Extended Indoor Layout Estimation using Monocular RGB for Efficient Path Planning and Navigation

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Shantanu Singh
2020701022
shantanu.singh@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
June 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled " **Extended Indoor Layout Estimation using Monocular RGB for Efficient Path Planning and Navigation** " by **Shantanu Singh**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. K. Madhava Krishna

To *my parents* and *friends*

Acknowledgments

I would like to extend my heartfelt thanks to my parents who have always been there for me, providing me with unwavering support and encouragement throughout my journey. I am also deeply grateful to Professor Madhava Krishna for his invaluable guidance and support during my time at the Robotics Research Center (RRC) at IIIT Hyderabad.

I began my research journey as an intern under Professor Madhava Krishna's mentorship at IIIT Hyderabad. I am truly thankful to him for providing me with the opportunity to work in a variety of areas within the field of robotics, which helped me to hone my interests before embarking on my Master's program at IIIT.

I am also thankful to my seniors Harshit, Mithun, Kaustubh, Udit, and Shubodh for their guidance and support in my robotics research endeavors. In addition, I would also like to express my gratitude to my friends Kinal, Omama, Mounika, Amit, and Krishna for being an integral part of my memorable journey at IIIT.

Finally, I would like to express my gratitude to Jaidev Shriram and Shaantanu Kulkarni for their valuable contributions to the publication that formed the basis of this thesis.

Abstract

In this work, we propose IndoLayout, a novel real- time approach for generating high-quality occupancy maps from an RGB image for indoor scenes. Such occupancy maps are often crucial for path-planning and mapping in indoor environments but are often built using only information contained in the ego view. In contrast, our approach also predicts occupancy values beyond immediately visible regions from just a monocular image, leveraging learnt priors from indoor scenes. Hence, our proposed network can produce a hallucinated, amodal scene layout that includes areas occluded in the RGB image, such as a navigable floor behind a desk. Specifically, we propose a novel architecture that uses self-attention and adversarial learning to vastly improve the quality of the predicted layout. We evaluate our model on several photorealistic indoor datasets and outperform previous relevant work on all metrics that measure layout quality, including newly adopted ones. Finally, we demonstrate the effectiveness of our method by showing significant improvements on the Point Goal navigation task over similar approaches using IndoLayout.

Contents

List of Figures

Chapter 1

Introduction

Figure 1.1: Indolayout - Amodal layout estimation for indoor environments

1.1 Motivation

Humans have a remarkable ability to navigate new indoor spaces based on knowledge acquired from traversing similar scenes in the past. As a result, one can easily infer multiple properties about a given location by leveraging these priors, such as the semantic configuration of a room (the various objects present) [3], proximity to adjacent spaces (a kitchen may occur near the dining room), and more relevant to our task, the layout of the scene.

Indoor layout estimation is the problem of estimating the occupancy map of a scene, its navigable and non-navigable areas. In recent years, this field has gained traction, motivated primarily by its applications in several robotics tasks, such as SLAM, Exploration, and Indoor Navigation. Layouts that are typically used for such tasks are limited to information contained in the ego-view and do not consider priors that humans can easily apply to a scene. Alternatively, humans can predict the *extended or amodal* layout of a scene and guess the presence of free space or obstacles behind occluding surfaces like furniture. While there has been sufficient traction for amodal layout estimation for on-road scenes in the context of Autonomous Driving [39, 42], there have been very limited efforts for indoor scenes like offices and home spaces. Further, this task is far from trivial as indoor layouts are arguably more complex and diverse in nature compared to typical outdoor layouts, where the shape of a vehicle and the surrounding environment are largely consistent. Further, in indoor scenes, the layout of a single room

itself can change drastically depending on the viewing angle, obstructions present, and the position of the robot.

In this thesis, we propose a learning based approach, *IndoLayout* , that uses attention to *amodally* predict the layout of indoor scenes given just an RGB image in such challenging environments. (Fig. 1.1)

1.2 Contributions

1. We present *IndoLayout*, a lightweight architecture that beats existing state-of-the-art on amodal occupancy map representation estimation by effectively leveraging attention and adversarial learning. (Fig. 5.1)

2. We demonstrate significant improvements over state-of-the-art on three large challenging indoor datasets - Gibson [50], Matterport 3D [4], and HM3D [38]. (Table. 6.1, Section 6.1)

3. We demonstrate the importance of analysing layout quality by adopting two new metrics for this task and also surpass prior work. (Section 6.1)

4. We generate an *IndoScene Layout* dataset to evaluate performance of indoor scene layout estimation methods.

5. Lastly, we apply *IndoLayout* to the Point Goal Navigation [41] task and show superior results over comparable methods. (Table. 6.3, Section 6.4)

1.2.1 Thesis Organization

This thesis is divided into six chapters. First chapter motivates the problem of generating occupancy grid maps using monocular RGB cameras and its applications in SLAM. Second chapter digs into the literature survey of the existing state of the art methods and various sub-components for occupancy map prediction. Third chapter introduces the IndoScene Layout dataset, diving into the details of data collection and preprocessing to generate quality samples for the chosen problem. Fourth chapter explores the use of existing self-supervised monocular depth estimation methods on Indoscene dataset for the task of layout estimation. Fifth chapter introduces the Indolayout model and describes in detail various components of its pipeline. It describes the experimentation settings, including the evaluation metrics and approaches compared. Sixth chapter presents the quantitative and qualitative results and comparison of Indolayout with various state of art methods for occupany prediction. It also presents the application it to the task of Point Goal navigation and Mapping, where again we compare the performance with previous methods. Seventh chapter concludes this thesis and motivates towards using monocular RGB camera for amodal layout estimation in downstream tasks.

1.3 Preliminary Material

1.3.1 Occupancy Grid Mapping

Occupancy grid mapping is a technique used in robotics and computer vision for creating a map of an environment. It involves dividing the environment into a grid of cells, with each cell representing a particular location within the environment. Each cell can be marked as occupied or unoccupied, based on whether there is an obstacle or other significant feature present at that location.

In occupancy grid mapping, the robot or camera system gathers data about the environment using sensors, such as lidar or stereo cameras. This data is used to estimate the likelihood that each cell in the grid is occupied or unoccupied. The resulting map can be used by the robot to navigate through the environment and avoid collisions with obstacles.

Occupancy grid mapping is a useful technique because it allows the robot to create a map of an unknown environment without the need for prior knowledge about the layout or geometry of the environment. It can also be used to update the map as the robot moves through the environment, allowing it to track changes and adapt to new obstacles.

1.3.2 Point Goal Navigation

Point goal navigation refers to the ability of a robot or autonomous system to navigate to a specific location within an environment. This type of navigation is typically used when the robot is given a specific target or goal location, and it must determine the best path to reach that location while avoiding obstacles and other hazards in the environment.

There are various algorithms and techniques that can be used to achieve point goal navigation, including path planning algorithms and local navigation algorithms. Path planning algorithms are used to determine the overall path that the robot should follow to reach its goal, while local navigation algorithms are used to help the robot navigate around obstacles and other hazards in its environment as it moves towards its goal.

Point goal navigation is an important capability for robots and autonomous systems, as it allows them to perform tasks such as delivery, exploration, and search and rescue in complex and dynamic environments. It is also a key component of many autonomous vehicles, such as self-driving cars and drones, which need to be able to navigate to specific locations with a high degree of accuracy and reliability.

Chapter 2

Related Works

In this chapter, we review the existing approaches for layout estimation for indoor environments. We follow that with the review of other approaches for monocular depth estimation and image to image translation that can be adapted to this task of layout generation given a monocular RGB image.

2.1 Indoor Layout Estimation

Layout is a catchall phrase that simultaneously refers to floorplans [28, 29, 34], Manhattan-world 3D room layouts [19,51,55], and occupancy maps [6]. Recent work on floorplans [29], [34] for instance, use a 3D point cloud as input to produce a polygonized floorplan of the indoor scene. Liu et al., 2015 [28] instead uses floorplans as a prior along with RGB images to predict the 3D room layout of a scene by exploiting the geometry of a scene. However, these representations often fail to capture the presence of obstacles in the scene, which is essential for downstream tasks such as robot navigation. Further, they often require the use of floorplans or 3D scans of the scene at inference, which is not easy to obtain. Instead, we focus on the occupancy map prediction from a monocular RGB image, which is easier to use on real robots.

Occupancy maps have been extensively used in robotics, particularly for mapping [6], navigation [25, 49], and planning [36]. Early approaches for mapping used LiDAR, and sensor fusion [20, 33] to build occupancy maps, but recently, deep learning approaches have shown great success using just RGB images, particularly in outdoor scenes [30,32,39]. Lu et al., 2019 [30] proposed using a variational encoder-decoder based architecture with a pretrained VGG-backbone as feature extractor for occupancy map prediction. They addressed the challenge of noisy and incomplete ground-truth layout available from the 3D point cloud at each timestep by using the variational sampling's robustness in their architecture. However, the variational bottleneck used in practice lead to incorrect blob-like shapes. Mani et al., 2020 [32] proposed an alternate formulation where they use encoder-decoder architecture with adversarial learning to address these issues. Further, they used registration to accumulate 3D point clouds from several timesteps to obtain lower noise and denser ground-truth layouts with information for occluded regions as well. Roddick et al., 2020 [39] proposed another architecture that used fea-

ture pyramid network (FPN) [27] backbone for extracting multiscale geometric and semantic features and then use stacked dense transformer layers to project these features from perspective space to BEV space. The dense transformer layers are essentially bottleneck layers that first collapse the features along the vertical (height) dimension and subsequently expand along the depth using 1-D convolution layers. The insight behind such an architecture choice was the observation of high dependency on vertical dimension for context relevant to occupancy prediction, whereas the horizontal dimension had a geometric mapping from perspective to BEV space. The bottleneck forced the network to transform relevant information only and become more robust to noise in the input image.

Learning based approaches for indoor layouts are however, relatively new [6,15,37,44] and are often used as a proxy for other tasks such as PointNav [24] and ObjectNav [1]. Chaplot et al., 2020 [6] used a Neural SLAM module that produces occupancy maps and estimates agent pose from input RGB images and motion sensors. The occupancy maps are supervised with ground-truth generated using the depth image at that timestep and is limited by occlusions and field of view of the depth sensor. Ramakrishnan et al., 2020 [37] overcame this limitation by extracting occupancy map from the 3D mesh with a custom sensor. The amodal layout estimation helps improve the performance on path planning significantly over [6]. Georgakis et al., 2021 [15] extends [6] by predicting semantics instead of just occupancy for the scene-layout. To speed up training by avoiding redundant samples in later stages of training, they used ensembles of their network for uncertainty estimation and selecting goal locations during exploration with maximum uncertainty. Shen et al., 2021 [44] extends [37] by extracting multi-layer semantic occupancy map from the Habitat simulator [41] using the ground-truth semantic labels available for the Matterport3D dataset [4]. A common limitation for all these approaches is the reliance on depth sensors at inference time, which incurs an additional computational expense and payload weight. In our work, we focus on improving layout predictions using just RGB images and surpassing relevant current state of the art, while also showing improvements on the PointGoal navigation task [24].

2.2 Amodal Layout Prediction

Occupancy maps generated from single views are often incomplete, lacking any information beyond what is immediately visible in the RGB image. Humans, however, can hallucinate beyond this and use prior information to reason about the occluded areas as well, predicting the *amodal* layout. To this end, [32, 42, 52] show how learning-based approaches can reasonably predict the presence of cars and roads beyond visible regions using just RGB images in outdoor scenes. Schulter et al.,2018 [42] proposed a two-stage network that takes masked RGB as input and first predicts in-painted Semantic and Depth maps reasoning about the geometry for occluded regions. These maps are then converted into an initial BEV-map which is refined by another network to get the final predicted layout. A key contribution of this paper was the use of the GPS location to extract layouts from Open Street Maps (OSM) [35] and use it to enforce amodal completion with a reconstruction loss between the predicted BEV and extracted layout. To deal with the noise in GPS location, they propose a CNN-network that

warps the OSM maps to the predicted BEV before computing the reconstruction loss. Mani et al. 2020 [32] highlighted the in-ability of [42] to use end-to-end learning in its two-stage approach and instead proposed a single encoder-decoder network that directly predicted the final BEV-map from the input RGB image. The ground-truth used for supervising the network was generated by registering the 3D semantic point-clouds over several timesteps before converting them into target BEV-maps. In addition, they used OSM [35] maps for adversarial learning and enforce amodal completion for the predicted layouts. Yang et al., 2021 [52] proposed the use of a cross-view transformer module for feature selection and enhancement to improve the layout prediction for RGB input. Their proposed architecture transforms the perspective-view features extracted by a backbone CNN to top-view using a block of stacked MLP layers, which are transformed back to perspective-view using another block with identical configuration. By using a cycle-consistency loss between the extracted features and the re-projected ones, they find that it improve the representativeness of features in both views which helps in improving the overall performance on the layout prediction task. They further use a transformer [47] module to compute an correlation map that is used for feature selection and aggregation at different stages in the network before the features are passed through the final decoder for predicting the BEV map. They emperically demonstrate significantly better amodal layout estimation compared to the prior approaches listed on the same datasets [14] [5].

Similarly, in indoor environments, [37, 43, 49] do amodal layout estimation, predicting occluded regions and, in some cases, semantic classes. Of these, Ramakrishnan et al., 2020 [37] is perhaps, the closest to our approach, and we adopt their work as our primary baseline. They supervise their network with ground-truth generated using the 3D mesh in the simulator with a custom sensor, hence performing amodal estimation for regions occluded in the input view. Seymour et al., 2021 [43] propose a network architecture that uses a transformer [47] for extracting features from an egocentric semantic map which is fused with features extracted from the RGB and Depth images using concatentation, followed by Dense layers to predict a multi-layer semantic occupancy map. A key limitation of their approach is that their network hallucinates occupancy only for a few selected semantic classes, while we do not discriminate between any, and predict occupancy for all objects. Wei et al., 2021 [49] attempt this as an inpainting problem and train a network to recreate the visible layout seen from a higher vantage point using an RGB image and the visible layout from a lower height, obtained using depth sensors. While the higher vantage point does offer more information for some occluded regions, the approach is costly since it involves multiple sensors for data collection and inference. In contrast, we only use a monocular RGB image and predict the true bird's eye view by training on publicly available datasets [4, 38, 50] and simulator [41].

2.3 Monocular Depth Estimation

A vast amount of research has been done in the field of supervised depth estimation where most of them frame the problem as a per-pixel regression ([11], [10], [23]) or as an ordinal regression problem

with depth discretization([12], [2]). While these methods deliver state of the art performance, they require huge amount of labeled data to train which has its own set of challenges in collection and curation. Particularly, we note that the depth obtained from simulators has significant noise due to the reconstruction techniques used for generating the 3D environment meshes from their corresponding sets of panoramic images. Hence, using supervised learning doesn't yield useful performance when relying on available simulator datasets for indoor environments, and we instead focus our efforts on self-supervised approaches for learning depth from monocular images.

Self-supervised learning of depth estimation is used to reduce the demand for large-scale labeled training data. Various approaches target stereo images or videos as training data for monocular depth estimation models with the objective of minimizing the photometric error between warped image,generated using the predicted depth, and real view. Garg et al., 2016 [13] proposed the first self-supervised method to use color consistency loss between stereo images to train a monocular depth model. Another alternative is to use monocular videos where the relative pose between two images also need to be estimated, either externally or as part of the model training, along with depth prediction for view warping. Zhou et al., 2017 [57] used separate networks for depth and pose each to construct the photometric loss across temporal frames. Many follow-up methods then try to improve the self-supervision by new loss terms based on semantic consistencies or geometric cues. For indoor environments, Zhou et al. 2019 [56] is a seminal work that proposed an optical-flow based approach and a data preprocessing step that required removing all the image pairs with pure rotation to handle large rotational motions. A more recent method, Yu et al., 2020 [53] defined a new approach using patch-match and plane-regularization to handle large textureless regions that can lead to unstable learning in indoor scenes.

We explore the use of self-supervised models, particularly Yu et al., 2020 [53], with our dataset to see if it can be reliable for occupancy prediction. We can use the monocular depth estimates to create a 3D point cloud and then discretize and project it based on the required size and resolution to get the desired 2D-occupancy grid map. Note that this method is limited since it doesn't perform amodal layout estimation and only predicts layout for regions visible in the current field of view.

2.4 Transformers for Image Synthesis

Generating bird's eye view images from a monocular camera is fundamentally ill-posed as RGB images lack concrete information about the depth of the scene. Hence, our work is more aligned with problems in the image translation domain where a new image is generated given a guiding image as input. Note that this becomes even more relevant when we take into account the shift in frame of reference for the input and output images and a one-to-one pixel correspondence isn't valid. Recent approaches that use attention [46, 54] are of particular relevance to us. Tang et al., 2019 [46] uses attention to guide their generative network that translates one view to another, given a semantic map of the target view as guidance. They propose a network architecture that uses multi-scale spatial pooling to enrich the extracted features from the input RGB and Semantic images. These features are then

passed through a custom multi-channel attention-based module for generating the target images. A key contribution of this work is the use of attention maps in uncertainty prediction which is utilized for regularization of the noise introduced by use of predicted semantic maps in the optimization process. Zhang et al., 2019 [54] cites the use of self-attention module to overcome the limitation of convolution layers in extracting long-range dependencies in an image for a given image translation task. They further propose using spectral normalization for both generator and discriminator networks based on empirical evidence that it helps stabilize the training for the generator by reducing large magnitudes for weights and consequently gradient updates.

Inspired by this body of work, we propose an attention-driven network to predict the bird's eye view map. However, unlike Tang et al., 2019 [46], we do not use any secondary image to guide our generation and directly predict the bird's eye view using just a monocular RGB image. Also, instead of opting for complex architecture with multi-channel aggregation for predictions, we simply use a self-attention block like Zhang et al., 2019 [54] to project the features from perspective-view to top-view in a context-dependent manner. Our empirical results demonstrate the effectiveness of our approach while still maintaining low memory-footprint and high throughput useful for practical solutions.

Chapter 3

Indoscene Layout Dataset

3.1 Introduction

Having a dataset of indoor environments available for researchers can facilitate research in the area of occupancy prediction by providing a common dataset that can be used by multiple researchers to compare and contrast different approaches. This can help to accelerate the development of new algorithms and techniques for layout estimation in indoor environments. Recent works like [37] and [6] use different indoor datasets with Habitat [41] simulator to train the layout prediction models in an interactive manner. The samples seen during the learning phase for the models vary in the way they explore the different scenes in these simulators. In addition, training the layout prediction model within an RL framework is highly inefficient and resource intensive. For instance, [6] uses 8 Nvidia V100 GPUs to train their layout prediction model along with other modules in their Neural-SLAM framework for 10 million steps. Using such a framework for evaluating different ideas is prohibitive due to involved resource costs, experiment time and even feasibility on consumer-grade hardware.

To address these issues and ensure a fair comparison between different approaches, we propose a new dataset *IndoScene Layout* with near exhaustive exploration of the scenes from different datasets available for the Habitat Simulator. Further we refine the layout map generated over the previous methods by taking into account the reconstructed mesh errors and semantics. Our key contributions with this dataset are as follows:

1. **Defined train-test samples for major indoor datasets:** Indolayout dataset contains RGB map, depth map, pose and generated ground-truth layouts for all major indoor datasets used with the Habitat [41] simulator at the time of its release. These included Matterport3D [4], Gibson Tiny and 4+ [50] and HM3D dataset [38]. The train-test split generated is based on the train-test split defined for scenes in the Point-Nav challenge series for the Habitat [41] simulator.

2. **Exhaustive coverage of scenes:** To ensure that the dataset consists of images that capture different viewpoints exhaustively for each scene in the different datasets, we have designed an exploration algorithm that uses the layout of the scene to select goals for the path planner that maximize

coverage of the entire scene. This helps reduce the size of the dataset without compromising on the diversity captured from the available resources.

3. **Low noise and redundancy:** To remove noisy samples from the simulator that occur due to reconstruction errors in involved datasets, we use several post-processing steps such as filtering based on percentage of missing values in RGB image, depth distribution range and variance and portion of floor visible in the corresponding generated layout.

We describe the procedure of collection and aspects of this dataset in detail in the following sections.

3.2 Simulator and Datasets

Similar to prior works, we also use the habitat [41] simulator for data collection. We repeat the process on three different datasets - Gibson [50], Matterport3D [4], and HM3D [38] to generate diverse samples. For the Gibson dataset, we further split the samples based on the Gibson Tiny split and Gibson 4+ [41], which is a filtered version of Gibson [50], consisting of scenes rated 4 or above by human evaluators, based on the texture and mesh quality of the scene.

3.3 Trajectory Generation

Since there has been little prior work that comprehensively evaluates layout on indoor scenes, we generate the training and validation splits for the aforementioned datasets ourselves by programming an agent in Habitat [41] simulator. Specifically, we spawn an agent one meter above the ground and capture a continuous trajectory as the agent maps the entire scene. The agent maximises coverage by choosing nearby areas yet to be mapped while avoiding movement near walls and obstacles. Our mapping objective ensures that our dataset includes a variety of different semantic classes and rooms, as well as diverse layouts that a typical robot may see during navigation. The agent is equipped with an RGB sensor of 512×512 resolution and a local map sensor that extracts a 128×128 occupancy map at a scale of 0.025 metres per pixel, or $3.2m \times 3.2m$.

3.4 Layout Generation

The raw output of the map sensor is ill-suited for our task as it includes areas that are outside the field of view. For instance, it is not feasible to guess the layout of rooms behind a closed door, which is included in the raw sensor output. Hence, we mask out such regions from the raw occupancy map using the technique proposed in [37] - we project rays from the agent until it hits a wall and exclude areas not covered by the ray. Note that we only use walls or other tall view-obstructing objects for masking purposes, due to which our layouts still include occlusions induced by furniture and other objects. We

shoot rays with a 120° FOV, as opposed to the camera's 90° FOV, which further increases the amount of hallucinated area. Finally, we dilate the mask to additionally increase coverage. All areas outside the mask are marked as *unknown*, and pixels within the mask are limited to either *occupied* or *free*. An example of the described pipeline is shown in Fig 3.1:

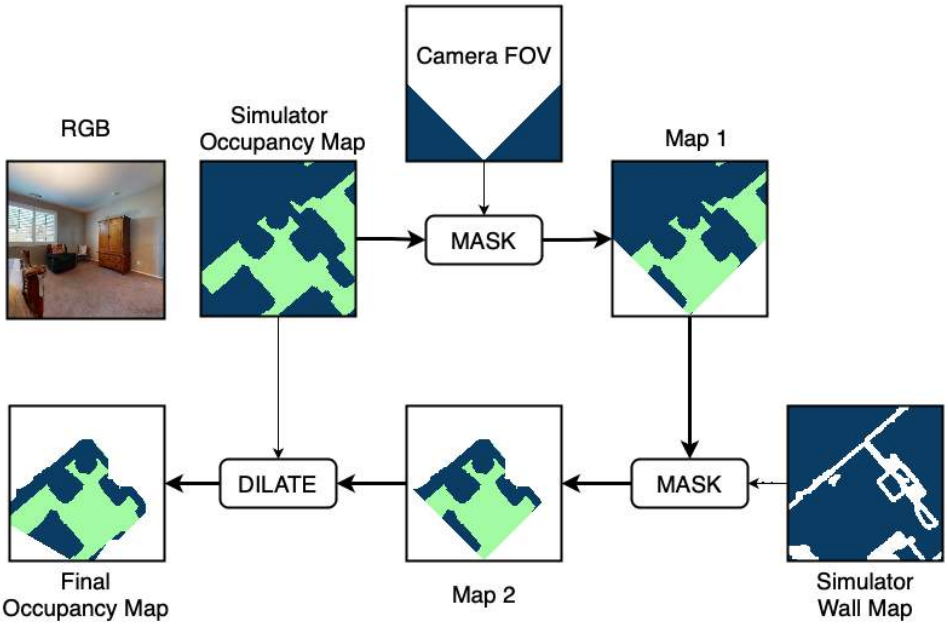

Figure 3.1: Partial Occupancy Map Generation Pipeline

After generating the trajectories and layouts for all scenes, we filter out potentially noisy samples by removing images with up-close obstacles based on the maximum depth visible and depth distribution's variance. We also remove images where the unknown region is more than 90% of the image, similar to [45, 49]. Here are some statistics for the final dataset used:

1. Gibson 4+ [41, 50]: It consists of 72 training and 12 validation scenes, (18,435 training images, 2955 validation images).

2. Gibson Tiny [50]: It consists of 25 training and 5 validation scenes, (8,176 training images, 1360 validation images).

3. Matterport [4]: It consists of 11 large and varying validation scenes, (7,885 validation images).

4. HM3D [38]: It consists of 100 validation scenes, (32,470 validation images).

We share a few samples from the above mentioned datasets in Fig. 3.2 and 3.3. To download the dataset or get more information, we request the reader to visit `https://github.com/indolayout/indolayout-dataset.git`.

Figure 3.2: Samples from Indoscene Dataset (Gibson-Tiny and HM3D)

Figure 3.3: Samples from Indoscene Dataset (Gibson 4+ and Matterport)

Chapter 4

Self-supervised monocular depth estimation

4.1 Introduction

Monocular depth estimation can be used to assist in occupancy grid mapping by providing estimates of the depth of objects in the scene. Given a single image of the scene, a monocular depth estimation algorithm can produce a depth map, which encodes the distance of each pixel in the image from the camera. This depth information can be used to infer the 3D positions of objects in the scene, which can be used to update the occupancy grid map.

Self-supervised learning can be a useful approach for monocular depth estimation, as it allows the model to be trained using a large dataset of images without the need for manual labeling of the depth maps. This can be more efficient and cost-effective than supervised learning, where a large dataset of labeled data is required. One approach is to use a dataset of images captured from a moving camera, and use the change in the appearance of the scene between consecutive frames to predict depth. For example, the photometric loss between the two frames can be used as the self-supervised objective function. The photometric loss measures the difference between the pixel intensities of the two frames, taking into account the geometric transformation between the frames. By minimizing the photometric loss, a CNN can learn to predict the depth map for a given input image. A few seminal works that utilize the above approach are [31], [17], [17]. These approaches work well for Outdoor datasets like KITTI but fail to learn proper depth in Indoor environments due to dominance of textureless surfaces that lead to noisy learning with photometric reconstruction as the objective. A recent paper [53] addresses this issue for indoor scenes by using patch-matching around distinct keypoints and plane regularization. We use the same method in layout estimation as explained in the sections below.

4.2 Overview of P^2 Net

P^2Net [53] argues that the poor performance of self-supervised approaches in monocular depth estimation task stem from use of non-discriminative point-matching. They address this issue by extracting keypoints with large local gradients and using patches around these keypoints instead of the entire image

for computing the photometric loss. However, using only patches provides limited supervision for the remaining textureless regions. To tackle this issue, they propose enforcing planar regularization for the depth prediction of these textureless regions defined using superpixels.

To understand their architecture better, we refer the readers to Fig 4.1 from [53] that showcases the various components they have used in their pipeline.

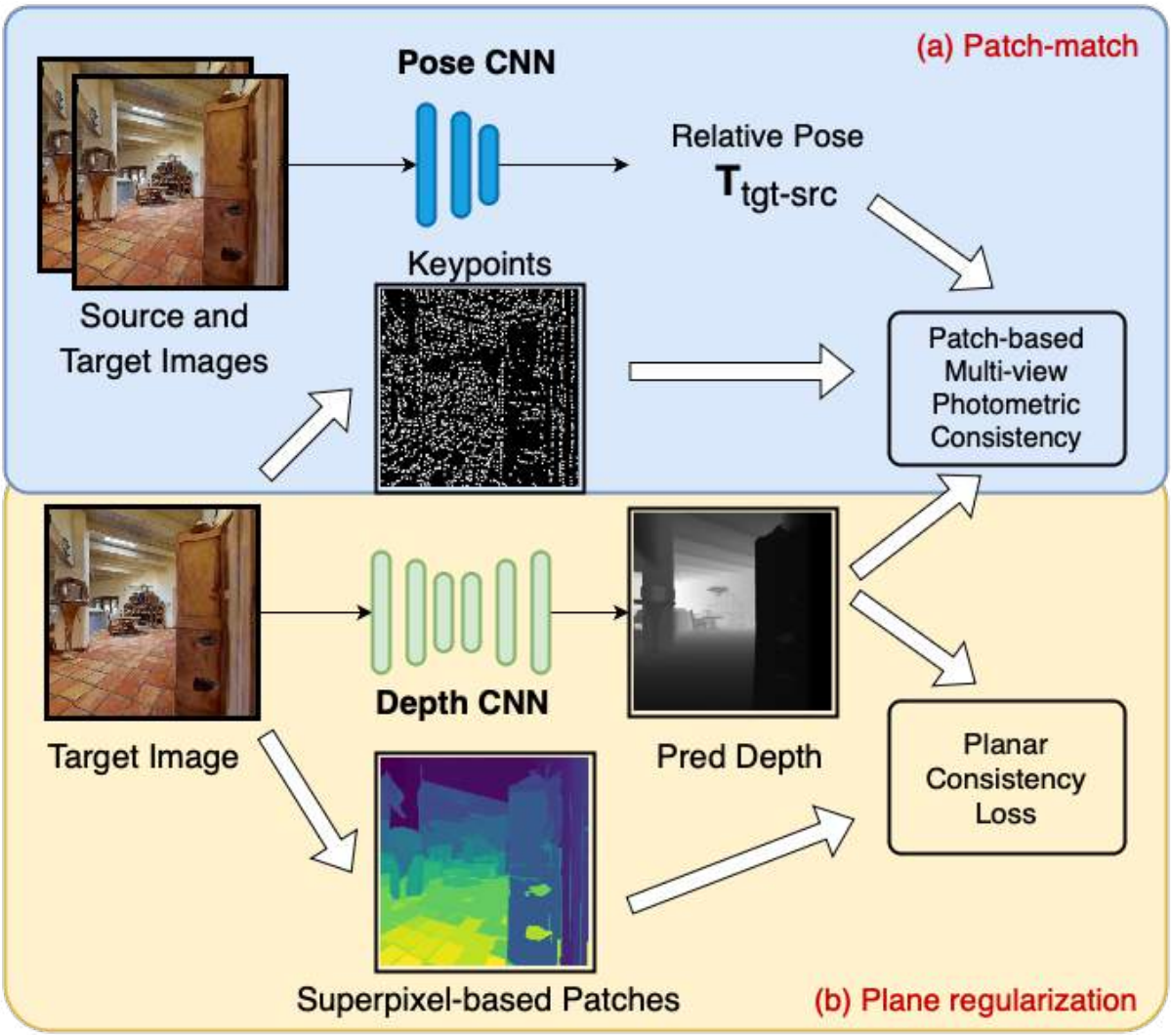

Figure 4.1: Architecture of P^2Net (From [53])

4.3 Evaluation on Indoscene dataset

The authors of P^2Net have shared both the pretrained model as well as their source code. We finetune their model on the training split of the Indoscene Layout dataset (3.1) and evaluate it on the test split. The datasets [4], [50] often have more artifacts and low quality textures due to improper reconstruction

techniques used for generating these meshes that are rendered for generating samples. This leads to poor supervision and subpar learning despite the proposed patch-based approach in P^2Net. We present a few challenging samples in Fig. 4.2 for the reader to verify the argument presented.

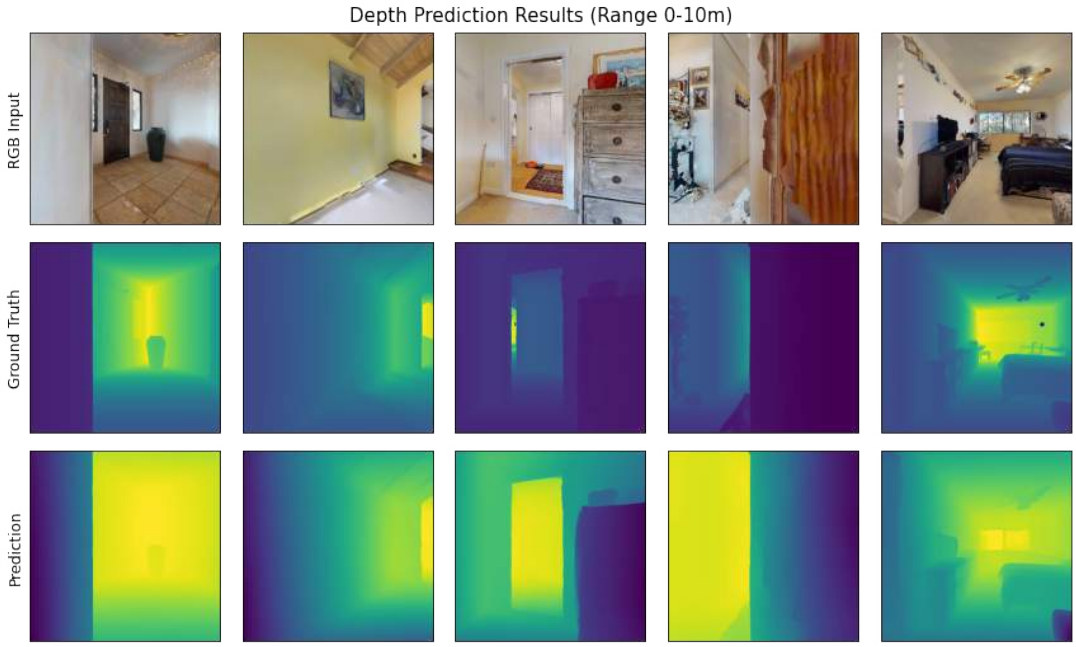

Figure 4.2: P^2Net Depth prediction outputs.

The above figure clearly highlights the scale ambiguity that leads to erroneous absolute depth values for a given input image. Further, for quantitative evaluation, we include the results on the standard evaluation metrics for depth estimation in 4.1. We include the scores of P2-Net reported in their paper as a separate row for comparison.

Dataset	**Metrics**				
	a1	**a2**	**a3**	**abs_rel**	**rmse**
Indoscene	0.356	0.624	0.808	0.369	7.202
NYUv2*	0.758	0.945	0.985	0.166	0.612

Table 4.1: Quantitative evaluation of P^2Net on Indoscene and NYUv2

The evaluation metrics reported in 4.1 are defined as follows:

$y, \hat{y}$ = ground-truth and predicted depth respectively

n = total number of pixels in the image

$a1, a2, a3$ = percentage of pixels that have $max(y/\hat{y}, \hat{y}/y) < 1.25, (1.25)^2, (1.25)^3$ respectively

$abs\_rel = \frac{1}{N} \sum \frac{|y - \hat{y}|}{y}$

$rmse = \sqrt{\frac{\sum (y - \hat{y})^2}{N}}$

Once again, the performance dip clearly indicates the challenge in learning depth for images obtained from simulator. Using such depth for occupancy map prediction would result in poor obstacle avoidance and suboptimal path planning. Hence, we avoid using the intermediate step of predicting depth and directly approach learning occupancy map prediction from the RGB input as show in subsequent chapters.

Chapter 5

Indolayout Architecture and Approach

5.1 Overview

Our proposed method not only tackles general layout prediction, but also attempts to reason beyond visible areas. This is a difficult problem in general due to the limited information present in an RGB image but even more so in the indoor scenario due to the varied spatial arrangement of objects and the complexity of layouts. Hence, we attempt to solve this problem by training a network that takes an RGB image as input and produces a three class image (corresponding to unknown, occupied, and free/navigable) after training on several large-scale photorealistic indoor scenes part of the Indoscene Dataset.

5.2 Network Architecture

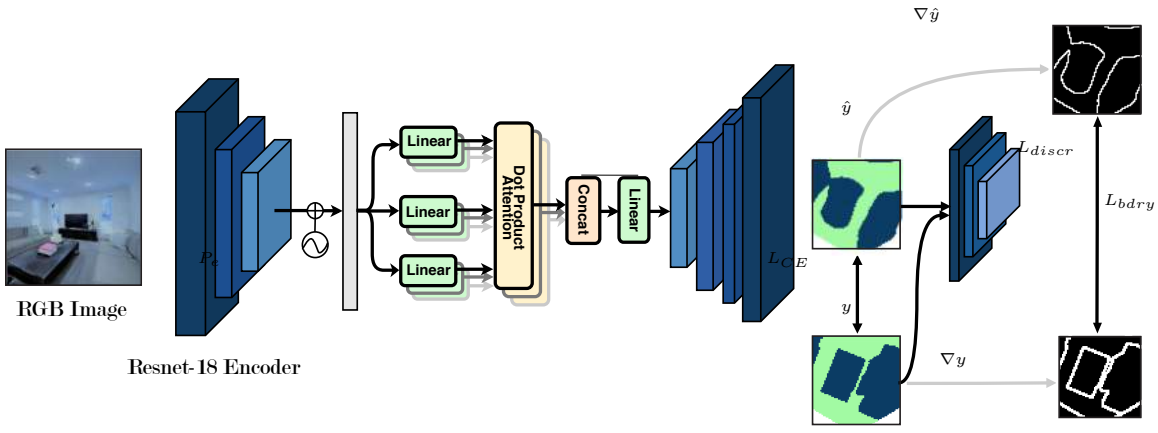

Figure 5.1: Indolayout architecture

The goal of our network is, given an input image, generate the corresponding top-view occupancy layout in metric scale. Given the nature of this task, we adopt the GAN [18] framework, and the

attention [47] module to leverage the benefits of each in our proposed model architecture(Figure 5.1). We describe the different components of our architecture in detail in the following sections.

5.2.1 Feature Extraction

We use the first 4 blocks of ResNet-18 (pre-trained on ImageNet [40]) as our encoder, followed by convolution and max-pooling layers to reduce the input map from a resolution of $3 \times 512 \times 512$ to $128 \times 8 \times 8$. We use a convolution-based encoder as the backbone for our model, instead of a transformer-based encoder such as ViT [9], to generate patch embeddings since they are more efficient for finetuning with small datasets due to their inductive biases. They also reduce the computation overhead by reducing the number of patches for the subsequent attention module, which allows for faster training and inference times.

5.2.2 Feature Encoding

The features extracted in the previous step carry the spatial nature of the perspective view and need to be transformed into a space more relevant to the top view. To this end, we propose using a self-attention module to project and aggregate these features across different patches in a context-dependent manner. We experimented with various configurations for the self-attention implementation based on ViT [9] and decided to use a single transformer block with multi-head attention, as empirical results with more number of blocks gave marginal improvements despite higher training/inference costs. We also add a learned positional embedding to the extracted features to provide additional context for feature aggregation in the self-attention module.

5.2.3 Occupancy Decoder

Our decoder takes the features computed by the transformer block and iteratively applies a series of convolutions, followed by BatchNorm [21], ReLU activation, and upsampling layers to produce a final output of shape $3 \times 128 \times 128$, where each channel corresponds to the probability of being unexplored, occupied, or free respectively, after applying the Softmax function.

5.2.4 Discriminator

Training a network with just a per-pixel loss such as binary cross entropy may not always produce outputs that are structurally coherent, as we typically perceive objects and layouts in groups of pixels or patches. Further, the shape of objects in the predicted layout may be irregular without including any priors about their typical shape. Motivated by this, we employ a patch-based discriminator [22] to distinguish between our predictions and the ground truth layouts, as such approaches have shown success in outdoor scenarios [32, 42, 52]. The discriminator takes the $3 \times 128 \times 128$ generated layout as input and outputs a label for various patches, corresponding to *real* or *fake*. Due to the high variance in

indoor layouts, we do not compute this adversarial loss using a distribution like [32] and [52]; instead we use the corresponding ground truth.

5.3 Loss Functions

For training, we use a combination of the following terms:

1. Weighted Cross Entropy computed over the three classes of our output using ground truth supervision.

$$L_{CE} = -\sum_{j=1}^{3} y_j \log(\hat{y_j})) \tag{5.1}$$

Here, y_j is the ground-truth probability of the j-th cell being occupied and $\hat{y_j}$ is the corresponding probability in the predicted layout.

2. Boundary Loss that penalises misclassification around the boundary of objects in particular. We calculate this by applying a L1 loss with the ground truth boundary and the spatial gradient of our output.

$$L_{bdry} = |\|\nabla \hat{y}\|_2 - y_{bdry}| \tag{5.2}$$

Here, y_{bdry} is the contours/boundary of the ground truth layouts, and $\hat{y}$ is the predicted layout. The spatial gradient ($\nabla \hat{y}$) computes the gradient in the x and y directions separately, which we then combine by calculating its norm.

3. GAN Loss that trains our patch-based discriminator and provides additional supervision to the generator.

$$\begin{aligned} L_{discr} &= \mathbb{E}_{y \sim p_{true}}[D(y)] + \mathbb{E}_{y \sim p_{fake}}[D(\hat{y})] \\ L_{gen} &= \mathbb{E}_{x \sim p_{true}}[D(\hat{y})] \end{aligned} \tag{5.3}$$

Here, $\hat{x}$ is the generated layout, corresponding to the fake distribution, and x is the ground truth layout, corresponding to the true distribution.

The final loss can be expressed as:

$$L = L_{CE} + \lambda_{bdry} L_{bdry} + \lambda_{gen} L_{gen} \tag{5.4}$$

where λ_x is the weight assigned to the loss term. We find that $\lambda_{bdry} = 0.001$ and $\lambda_{gen} = 0.01$ gives us the best results.

5.4 Evaluation Metrics

We quantitatively evaluate the performance of all approaches against the ground truth layouts. In line with prior work on layout estimation, we report the mean Intersection-over-Union (mIoU) and mean Average Precision (mAP) metrics.

The mean Intersection over Union **(mIoU)** is defined as:

$$mIoU = \frac{1}{n} \sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{5.5}$$

where:

n = number of classes
TP_i = number of true positive pixels for class i
FP_i = number of false positive pixels for class i
FN_i = number of false negative pixels for class i

Average Precision **(AP)** is defined as:

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{5.6}$$

where:

n = index over list of thresholds
R_n = Recall at nth threshold
P_n = Precision at nth threshold

and **(mAP)** is the mean of average precision over all classes:

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{5.7}$$

The mIoU metric effectively captures the minimum of precision and recall; hence we additionally report the mean **F1 score** defined as follows:

$$P = \frac{TP}{TP + FP} \tag{5.8}$$

$$R = \frac{TP}{TP + FN} \tag{5.9}$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{5.10}$$

where:

TP = number of True Positives
FP = number of False Positives
FN = number of False Negatives
P = Precision
R = Recall

We report each metric for the occupied and navigable/free class in the layouts. While these metrics capture the efficacy of our method, they may not capture the visual quality of the layouts well. Optimising for IoU or F1 in particular, can have the effect of producing rounded edges, when the layout of

objects in bird's eye view are typically sharper. This is particularly true for large objects, where small errors along the boundary will have a minimal contribution to the loss function. Therefore, we report the Boundary IoU **(bIoU)** [7] for each class, a more sensitive metric that focuses on boundary quality alone. [16] also proposed a boundary metric for segmentation that measured the difference in tangent angles between contours on the predicted and ground truth segmentation, but such a metric is more suited for building segmentation than our scenario due to the polygonal nature of buildings.

$$bIoU = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{5.11}$$

where, similar to **mIoU**, we have:

n = number of classes
TP_i = number of true positive pixels for class i
FP_i = number of false positive pixels for class i
FN_i = number of false negative pixels for class i

In addition to boundary IOU, we report the SSIM [48] score, which considers the similarity in structure between two images.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{5.12}$$

where:

x, y = Pair of Images being compared
μ_x, μ_y = mean of x, y
σ_x^2, σ_y^2 = variance of x, y
σ_{xy} = covariance of x and y
C_1, C_2 = constants to avoid division by zero

Chapter 6

Performance Evaluation

To evaluate the effectiveness of our proposed method, we compare *IndoLayout* against the current state-of-the-art models:

1. **ANS RGB**: The monocular indoor layout estimation method proposed in [6] referred to as 'Active Neural SLAM (ANS)'. They approach layout estimation as a subtask along with multiple other learnable modules including global and local policies. Inspired by traditional SLAM, the authors propose a hierarchical framework with the objective of maximizing exploration for a given scene in a fixed number of timesteps. The entire learning algorithm is framed as an optimization problem in the framework of reinforcement learning. Since we are going with supervised learning, we retrain their layout module selectively from scratch on the Indoscene layout dataset (3.1) for fair comparison. For brevity, we avoid discussing the architecture details of their model and request readers to refer their paper for implementation details.

2. **OccAnt RGB**: The amodal monocular indoor layout estimation method proposed in [37] referred to as 'Occupancy Anticipation (OccAnt)'. Their approach builds upon the Active Neural SLAM [6] approach by extending the architecture to learn amodal layout estimation. The monocular model proposed use the layout prediction module from ANS and extracts features from its output and merges it with features extracted from the RGB image in a pyramidal fashion for amodal layout generation. We once again request the reader to refer to [37] for proper implementation details. The authors report significant improvement in downstream tasks such as exploration and point to point navigation over the ANS models with their approach. For fair comparison, we once again retrain their layout prediction model from scratch on the Indoscene layout dataset (3.1).

We additionally report the scores of OccAnt RGBD [37], the state-of-the-art on indoor layout estimation, as a benchmark. Note that this approach uses both RGB and depth information during training and at inference time. While this is an unfair comparison, we include this to report the best-known performance for this task.

6.1 Layout estimation

			mIoU %			mAP %			F1 %			SSIM	Boundary IOU %
Dataset	Method	Only RGB?	Occ	Free	Mean	Occ	Free	Mean	Occ	Free	Mean		
Gibson 4+	ANS (RGB) [6]	✓	33.04	32.67	32.85	**77.81**	69.72	73.76	48.23	45.84	47.03	49.23	14.65
	OccAnt (RGB) [37]	✓	52.87	61.32	57.09	70.02	72.33	71.17	68.07	**74.08**	71.07	66.19	36.42
	***IndoLayout* (Ours)**	✓	**59.06**	**67.84**	**63.45**	71.92	**83.01**	**77.46**	**72.96**	74.02	**73.49**	**69.37**	**39.06**
	OccAnt (RGBD) [37]	✗	69.63	71.54	70.58	83.01	81.02	82.01	81.5	82.15	81.82	74.75	54.02
Gibson Tiny	ANS (RGB) [6]	✓	29.21	36	32.60	**70.27**	72.27	**71.27**	43.84	49.6	46.72	47.99	14.46
	OccAnt (RGB) [37]	✓	47.6	61.1	54.35	63.13	72.56	67.84	63.22	73.84	68.53	64.16	33.16
	***IndoLayout* (Ours)**	✓	**52.3**	**64.49**	**58.39**	64.13	**77.8**	70.96	**67.53**	**76.94**	**72.23**	**66.45**	**34.96**
	OccAnt (RGBD) [37]	✗	70.8	71.96	71.38	85.45	80.5	82.975	82.36	82.24	82.3	73.2	51.25
Matterport	ANS (RGB) [6]	✓	24.1	34.32	29.21	**66.29**	77.06	71.67	37.24	48.06	42.65	42.62	12.11
	OccAnt (RGB) [37]	✓	43.02	63.3	53.16	63.53	76.45	69.99	58.08	75.65	66.86	63.79	33.44
	***IndoLayout* (Ours)**	✓	**49.48**	**66.39**	**57.93**	64.61	**81.62**	**73.11**	**64.55**	**78.12**	**71.33**	**67.34**	**36.44**
	OccAnt(RGBD) [37]	✗	67.53	74.68	71.10	82.04	83.25	82.64	79.69	84.12	81.90	74.92	53.33
HM3D	ANS (RGB) [6]	✓	31.53	35.61	33.57	**80.67**	70.84	**75.755**	46.65	48.88	47.765	49.73	13.52
	OccAnt (RGB) [37]	✓	53.17	62.85	58.01	71.9	71.68	71.79	68.26	75.15	71.705	66.61	35.79
	***IndoLayout* (Ours)**	✓	**57.02**	**66.23**	**61.625**	71.64	**76.19**	73.915	**71.57**	**77.86**	**74.715**	**69.22**	**37.6**
	OccAnt (RGBD) [37]	✗	71.55	71.68	71.615	85.91	78.84	82.375	82.84	81.9	82.37	74.98	53.13

Table 6.1: Indolayout outperforms prior RGB baselines on all datasets.

For the layout estimation task, we use our proposed *IndoScene Layout* for training and evaluation of the different approaches listed above. We divide the analysis based on different sources such as Gibson-Tiny, Gibson 4+ [50], Matterport3D [4] and HM3D [38] used for generating our dataset. This helps in the evaluation of the generalization capability of different models as described in the following section.

6.1.1 Performance on Evaluation Metrics

To evaluate the performance of the models on the task of layout estimation, we compare the predicted local occupancy maps against the ground truths and quantify the correctness of these predictions using IoU and F1 score metrics described in section 5.4. In Table 6.1, we compare the performance of our models on the Gibson4+ and Gibson Tiny datasets. All the models are trained on the training split for both datasets and then evaluated on a separate validation split. As observed, among the RGB-only models, *IndoLayout* model is substantially better than the other baselines in its prediction for both the

occupied as well as the free space. Our model reduces the gap in the performance compared to the RGBD model, thus alleviating the penalty incurred for tasks where only a monocular setup can be used.

In Table 6.1, we report the out-of-the-box performance on the validation splits for Matterport and HM3D datasets. Again, we observe that our model outperforms the other RGB models and generalizes better to novel scenes.

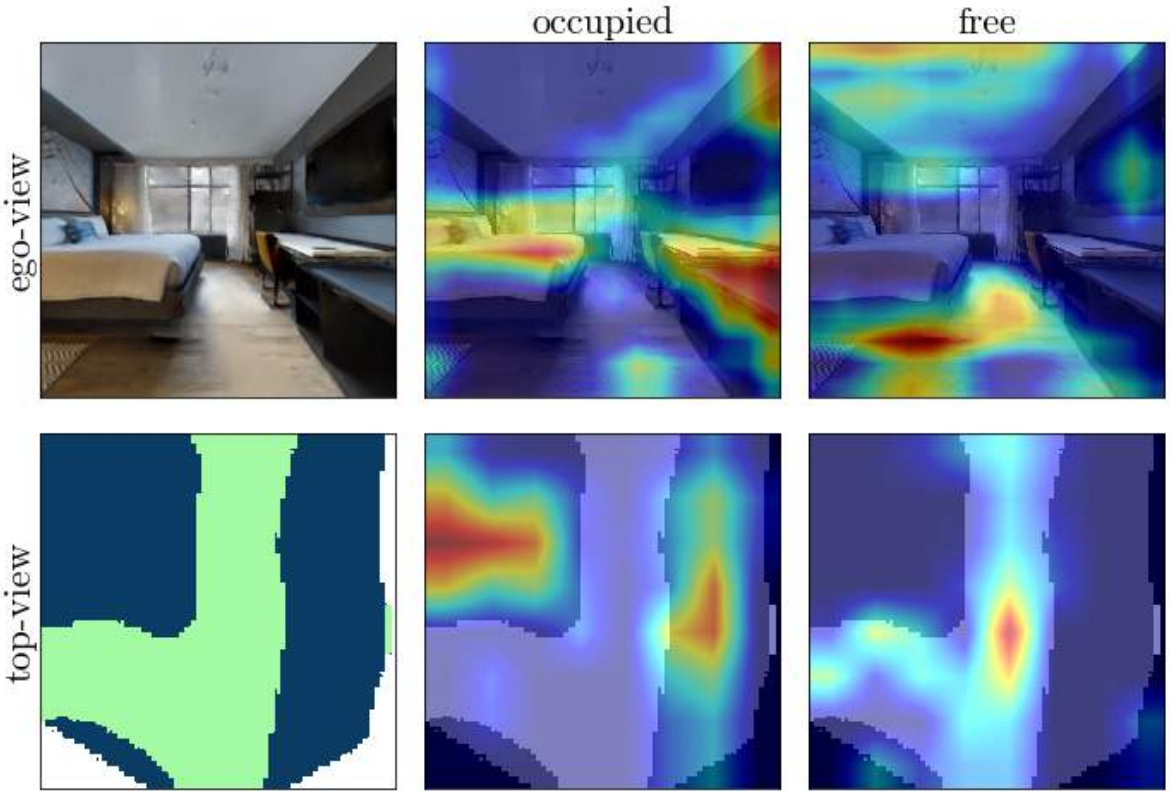

Figure 6.1: Exploring saliency maps to visualize Indolayout's interpretation of the input image

To better understand the reason why our model does better, we inspect the saliency maps computed using GradCAM for all the models, as well as the attention map for our model in Figure 6.1. This gives an insight into which regions the models focus on, for a given image while generating the corresponding local occupancy maps. From the figure, it is clear that due to attention, our model is able to focus on relevant surfaces while predicting the target classes.

For the occupied class, the saliency map in the ego-view row shows that the model pays attention to the surfaces of the table and bed in the input RGB image. The corresponding highlights are also seen in the top-view which indicates that the *IndoLayout* model has learnt relevant feature mapping for the task of layout generation. A similar case is seen for the free class. An interesting observation, that can be made in this sample, is that the *IndoLayout* model learns to use the ceiling to estimate the free regions in the room. This makes sense since the ceiling is relatively less occluded than the floor and still provides

a good estimate for the extents of the room. This highlights the strength of deep learning in learning relevant features implicitly for the task of layout generation given enough data and an architecture that can leverage the available information efficiently.

6.1.2 Quality of Generated Outputs

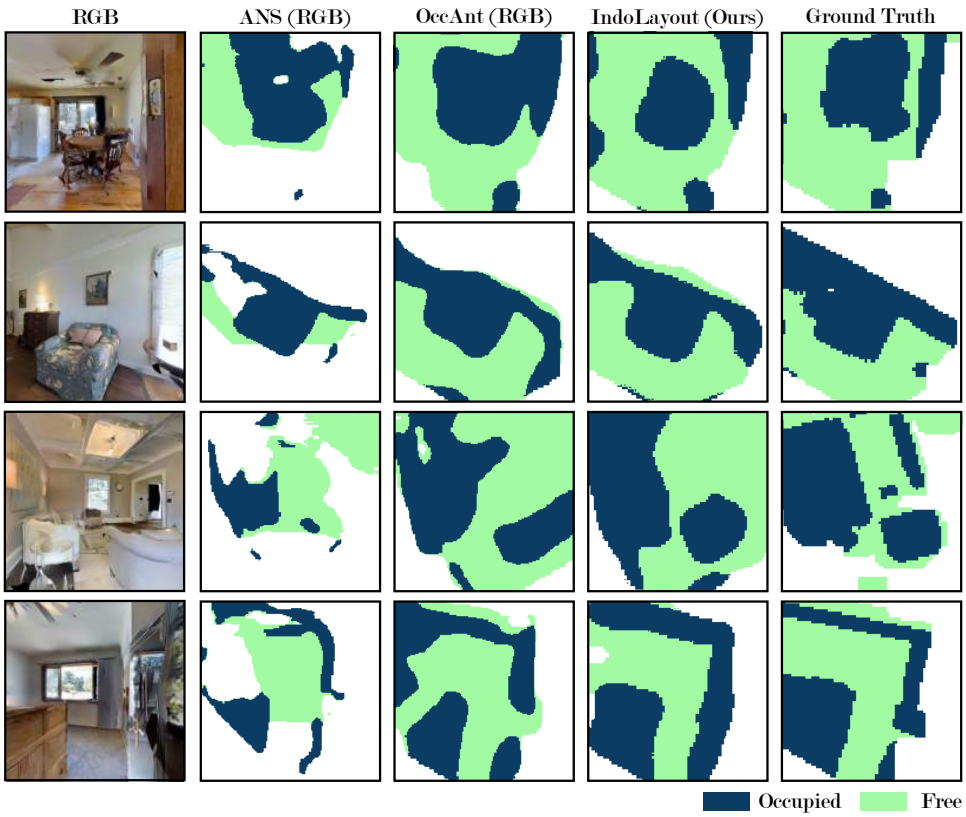

Figure 6.2: Qualitative comparison of layout prediction results

IndoLayout produces vastly better layouts qualitatively compared to prior art, as shown in Figure 6.2. We find that OccAnt [37] often produces blurry outputs with more rounded edges, as compared to our model. ANS [6] predicts only visible occupancy and hence suffers from occlusions. Even for the free regions, *IndoLayout* captures the narrow spaces more accurately which is important for navigation in tight spaces. This analysis is confirmed by the reported SSIM and Boundary IoU metrics, on which we also show significant improvements over [37] and [6]. We attribute this gain to the use of self-attention and a discriminator that operate at the patch level while attending to the global context. Note, we include the last row as failure case to highlight how highly complex scenes can still be challenging for our proposed model. We dive into the details of the failure cases in section 6.7.

6.2 Amodal estimation

We try to investigate the reason behind the performance boost by evaluating the hallucinated and visible occupancy regions separately. We find that the percentage of pixels hallucinated by our approach is 4% less than OccAnt RGB [37] but 6% more accurate within this area. Since the percentage of hallucinated pixels was calculated using the model's output, the higher accuracy obtained by our approach shows that we perform better within the hallucinated area. This improvement is significant for planners that may use the amodal layout predicted. Further, we find that our model is 4% more accurate within the visible occupancy region, which is also critical for navigation purposes.

6.3 Ablation Studies

	mIoU %			**mAP %**		
Method	Occ	Free	Mean	Occ	Free	Mean
Base	52.59	64.09	58.34	70.84	73.13	71.98
w/ Boundary Loss	54.2	63.3	58.75	69.9	74.8	72.35
w/ Discriminator	54.4	64.4	59.4	**72.5**	75.3	73.9
w/ Self-Attention	56.79	64.24	60.51	68.97	77.85	73.41
IndoLayout (All)	**59.06**	**67.84**	**63.45**	71.92	**83.01**	**77.46**

Table 6.2: We examine the role of each component in *IndoLayout* by comparing the performance gain over the base model (encoder-decoder architecture) for both IOU and mAP scores. We observe that self-attention has a substantial impact on the model's overall performance, followed by the boundary loss and the discriminator.

6.3.1 Importance of Attention

We find that adding self-attention to our network significantly improves performance, as shown in Table 6.2, with the results most pronounced on the Gibson 4+ dataset. We attribute this gain to the expressive power of self-attention, which has been well-established for vision-related tasks in recent years [8] [26]. Using self-attention, our model can learn the global context in addition to local features, which is critical for such a task as distant pixels may help *contextualise* local patches. As mentioned earlier, the saliency maps visualised in Figure 6.1 further support our hypothesis that attention enables our model to focus on relevant regions in the input image more effectively.

6.3.2 Effect of Adversarial Learning

By using a patch-based discriminator [22], we observe a 2% improvement in IoU for the occupied class. In addition, we also notice an improvement in the visual quality for several images, as shown in figure 6.2. However, the visual improvements are not as pronounced as in outdoor scenarios [32] [52], such as the KITTI [14] and Argoverse [5] datasets. We attribute this to the typical shape of vehicles and road layouts, which belong to a smaller distribution than indoor layouts. Upon inspection, we find that objects belonging to the same semantic class, such as a dining table, can have vastly varying layouts, in contrast to the outdoor scenario, where vehicle shapes are largely similar. Further, the shape of navigable areas in indoor scenes is highly dependent on furniture placements and room size, making it more challenging to regularise the output using existing layouts, as originally proposed in [42].

6.4 Application: PointNav

Difficulty	Method	Success Rate ↑	SPL ↑	Time ↓
Easy	ANS RGB [6]	0.851	0.676	154.248
	Occant RGB [37]	0.888	0.715	135.498
	***IndoLayout* (Ours)**	**0.913**	**0.731**	**127.105**
Medium	ANS RGB	0.626	0.488	283.329
	Occant RGB	0.698	0.532	261.636
	***IndoLayout* (Ours)**	**0.763**	**0.566**	**233.803**
Hard	ANS RGB	0.303	0.239	429.541
	Occant RGB	0.339	0.248	417.312
	***IndoLayout* (Ours)**	**0.431**	**0.337**	**383.33**
Overall	ANS RGB	0.7	0.552	236.51
	Occant RGB	0.752	0.59	217.288
	***IndoLayout* (Ours)**	**0.8**	**0.621**	**198.246**

Table 6.3: Using *IndoLayout* as the mapping module in previous layout based state of the art [37] on the PointNav task [41], we show considerable improvement over all baselines, showing the significance and utility of our approach.

To establish the significance of our improvements, we apply *IndoLayout* to the PointGoal navigation task from the Habitat Challenge 2020 [41] where an agent in Habitat simulator [41] has to pathfind and move to a target location without any prior global map. [37] showed that hallucination could significantly help in path planning and navigation for this task, and successfully beat prior work [6] that only used the visible occupancy. Hence, we simply replace the layout module used in the RL pipeline of [37] with *IndoLayout*, trained on Gibson4+, and evaluate the performance of our model. We only compare against the RGB variants of [6], [37] to be fair.

We find that by simply replacing the layout module, we observe a 5% improvement in success rate and SPL, standard metrics used for this task. Further, we do significantly better for the *medium* and *hard* episodes in the validation set, with a 7% higher success rate than OccAnt RGB on *medium* episodes and 9% higher success rate on *hard* episodes. We also observe similar trends for other reported metrics, as shown in Table 6.3. Since these episodes involve navigation across longer distances, the performance improvements suggest that our layouts are more suited for planning and navigation purposes. Once again, we note that we did not train the policy from scratch alongside our model, and expect even greater improvements if trained with *IndoLayout* in an end-to-end manner.

6.5 Application: Mapping

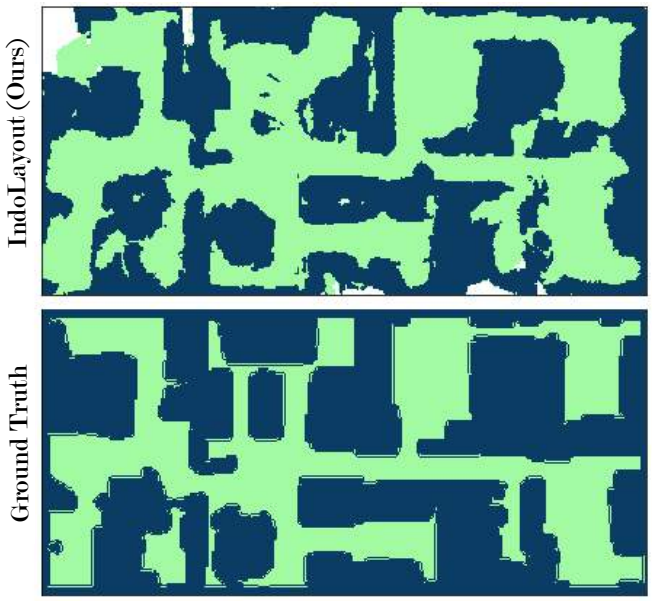

Figure 6.3: Results of mapping with Indolayout predicted occupancy maps and ground truth poses.

Since our dataset consists of a continuous trajectory per scene, we use the predicted layouts for the validation split and register the maps using the raw probability values predicted by our model. We filter out low-confidence predictions using a threshold and aggregate information using a moving average.

We find that the results, as shown in figure 6.3 closely resemble the ground truth, despite never being trained on it. While there is room for improvement, this shows the efficacy of *IndoLayout* for potential mapping tasks despite being an RGB only model.

6.6 Timing Analysis

Method	Parameters (M)	FPS
Occant (RGB)	19.86	33.1
IndoLayout (Ours)	**14.35**	**61.02**

Table 6.4: Timing analysis of *IndoLayout* against state-of-the-art

In addition to the above, we also report additional model statistics like model parameter count (in millions) and inference speed (frames-per-second) in table 6.4. We evaluate all the models with an input size of 3 x 512 x 512 and an output size of 3 x 128 x 128 on an NVIDIA GeForce GTX 1080Ti GPU. *IndoLayout* layout is twice as fast, with a lower memory footprint, while showing superior performance.

6.7 Failure Cases

While our model shows improvements on several fronts, we also notice certain instances where the network either incorrectly hallucinates regions correctly or fails to produce sharp outputs. We display one such instance in Figure 6.2. Further, in some instances, multiple objects are combined together, suggesting that the small gaps between objects can confuse the network.

We provide an additional reference to highlight further shortcomings of our model in Fig. 6.4. We note the following conditions that our model is susceptible to for making erroneous predictions:

1. **Difference in texture of floor, wall or furniture:** The model seems to overrely on textures, rather than shapes, in the observed image for extracting information relevant to occupancy prediction.

2. **Poor contrast between foreground and background objects:** If the color of walls or floor matches that of the obstacles, the model confuses and merges them at various points inspite of a visible boundary separating them in the input image.

3. **Difference in room setting or construction:** The model does not generalize well to significantly different unseen environments. The first two rows belong to a church which has a markedly different layout than that of an apartment which leads to the poor performance observable in the predictions.

4. **Difference in lighting:** The model's accuracy in occupancy prediction takes a performance hit when the lighting of the room differs as shown in the second last row. The use of data augmentation techniques such as color shift are not enough to provide robustness against directional lighting changes.

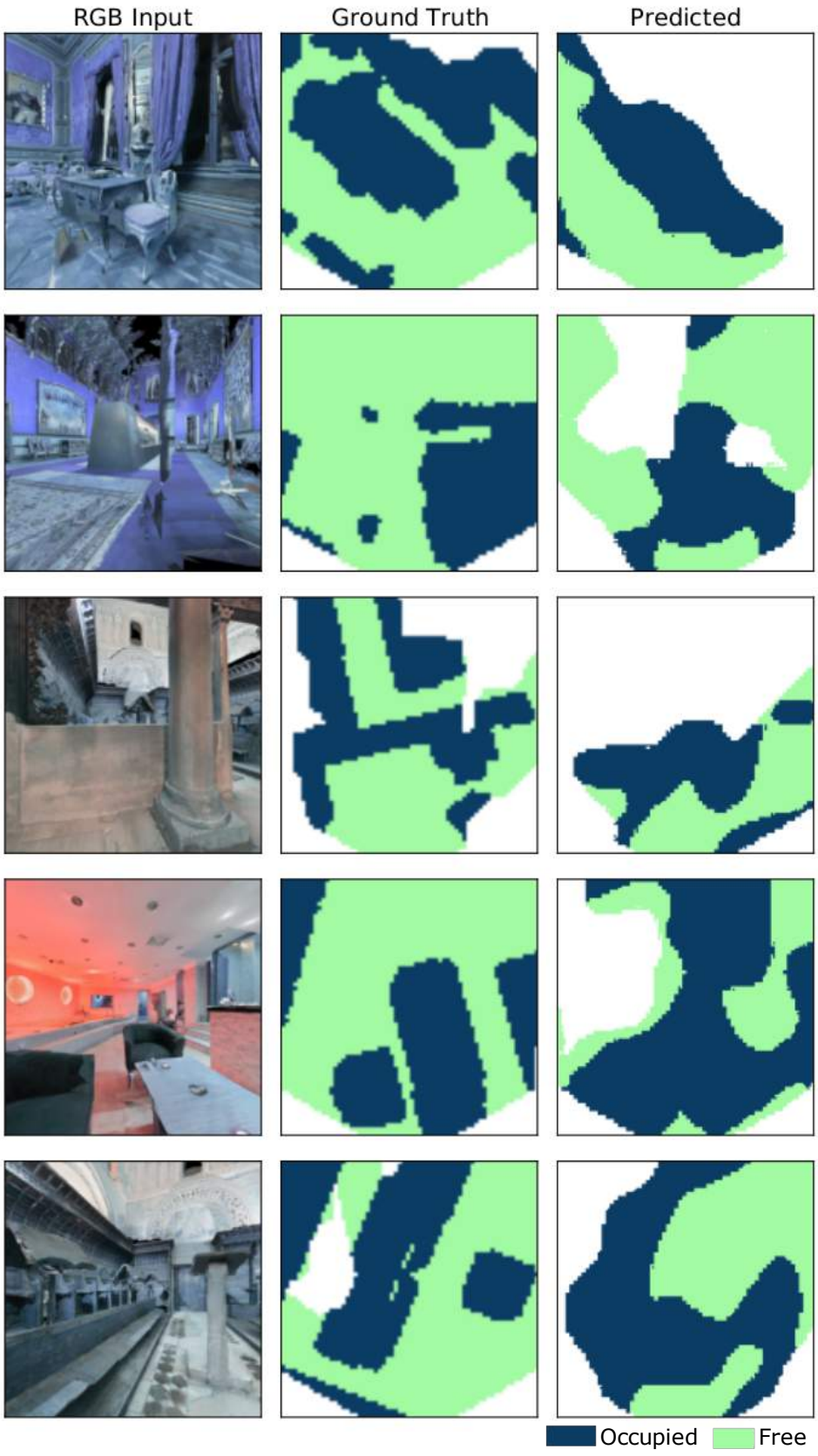

Figure 6.4: Failure cases

Chapter 7

Conclusions and Future work

In this thesis, we propose *IndoLayout*, a real-time network that predicts amodal layouts in indoor scenes using just an RGB image. Our analysis across multiple photorealistic indoor datasets shows the efficacy of our proposed network, which leverages attention and surpasses prior work by a significant margin in quantitative and qualitative studies. We further validate the model's task relevant learning by visualizing its saliency maps as well as monitoring its performance in the downstream task of point to point navigation and mapping. This helps in establishing that the architecture we propose for Indolayout is useful for learning better representations in indoor environments and has relevant applications in the field of robotic vision and navigation.

Further as part of this thesis, we provide a new dataset *Indoscene layout* (3.1) that can be used to benchmark layout estimation models for indoor environments collected from several different datasets. Using occupancy maps directly from the simulator as labels would be ill-posed as it will force the models to unreasonably predict for regions beyond walls and or behind obstacles that completely occlude the view. In order to alleviate this, we propose a preprocessing pipeline that takes into account the wall boundaries and tall obstacles to limit the occupancy prediction to reasonable regions. We share the details of our entire pipeline as well as the generated maps for over 200 scenes and 150k samples as part of the dataset.

Lastly, we provide a short insight into the limitations of monocular depth estimation methods for indoor scenes with regards to the task of layout estimation. Despite significant progress, the results obtained using such approaches are still far from usable for obstacle avoidance and efficient planning. Further research is required that leverages better geometric cues to help these algorithms generalize better to unseen environments.

Related Publications

7.1 Relevant Publications

1. **Shantanu Singh***, Jaidev Shriram*, Shaantanu Kulkarni, Brojeshwar Bhowmick, and K. Madhava Krishna, *"IndoLayout: Leveraging Attention for Extended Indoor Layout Estimation from an RGB Image,"* **2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)**

Bibliography

[1] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. *Computing Research Repository (CoRR)*, abs/2006.13171, 2020.

[2] S. F. Bhat, I. Alhashim, and P. Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[3] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic slam. In *International Conference in Robotics and Automation (ICRA)*, pages 1722–1729, 2017.

[4] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[5] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8740–8749, 2019.

[6] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural slam. In *International Conference on Learning Representations (ICLR)*, 2020.

[7] B. Cheng, R. Girshick, P. Dollár, A. C. Berg, and A. Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15334–15342, 2021.

[8] J.-B. Cordonnier, A. Loukas, and M. Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[10] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[11] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[13] R. Garg, V. K. Bg, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

[14] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[15] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis. Learning to map for active semantic goal navigation. In *International Conference on Learning Representations (ICLR)*, 2022.

[16] N. Girard, D. Smirnov, J. Solomon, and Y. Tarabalka. Polygonal building segmentation by frame field learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[17] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 27, 2014.

[19] V. Hedau, D. Hoiem, and D. A. Forsyth. Recovering the spatial layout of cluttered rooms. *International Conference on Computer Vision (ICCV)*, pages 1849–1856, 2009.

[20] F. Homm, N. Kaempchen, J. Ota, and D. Burschka. Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection. In *2010 IEEE Intelligent Vehicles Symposium*, pages 1006–1013, 2010.

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017.

[23] J. Jiao, Y. Cao, Y. Song, and R. Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[24] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? *Robotics and Automation Letters (RA-L)*, 5:6670–6677, 2020.

[25] K. D. Katyal, A. Polevoy, J. L. Moore, C. Knuth, and K. M. Popek. High-speed robot navigation using predicted occupancy maps. *International Conference in Robotics and Automation (ICRA)*, pages 5476–5482, 2021.

[26] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*.

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[28] C. Liu, A. G. Schwing, K. Kundu, R. Urtasun, and S. Fidler. Rent3d: Floor-plan priors for monocular layout estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3413–3421, 2015.

[29] C. Liu, J. Wu, and Y. Furukawa. Floornet: A unified framework for floorplan reconstruction from 3d scans. In *European Conference on Computer Vision (ECCV)*, pages 201–217, 2018.

[30] C. Lu, M. van de Molengraft, and G. Dubbelman. Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. *Robotics and Automation Letters (RA-L)*, 4:445–452, 2019.

[31] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5667–5675, 2018.

[32] K. Mani, S. Daga, S. Garg, S. S. Narasimhan, M. Krishna, and K. M. Jatavallabhula. Monolayout: Amodal scene layout from a single image. In *Winter Conference on Applications of Computer Vision WACV)*, pages 1689–1697, 2020.

[33] P. Moghadam, W. S. Wijesoma, and D. J. Feng. Improving path planning and mapping based on stereo vision and lidar. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pages 384–389, 2008.

[34] C. Mura, O. Mattausch, A. Jaspe Villanueva, E. Gobbetti, and R. Pajarola. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics*, 44:20–32, 2014.

[35] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[36] J. Philion and S. Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision (ECCV)*, 2020.

[37] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman. Occupancy anticipation for efficient exploration and navigation. In *European Conference on Computer Vision (ECCV)*, pages 400–418. Springer, 2020.

[38] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. M. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, and D. Batra. Habitat-matterport 3d dataset

(HM3d): 1000 large-scale 3d environments for embodied AI. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2021.

[39] T. Roddick and R. Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11138–11147, 2020.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115:211–252, 2015.

[41] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *International Conference on Computer Vision (ICCV)*, 2019.

[42] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker. Learning to look around objects for top-view representations of outdoor scenes. In *European Conference on Computer Vision (ECCV)*, 2018.

[43] Z. Seymour, K. Thopalli, N. C. Mithun, H.-P. Chiu, S. Samarasekera, and R. Kumar. Maast: Map attention with semantic transformers for efficient visual navigation. *International Conference in Robotics and Automation (ICRA)*, pages 13223–13230, 2021.

[44] Z. Shen, L. Kästner, and J. Lambrecht. Spatial imagination with semantic cognition for mobile robots. *International Conference on Intelligent Robots and Systems (IROS)*, pages 2174–2180, 2021.

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

[46] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2417–2426, 2019.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[48] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[49] M. Wei, D. Lee, V. Isler, and D. Lee. Occupancy map inpainting for online robot navigation. In *International Conference in Robotics and Automation (ICRA)*, pages 8551–8557, 2021.

[50] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9068–9079, 2018.

[51] S.-T. Yang, F.-E. Wang, C.-H. Peng, P. Wonka, M. Sun, and H. kuo Chu. Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3358–3367, 2019.

[52] W. Yang, Q. Li, W. Liu, Y. Yu, Y. Ma, S. He, and J. Pan. Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[53] Z. Yu, L. Jin, and S. Gao. P^2 net: Patch-match and plane-regularization for unsupervised indoor depth estimation. In *European Conference on Computer Vision*, pages 206–222. Springer, 2020.

[54] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 7354–7363. PMLR, 2019.

[55] Y. Zhao and S.-c. Zhu. Image parsing with stochastic scene grammar. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24. Curran Associates, Inc., 2011.

[56] J. Zhou, Y. Wang, K. Qin, and W. Zeng. Moving indoor: Unsupervised video depth learning in challenging environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8618–8627, 2019.

[57] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.