arrangement. *ACM Transactions on Graphics (TOG)-Proceedings of ACM SIGGRAPH 2011, v. 30,(4), July 2011, article no. 86 30*, 4 (2011). 16, 17

[YYT15] YU L.-F., YEUNG S.-K., TERZOPOULOS D.: The clutterpalette: An interactive tool for detailing indoor scenes. *IEEE transactions on visualization and computer graphics 22*, 2 (2015), 1138–1148. 16, 17

[YZD*21] YANG C., ZHENG J., DAI X., TANG R., MA Y., YUAN X.: Learning to reconstruct 3d non-cuboid room layout from a single rgb image. *arXiv preprint arXiv:2104.07986* (2021). 19

[YZK21] YIN T., ZHOU X., KRAHENBUHL P.: Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 11784–11793. 9

[ZBK*] ZHANG Y., BAI M., KOHLI P., IZADI S., XIAO J.: Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding. In *IEEE International Conference on Computer Vision*, pp. 1201–1210. 9

[ZCSH18] ZOU C., COLBURN A., SHAN Q., HOIEM D.: Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 2051–2059. 14

[ZCZ*21] ZHANG C., CUI Z., ZHANG Y., ZENG B., POLLEFEYS M., LIU S.: Holistic 3d scene understanding from a single image with implicit representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 8833–8842. 4, 12, 13, 14, 15, 22

[ZFF*21] ZHOU H., FENG Y., FANG M., WEI M., QIN J., LU T.: Adaptive graph convolution for point cloud analysis. In *2021 IEEE/CVF International Conference on Computer Vision* (2021), pp. 4945–4954. 10, 11

[ZHG*16] ZHAO X., HU R., GUERRERO P., MITRA N., KOMURA T.: Relationship templates for creating scene variations. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 1–13. 17

[ZHPY21] ZHANG Y., HUANG H., PLAKU E., YU L.-F.: Joint computational design of workspaces and workplans. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–16. 23

[ZJFJ19] ZHAO H., JIANG L., FU C., JIA J.: Pointweb: Enhancing local neighborhood features for point cloud processing. In *IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 5565–5573. 10, 11

[ZJJ*21] ZHAO H., JIANG L., JIA J., TORR P. H., KOLTUN V.: Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 16259–16268. 1, 10, 11, 12

[ZSYH20] ZHANG Z., SUN B., YANG H., HUANG Q.: H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision* (2020), Springer, pp. 311–329. 7, 8, 9

[ZTF*18] ZHOU T., TUCKER R., FLYNN J., FYFFE G., SNAVELY N.: Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817* (2018). 6

[ZWK14] ZHAO X., WANG H., KOMURA T.: Indexing 3d scenes using the interaction bisector surface. *ACM Transactions on Graphics (TOG) 33*, 3 (2014), 1–14. 15, 16, 22

[ZYM*20] ZHANG Z., YANG Z., MA C., LUO L., HUTH A., VOUGA E., HUANG Q.: Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG) 39*, 2 (2020), 1–21. 18, 21

[ZZL*20] ZHENG J., ZHANG J., LI J., TANG R., GAO S., ZHOU Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. In *European Conference on Computer Vision* (2020), Springer, pp. 519–535. 5