2020

Online Synthesis Of Speculative Building Information Models For Robot Motion Planning

Armon Shariati
University of Pennsylvania

Online Synthesis Of Speculative Building Information Models For Robot Motion Planning

Abstract

Autonomous mobile robots today still lack the necessary understanding of indoor environments for making informed decisions about the state of the world beyond their immediate field of view. As a result, they are forced to make conservative and often inaccurate assumptions about unexplored space, inhibiting the degree of performance being increasingly expected of them in the areas of high-speed navigation and mission planning. In order to address this limitation, this thesis explores the use of Building Information Models (BIMs) for providing the existing ecosystem of local and global planning algorithms with informative compact higher-level representations of indoor environments. Although BIMs have long been used in architecture, engineering, and construction for a number of different purposes, to our knowledge, this is the first instance of them being used in robotics. Given the technical constraints accompanying this domain, including a limited and incomplete set of observations which grows over time, the systems we present are designed such that together they produce BIMs capable of providing explanations of both the explored and unexplored space in an online fashion. The first is a SLAM system that uses the structural regularity of buildings in order to mitigate drift and provide the simplest explanation of architectural features such as floors, walls, and ceilings. The planar model generated is then passed to a secondary system that then reasons about their mutual relationships in order to provide a water-tight model of the observed and inferred freespace. Our experimental results demonstrate this to be an accurate and efficient approach towards this end.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Computer and Information Science

First Advisor

Camillo J. Taylor

Keywords

Building Information Modeling, Mapping, Robot Perception, SLAM

Subject Categories

Computer Sciences | Robotics

ONLINE SYNTHESIS OF SPECULATIVE BUILDING INFORMATION MODELS FOR ROBOT MOTION PLANNING

Armon Shariati

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2020

Supervisor of Dissertation

Camillo J. Taylor, Raymond S. Markowitz President's Distinguished Professor

Graduate Group Chairperson

Rajeev Alur, Zisman Family Professor and Graduate Group Chair

Dissertation Committee:

Kostas Daniilidis, Ruth Yalom Stone Professor, University of Pennsylvania
Yasutaka Furukawa, Associate Professor, Simon Fraser University
Jean Gallier, Professor, University of Pennsylvania
Jianbo Shi, Professor, University of Pennsylvania

ONLINE SYNTHESIS OF SPECULATIVE BUILDING INFORMATION MODELS FOR ROBOT MOTION PLANNING

For my parents, Nazly and Saeid, the giants on whose shoulders I stand.

ACKNOWLEDGEMENT

In addition to my committee, I would like to begin by thanking all those other incredible teachers and mentors I have had over the years that have brought me to this point. Robert Cherdack, for believing in a ne'er-do-well and being the first to spark that flame of curiosity within me 10 years ago. Sharon Kalafut, for introducing me to my field and helping me navigate my early academic career. Michael Spear, my friend and first advisor, for setting a fledgling computer science student on the path to realize his passion with so much personal care and support. John Spletzer, for providing me with all the means to explore my discipline despite my inexperience. Kostas Daniilidis, Jean Gallier, Katherine Kuchenbecker, and Max Mintz for continuously going out of their way to provide me with knowledge, resources, and friendship. Finally, my advisor, CJ Taylor, not only for giving me the opportunity of a lifetime to pursue my education at the highest levels, but for being such an endless source of patience, wisdom, and assurance.

I would also like to thank my Penn family of peers, who were constant sources of support and camaraderie. Carlos Esteves, Steven Phillips, Alex Zhu, and Chao Qu, my colleagues in name and older brothers in spirit, for helping me steer through the challenging waters of the PhD program and providing me with countless hours of tutelage on the myriad of complex topics we encountered. Bernd Pfrommer, for playing a critical role in collecting the data necessary to showcase my findings.

Finally, I would like to thank my closest friends and family. Casey, Cooper, Dan, Eric, Jake, Joey, Max, and Tony, for lifting me up after every trial I encountered and being such everlasting sources of joy and laughter in the best and worst of times. My love, Alexandra, for always being by my side and reminding me of the light at the end of the tunnel. My brother, PJ, for helping me to believe in myself and for making me become a better role model along the way. Lastly, my parents, Nazly and Saeid, for instilling in me the passion, fortitude, and unquenchable thirst for knowledge, which made realizing this dream possible.

ABSTRACT

ONLINE SYNTHESIS OF SPECULATIVE BUILDING INFORMATION MODELS FOR ROBOT MOTION PLANNING

Armon Shariati

Camillo J. Taylor

Autonomous mobile robots today still lack the necessary understanding of indoor environments for making informed decisions about the state of the world beyond their immediate field of view. As a result, they are forced to make conservative and often inaccurate assumptions about unexplored space, inhibiting the degree of performance being increasingly expected of them in the areas of high-speed navigation and mission planning. In order to address this limitation, this thesis explores the use of Building Information Models (BIMs) for providing the existing ecosystem of local and global planning algorithms with informative compact higher-level representations of indoor environments. Although BIMs have long been used in architecture, engineering, and construction for a number of different purposes, to our knowledge, this is the first instance of them being used in robotics. Given the technical constraints accompanying this domain, including a limited and incomplete set of observations which grows over time, the systems we present are designed such that together they produce BIMs capable of providing explanations of both the explored and unexplored space in an online fashion. The first is a SLAM system that uses the structural regularity of buildings in order to mitigate drift and provide the simplest explanation of architectural features such as floors, walls, and ceilings. The planar model generated is then passed to a secondary system that then reasons about their mutual relationships in order to provide a water-tight model of the observed and inferred freespace. Our experimental results demonstrate this to be an accurate and efficient approach towards this end.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Chapter 1

Introduction

Upon entering a building we have never visited, our brain readily anticipates (at least roughly) where any adjacent hallway may lead. We are readily able to disentangle vast amounts of clutter from the room layout. We can recognize objects even in the face of severe occlusion and reason about their spatial extent. More fascinating still is how our understanding of the space and our ability to speculate continues to improve as we move through it. It is our understanding of indoor spaces as human beings which provides us with the necessary context for inferring the nature of space beyond our field of view. Given a door, we can be reasonably certain it leads to another hallway or room. When approaching an intersection of two hallways, we expect them to continue even though we may not be able to see around the corner. Barring the possibility we find ourselves in a Hollywood movie, we do not think to look for an exit behind a book case or under a table. Unfortunately, this level of intelligence is still grossly lacking in autonomous mobile robots today, which greatly degrades the high degree of performance expected of them.

In this dissertation, we aim to bridge this gap by providing robots with the ability to incrementally construct water-tight models of the buildings in which they operate, which includes the position and extent of all walls, floors, ceilings, and doors, as well as objects from a

small set of classes. These models, often referred to as Building Information Models (BIMs) (Tang et al., 2010) have often been used in architecture, engineering, and construction for visualization, design, and space planning (Administration, 2003), but to the extent of our knowledge, have not been used in the context of robotics. We assert that equipped with these compact higher-level representations of the environment, existing motion planning algorithms can be made more efficient and robust, as BIMs not only provide a semantic understanding of disjoint spaces, such as rooms and hallways, but also provide an intuitive prior for inferring the presence of various structures beyond the visible portions of the scene.

We proceed by constructing a structure-aware Simultaneous Localization and Mapping (SLAM) system, described in Chapter 3, which uses infinite layout planes as landmarks for mapping. After detecting, associating, and localizing the different planes, they are then passed to an algorithm, presented in Chapter 4, which reasons about their mutual relationships in order to bound and infer freespace corresponding to distinct rooms and hallways, producing a best estimate for the building information. We will see that by providing this algorithm with an understanding of structural continuity in the form of planes, it is able to hallucinate regions of space that are occluded or yet to be explored. The efficacy of our system will be evaluated qualitatively on its ability to produce BIMs which are compact, flexible, and descriptive, and quantitatively on its computational speed and the accuracy of its estimated layouts.

1.1 Problem Statement

Input: A sequence of depth and/or intensity images $\{(I_{depth}, I_{intensity})^{(i)}\}_{i=1,\dots,n}$ as keyframes, alongside initial estimates of frame-to-frame motion $\{G^{(i)} = (R,t)^{(i)}\}_{i=1,\dots,n-1}$.

Output: A complete Building Information Model (BIM), which includes:

- A set of disjoint polyhedral regions $\mathcal{R}$, where each region is defined by a set of 3D layout planes $P = \{\pi_j\}$ bounding the inferred freespace it spans and a class label s denoting the region type, *i.e.* room or corridor;

- A set of objects $O = \{o_k\}$, each parameterized by its bounding box width w, height h, length l, center (m^x, m^y, m^z), and orientation (ϕ, ψ, θ), that respects real-world physics (no interpenetration with the layout or other objects), as well as its object class label s;

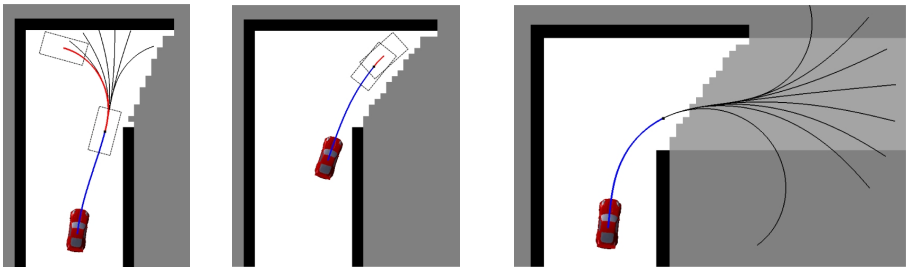
- A set of doorways $\mathcal{D}$, each parameterized by its width w, height h, and center (m^x, m^y), embedded within a unique plane π with which it is associated;

Observe that this representation fits neatly in conventional mapping-planning frameworks, as transforming a BIM to the corresponding voxel map representation amounts to simple discrete spatial sampling.

1.2 Motivation

Many advanced high-speed autonomous systems today, including the Fast Lightweight Autonomy platform (FLA) developed by our own research group (Mohta et al., 2018), continue to rely on costly overparameterizations of space such as voxel grids. This is primarily due to the flexibility they provide. Making no assumptions about the environment, voxel grids are easily updated through a simple ray tracing procedure of the 3D points provided by onboard visual sensors. However, these representations are not only resource intensive, but also suffer several other major shortcomings which fetter performance.

Chief among them is that extrapolating spatial structure beyond the range of the onboard sensors can perceive is challenging due to the inherent complexity of such models. Without this faculty and no prior map to rely on, even the most advanced planning systems are forced to make inaccurate and unhelpful assumptions that naturally either compromise the safety or inhibit the performance of these robotic systems. Typically, if speed is a priority then one often assumes that unobserved space is free, but this often results in dangerous and unsustainable behavior. Alternatively, one may assume that the unobserved space is occupied if safety is the primary concern, however, prohibitively cautious behavior tends to occur as a result. An example of such a performance degradation can be seen in Figure



(a) (b) (c)

Figure 1: Implicit safety constraints can cause a robot to decrease its speed from 4 m/s in (a) down to 1 m/s in (b) as it approaches a blind corner such as this one. Alternatively, we would like to leverage what we understand about the environment in order to provide the planner with a prediction of what could lie ahead so that it may commit to a trajectory with a constant speed into the unknown with a well informed notion of risk as seen in (c) (Richter et al., 2018).

1, where a robot must drastically reduce its speed as it approaches a blind corner in order to ensure its safety. Ultimately, we would like our robots to understand enough about the environment so that they may balance speed and risk intelligently.

A naive approach in attempting to build such a predictive system would be to try and directly model an accurate prior distribution over the probabilistic space of all real-world environments. This alone is an incredibly challenging undertaking due to the high dimensionality of even modest sized occupancy maps and the richness of natural and made-made environments. The approach demonstrated by Richter et al. (Richter et al., 2018), proposes to address the problem in planning space by learning a mapping from images to actions. In contrast, we recognize that the set of maps corresponding to environments we would expect to see does not evenly cover the space of possible maps. There is in fact an abundance of structure and a good degree of foreseeable patterns in real-world environments which induces dependencies between observations and reduces the probabilistic space to a set of concentrated regions.

Another limitation of voxel grids which we seek to address is that the use of ray tracing for updating voxel occupancy creates a dependence on the quality of the given sensor providing the 3D points for producing accurate maps of the environment. Unfortunately, in practice

even the highest quality sensors are subject to noise, which yields discontinuous non-smooth surfaces. More detrimental still for motion planning are the phantom obstacles that are often introduced as well. These phantom obstacles, which arise from phase ambiguities, can greatly compromise safety when navigating in sensitive areas such as doorways. We assert that higher-level representations can be used to mitigate noise through the regularization effect provided by abstraction.

In addition to the limitations imposed by the use of voxel grids at the local planning level, the absence of a semantically meaningful map representation means that global planning problems, which should be in the form of "find a path from Room 1 to Room 2" take the form of "find a path from coordinate (x_1, y_1, z_1) to (x_2, y_2, z_2)". Notice that the former leads to a more functional view of space, in that doorways and adjacent corridors act as natural transitions between disjoint rooms. In principle, this should lead to more efficient paths as the system can now avoid common pitfalls such as planning a path through occluded room boundaries. Observe that this is not a problem which is easily overcome with a simple semantic segmentation of the voxel map.

1.3 Challenges

In order to serve as an effective holistic solution to the problem at hand, our system must be able to perform the following three tasks concurrently: (1) localize objects in 3D; (2) model building infrastructure; and (3) integrate these estimates from each frame into a consistent water-tight 3D model of both the observed and unobserved space over time. However, each of these components in isolation faces its own set of obstacles.

Although much research has been done in the vein of 3D object detection over the past decade, indoor environments pose several unique obstacles to even state of the art detectors. In the face of excessive clutter, for instance, it becomes more difficult to disentangle different object instances due to different objects occluding one another. This effect is further compounded by the amodal nature of first-person perception, which introduces additional self-occlusions. The objects contained within indoor environments can also span a

large number of classes, each with a potentially large degree of intra-class variation between different instances.

Constructing BIMs typically involves approximating the building structure using sets of constrained geometric primitives such as lines and planes, where the constraints arise from different assumptions made about how the building layout is expected to unfold. However, despite the highly regular structural patterns among buildings, variations and pathological cases may still exist. In which case, overly constrained or simplified models may no longer be able to adequately capture the richness of the space. Clutter also factors into layout estimation as it typically accumulates around the edges of room boundaries, which makes extracting individual layout components for approximation difficult. Moreover, supposing that different components can be easily detected and approximated, constructing a complete model of the environment online, with only limited information, remains a challenging problem.

Integrating 3D measurements across multiple fields of view has fortunately been a long studied problem. However, despite the advances made in SLAM, the issue of data association has received limited attention. Without a means for robustly associating observations in one frame to those seen at another point in time, the possibility of a false alignment or incorrect loop closure, which can irrevocably mar the correctness of the reconstruction, remains ever present. This problem is made more difficult in the presence of drift, which places different observations of the same landmark at different positions. Although certain landmarks are highly distinct, others, such as planar surfaces corresponding to walls, floors, or ceilings, often lack visibly distinguishable characteristics. One solution may be to detect sufficient spatial overlap between sequential frames, however, how does one associate different observations over extended gaps in time besides using a distance-based heuristic? This further suggests that in order for a system to be a robust solution for the task at hand, it must be able to revise past decisions in the light of new information.

Chapter 2

Related Work

2.1 Building Information Modeling

Given a complete set of registered point cloud scans of a building interior, methods for constructing Building Information Models (BIMs) aim to extract all the major architectural components including walls, floors, ceilings, and sometimes doors and windows; effectively compressing the information contained in the point cloud into a compact water-tight model of the environment that requires few parameters to encode (Tang et al., 2010). These models are often used to support some task in architecture, engineering, or construction, such as visualization during design, detecting construction errors, and simulation and space planning during facility management (Administration, 2003). However, in this thesis we will explore their use for robot navigation and planning.

As indoor environments typically conform to a highly regular structure, constructing BIMs is typically achieved by approximating layout components with a set of geometric primitives such as lines, planes, and cuboids. Although true BIMs are supposed to be three dimensional, we consider methods for constructing 2D floor plans as a subset of scan-to-BIM systems.

2.1.1 Floor Plan Models

Arguably the simplest model for indoor environments are floor plans. Given that floor plans model a 2D projection of the space, line segments are often used to approximate the location of walls (Okorn et al., 2010; Turner and Zakhor, 2014; Cabral and Furukawa, 2014). Okorn et al. filter points corresponding to the floor and ceiling, project the remaining points to the ground plane, and then detect line segments using a Hough transform in a histogram of point densities (Okorn et al., 2010). Cabral and Furukawa follow a similar approach, but instead of using a Hough transform, they compute a shortest path over points to detect wall boundaries (Cabral and Furukawa, 2014).

Although these methods may produce a model from which a human may infer freespace, a collection of line segments alone cannot explicitly provide this level of understanding. In order for a system to do so, it must be able to reason at the level of rooms (Turner and Zakhor, 2014; Liu et al., 2018a; Luperto et al., 2019). (Turner and Zakhor, 2014) perform room segmentation after triangulating boundary points with scanner positions. In (Liu et al., 2018a), the authors demonstrate a less engineered method than those mentioned so far by taking a data driven approach using the latest advancements in deep learning (Liu et al., 2018a). They first train a network to predict the locations of various types of corners in a map of projected point density image, which are then grouped together to form rooms using a linear program. Although their system displays impressive results in producing complete floor plans, which even include object models, acquiring detailed floor plan models in real-world environments for supervised training can be time consuming and expensive. Luperto et al. are among a few that have begun addressing the problem of predicting layout given only partial observations though take a more traditional approach starting from line segments denoting walls and then estimating room boundaries (Luperto et al., 2019).

In summary, while floor plans are among the most succinct representation for a building, their fundamental limitation is their inability to represent 3D space.

2.1.2 3D Building Models

Addressing the limitations of 2D floor plans, numerous methods have proposed for approximating the complete 3D space.

As indoor environments typically conform to a highly planar structure, sets of oriented planes are a natural choice for modeling the planar surfaces inherent in indoor environments (Hähnel et al., 2003; Thrun et al., 2004; Furukawa et al., 2009a; Sinha et al., 2009; Chauve et al., 2010; Budroni and Boehm, 2010; Sanchez and Zakhor, 2012; Vanegas et al., 2012; Xiong et al., 2013; Mura et al., 2014; Oesau et al., 2014; Monszpart et al., 2015; Ochmann et al., 2016; Armeni et al., 2016). A relatively straightforward approach is to directly fitting planes to point cloud data by region growing (Hähnel et al., 2003; Chauve et al., 2010), RANSAC (Sanchez and Zakhor, 2012), or detecting peaks in histograms of point density (Sinha et al., 2009). Although these methods yield a reasonable approximation, without some form of regularization to bound model complexity in terms of the number of planes (Thrun et al., 2004) general unconstrained planar models such as these are susceptible to noise from clutter. Furthermore, they do not easily lend themselves to interpretation. Given a set of arbitrarily oriented planes it may be challenging to infer which truly correspond to walls, floors, and ceilings, and which simply arise from clutter.

In order to reduce model complexity, increase robustness to clutter, and provide greater understanding, it is useful to leverage prior knowledge concerning the regularity of indoor environments in order to impose constraints on plane orientations and their mutual relationships. Assuming walls are typically aligned to one of two orthogonal axis, *i.e* a Manhattan World (Coughlan and Yuille, 1999), is a common approach to this end (Furukawa et al., 2009a; Budroni and Boehm, 2010; Vanegas et al., 2012; Xiong et al., 2013; Mura et al., 2014; Oesau et al., 2014; Monszpart et al., 2015; Armeni et al., 2016). After detecting this Manhattan frame, Budroni and Boehm use two planar sweeps, one for each orthogonal axis, in order to find the offset of dominant vertical walls (Budroni and Boehm, 2010). Instead of detecting a single activation meant to correspond with a single wall surface, Armeni

et al. use a bank of convolutional filters to detect a volumetric wall signature: two peaks in the 1D histogram of point density, which corresponds to two surfaces bounding void space (Armeni et al., 2016). We note that they are among the few that also focus on modeling object-layout relationships in their BIMs (Armeni et al., 2019). In (Vanegas et al., 2012), the authors propose using a set of predefined shape templates in order to classify points in local neighborhoods as belonging to a Manhattan-aligned wall, corner, or edge, and then clustering groups of similarly labeled points to recover the greater geometry of the building. Xiong et al. detect planar patches in a voxel grid, after discretizing the input point cloud, and then classify each patch as wall, floor, ceiling, or clutter using a context-based machine learning algorithm (Xiong et al., 2013). The approach most similar to our own in the context of constrained plane detection is the work of Monszpart et al.. Formulating the problem of modeling as one of model selection, their system detects the minimum set of planes required to explain the observations (Monszpart et al., 2015). In contrast to our approach however, they perform their search over the space of all possible models enumerated explicitly, whereas we search over the space of enumerated correspondences.

Observe that much like how a collection of line segments alone cannot represent freespace in the case of floor plans, a collection of planes alone cannot represent freespace in the 3D setting. However, both lines and planes are still useful as they can be used as a guide for volumetric reasoning. Mura et al. first project and cluster detected planar patches to form a set of a representative lines partitioning the space into a cell complex, then compute diffusion distances to cluster grid cells into separate rooms (Mura et al., 2014). Oesau et al. follow a similar approach, except that their lines are extracted from a 2D projection of points (Oesau et al., 2014). In (Ochmann et al., 2016), the authors continue to build upon these previous works to model wall thickness and the mutual arrangements of neighboring rooms. Note than this space partitioning approach is non-recursive, as compared to the familiar *binary space partitioning* scheme (Chauve et al., 2010). Ultimately, their approach to modeling freespace is to label all cells in the cell complex as either free or occupied. Instead of this standard labeling scheme, Furukawa et al. propose labeling cells in a voxel

grid as either on the interior of the bounding surface, or the exterior (Furukawa et al., 2009a). Xiao and Furukawa use lines to enumerate rectangles and subsequently cuboids to cover groups of free voxels in a scheme called Constructive Solid Geometry (Xiao and Furukawa, 2014). Although we explore a similar greedy algorithm, we make two major modifications in order to enable our system to perform speculation into unexplored regions of the environment. The first is that our cuboids are enumerated based on infinite layout planes, which extend beyond our field of view, instead of finite segments. And second, our cost function for a given cover does not penalize cuboids that encompass unobserved voxels. As a part of our future proposed work, we also seek to generalize our model based on cuboid primitives to include convex polygons. In (Mura et al., 2016), the authors detect planar patches and then use a ceiling-to-floor structural grammar to constrain the structural relationships between adjacent planes in order to recover individual rooms. This approach has been demonstrated to provide a significant amount of robustness as it is able to utilize local appearance information and global geometric information for inference. We note that (Mura et al., 2016) is the only approach capable of recovering a complete 3D representation of the space, as opposed to the more standard 2.5D representation which assumes floors and ceilings are always orthogonal to the gravity vector. While Ikehata et al. also utilize a structural grammar, except in their case for inferring the relative arrangements of rooms, they perform volumetric segmentation of rooms prior to estimating wall boundaries (Ikehata et al., 2015), much like (Jung et al., 2018).

Although it produces a more complex mesh representation and, in practice, is better thought of an extension of the scene completion methods to be discussed in Section 2.2 as opposed to a method for building information modeling, the work of Dai et al. is worth noting as it may signal the introduction of deep learning towards solving the scan-to-BIM problem as the scalability of 3D learning continues to improve (Dai et al., 2018). Their method performs a semantic segmentation of the voxels and demonstrates a reasonable capacity for filling in large missing spatial extents in 3D scans.

2.1.3 Summary

Over the years these scan-to-BIM methods have continued to improve in terms of robustness (Liu et al., 2018a) and accuracy (Jung et al., 2018) in constructing compact representations of indoor environments. However, in contrast to the methods thus described, with the exception of a few (Thrun et al., 2004), our primary goal is to synthesize BIMs for robotic applications, which in turn imposes constraints and requirements not often recognized in this current body of literature.

The most important of which is the requirement that our system be able synthesize BIMs given only the types of incomplete sequential measurements collected during robotic exploration. While a small body of research is beginning to emerge which includes online interactive floor plan generation using mobile devices (Angladon et al., 2018), to our knowledge all of the methods for large-scale automatic BIM construction are designed as batch systems requiring all of the relevant data be available at once. In contrast, our solution must be able to revise and update the model as new information becomes available in a timely manner.

The second major requirement of a scan-to-BIM system for it to be useful for path planning is that it needs to provide an explicit representation of free and occupied space. Moreover, the system must be able to not only infer the nature of unobserved space that arises from occlusion due to amodal perception, but it must also actively speculate as to the presence of freespace beyond the range of the given set of measurements. Although methods such as (Chauve et al., 2010; Oesau et al., 2014; Mura et al., 2014; Xiao and Furukawa, 2014; Turner and Zakhor, 2014; Ikehata et al., 2015; Armeni et al., 2016, 2019; Mura et al., 2016; Ochmann et al., 2016; Liu et al., 2018a; Jung et al., 2018) model freespace, they do not reason outside the observed boundaries of the space. The only structural elements used to bound the freespace are those that are directly seen and are the only ones that appear in the final model. Though methods for predicting 2D layout beyond a field of view (Luperto et al., 2019) and large-scale 3D scene inpainting (Dai et al., 2018) have begun to appear in the

literature, the problem thus far has received limited attention from the broader community. The idea of using infinite layout planes as a *ghost prior* for anticipating the location of occluded walls is proposed in (Liu et al., 2001) and (Chauve et al., 2010). However, our contribution is to extend this intuition for inferring the presence of entire volumetric entities such as rooms, given only partial observations of its bounding segments.

Other more minor requirements include the ability of the scan-to-BIM system to construct representations of multi-floor buildings and to include doors in the final model. As is the case with almost all sensing modalities, laser scanners are susceptible to noise, which can result in them producing phantom obstacles in the form of spurious points in the point cloud. If a robot is in the process of navigating through a doorway, these noisy points can significantly degrade performance as the robot can be made to believe there is no gap large enough for it to traverse through. We believe including an explicit representation of doorways in the manner of (Xiong et al., 2013; Ikehata et al., 2015; Ochmann et al., 2016; Jung et al., 2018) can provide a greater degree of robustness in performing such tasks.

2.2 3D Indoor Scene Understanding

3D understanding of indoor scenes from a single view has long been at the forefront of computer vision research. While most methods are often focused entirely on either object detection or layout estimation, in recent years much effort has been expended in approaches which estimate both jointly. Unlike the methods in Section 2.1, which take a complete set of registered building scans as input, those presented here process a single RGB, panoramic RGB, or RGBD image captured at an unknown pose, and are primarily focused on maximizing what can be understood of the scene given the limited scope. Due to the nature of this type of data, these system must also contend with occlusion due to amodal perception. Furthermore, while most scan-to-BIM methods typically treat clutter as noise, many of the methods discussed here often model objects explicitly.

2.2.1 Object Detection

Detecting objects in 2D images had long been an open problem since the early 2000's (Viola and Jones, 2001) until the seminal work of Girshick et al. (Girshick et al., 2014). While many early techniques garnered much success using hand-crafted features in tandem with traditional machine learning (Felzenszwalb et al., 2010), modern state of the art approaches, such as Faster-RCNN (Ren and Sudderth, 2016), YOLO (Redmon et al., 2016), and Mask-RCNN (He et al., 2017), rely on end-to-end deep neural networks; many of which build upon the original RCNN system (Girshick et al., 2014). For the sake of brevity, we provide only a cursory review of these methods here.

Unfortunately, many of these methods are limited to producing only 2D bounding boxes. Although there exist monocular approaches capable of generating 3D object detections (Tekin et al., 2018), and even those that can do so to-scale using additional context and size priors of semantic classes (Chen et al., 2016), the problem is typically more difficult without additional geometric information. As a result, systems for 3D object detection are typically designed assuming RGBD information is available; leveraging the depth information they provide. Among the first of these systems is that of Song and Xiao, who use a sliding window approach for detecting objects in the 3D point cloud (Song and Xiao, 2014). Of course, given the dimensionality of the space, this technique is rather slow (25 min $\times$ number of object categories). Despite the increase in performance and efficiency in their follow up work which utilizes a trained Region Proposal Network (RPN), inference time is still takes about 20 seconds for each forward pass (Song and Xiao, 2016). In (Gupta et al., 2015), the authors propose an alternative approach which entails aligning 3D CAD models to a 2D instance segmentation mask computed using (Gupta et al., 2014). They are the first to note that using 2D information in order to constrain the 3D search is both more efficient and accurate than reasoning in the 3D space directly. Lahoud and Ghanem leverage the same insight to design a system that uses Faster-RCNN (Ren and Sudderth, 2016) to compute a 2D bounding box, which in turn defines a frustum in a point cloud within which they

regress an oriented 3D bounding box (Lahoud and Ghanem, 2017). This methodology of leveraging mature technologies for 2D bounding box detection as a seed for 3D bounding box regression has increasingly become standard for detecting 3D objects at arbitrary heights (Qi et al., 2018; Xu et al., 2018). These later methods have achieved state of the art results using network architectures which operate directly on point clouds (Qi et al., 2017). While some have proposed regressing 3D bounding boxes using the 2.5D RGBD representation, instead of converting the depth information into a point cloud first and losing local geometry (Deng and Latecki, 2017; Luo et al., 2017), others have argued that these techniques may be vulnerable to foreground noise and occlusion. A lighter-weight alternative, which can be thought of integrating the two approaches, is to integrate multi-view information from both the 2D forward view and the 2D Birds-Eye-View (BEV) perspective for 3D regression (Chen et al., 2017). Though these methods are limited to only detecting objects on the ground plane, this can be an acceptable assumption within certain contexts such as autonomous driving.

While appearance information from RGB images is quite useful for discerning smaller objects amongst clutter, given a few semantic classes corresponding to objects of a larger size, the geometric information available from point clouds has been demonstrated to be sufficient for robust 3D detection, particularly in the context of autonomous driving (Simon et al., 2018; Zhou and Tuzel, 2018). While the work of Simon et al., demonstrates a real-time approach using a Birds-Eye-View projection of the point cloud as input to a modified YOLO detector (Redmon et al., 2016), it assumes all objects reside on the ground plane (Simon et al., 2018). Meanwhile, VoxelNet (Zhou and Tuzel, 2018) imposes no such restriction. The primary drawback of their approach however is that they must first convert the sparse point cloud into a voxel grid, which is not only memory intensive, but requires greater care during training the CNN weights due to the large number of empty voxels. The state of the art among pure point cloud based approaches can be found in (Wang et al., 2018), where the authors train a PointNet (Qi et al., 2017) architecture for instance segmentation by learning a similarity matrix relating points.

2.2.2 Layout Estimation

While they are not designed specifically for modeling indoor layouts, methods such as (Hoiem et al., 2005a,b; Saxena et al., 2009; Liu et al., 2018b) provide valuable insight as to how appearance information from a single RGB image can be used to infer geometric properties of arbitrary planar scenes. These models are among the most complex discussed as they are intended to represent less structured environments. The pioneering work of Hoiem et al. propose learning appearance based models of geometric classes in order to label major surfaces such as building facades and the ground plane and infer their orientation in the image (Hoiem et al., 2005a,b). Saxena et al. train a Markov Random Field to infer a set of plane parameters corresponding to various patches in the image, as well as their mutual relationships (Saxena et al., 2009). Gupta et al. demonstrate a volumetric approach by using projected blocks to approximate segmented building structures in the image in order to infer additional 3D properties such as occlusion and stability (Gupta et al., 2010a). Most recently, Liu et al. demonstrate an approach which uses a scene attribute grammar to explicitly model a rich hierarchy of semantic regions and the geometric attributes of each component jointly (Liu et al., 2018b).

Often in a single image taken in an indoor environment, however, there exist far fewer visible planar segments compared to these outdoor scenes. Furthermore, planar regions seen indoors are often arranged in highly predictable patterns, which in principle should simplify layout inference.

Leveraging the 2.5D structure of most buildings, among the more successful and simple approaches is to detect the piecewise-linear floor-wall boundary (Delage et al., 2006). While the authors make no assumption regarding the relative orientation of adjacent walls, they do assume orthogonality between the ground plane and the vertical wall segments. However, assuming a Manhattan World (Coughlan and Yuille, 1999) can be useful as it can lead to an even simpler estimation task (Lee et al., 2009; Flint et al., 2011; Furukawa et al., 2009b). Recognizing clutter typically accumulates along the floor-wall boundary, Lee et al. model

the ceiling wall boundary (Lee et al., 2009). Their system detects sets of parallel lines aligned with one of the three dominant axes in order to constraint the position of corners where pairs of adjacent walls intersect one another. In (Flint et al., 2011), the authors utilize the same model but incorporate additional appearance information and information from multiple viewpoints into a Bayesian framework for more robust boundary inference. Furukawa et al. compute a point cloud from a set of calibrated images using a multi-view stereo algorithm, from which they then extract dominant Manhattan plane directions, enumerate planar hypotheses, and use a MRF to select the optimal configuration of planes to reconstruct the scene (Furukawa et al., 2009b). While some of these methods, such as (Flint et al., 2011), make attempts towards maintaining robustness against clutter, the majority do not discriminate between clutter and layout, as they are designed for to model general indoor scenes, as opposed to rooms specifically where clutter is more prevalent.

Although the first system that proposes using the single cuboid model is used to approximate extended corridors (Shakunaga, 1992), the model is more commonly used to provide the necessary robustness to clutter when approximating single room layouts (Hedau et al., 2009; Schwing et al., 2012; Schwing and Urtasun, 2012; Dasgupta et al., 2016; Lee et al., 2017a). This is due to the fact that, in addition to being subject to fewer parameters, appearance information from surfaces can be used to support the layout model instead of boundary edges alone, which often suffer from a greater degree of noise. Observe that implicit in the cuboid model is the assumption of a Manhattan World as well as convexity. Hedau et al. iteratively estimate box parameters and compute surface labels using a modified version of Hoiem et al. surface layout algorithm (Hoiem et al., 2007). Schwing et al. propose a method for structured prediction over a simplified parameterization of the cube model (Schwing et al., 2012). The authors improve upon their approximate inference scheme and present a method for exact inference in their follow up work (Schwing and Urtasun, 2012). Extending (Schwing et al., 2012) as well, in Zhang et al. the authors include depth information in order to jointly model layout and label clutter (Zhang et al., 2013).

With the advent of mature Convolutional Neural Network (CNN) architectures, which have achieved state of the art performance for image classification, many have been adapted for other tasks including layout estimation (Yang et al., 2016a; Dasgupta et al., 2016; Lee et al., 2017a). Yang et al. present a real-time approach for detecting floor-wall boundary model of Delage et al. by labeling pixels on the ground plane using a CNN, and fitting line segments to their boundary (Yang et al., 2016a). However, like other models of this variety, it is not robust to clutter which accumulates along the edges of rooms. As such it is primarily designed for corridor-like environments. Dasgupta et al. retrain a CNN designed for semantic segmentation in order to classify surface pixels according to which face of the cuboid they reside on (Dasgupta et al., 2016). The output belief maps output from the network are then passed to an optimizer to recover the box parameters. Lee et al. propose an end-to-end learning approach, where their network predicts a set of layout keypoints which completely specify the cuboid parameters (Lee et al., 2017a). As panoramic images provide greater visibility of the bounding layout components, some networks have been trained to recover more complex room layouts by predicting boundary edges through occlusion (Fernandez-Labrador et al., 2018; Zou et al., 2018). Due to the richness of the internal representations in these networks, they are often more robust to clutter than hand-crafted approaches.

In addition to Zhang et al., others have also turned to RGBD sensors in order to address this highly geometric estimation task (Taylor and Cowley, 2012; Guo and Hoiem, 2013). Taylor and Cowley extract dominant Manhattan aligned planes from the RGBD image and use Dynamic Programming in order to determine the optimal set of non-overlapping vertical wall segment intervals (Taylor and Cowley, 2012). While Taylor and Cowley are primarily focused on detecting vertical layout components, Guo and Hoiem estimate horizontal support surfaces (Guo and Hoiem, 2013). Much like us, they seek to use such surfaces in order to infer the free space extent outside their field of view.

2.2.3 Joint Object Detection and Layout Estimation

In contrast to the methods presented in the previous two subsections, those presented here aim to detect objects and estimate the scene layout in tandem. As we will see, performing both tasks simultaneously can in fact result in a simplified and more accurate inference procedure than performing each task independently. In addition, we will see how 3D models for objects and layout can be used to infer free space in occluded regions.

Many early approaches rely on first estimating the layout, and then using the layout as a context for inferring object locations (Hoiem et al., 2008; Hedau et al., 2010, 2012). Hoiem et al. estimate rough surface geometry in outdoor scenes as well as the camera viewpoint in order to constrain the locations in which pedestrians may reside (Hoiem et al., 2008). Hedau et al. use their method for estimating room layout (Hedau et al., 2009) in order to impose spatial constraints based on the layout in a probabilistic model for 3D object detection (Hedau et al., 2010). While their original approach is designed to only detect beds, in their follow up work (Hedau et al., 2012) they extend their cuboid object model to build a generic class of box-like objects. Their latter method not only provides multiple object detections, but it also explicitly models the mutual relationships between them, as well as their spatial relationship with the layout, in order to infer the presence occluded free space.

In more recent years, research has demonstrated that, instead of a sequential approach, performing both inference tasks jointly provides a greater degree of robustness (Gupta et al., 2010b; Schwing et al., 2013; Zhang et al., 2014; Izadinia et al., 2017; Tulsiani et al., 2018; Du et al., 2018). Gupta et al. enumerate a set of possible of layouts and a set of 3D object detections, train an SVM to learn a scoring function for a given configuration, then perform a beam search to find the optimal configuration (Gupta et al., 2010b). Schwing et al. propose a similar method, except they use a branch and bound approach to prune the combinatorial search space of possible configurations (Schwing et al., 2013). Instead of formulating the problem as a search problem, Zhang et al. propose training an SVM to predict the optimal

room configuration using ground truth labels (Zhang et al., 2014). However, more recently, deep neural networks have replaced SVMs in data driven approaches (Izadinia et al., 2017; Tulsiani et al., 2018; Du et al., 2018). Izadinia et al. use two networks: one for labeling pixels according to which box face they belong to, and the other for 2D bounding box detection (Izadinia et al., 2017). After searching for the appropriate 3D CAD model corresponding to the object observation from a library, the position of the room layout and objects are optimized. Tulsiani et al. take a similar approach, except instead of aligning object CAD models, they predict the full 3D shape and pose of each object in a voxel representation (Tulsiani et al., 2018). Du et al. build upon the work of Tulsiani et al. (Tulsiani et al., 2018). However, they use a more compact cuboid and plane model for objects and layout respectively, and introduce a module for inferring physically stable arrangements (Du et al., 2018).

Given the highly cluttered nature of indoor room scenes, it is useful to have additional geometric information such as depth in order to disentangle different elements, instead of relying on appearance information alone (Kim et al., 2012; Mattausch et al., 2014; Lin et al., 2013; Kim et al., 2013; Ren and Sudderth, 2016; Song et al., 2017; Zou et al., 2019). Although their method follows a more sequential approach as they segment the layout first, Kim et al. detect repeated object instances in a 3D point cloud, and replace them with a lower complexity deformable model (Kim et al., 2012). Mattausch et al. take a similar approach, but use planar features to describe objects and replace the repeated object instances with CAD models instead (Mattausch et al., 2014). Note that unlike (Kim et al., 2012), their method is considered a joint approach as layout components are considered to be just another class of object. Lin et al. continue to build upon the work of Schwing et al. by incorporating the additional depth information into a Conditional Random Field for modeling object cuboid positions, where the pairwise potentials capture the contextual relations between the scene and the objects (Lin et al., 2013). Kim et al. model scenes using a Voxel-CRF in order to refine raw depth values based on a class segmentation (Kim et al., 2013). Song et al. extend this approach by using a CNN in order to simultaneously label

voxels according to their semantic class as well as to predict the occupancy of unobserved voxels (Song et al., 2017). Ren and Sudderth propose two extensions of the traditional voxel representation for feature learning, including Clouds of Oriented Gradients (COG) for detecting objects and Manhattan voxels for predicting layout (Ren and Sudderth, 2016). In Zou et al., the authors propose a similar approach as (Izadinia et al., 2017), except they use a CNN for CAD model retrieval and they model layout using a more general Manhattan-aligned planar model (Zou et al., 2019). They are among the few which emphasize parsing both visible and occluded portions of the scene.

2.2.4 Summary

Despite the impressive results these systems have demonstrated in interpreting 3D scenes, they suffer several limitations which prevents them from being readily applied for online building information modeling.

First and foremost is that many of these systems are designed to only process information from a single limited field of view. In contrast, our system requires the ability to consider information from multiple frames arriving in a sequence in order to build a model of the entire environment. While systems such as (Flint et al., 2011) and (Saxena et al., 2009) would appear to offer multi-frame support, in that it is possible to include additional nodes and edges into their probabilistic graphical models, doing so makes performing inference considerably more expensive with each additional frame.

Second, is that in a robotics context, we require a to-scale metric map of the environment for navigation and planning. However, without a known reference height (Criminisi et al., 2000) or size priors for semantic classes (Chen et al., 2016), many of the RGB only approaches such as (Tekin et al., 2018; Hoiem et al., 2008; Hedau et al., 2010, 2012; Gupta et al., 2010b; Schwing et al., 2013; Zhang et al., 2014; Izadinia et al., 2017; Tulsiani et al., 2018; Du et al., 2018) can only provide reconstructions up to an unknown scale factor due to the fundamental depth ambiguity in projective geometry. On the other hand, the additional metric depth information provided by RGBD sensors can not only provide scale, but additional robustness

as well (Zhou and Tuzel, 2018; Wang et al., 2018; Taylor and Cowley, 2012; Guo and Hoiem, 2013; Kim et al., 2013; Lin et al., 2013; Ren and Sudderth, 2016; Song et al., 2017; Zou et al., 2019).

Among the methods which produce metric reconstructions, an outstanding issue of model compactness and generalizability still remains. While CAD models for objects are relatively compact and provide the ability to model shape (Mattausch et al., 2014; Zou et al., 2019), they limit the variety of objects which can be modeled, are often expensive to procure, and can be computational expensive to use as every detection requires a search over the entire library and an alignment procedure in order to recover the 3D model. Moreover, while these methods produce visually appealing results, the reconstructions provided may include more detail than what is necessary for a robot to navigate through the world. While using cuboids to model objects (Wang et al., 2018; Ren and Sudderth, 2016; Lin et al., 2013) provides greater generalizability with an even greater degree of compactness, using a single cuboid model for the layout (Ren and Sudderth, 2016) may often be too restrictive for modeling environments larger than a single room. Although the voxel representations of (Kim et al., 2013) and (Song et al., 2017) provide full generalizability and fit nicely into many robot perception pipelines, a major argument of this thesis is that even with semantic annotations, these complex models are not only memory intensive, but they fundamentally fail to deliver a true understanding of the environment. We assert that a compact yet suitably expressive models for objects (Kim et al., 2012) and layout (Zou et al., 2019), including those more akin to the ones we discuss in Section 2.1, can provide robots with the ability to generate more efficient global plans and the confidence to navigate at high speeds.

2.3 Simultaneous Localization and Mapping

Simultaneously estimating pose and integrating scene information across multiple viewpoints into a single map representation is arguably one of the oldest problems in computer vision (Marr and Poggio, 1979; Longuet-higgins, 1981; Harris and Pike, 1988) and in robotics (Smith and Cheeseman, 1986). Although these techniques have continued to evolve over the

years, the anatomy of almost all modern SLAM systems remains the same, consisting of a *front end* and *back end*. While the front end is responsible for abstracting sensor data into a model useful for estimation, the back end is responsible for performing inference on the abstracted data to recover the state of the world. In the following subsections, we discuss different types of feature abstractions, the effects on the back end optimization, and the level of understanding each provides.

2.3.1 Feature-Based SLAM

The front end of most classical SLAM systems detect and match a sparse set of point features across a sequence of images or point clouds, which are then fused together into a single model of the environment using a probabilistic back end such as an Extended Kalman Filter, Rao-Blackwellized particle filters, and maximum likelihood estimation (MLE); a complete overview of which can be found in the survey by Durrant-Whyte and Bailey (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006) and the book by Thrun et al. (Thrun et al., 2005). Although these methods are designed for range-based sensors, which provide 3D measurements implicitly, they have been adapted to use triangulated features from both stereo (Davison and Murray, 2002) and monocular cameras (Davison et al., 2007).

While computationally efficient, these probabilistic filtering methods rely heavily on a series of linearizations of nonlinear functions at every time step in order to estimate the current state, which makes them error prone and incapable of correcting past updates in the light of future observations. This has led to the modern *de-facto* SLAM formulation based on the seminal work of Lu and Milios (Lu and Milios, 1997), which employs maximum a posteriori (MAP) estimation over pose-graphs. Although the underlying optimization problem may seem daunting at first, modern solvers such as iSAM (Kaess et al., 2008) and g2o (Kummerle et al., 2011), have exploited the sparsity and topology of most pose-graphs in order to perform fast incremental inference. This optimization as a back end design has also been leveraged in visual SLAM (Nister et al., 2004), also known as incremental Structure-from-Motion (Snavely et al., 2006), where the Bundle Adjustment (BA) (Agarwal et al., 2010)

step, which is typically used for global refinement after incremental map updates, is run continuously in a separate mapping thread from the front end tracking process (Klein and Murray, 2007; Mur-Artal et al., 2015).

Landmark features, detected by the front end of the SLAM system, are ideally highly salient and easy to track over time. Given a range scan, features correspond to locations in the point cloud at which a particular geometric signature is detected. Whereas in an image, feature transforms, such as SIFT (Lowe, 2004), are used in order to extract keypoints, where each is provided its own unique feature descriptor, which encodes additional appearance information to be used for matching. However, the primary limitation of appearance-based feature descriptors such as these, is that they are dependent on the amount of available texture in the environment. In such settings, it is common to use higher-order geometric features including lines (Smith et al., 2006; Li et al., 2017; Pumarola et al., 2017) and planes (Trevor et al., 2012; Lee et al., 2012; Kaess, 2015; Taguchi et al., 2013; Ma et al., 2016; Hsiao et al., 2017). A unified model for handling lines in addition to points can be found in (Smith et al., 2006) and (Li et al., 2017). Pumarola et al. demonstrate a similar approach in (Pumarola et al., 2017) using lines and is built on ORB-SLAM (Mur-Artal et al., 2015). Leveraging the abundant amount of planar structures in indoor environments, in (Trevor et al., 2012; Lee et al., 2012; Kaess, 2015), the authors map only planar features. While a general planar model may be more challenging to optimize, each of the three proposes a different representation for a plane in order to make the optimization efficient. Taguchi et al. extend planar maps in order to demonstrate a method for registering sequential frames given any combination of three point and plane primitives (Taguchi et al., 2013). While their final map representation is not feature based, but instead a dense map of points, in both (Ma et al., 2016) and (Hsiao et al., 2017), the authors employ a global optimization over planar landmarks. They also use dense frame-to-frame alignment for initial motion estimation. Dense methods are further discussed in Section 2.3.4. Using lines and planes for mapping is also useful as they provide additional geometric constraints on motion. However, unlike our approach, and the other Structural SLAM methods, described in Section 2.3.3, which

also use planar features for mapping, none of the methods described above go on to leverage the structural regularity of the building in order to impose constraints between the features themselves. As we will see in the sequel, doing so can not only simplify the mapping problem itself, but in fact lead to more accurate reconstructions by compensating for the noise which leads to the ever present drift characteristic in general SLAM systems.

While point features, in general, may be easy to compute and match, they offer little understanding of the environment itself in terms of objects, layout, free space, etc. In contrast, in Section 2.3.3, we will see that imposing constraints between planar regions can lead to a highly informative map of structural components in the building. Similarly, instead of constructing a map of arbitrary points, Semantic SLAM, which is further discussed in Section 2.3.2, constructs a map of detected objects. Observe that not only do these representations provide a greater degree of understanding, but they are also inherently more compact, as there are typically far less objects and structural components in a building than the arbitrary number of keypoint features computed from an image.

2.3.2 Semantic SLAM

In contrast to Semantic Mapping techniques, such as (Nüchter and Hertzberg, 2008; Kundu et al., 2014; Sunderhauf et al., 2017; McCormac et al., 2017; Zhi et al., 2019), which merely provide a semantic labeling to the dense reconstructions output by Dense SLAM (discussed further in Section 2.3.4), Semantic SLAM detects specific object instances, whose point locations are then passed to the same probabilistic or pose-graph optimization back ends used in typical feature-based SLAM methods. Not only is this set of point features more sparse, which results in more efficient optimization, but it provides a more informative representation of the environment as opposed to points from conventional feature detectors.

Castle et al. propose the first Semantic SLAM system in (Castle et al., 2007), which detects planar objects and incorporates points along their boundary as features in an EKF back end. Recognizing that using planar objects alone is rather limited, Civera et al. follow a similar approach, but use the centroid of 3D objects, detecting using a database of object

models, as point features instead (Civera et al., 2011). In (Bao and Savarese, 2011), Bao and Savarese demonstrate an MLE formulation of the problem, where they perform joint optimization over camera parameters, 3D object bounding boxes, and object classes. More recent approaches using pose-graphs have also included richer shape representations such as meshes from a library of object models (Salas-Moreno et al., 2013), point clouds (Gálvez-López et al., 2016), voxels (Mccormac et al., 2018), and quadrics (Nicholson et al., 2019; Hosseinzadeh et al., 2019; Ok et al., 2019).

Data Association

At its core, the problem of data association involves identifying which measurements or observations taken at a particular timestep correspond to which landmarks currently in the map. Developing robust solutions to such a problem is of paramount importance as it is widely known that a single false data association can irrevocably mar reconstruction accuracy. As a result, it is a challenge that arises within the context of all types of SLAM systems, although it has garnered significant attention from the Semantic SLAM community in recent years as researchers seek ways to robustly incorporate discrete noisy measurements from object detectors. Therefore, we provide an overview of these methods here.

Traditionally, most solutions for the data association problem involve appearance-based matching and/or tracking of features across multiple frames. However, even when supplemented with methods for outlier removal such as RANSAC (Hartley and Zisserman, 2004) this process is naturally error prone due to noise, perceptually aliasing, variations in visual texture, and viewpoint dependencies. Note these are the same challenges also facing distance-based similarity heuristics used when working with 3D landmarks from LIDAR data. Meanwhile, methods for loop closure, a subset of data association, which often rely on a separate module for place recognition (Lowry et al., 2016) also fall prey to the same pitfalls.

As mentioned previously, when an incorrect data association occurs, the result can be catastrophic for reconstruction. This is primarily due to the fact that information flows in one

direction from the SLAM front end, which is responsible for managing data associations, to the SLAM back end, which is responsible for model optimization, making it impossible for either component to consider the impact when deciding which of the various data associations to enforce.

Bowman et al. address the issue by eliminating the hard decision making component and instead reasoning about the space of data associations probabilistically using Expectation Maximization (EM) to assign different weights to certain measurements (Bowman et al., 2017). Doherty et al. provide an alternative probabilistic formulation capable of entertaining multiple modes in (Doherty et al., 2019). Orthogonal to these methods are those that pass sole responsibility of data association entirely to the SLAM back-end (Sunderhauf and Protzel, 2012; Carlone et al., 2014b; Graham et al., 2015). Sunderhauf and Protzel introduce additional *switch variables* (i.e. indicator variables residing on the 0-1 interval) into the traditional non-linear least-squares optimization in order to determine which constraints to keep active and which to disable when minimizing the reconstruction objective (Sunderhauf and Protzel, 2012). Their method has also been referred to as line process (Choi et al., 2015; Lee et al., 2017b). Carlone et al. present a solution most closely related in spirit to our own which involves a linear program (LP)-relaxation of a sparse objective in order to determine the largest set of inlier measurements in (Carlone et al., 2014b). While their approach is fast and effective, it is unfortunately limited to 2D planar environments with linear measurement functions. However, Graham et al. present a similar approach that generalizes to the full 3D SLAM problem using residual gating instead of linear programming.

Another noteworthy method also involving the minimization of a sparse objective, though it is specifically used for uncovering loop closures, can be found in (Latif et al., 2017). Their approach involves searching through a set of basis corresponding to each location in order to uncover that which has the greatest degree of correlation with a particular observation.

2.3.3 Structural SLAM

In an effort to preserve generalizability across different environments, traditional feature-based SLAM systems often refrain from making any assumptions about the environment in which the system will operate in. However, the large degree of structural regularity characteristic of indoor environments can often be exploited in order to reduce the pervasive drift, which is ever present in even state of the art visual SLAM systems. To this end, albeit at the cost of universality, Structural SLAM systems introduce additional geometric constraints between higher-order geometric features such as lines and planes, based on certain assumptions about the world. However, doing so also leads to a meaningful representation of the environment that is still compact, as certain primitives begin to take on additional semantic meaning, *e.g.* orthogonal planes become walls, floors, and ceilings.

Lines have long been an element of interest in computer vision, which made them one of the first types of primitives to be explored for use in visual structural SLAM (Nguyen et al., 2006; Zhou et al., 2015; Ikehata et al., 2016; Li et al., 2019; Camposeco and Pollefeys, 2015). In (Nguyen et al., 2006), Nguyen et al. compute a histogram over line segments, and select only those belonging to one of three detected orthogonal directions as landmark features. Zhou et al. use orthogonal lines for mapping as well, except they first detect the three orthogonal vanishing points corresponding to the dominant axes of the building, and use only those lines aligned to these directions (Zhou et al., 2015). In addition to orthogonality, Ikehata et al. also include adjacency and length constraints between different pairs of line segments (Ikehata et al., 2016). It is worth noting that they assume that their camera orientation is known a priori, which makes their optimization linear over a set of translations. However, while we estimate rotation from the visible structure, they simply use the estimate of rotation provided by the IMU. Li et al. relax the Manhattan World (Coughlan and Yuille, 1999) constraint between lines to an Atlanta World (Schindler and Dellaert, 2004) constraint (Li et al., 2019). Camposeco and Pollefeys propose an alternative use for lines than for tracking as features in (Camposeco and Pollefeys, 2015). Instead,

they use lines in order to estimate vanishing points, which they then track within their filtering-based back end. They note that tracking these stable indicators of rotation greatly reduced the amount of drift their system incurs.

While these methods use arbitrary lines for tracking, several methods inspired by the advances in single view scene understanding (reviewed in Section 2.2) track lines corresponding the piece-wise linear floor-wall and ceiling-wall boundaries (Flint et al., 2010; Tsai et al., 2011; Yang et al., 2016b). Flint et al. extend the work of Lee et al., and provide the means for including information from multiple sequential views for their hypothesis testing framework (Flint et al., 2010). Tsai et al. build upon the work of Delage et al. by tracking the piece-wise linear floor-wall boundary, but in a Bayesian filtering framework (Tsai et al., 2011). Yang et al. also utilize the model proposed by Delage et al. in their Pop-up SLAM (Yang et al., 2016b) system, which uses the boundary lines in order to construct a planar pop-up model of the environment. They use a CNN in order to detect boundary components (Yang et al., 2016a) and bootstrap a direct visual SLAM system (Engel et al., 2014) to provide initial estimates of the camera trajectory.

Although detecting boundary segments can implicitly lead to a planar model, several systems detect and track planes explicitly. However, unlike the general planar models of (Trevor et al., 2012; Lee et al., 2012; Kaess, 2015; Ma et al., 2016; Hsiao et al., 2017), the following methods incorporate a Manhattan World constraint between planes in order to reduce camera motion drift and obtain greater semantic meaning from landmarks (Nguyen et al., 2007; Hsiao et al., 2018; Hosseinzadeh et al., 2018; Yang and Scherer, 2019). In (Nguyen et al., 2007), the authors provide a reformulation of their Orthogonal SLAM system (Nguyen et al., 2006) to use orthogonal planes instead of lines. Meanwhile in (Hsiao et al., 2018), Hsiao et al. integrate the parallelity and orthogonality constraints into their previous dense planar SLAM method (Hsiao et al., 2017). Notice that by introducing the orthogonality constraint between planes, they begin to take on additional semantic meaning as floors, walls, and ceilings. Hosseinzadeh et al. and Yang and Scherer expand the

semantic richness of these constrained planar layout models by providing frameworks for mapping ellipsoid (Hosseinzadeh et al., 2018) and cuboid (Yang and Scherer, 2019) object models alongside orientation-constrained planes.

Instead of using only one or the other, it is also possible to use both lines and planes in a constrained optimization in order to take advantage of the benefits offered by tracking both (Amayo et al., 2016; Lu and Song, 2015; de la Puente and Rodriguez-Losada, 2014). Amayo et al. present a unified framework for mapping with points, lines, and planes with architectural constraints simultaneously (Amayo et al., 2016). A similar method can be found in (Lu and Song, 2015). Meanwhile, de la Puente and Rodriguez-Losada not only include additional geometric primitives such as circles, but they also introduce an EM optimization which automatically detects which structural constraints to enforce (de la Puente and Rodriguez-Losada, 2014). In this respect, their method is closely related to our approach. In their follow up work (de la Puente and Rodriguez-Losada, 2015), they extend their hierarchical representation of structure in order to include rectangles, which bears further similarity to our work. However, our approach uses rectangular primitives as an explicit approximation of the free space itself, whereas they use rectangles as just another higher-order geometric landmark.

Another line of research among this body of literature which is most relevant to our work presented here, includes those methods which leverage the observability of structural orientations within buildings in order to decouple rotation and translation estimation in SLAM delivering fast and accurate results (Carlone et al., 2014a; Agarwal et al., 2017, 2019; Li et al., 2018; Kim et al., 2018b,a). In fact, if robot orientations are known the SLAM problem becomes a linear-least squares problem (Carlone et al., 2015). Carlone et al. demonstrate a closed-form solution to the SLAM problem in the planar case called the Linear Approximation for Pose Graph Optimization (LAGO) by first estimating robot orientations in order to subsequently estimate robot positions (Carlone et al., 2014a). The authors show that in such a setting both orientation estimation and position estimation can be formulated as

quadratic optimization problems giving rise to the closed-form solution of each. Although the LAGO formulation has been shown to provide very fast, accurate, and robust initial estimates to the SLAM problem, its primary limitations are that it does not include feature-based measurements and it does not readily extend to 3D. In (Agarwal et al., 2017), Agarwal et al. achieve an even greater degree of accuracy and reliability, however at the cost of computational speed as their method RFM-SLAM relies on iterative non-linear on-manifold optimization in order to estimate orientations. In their subsequent work (Agarwal et al., 2019), the authors include absolute orientation sensing (in addition to the standard relative orientation sensing) to solve the linearized least-squares problem. In (Li et al., 2018), Li et al. bootstrap a general visual SLAM framework to provide an initial trajectory and landmark positions, which they then refine using their two part optimization strategy. They first align all camera rotations until the vanishing points in each image (which correspond to the Manhattan frame) are consistent with the location at which they are detected in the first frame. Afterwards, frame-to-frame translations are re-expressed in each of the corrected frames and optimized with respect to the motion of points and orthogonal lines in the image. The Visual Odometry system proposed by Kim et al. follows a similar approach, first de-rotating camera orientations with respect to the detected Manhattan frame in each image then re-expressing frame-to-frame translations before optimization (Kim et al., 2018b). In their follow up work (Kim et al., 2018a), they extend their VO system (Kim et al., 2018b) to a SLAM formulation, in which planar features are modeled as landmarks in order to further constrain the camera motion and consequently, reduce drift in translation. Although their approach to this end is nearly identical to our own, we integrate the 1-D planar range measurements to infer state using a pose graph optimization instead of a filtering framework. This formulation allows us to entertain additional constraints between the planar landmarks retroactively through our sparse optimization procedure. Observe that all of the methods just described require some method for per-frame rotation estimation. A survey of such methods can be found in (Carlone et al., 2015).

2.3.4 Dense SLAM

With the introduction of low-cost dense laser scanners such as the Microsoft Kinect at the start of the decade, many sought to leverage these technologies in order to construct rich visually appealing dense reconstructions of indoor environments mostly for AR and VR applications. Unlike feature-based SLAM, Dense SLAM aims to utilize the entirety of the data available for estimation as opposed to sparse subset of features, which typically yields more accurate estimation.

Despite the renewed interest in the problem at the time, the problem of estimating the relative transformation between two point sets can be traced back to the seminal work of Besl and McKay who introduce Iterative Closest Point (ICP) (Besl and McKay, 1992); a method which is considered standard practice today. However, ICP is designed for registering only a single pair of point sets. Given a sequence of scans, simply integrating the relative frame-to-frame transformations would result in a significant amount of accumulated drift in the final reconstruction. As a result, ICP is typically used as a tool in both depth fusion and large-scale registration.

The task of depth fusion involves taking a sequence of depth scans and integrating the measurements into a single 3D model of the scanned space. Newcombe et al. demonstrate the first modern real-time method for such a problem in their KinectFusion (Newcombe et al., 2011) pipeline. Despite its impact, KinectFusion (Newcombe et al., 2011) is only limited to mapping small desktop scenes due to its intensive memory requirements. In order to address this limitation, Nießner et al. propose a voxel hashing technique in order to optimize speed by only considering visible voxels (Nießner et al., 2013). Keller et al. propose an alternative to using voxel grids, and instead perform fusion using points alone (Keller et al., 2013). The methods described above achieve a high degree of accuracy using ICP (Besl and McKay, 1992) by aligning new scans to the scene model constructed up to that point. While this frame-to-model approach avoids the need for a later global correction, it can only produce accurate reconstructions of relatively small scenes. This is primarily

due to the fact that it is impossible to reintegrate previous scans once they have been incorporated into the model, which also leads to the accumulation of drift over time.

Unlike fusion, the task of pose estimation, also known as registration in this context, is as concerned with the quality of the reconstructed trajectory as it is with the final map. Henry et al. propose a coarse-to-fine scheme, which uses a sparse set of points to compute an initial frame-to-frame alignment using ICP, followed by a secondary dense ICP for refinement (Henry et al., 2012). They then use Sparse Bundle Adjustment (SBA) for a global correction. Many subsequent methods are modeled on their overall approach. In (Kerl et al., 2013), the authors combine appearance and geometric information for frame-to-frame registration for increased accuracy, followed by pose graph optimization. Endres et al. use image features for coarse alignment, then the point cloud for fine scale alignment and transform verification, followed by a pose graph optimization for global correction as well (Endres et al., 2014). While Whelan et al. continue this trend and introduce further refinements to the technique for mapping larger scenes in (Whelan et al., 2015a), in their ElasticFusion (Whelan et al., 2015b) paper, they propose an alternative approach to using pose graph optimization for global refinement. Instead, they employ a series of local BAs in order to maintain accuracy over larger trajectories. However, most techniques continue to use pose-graphs due to its greater degree of robustness. In their hybrid fusion-registration framework, Dai et al. use a chunked version of the voxel representation employed by KinectFusion (Newcombe et al., 2011) which they refine with pose graph optimization in their BundleFusion (Dai et al., 2017) pipeline. This not only allows BundleFusion to map large scenes, but to do so accurately through its ability to re-integrate scans in the final model.

Although many of these methods are able to produce rich visually appealing models of indoor environments, they ignore the pervasive structural elements in the scene which can be used in order to increase accuracy and refine the final map (Salas-Moreno et al., 2013; Zhang and Singh, 2014; Nardi et al., 2019; Zhang et al., 2015; Dzitsiuk et al., 2017; Straub et al., 2017; Lee et al., 2017b; Lin et al., 2018; Halber and Funkhouser, 2017). Salas-Moreno

et al. detect a set of planar landmarks, which are the only portions of the dense map that are refined in subsequent frames (Salas-Moreno et al., 2014). Similarly, Zhang and Singh use only point sets corresponding to lines or planes for frame-to-frame tracking in LOAM (Zhang and Singh, 2014). In (Nardi et al., 2019), the authors provide a unified representation of points, lines, and planes, to be used as constraints in ICP. Zhang et al. extend the work of (Nießner et al., 2013) and (Newcombe et al., 2011) in order to include object and layout labels for voxels (Zhang et al., 2015). These labels are then used to refine planar surfaces and object boundaries, as well as to fill in holes in the final reconstruction. Dzitsiuk et al. follows a similar approach, except their planes are represented implicitly within their volumetric representation instead of maintaining the set of planes separately (Dzitsiuk et al., 2017). While these systems all use planar models to refine their dense estimates, like other feature-based SLAM methods which simply use unconstrained planes for mapping (Trevor et al., 2012; Lee et al., 2012; Kaess, 2015; Ma et al., 2016; Hsiao et al., 2017), they do not enforce constraints between the planes themselves.

On the other hand, several recent dense methods have produced highly accurate reconstructions across large-scale indoor environments by including such constraints (Straub et al., 2017; Lee et al., 2017b; Lin et al., 2018; Halber and Funkhouser, 2017); effectively dense analogs of the Structural SLAM systems described in Section 2.3.3. Straub et al. propose a novel method for inferring the directional Stata Center Wold (Straub et al., 2015b) segmentation of a semi-dense surfel representation of the environment, while simultaneously performing real-time camera motion estimation (Straub et al., 2017). The extension we propose for our structural SLAM method in Chapter 3 utilizes the same SCW prior assumption. In (Lee et al., 2017b), Lee et al. propose a system that uses planes corresponding to walls, floors, and ceilings, in order to refine the fused depth maps provided by KinectFusion (Newcombe et al., 2011). Their approach assumes that walls are orthogonal to parallel floors and ceilings, but not necessarily to one another. Whereas their refinement procedure occurs after registration, Lin et al. use local floor plan priors to guide the registration of individual scan, followed by a complexity-reducing (in terms of the number of corners

formed by planar segments) global optimization for refinement (Lin et al., 2018). Halber and Funkhouser employ a fine-to-coarse registration scheme, where they detect edges and planes with orthogonality constraints in local regions, which are then propagated to larger windows, merged, and used to refine global alignments (Halber and Funkhouser, 2017).

In contrast to Semantic and Structural SLAM, while Dense SLAM methods are able to produce accurate and highly detailed reconstructions of building interiors, their underlying map representations, which typically include point clouds or voxels, are incredibly complex (requiring many parameters to encode) and yet provide little to no higher-level understanding of the environment itself. Moreover, the added complexity required also leads to a significant memory and computational overhead.

2.3.5 Summary

General feature-based SLAM methods are fast and mature technologies (Mur-Artal et al., 2015). However, their representations of space, which often consists of simple unconstrained primitives including points (Mur-Artal et al., 2015), lines (Smith et al., 2006; Li et al., 2017; Pumarola et al., 2017), and planes (Trevor et al., 2012; Lee et al., 2012; Kaess, 2015; Taguchi et al., 2013; Ma et al., 2016; Hsiao et al., 2017), provide little understanding of the environment itself. While Structural SLAM methods, such as (Nguyen et al., 2006; Zhou et al., 2015; Ikehata et al., 2016; Li et al., 2019; Camposeco and Pollefeys, 2015; Flint et al., 2010; Tsai et al., 2011; Yang et al., 2016b; Nguyen et al., 2007; Hsiao et al., 2018; Hosseinzadeh et al., 2018; Yang and Scherer, 2019; Amayo et al., 2016; Lu and Song, 2015; de la Puente and Rodriguez-Losada, 2014), use the same types of primitives, the mutual constraints imposed on them by introducing prior assumptions about the environment results in the primitives taking on semantic meaning. For instance, orthogonal planes can be interpreted as layout components, *e.g.* walls, floors, and ceilings. Semantic SLAM methods, including (Castle et al., 2007; Civera et al., 2011; Bao and Savarese, 2011; Salas-Moreno et al., 2013; Mccormac et al., 2018; Nicholson et al., 2019; Bowman et al., 2017), provide a similar semantic interpretation of the environment by directly providing a map of objects.

While these methods make great strides towards a SLAM system capable of providing lifelong understanding (Cadena et al., 2016), we are yet to see these features used to support some later task, such as inference and modeling. Though Gaussian Processes Occupancy Maps (O'Callaghan and Ramos, 2012), Hilbert Maps (Ramos and Ott, 2016; Guizilini et al., 2019), and more recently, Deep Generative Networks (Katyal et al., 2019) have been shown to be effective at providing robust probabilistic predictions as to the state of occupancy of unobserved cells by explicitly modeling the statistical dependencies between neighboring voxels, their underlying representation cannot be interpreted in terms of higher-level scene elements. This is due to the inherent complexity of dense representations including point clouds (Henry et al., 2012; Kerl et al., 2013; Keller et al., 2013; Salas-Moreno et al., 2013; Zhang and Singh, 2014; Lin et al., 2018; Halber and Funkhouser, 2017), OctoMaps (Endres et al., 2014), surfels (Whelan et al., 2015b; Straub et al., 2017), and voxels (Newcombe et al., 2011; Nießner et al., 2013; Whelan et al., 2015a; Zhang et al., 2015; Dzitsiuk et al., 2017; Dai et al., 2017; Lee et al., 2017b). Although Semantic and Structural SLAM do provide compact and interpretable maps, they do not contain the same level of building details as is seen in Building Information Modeling, which is discussed in Section 2.1. In this dissertation, we propose a method for bridging this gap.

Chapter 3

Layout Modeling with Infinite Planes

This chapter describes a novel approach to SLAM that leverages the observability of a building's dominant structural orientations in order to reformulate the reconstruction task as a model selection problem. While there exist several SLAM approaches that try to incorporate the rectilinear structure of indoor man-made environments within their models by tracking more semantically meaningful features such as lines and planes, to our knowledge, we are the first to frame the mapping aspect of the problem entirely as one of model selection. Our ideal model is one that would resemble an architect's floor plan which outlines the location of all large static *layout structures*, namely walls, floors, and ceilings. Such a generative model, even partially complete, could not only enable a robot to track its position and orientation within the environment with much greater precision, but could also serve as a strong prior for inferring structure beyond the current field of view. We demonstrate on real data that our novel convex formulation based on the principle of Occam's razor identifies the simplest model of the environment that is most consistent with the measurement constraints. By directly incorporating the search over possible data associations into the back end reconstruction objective, our system produces an optimized trajectory and a

compact map representation simultaneously.

3.1 Manhattan Case

The work addressing the simplified Manhattan case presented in this section can also be found in the corresponding publication (Shariati et al., 2019).

To ground our subsequent discussion we begin with a brief description of the input data and the hardware systems used to acquire it in our subsequent experiments. Figure 2 shows an example of one of our sensor rigs. Despite slight differences among sensor configurations, every rig features a stereo pair of cameras, hardware synchronized to an inertial measurement unit (IMU), as well as a depth sensor that captures low-resolution depth images up to a range of about 6 meters. The data from the stereo cameras and IMU are used to drive a stereo based MSCKF Visual-Inertial Odometry (VIO) system (Sun et al., 2018). Further details surrounding each sensor are provided in Section 3.4.

The end result is that our sensor suites provides the analysis algorithm with a set of depth maps along with initial estimates for the relative motion of the sensor rig over time and an estimate of the gravity vector in each frame.

We note that similar datasets could be acquired using other means. One could acquire depth maps using a LIDAR sensor like the Velodyne puck or from a passive stereo system. Similarly pose information could be derived from monocular Visual-Inertial Odometry or from wheel encoders on a moving platform. The proposed analysis would still be applicable in all of these cases.

3.1.1 Manhattan Structure Detection

The first stage in the analysis involves processing each frame in the depth map separately to extract salient axis-aligned planar fragments of layout structure, called *layout segments*. The first step in this process involves projecting the depth points into the plane defined by the measured gravity vector and then rotating the resulting 2D point set in one degree increments to find a yaw orientation that minimizes the entropy of the resulting point

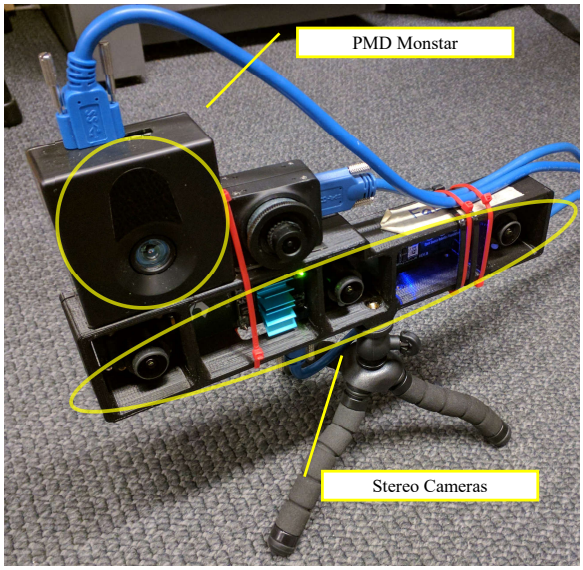


Figure 2: Sensor rig used to acquire data. Annotated in yellow are the PMD Monstar depth sensor and a custom stereo pair. The camera to the right of the Monstar and the center stereo camera are not used.

distribution. This algorithm is described in a number of previous works including (Bazin et al., 2013) and (Taylor and Cowley, 2012), where it is referred to as an entropy compass. Upon completion, this procedure recovers the orientation of the frame with respect to the prevailing Manhattan structure. We note that one could also use other means for Manhattan frame estimation such as (Straub et al., 2015a) or (Joo et al., 2019).

Once this has been done, the system labels each pixel in the depth map according to the axis alignment of the surrounding $k \times k$ patch, and then groups them together using a connected components procedure as shown in Figure 3. Finally, using an inverse perspective projection, each cluster of pixels is projected into the aligned sensor frame as a point cloud where we fit an orientation-constrained planar model using RANSAC. Planar segments with insufficient extent are discarded to favor the detection of dominant structures.

In addition to the entropy compass procedure which is applied to each frame individually, the system has an estimate for the relative orientation between each frame derived from the visual-inertial odometry system. These two sources of information are fused to provide a

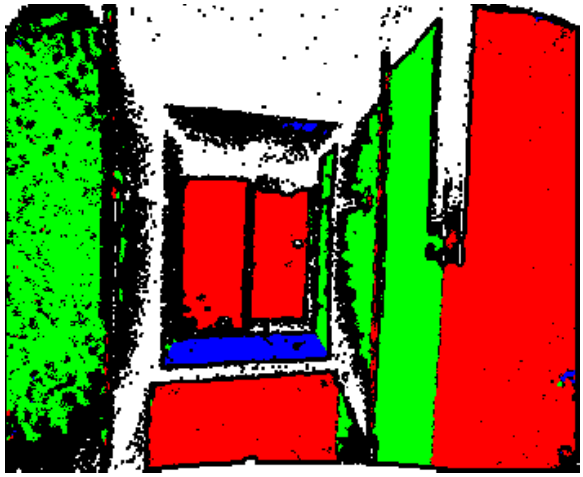


Figure 3: Depth map broken into salient surfaces. Red, green, and blue pixels represent x, y, and z-axis alignment. A pixel p is assigned to the major axis which maximizes the number of pixels in its $k \times k$ neighborhood that would reside on the plane centered at p with the given major axis orientation. If no axis can be assigned with sufficient confidence or no depth information is recorded at p, it is colored black and white respectively.

final estimate of the orientation of each frame in the sequence. The relative yaw estimates from the VIO system are used to constrain the range of angles considered in the entropy compass phase and to provide orientation estimates during periods where no axis-aligned surfaces are visible.

The end result of the procedure is described in Figure 4, which shows a top down view of a set of camera frames. Each frame is associated with two coordinate frames of reference, one which indicates the actual orientation of the sensor head and the other which indicates an axis-aligned frame derived from the entropy analysis. Each layout measurement, denoting an estimate for the minimum distance between the sensor and the corresponding layout segment observed at that frame, is depicted by a red or green dotted line.

This system of measurements can be abstracted into a pose-graph (Lu and Milios, 1997) shown in Figure 5. Here the circular nodes on top correspond to axis-aligned frame positions while the triangular and rectangular nodes on the bottom correspond to layout segments. The links between frames correspond to the estimates for inter-frame motion while the links between the frames and the layout segments correspond to the distance measurements

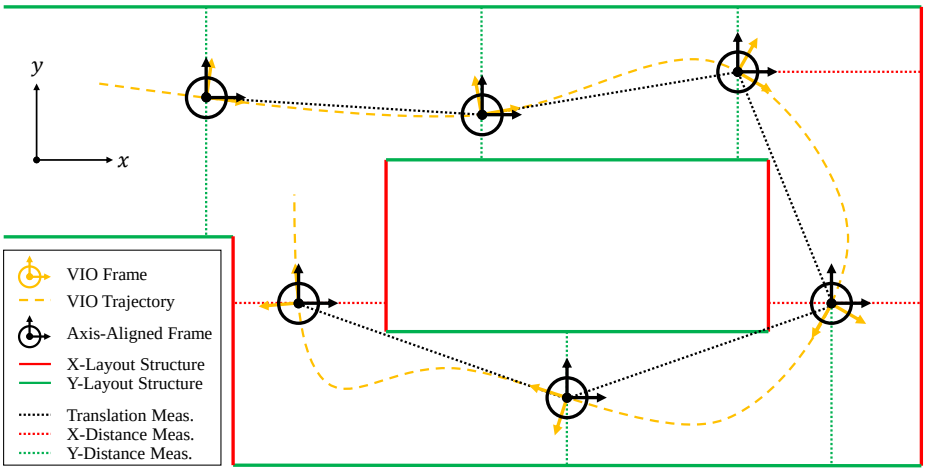


Figure 4: A 2-dimensional geometric representation of our model which illustrates a sensor moving through a Manhattan environment making periodic range measurements to various layout structures. Solid lines correspond to layout structures, while dotted lines correspond to measurements. Each distance measurement to a particular layout structure corresponds to the distance computed to the visible layout segment within the depth map captured at that frame.

described in Figure 4. Expressing each estimate of inter-frame translation, provided by the VIO subsystem, with respect to the previous estimate of the Manhattan frame, provided by the orientation estimation procedure, yields an axis-aligned and – in principle – drift-reduced trajectory.

3.1.2 Model Optimization

In the sequel we will use the following notation to describe the elements of the model shown in Figure 5. Let $\mathbf{p}_i \in \mathbb{R}^3$ denote the position of frame i in the axis-aligned trajectory while $R_i \in \mathbf{SO}(3)$ denotes the orientation of the axis-aligned frame with respect to the corresponding sensor frame. Each of the layout segments that we observe will ultimately be associated with a structural supporting *layout plane*, which is modeled as an axis-aligned surface with infinite extent. Each such layout plane will be modeled with a single parameter. More specifically we will let m_j^x denote the x coordinate of a layout plane with index j that is perpendicular to the x-axis of the model, similarly m_k^y denotes the y coordinate of a y aligned layout plane with index k and m_l^z denotes the z coordinate of a z aligned layout plane with index l.

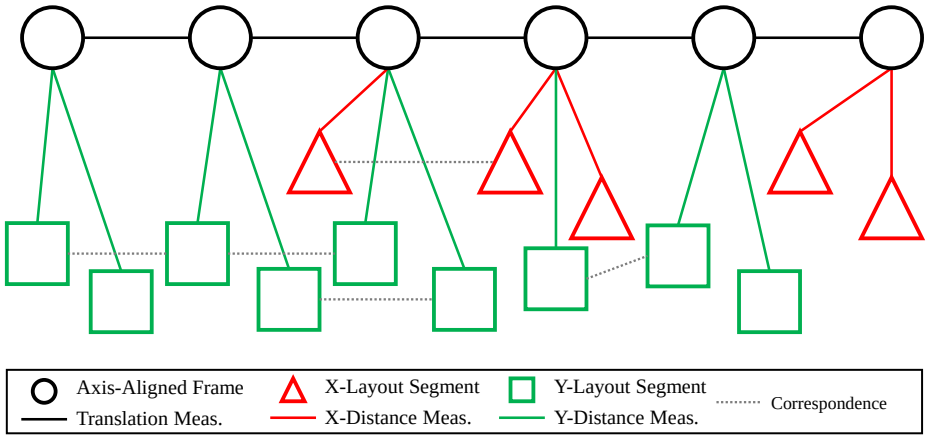


Figure 5: A functional representation of our model, in two dimensions, as a factor-graph. Circles correspond to robot locations, while triangles and squares correspond to x and y aligned layout segments respectively. Solid lines correspond to measurement factors derived from the VIO, entropy analysis, and depth map processing. We extend the traditional factor-graph formulation by including binary correspondence edges, represented by dotted lines. Initially generated by a temporal analysis, the set of hypothetical correspondence edges is also augmented by a user defined heuristic. Our sparse optimization procedure ultimately determines which of these constraints to enforce and discard.

Correspondences between layout segments are denoted by the dotted lines in Figure 5. These correspondences amount to asserting that two extracted segments lie on the same axis-aligned layout plane. Note that these correspondences would typically link layout segments extracted in different frames but could also link two segments extracted in the same frame. At this stage of the analysis procedure a simple temporal analysis procedure is used to establish correspondences between segments seen in one frame and segments seen in the subsequent frame that have sufficient overlap. This initial set of correspondences will be augmented with longer range correspondences that are automatically discovered in a subsequent step of the process.

We will let the vector $\mathbf{t}_i \in \mathbb{R}^3$ denote the estimate for the translation between subsequent axis-aligned frames in the sequence that is derived from the visual odometry system and corrected by the orientation estimation procedure. That is $\mathbf{t}_i$ denotes an estimate for the quantity $\mathbf{p}_{i+1} - \mathbf{p}_i$.

We will let ξ denote a vector formed by stacking the free parameters of our model, that is $\mathbf{p}_i$, m_j^x, m_k^y, and m_l^z, for all i, j, k, and l. Note that we assume that the camera orientations that align the frames with the Manhattan model, R_i, have been estimated using the entropy compass procedure described previously.

In this case the measurement system takes on a particularly simple linear form. Namely for each measurement from a frame to an x-aligned layout segment we have an equation of the form

$$m_j^x - p_i^x = d_{ij} \tag{3.1}$$

where m_j^x denotes the x coordinate associated with the x-aligned layout plane associated with the layout segment, p_i^x denotes the x coordinate of the position of frame i, and d_{ij} denotes the measured offset between the layout segment and the camera as depicted in Figure 4. Note d_{ij} can be signed depending upon where the frame is relative to the layout segment.

For layout segments aligned with the y axis and z axis we would have exactly analogous equations

$$m_k^y - p_i^y = d_{ik} \tag{3.2}$$

$$m_l^z - p_i^z = d_{il} \tag{3.3}$$

As previously discussed, the measurements of interframe motion derived from the VIO system and entropy analysis can be modeled as follows

$$\mathbf{p}_{i+1} - \mathbf{p}_i = \mathbf{t}_i \tag{3.4}$$

Given this system of measurements the task of finding the optimal estimate for the structure of the scene and the trajectory of the sensor based on the factor-graph simply amounts to

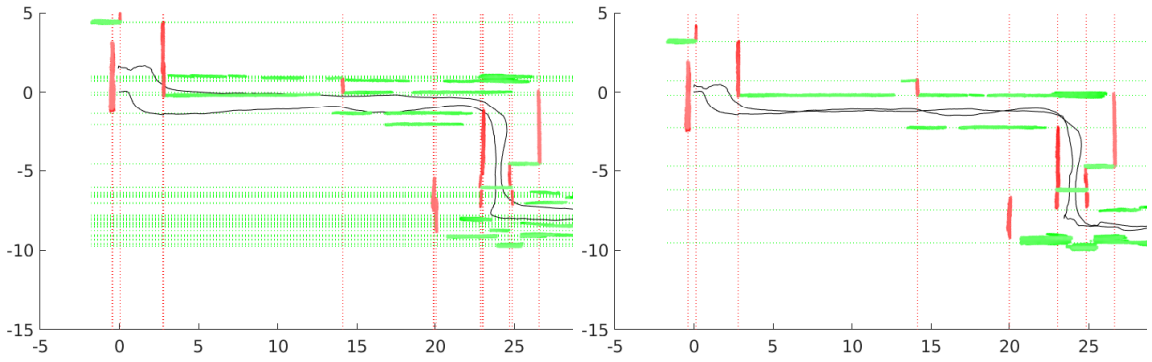


Figure 6: A birds-eye illustration of how our convex solution (below) can improve reconstruction by eliminating the drift still present in the least-squares solution (above). Notice the reduction in the total number of layout planes, which are denoted with red and green denoted lines corresponding to each axis-alignment.

solving a sparse linear system $A\xi = b$ in a least squares sense.

$$\underset{\xi}{\text{minimize}} \quad \|A\xi - \mathbf{b}\|_2 \tag{3.5}$$

This is simply the system formed by stacking the measurement equations, namely Equations 3.1, 3.2, 3.3, and 3.4, into a single sparse system. The vector $\mathbf{b}$ aggregates the right hand sides of the equations including the distance measurements, d_{ij}, d_{ik}, d_{il}, and the translation estimates $\mathbf{t}_i$. This sparse system can be solved extremely efficiently even for relatively large systems of measurements.

3.1.3 Automatic Model Selection

Running the optimization in Equation 3.5 yields the result shown in the second column of Figure 9. Each entry shows a result that captures the overall structure of the hallway but also exhibits the kind of drift typically associated with SLAM solutions; an artifact further highlighted in Figure 6. These reconstruction errors stem from the fact that the initial set of correspondences derived from the stream of depth frames is necessarily incomplete. While correspondences derived from frame to frame analysis are typically correct they fail to capture salient long term matches. For example, when one enters then exits a room it is important to encode the fact that walls in the hallway were in fact previously

seen and are not new features in the map. Similarly it is entirely possible to encounter a structural wall, then an opening, and then an entirely new section of the same wall. This problem of establishing long range correspondences is exacerbated by the fact that layout structures, unlike visual point feature landmarks, are extended structures and are rarely visually distinctive. Different sections of the same structure can have different appearances in different locations which can frustrate simple techniques that attempt to establish these long range correspondences.

We note that this problem of establishing long range correspondences subsumes the problem of loop closure which also revolves around the issue of deciding that one structure, a wall in this case, corresponds to another observed previously.

We propose a novel method that allows us to solve this problem by re-imagining this problem as one of model selection where our goal is to derive the simplest model that is consistent with our observations.

We begin by noting that solutions to Equation 3.5 suffer from having too many wall surfaces. This is because when a layout structure is encountered again after an intervening break it will be entered again in the map as a new structural layout plane. Our goal then is to discover which of the segments in our overly large model could actually be coincident. Identifying two or more layout structures with each other effectively reduces the number of parameters associated with the model since all of the displacement parameters associated with that set are collapsed to a single value. In this way we effectively compress the model leading to a simpler solution.

We begin by encoding all of the possible or suspected equivalences between layout segments in a set of equations of the following form

$$m_a^x - m_b^x = 0 \tag{3.6}$$

As you would expect Equation 3.6 encodes the idea that the x aligned layout plane with index a and the one with index b are in fact the same. Analogous equations are defined for y and z aligned layout planes.

These possible equivalences can be readily accumulated into a single sparse linear system $E\xi = 0$ where E is a sparse matrix encoding the relationship and ξ is the vector of model parameters used in Equation 3.5.

One possible approach to generating equivalence hypotheses, is to simply enumerate all possible equivalences between segments which face the same direction (north, east, south, west). However, this approach leads to an unnecessarily large E matrix that contains numerous spurious hypotheses; the effects of which we discuss more thoroughly in Section 3.4. For now, we adopt the heuristic of enumerating all possible equivalences between segments facing in the same direction that are within some distance of each other. Depending on the length of the path, the expected amount of drift, and the initial number of planes detected, this value can vary between 0.5-3 meters.

At this point we are not sure which of the equivalences are correct and which are false. This leads to a model selection problem. If there are k possible equivalence relations then there are in principle 2^k possible models depending on which of the equivalence relations are enforced, modulo independence issues related to transitive closures among the equivalence relations.

How then can we go about selecting which relations are correct from this exponentially large set of possibilities?

We begin by using the original reconstruction problem as a system that defines a set of possible solutions. We do this by considering the set of ξ values that satisfy:

$$\|A\xi - \mathbf{b}\|_2 \leq \delta \tag{3.7}$$

where δ encodes the discrepancy between a proposed solution and the available measurements.

One way to choose delta is simply by setting it to

$$(1+\epsilon)\|A\xi^*_{\text{lin}} - \mathbf{b}\|_2,$$

where ξ^*_{lin} is the optimal value of ξ after solving Equation 3.5. Alternatively, one can relate δ to the error that one expects in the measurements based on the sensor model. We could also imagine replacing the ℓ_2 norm with the ℓ_1 or ℓ_∞ norms. In each case this inequality defines a convex set in parameter space corresponding to solutions that are sufficiently consistent with the original set of measurements.

We then view our problem as finding the point in the set that maximizes the number of equivalence relations we can satisfy. Note that maximizing the number of equivalences is equivalent to minimizing the number of parameters in the final model, so our goal is to effectively apply the principle of Occam's razor to find the simplest model that explains our data.

Formally we can state our goal as follows

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & \|E\xi\|_0 \\ \text{subject to} \quad & \|A\xi - \mathbf{b}\|_2 \leq \delta \end{aligned} \tag{3.8}$$

In this expression, the ℓ_0 norm of a vector simply counts the number of non-zero entries in its input. This problem formulation is reminiscent of the kinds of problems one encounters in compressed sensing.

While this formulation is what we would ideally like to tackle, the discontinuous nature of the ℓ_0 norm makes it intractable so we resort instead to the ℓ_1 norm which we can view as a convex relaxation of our original problem.

Our new goal then can be stated as follows

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & \|E\xi\|_1 \\ \text{subject to} \quad & \|A\xi - \mathbf{b}\|_2 \leq \delta \end{aligned} \tag{3.9}$$

Many may notice the similarity between our formulation and the LASSO procedure (Tibshirani, 2011). LASSO performs subset selection over model coefficients by forcing as many of them to zero by bounding the sum of the absolute values of regression coefficients. Our approach to model simplification is different as our procedure reduces model complexity by enforcing equivalence relations encoded in the E matrix.

At this point we note that the optimization problem stated in Equation 3.9 involves minimizing a convex function subject to a convex constraint which places us squarely in the domain of convex optimization. The resulting problem can be reformulated as solving for the optimal value of a linear objective function subject to a set of linear and convex quadratic constraints. We note that we can solve problems involving hundreds of variables in a matter of seconds due to the sparseness of the underlying systems.

Once the problem has been solved we examine the resulting vector $E\xi$ and apply a threshold μ to decide which of the equivalences should be enforced. We then re-solve the optimization problem enforcing these equivalences

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & \|A\xi - \mathbf{b}\|_2 \\ \text{subject to} \quad & E'\xi = 0 \end{aligned} \tag{3.10}$$

where E' denotes the reduced set of enforced equivalences. The extent of the new layout structures are determined by computing the boundary around the individual corresponding layout segments residing on the same plane.

We can also introduce an additional hard constraint on the trajectory of the sensor relative to

the layout segments. For instance, a measurement d_{ij} to layout segment m_j^x also introduces the following linear inequality constraint

$$-\,\mathbf{sign}(d_{ij})(m_j^x - p_i^x) \leq 0 \tag{3.11}$$

This constraint ensures that the recovered model is topologically consistent with the range observations. Accumulating these inequalities yields an additional convex constraint $D\xi \leq 0$, which is added to Equations 3.9 and 3.10. However, in practice we notice that the inclusion of these constraints can increase computation time for a limited gain as they are not typically violated given a reasonably accurate set of measurements.

3.2 General Case

In this section we seek to address the limitations of our previous approach in order to develop a system better capable of generalizing to a larger set of SLAM problems. In particular, we seek to relax the assumption of a strict Manhattan World. Instead, we will assume that the building structure must conform only to a finite set of *principal directions*, i.e. a Stata Center World (Bosse et al., 2003). We also address many of the limiting performance factors originally preventing true real-time application and introduce methods for increasing robustness. A real-time open source Python implementation of our updated system, which we call OccamSAM, can be found at `https://github.com/ashariati/occamsam`.

Our technical approach discussed here mirrors that of (Shariati et al., 2019). To begin, we provide an overview of the slight differences between the sensor configuration that we use for the latest experiments and that of the previous work.

Given that the primary landmark features we are interested in mapping are large planar structures corresponding to the building layout, we have replaced our original 3D sensors with a Velodyne, as the former are constrained by a extremely limited fields of views providing little coverage of the scene. In order to further expand the Velodyne's $30^\circ \times 180^\circ$ field of view, we attach the sensor to a upward-facing actuated mount, illustrated in Figure

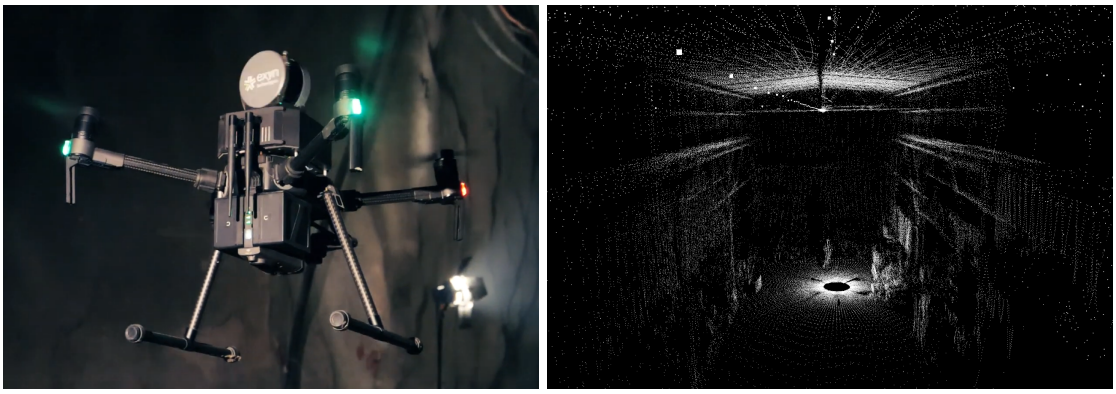


Figure 7: Quadrotor platform with a rotating Velodyne LIDAR configuration on top and resulting 360° scan of the scene (*courtesy of Exyn Technologies*).

7, which provides an additional axis of continuous rotation during each sweep. The result is a scan providing complete 360° coverage of the scene around the robot. Observe that the extensive coverage afforded by this new configuration results in a higher degree of visual contrast between the layout structures and surrounding clutter, and as a result, a simplified segmentation task.

Just as in the previous approach, we bootstrap our SLAM system to a separate state estimation subsystem which provides initial estimates of frame-to-frame motion. As we now use 3D LIDAR data exclusively for sensing, we use a customized inertial LOAM (Zhang and Singh, 2014) implementation instead of VIO from image sequences.

3.2.1 Structure Detection

Our analysis begins as before by identifying the dominant orientation of the building using the entropy compass method described in (Bazin et al., 2013) and (Taylor and Cowley, 2012). Given the measured gravity vector provided by our state estimation pipeline, we incrementally rotate the orbital scan in fractional degree increments to find a yaw orientation that minimizes the entropy of the projected point distribution along the x and y axis. Though this approach relies on a Manhattan World assumption, we note that any such method for tracking a stable fixed principal direction would suffice. In addition, in practice, most buildings contain at least one Manhattan frame that is easily observable at all times.

Given the rotation between the robot frame and the detected Manhattan frame in each keyframe, we then enumerate an additional set of principal directions by taking linear combinations of the three basis vectors comprising the Manhattan frame. In our current implementation, we let the set of principal directions be defined as

$$\mathcal{V} = \{\mathbf{v}_z, \mathbf{v}_x, \mathbf{v}_y, (\mathbf{v}_x + \mathbf{v}_y)/2, (\mathbf{v}_x - \mathbf{v}_y)/2\},$$

however, any finite number of combinations of the three Manhattan axis $\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z$ could be included in the set in order to model more complex building layouts. It is worth noting that methods for automatic surface normal clustering (Straub et al., 2015b) or directional segmentation (Straub et al., 2017) can also be used in order to generate the set of principal directions $\mathcal{V}$. In section 3.3.1 we provide an outline of another automatic approach based on our own model selective method.

Once a set of principal directions has been identified, we can then extract a set of layout planes by performing a series of planar sweeps, identifying peaks in the histograms of point density along each axis as plane locations (Budroni and Boehm, 2010). This approach has the added bonus of identifying layout planes supporting multiple discontinuous segments implicitly, which cuts down on the total number of measurements requiring data association. In addition, we have observed this method to be more robust to clutter especially when planes are extracted from our extensive 360° scans. The result is a fewer number of rows in our linear system during optimization, alleviating a previously observed performance bottleneck.

After re-expressing the frame-to-frame translations estimates provided by our LIDAR odometry within the frame of each identified building frame, and having detected a set of planes along each axis, we are again left with a pose-graph (Lu and Milios, 1997) comprised entirely of linear odometric and observational measurement factors.

3.2.2 Model Optimization

Our first major modification of the model optimization step includes the relaxation of the Manhattan World assumption. Moreover, we enable the incorporation of infinite planar landmarks aligned to any of the dominant principal directions within the building $\mathcal{V}$. The new linear observation constraints take the following form,

$$m_j^r - \mathbf{v}_r^T \mathbf{p}_i = d_{ij} + \epsilon_d, \tag{3.12}$$

where $\mathbf{v}_r$ represents one of the unit principal directions in $\mathcal{V}$, m_j^r corresponds to plane j's one dimensional offset in the direction of $\mathbf{v}_r$, and $\mathbf{p}_i$ once again represents the robots position at keyframe i. Notice the inclusion of the Gaussian noise characteristic for the range measurement $\epsilon_d \sim \mathcal{N}(0, \sigma_d)$ in Equation 3.12. As we will see in the following section, doing so not only provides for a more accurate reconstruction during model optimization, but a more principled way of defining reconstruction bounds during automatic model selection. Similarly, our linear odometric constraints are now expressed as

$$\mathbf{p}_{i+1} - \mathbf{p}_i = \mathbf{t}_i + \epsilon_t. \tag{3.13}$$

Given this new modified set of constraints, solving for the optimal set of plane locations m_j^r and sensor trajectory $\mathbf{p}_i$ amounts to solving the following weighted linear least-squares problem,

$$\underset{\xi}{\text{minimize}} \quad \|\Sigma^{-1} (A\xi - \mathbf{b}) \|_2, \tag{3.14}$$

where ξ is the vector formed by stacking the free parameters of the model, and the rows of A and $\mathbf{b}$ are formed by stacking the remaining components of the left and right-hand side, respectively, of Equations 3.12 and 3.13. Note the inclusion of the diagonal matrix Σ^{-1} containing the inverse of the estimated variance of the noise σ^2 for each measurement along its main diagonal.

Reducing Problem Size

Though the sparse linear system in Equation 3.14 is already efficiently solvable, the following marginalization scheme allows for scalable real-time performance. Like other incremental graph-based SLAM solvers, such as iSAM (Kaess et al., 2008), our method prevents the optimization from repeatedly updating portions of the state that are not affected by the latest measurements and have already realized values close to their optimum.

We begin with a simple factorization of the linear system in Equation 3.14 ignoring the weighting factor of Σ^{-1} for now without any loss of generality. Observe that the linear objective can be trivially rewritten as simultaneous optimization over the two types of free variables, namely planar landmark offsets and sensor positions,

$$\underset{\mathbf{m},\mathbf{p}}{\text{minimize}} \quad \left\| \begin{pmatrix} A_m & -A_p \\ 0 & B_p \end{pmatrix} \begin{pmatrix} \mathbf{m} \\ \mathbf{p} \end{pmatrix} - \begin{pmatrix} \mathbf{d} \\ \mathbf{t} \end{pmatrix} \right\|_2, \tag{3.15}$$

where the value of each $\mathbf{m}$ and $\mathbf{p}$ can be optimized independent of one another. Moreover, suppose that after solving the above optimization problem over $\mathbf{p}$ alone, that $\mathbf{p}^*$ corresponds to the optimal set of keyframe positions. Then, the optimization over planes $\mathbf{m}$ becomes,

$$\underset{\mathbf{m}}{\text{minimize}} \quad \left\| \begin{pmatrix} A_m \\ 0 \end{pmatrix} \mathbf{m} + \left(\begin{pmatrix} -A_p \\ B_p \end{pmatrix} \mathbf{p}^* - \begin{pmatrix} \mathbf{d} \\ \mathbf{t} \end{pmatrix} \right) \right\|_2. \tag{3.16}$$

Similarly, assuming that after we perform the optimization in Equation 3.15 over planes alone we have an optimal set of plane positions $\mathbf{m}^*$, we can reformulate the optimization over keyframe positions as,

$$\underset{\mathbf{p}}{\text{minimize}} \quad \left\| \begin{pmatrix} -A_p \\ B_p \end{pmatrix} \mathbf{p} - \left(\begin{pmatrix} -A_m \\ 0 \end{pmatrix} \mathbf{m}^* + \begin{pmatrix} \mathbf{d} \\ \mathbf{t} \end{pmatrix} \right) \right\|_2. \tag{3.17}$$

Observe however that since solving optimization defined in Equation 3.17 simply amounts

to solving the equivalent linear least-squares problem defined by squaring the objective function, we can express the value of $\mathbf{p}^*$ in closed form as a function of any given $\mathbf{m}$,

$$\begin{aligned}\mathbf{p}^* &= \left(\begin{pmatrix}-A_p & B_p\end{pmatrix}\begin{pmatrix}-A_p \\ B_p\end{pmatrix}\right)^{-1}\begin{pmatrix}-A_p & B_p\end{pmatrix}\left(\begin{pmatrix}-A_m \\ 0\end{pmatrix}\mathbf{m} + \begin{pmatrix}\mathbf{d} \\ \mathbf{t}\end{pmatrix}\right) \\ &= A_m^{\dagger}\mathbf{m} - \mathbf{c}\end{aligned} \tag{3.18}$$

Substituting this expression for $\mathbf{p}^*$ back into the optimization over planes defined in Equation 3.16 then yields the following marginalized problem,

$$\underset{\mathbf{m}}{\text{minimize}} \quad \left\| \left(\begin{pmatrix}A_m \\ 0\end{pmatrix} + \begin{pmatrix}-A_p \\ B_p\end{pmatrix} A_m^{\dagger}\right)\mathbf{m} - \left(\begin{pmatrix}-A_p \\ B_p\end{pmatrix}\mathbf{c} + \begin{pmatrix}\mathbf{d} \\ \mathbf{t}\end{pmatrix}\right)\right\|_2 . \tag{3.19}$$

Of course this can be written more compactly as,

$$\underset{\mathbf{m}}{\text{minimize}} \quad \|C_m\mathbf{m} - \mathbf{e}\|_2 . \tag{3.20}$$

As we expect the number of planes to grow more slowly as compared to the number of keyframes, the effect of this marginalization approach is that the size of our final optimization problem remains relatively fixed throughout the course of the exploration. This in turn provides real-time scalability without compromising an exact solution. Notice that at any point we can obtain the optimal trajectory by simply plugging in the value of $\mathbf{m}^*$ found by solving the optimization in Equation 3.20 back into Equation 3.18.

The formulation in Equation 3.20 can also be used in order to provide additional analytical insights about the nature of the optimization itself. Squaring the objective function again results in an equivalent optimization, but allows us to formulate the problem as a minimization of a quadratic function,

$$\underset{\mathbf{m}}{\text{minimize}} \quad \mathbf{m}^T C_m^T C_m \mathbf{m} - 2\mathbf{e}^T C_m \mathbf{m} + \mathbf{e}^T \mathbf{e}, \tag{3.21}$$

The key revelation provided by this reformulation is that given certain observability assumptions are met, the inverse of the Hessian, namely $(C_m^T C_m)^{-1}$, encodes the directions of greatest variance for the optimization. In other words, by examining the Eigen decomposition of this matrix, we can characterize the uncertainty in our final reconstruction.

3.2.3 Automatic Model Selection

For any pair of planar landmarks aligned with the same principal direction $\mathbf{v}_r$, we can assert that they correspond to the same layout structure in the real world if, after solving for the optimal set of model parameters best satisfying the given measurements, the following constraint is satisfied,

$$m_a^r - m_b^r = 0, \tag{3.22}$$

i.e. they reside at the same 1D offset. Unfortunately, even if such is the case in the real world, it is rather unlikely for the constraint to be satisfied exactly in practice, even after running the robust optimization in Equation 3.14 with modeled noise characteristics. This observation is what leads to our novel reformulation of the SLAM problem, which aims to maximize the number of satisfiable equivalence relations. This is achieved by enumerating all possible equivalences and searching through the resulting space of data associations by solving the following sparse optimization problem,

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & \|E\xi\|_1 \\ \text{subject to} \quad & \|A\xi - \mathbf{b}\|_2 \leq 2\|\Sigma\|_2. \end{aligned} \tag{3.23}$$

Recall that E is formed by stacking equivalence constraints between plane parameters of the same directional class – the form of which is given in Equation 3.22.

Observe that what prevents the optimization in Equation 3.23 from producing the trivial solution of all landmark parameters being equivalent is the bound on the reconstruction error. In contrast to our previous approach, which uses a hand-tuned parameter bounding the reconstruction error with respect to the error of the ordinary least-squares solution,

we now bound our reconstruction error in a more principled manner as it relates to the expected noise in each measurement. If we assume that the distribution of measurement errors is in fact Gaussian, the new bound affords enough slack for each measurement to maintain a 95% likelihood of occurring. In both simulation and the real world, this bound has proven to be both flexible and informative for consistent and accurate convergence.

Although this formulation provides an effective means for exploring the space of data associations and loop closures, an issue in practice is that all equivalences contained in the rows of E are given equal consideration by the optimization. That is, the value of associating one pair of landmarks is the same as any other. We can readily see that this behavior may be less than desirable in many circumstances as the optimization might match two relatively insignificant planar fragments at the expense of two major ones, so long as doing so continues to respect the reconstruction error bound. Therefore, this newest version of our system includes the assignment of a different weight to each equivalence relation in E based on its importance with respect to various criteria.

The first of these criteria relates the intensity, mass, or any other unit of size corresponding to each landmark using a simple summation as follows,

$$w_H(a, b) = h_a + h_b, \tag{3.24}$$

where h_a and h_b denote the size of planes a and b, respectively. In this case, the size of a layout plane refers to the size of its underlying layout segments. Applying this weighting scheme to every row of E results in a "gravity" effect in which planes with a larger amount of support attract more associations than planes with less support. For instance, the optimization considers the association of two large planes as very lucrative, while the association of two small planes as less so. Meanwhile, the association of a small plane and a large plane may still yield a modest reward somewhere in the middle.

The second of these criteria relates the pairwise distance between planes using the traditional

radial basis function,

$$w_D(a,b) = \exp\left(\frac{(m_a^r - m_b^r)^2}{2\sigma^2}\right). \tag{3.25}$$

The effect of applying this weighting function to each equivalence constraint is that as the distance between planes a and b increases, the value of their association rapidly approaches 0. Furthermore, by varying the value of σ one can manage the slope of this decline. Notice that we can use the weighting scheme in Equation 3.25 as a more elegant alternative to the hard decision threshold used originally to discard hypotheses whose initial pairwise distance between planes exceeded the fixed amount. Instead, we now only discard a hypothesis when its corresponding weight is approximately 0.

By defining weighting functions we have provided a framework for generalizing other criteria used in our previous method as well. For instance, our original constraint, that only planes observed to be facing the same way with respect to their principal direction may be associated with one another, can be enforced by using the following weighting function,

$$w_S(a,b) = \overline{\mathbf{sign}(d_a) \oplus \mathbf{sign}(d_b)}, \tag{3.26}$$

where $\oplus$ denotes the exclusive-or operation.

Ultimately, the true power of weighting functions comes to fruition as we begin to compose them with one other. For example, if we let our compositional weighting function be defined as,

$$w_C(a,b) = w_H(a,b) * w_D(a,b) * w_S(a,b), \tag{3.27}$$

we can blend the effects of each seamlessly into a single implicit constraint on the optimization procedure. Letting W represent the diagonal matrix whose i^{th} entry along its main diagonal be the value of $w_C(a,b)$ applied to the planes related by the equivalence in the i^{th}

row of E, our weighted optimization procedure becomes,

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & \|WE\xi\|_1 \\ \text{subject to} \quad & \|A\xi - \mathbf{b}\|_2 \leq 2\|\Sigma\|_2. \end{aligned} \tag{3.28}$$

The result of these extensions to the original method for automatic model selection is a more intelligent optimization procedure that is devoid of any user-defined parameters besides those that may be sparingly used to define new weighting functions. Once the optimization in Equation 3.28 is solved, we can then enforce the set of discovered associations E', i.e. the rows of E where $E\xi^*$ is close to zero, as constraints in Equation 3.14,

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & \|\Sigma^{-1}\left(A\xi - \mathbf{b}\right)\|_2 \\ \text{subject to} \quad & E'\xi = 0. \end{aligned} \tag{3.29}$$

3.3 Extensions of the Model Selection Approach

This section highlights a few noteworthy extensions of our model selective approach. Although these formulations are yet to be rigorously tested, we believe that they provide clear avenues for future research. In addition, they serve as examples demonstrating the flexibility of our method – using a linearized reconstruction error bound in order to guide a sparse optimization over possible data associations – and how it may be used to solve other various types of related problems.

3.3.1 Bearing Landmarks

The first problem we seek to address here is originally encountered in Section 3.2.1, where we were interested in a method for automatically detecting the set of principal directions within the building.

We begin again with the assumption that we have an estimate for the gravity vector at all keyframes and that the set of building principal directions $\mathcal{V}$ all reside in the plane defined

by this vector. Observe that in such a setting, identifying the set of unit vectors in $\mathcal{V}$ is equivalent to finding the corresponding yaw angle of each about the gravity vector. This one dimensional space is now analogous to the one dimensional space of plane offsets. However, although both spaces have the same topology (that of a 1D line), in the circular manifold of 1D angles, an infinite number of angles correspond to the same orientation, i.e. all those that are the same modulo 2π. This fact forces us to revise our definition of equivalence as well as our measurement model. Though we will lose the benefit of convexity in our modified version of the optimization problem, if the problem size is kept reasonably small, an efficient solution via branch and bound still exists.

We use the following notation to describe each component of our model in this setting. Let $\alpha_i \in \mathbb{R}$ denote the yaw angle of the sensor about the gravity vector in keyframe i while $\beta_j \in \mathbb{R}$ denotes the angular position of the j^{th} landmark. Note that both of these angles exist within the same axis-aligned frame. Thus, like our original approach, this formulation subsumes a method for tracking the orientation of some stable fixed frame that is observable across most keyframes. We remark that we deliberately keep the concept of a bearing a landmark general. These landmarks in fact could correspond to stable points along the horizon in an image. However, for the purpose of this example we say that they correspond to the orientation of a particular layout segment orthogonal to the xy-plane.

Furthermore, let the measurement ϕ_i represent the estimate for the angular difference between sequential sensor orientations. Precisely, this quantity is defined as,

$$\alpha_{i+1} - \alpha_i = \phi_i + \epsilon_\phi. \tag{3.30}$$

Similarly, we let γ_{ij} represent the estimate for the angular difference between the orientation of keyframe i and angular position of landmark j, i.e.,

$$\beta_j - \alpha_i = \gamma_{ij} + \epsilon_\gamma. \tag{3.31}$$

At this point we note the first major difference between the approach described here and that which we discussed earlier. That is, that this constraint alone cannot account for the fact that an infinite number of landmark bearing angles β_j are valid in reality by adding any factor of 2π. Therefore, we must introduce an additional integer variable $m_{ij} \in \mathbb{Z}$ in order to provide that flexibility. Thus, our observation measurement model becomes,

$$\beta_j - \alpha_i - 2\pi m_{ij} = \gamma_{ij} + \epsilon_\gamma. \tag{3.32}$$

The same limitation also arises when using our previous definition of equivalence, which is restated below using our angular notation as follows,

$$\beta_a - \beta_b = 0. \tag{3.33}$$

In this case, we would like to be able to be able to model the fact that any angular difference between two landmarks that is a multiple of 2π should imply that the landmarks correspond to the same entity. Therefore, we introduce a second integer variable $n_{ab} \in \mathbb{Z}$ to address this issue. As a result, our new equivalence constraint becomes,

$$\beta_a - \beta_b - 2\pi n_{ab} = 0. \tag{3.34}$$

Stacking the measurements and equivalence constraints defined in Equations 3.30, 3.32, and 3.34 using the same conventions to obtain our familiar sparse linear systems, we can define the new model selection procedure as,

$$\begin{aligned} \underset{\boldsymbol{\beta},\boldsymbol{\alpha},\mathbf{n},\mathbf{m}}{\text{minimize}} \quad & \|E\boldsymbol{\beta} - 2\pi\mathbf{n}\|_1 \\ \text{subject to} \quad & \|\left(A_\beta\boldsymbol{\beta} - A_\alpha\boldsymbol{\alpha} - 2\pi\mathbf{m}\right) - \boldsymbol{\gamma}\|_2 \leq 2\|\Sigma_\gamma\|_2 \\ & \|B_\alpha\boldsymbol{\alpha} - \boldsymbol{\phi}\|_2 \leq 2\|\Sigma_\phi\|_2 \end{aligned} \tag{3.35}$$

Unfortunately, due to the fact that the entries of $\mathbf{n}$ and $\mathbf{m}$ reside in $\mathbb{Z}$, Equation 3.35 is no longer a convex optimization problem, but a mixed integer linear program. Although it is still solvable with methods such as branch and bound, a greater degree of consideration must be taken in order to keep the problem tractable.

3.3.2 3D Landmarks

While the ability to quickly search through the vast space of possible data associations is a powerful tool, the formulations thus described are limited to optimization over 1D subspaces preventing our method's generalization as a solution to the full 3D SLAM problem. To this end, we provide the most exciting extension to our approach thus far, which enables optimization over the complete space of 3D points.

Before presenting the full details of this new formulation, we reassert the assumption that we have available some method for detecting some fixed orientation of the building across all keyframes.

Given this orientation with respect to keyframe i, R_i, we not only re-express the initial estimate of frame-to-frame translation within this frame, which we denote as $\mathbf{t}_i$, but also the coordinates of every detected point $\mathbf{m}_j \in \mathbb{R}^3$. Letting $\mathbf{p}_i \in \mathbb{R}^3$ again denote the position of keyframe i, we again obtain a set of linear odometric constraints,

$$\mathbf{p}_{i+1} - \mathbf{p}_i = \mathbf{t}_i + \epsilon_t, \tag{3.36}$$

and a set of linear landmark constraints,

$$\mathbf{m}_j - \mathbf{p}_i = \mathbf{d}_{ij} + \epsilon_d. \tag{3.37}$$

Also identical to our original formulation, is our definition of equivalence. Namely, that two landmarks are considered identical if they share the same position, though we extend this notion to 3D as follows,

$$\mathbf{m}_a - \mathbf{m}_b = 0. \tag{3.38}$$

Observe however, that for each equivalence relation defined in Equation 3.38 there exists three separate constraints – one for each dimension. Therefore, we cannot directly utilize our original objective function $\|E\mathbf{m}\|_1$ as doing so might result in only a subset of the three required constraints being satisfied as each constraint is treated as independent of one another by the optimization. In other words, if we are to conclude that a specific equivalence relation is satisfied, all three of its implicit constraints must be satisfied simultaneously.

We address this issue by using a generalization of the L_1-norm. The $L_{2,1}$ matrix norm for a $D \times Q$ matrix X is defined as

$$\|X\|_{2,1} = \sum_{j=1}^{Q} \left(\sum_{i=1}^{D} x_{ij}^2 \right)^{\frac{1}{2}}. \tag{3.39}$$

Notice that this norm is still a convex function in the columns of X as the operation amounts to a sum of convex functions, i.e. L_2 norms. Of particular interest however, is that minimizing this norm has the tendency to zero out entire columns of coefficients making its use as a convex relaxation to the original sparse problem appropriate. We leverage this insight into redefining our objective function.

Let M be the $3 \times N$ matrix formed by rearranging the set of landmarks such that the j^{th} column of M is $\mathbf{m}_j$. Furthermore, let E be defined as it was in the case of 1D landmarks, in that each row contains a 1 in column a and -1 in column b for each equivalence relation defined by Equation 3.38. For the sake of clarity, if there are Q possible landmark equivalences and N landmarks in total, then E is a $Q \times N$ matrix. Given this notation, we can state our new optimization problem as,

$$\begin{aligned} \underset{M,\mathbf{p}}{\text{minimize}} \quad & \|ME^T\|_{2,1} \\ \text{subject to} \quad & \|A_m \text{vec}(M) - A_p\mathbf{p} - \mathbf{d}\|_2 \leq 2\|\Sigma_d\|_2 \\ & \|B_p\mathbf{p} - \mathbf{t}\|_2 \leq 2\|\Sigma_t\|_2. \end{aligned} \tag{3.40}$$

This new objective now yields our desired behavior, which is to maximize the number

of enforceable 3D equivalences while still maintaining a reasonable bound on the total reconstruction error.

Given that we now possess the ability to map and associate arbitrary 3D points, we can readily see the implications of the optimization defined in Equation 3.40 towards solving the problem of Semantic SLAM. In Semantic SLAM, the 3D points often correspond to the location of various detected objects in the world. However, due to the discrete nature of object labels, it becomes unclear as to how to best associate the different landmarks with one another in the event of an erroneous classification. We now describe how our optimization may be modified in order to address this challenge.

Recall from Section 3.2.3 how we can introduce various types of weighting functions into our objective function in order to assign a different value for each suspected equivalence relation selected for enforcement by the optimization. In the case of Semantic SLAM, we would ideally like to assign weights based on our classification confidence for each landmark. Therefore, let us define a confidence-based weight for each pair of landmarks with the same class label s as a sum of both confidences,

$$w(a,b) = p(m_a^s = s) + p(m_b^s = s), \tag{3.41}$$

where $p_a(s)$ and $p_b(s)$ denote the probability mass function over labels for each landmark. Furthermore, we can also define weights associating pairs of landmarks with different class labels based on a conditional confidence as,

$$w(a,b) = p(m_b^s | m_a^s) + p(m_a^s | m_b^s), \tag{3.42}$$

We note that these conditional probability distributions may be obtained from the corresponding confusion matrix for each object detector. Given this set of weight functions, we can incorporate meaningful pairwise weights between possible equivalence relations as

before with the introduction of a diagonal weight matrix W into the objective function,

$$\begin{aligned} \underset{M,\mathbf{p}}{\text{minimize}} \quad & \|ME^T W\|_{2,1} \\ \text{subject to} \quad & \|A_m \text{vec}(M) - A_p\mathbf{p} - \mathbf{d}\|_2 \leq 2\|\Sigma_d\|_2 \\ & \|B_p\mathbf{p} - \mathbf{t}\|_2 \leq 2\|\Sigma_t\|_2. \end{aligned} \tag{3.43}$$

Although we are yet to provide a deeper quantitative analysis of this formulation's performance, preliminary results on simulated data indicate an ability to converge to compact accurate reconstructions with the same efficiency as our original approach. In fact, our open source implementation of OccamSAM utilizes the problem formulation in Equation 3.43 by default as it is simply a generalization of the 1D case.

Heterogeneous Landmarks

The optimization described in Equation 3.43 fully enables the mapping of general 3D points. We can exploit this property in order to also include other geometric primitives within our SLAM system that can be represented by a 3D point, such as planes.

The closest point representation of a plane is a three parameter representation which selects the point along the plane closest to the origin as its sole descriptor. Given a normal vector $\mathbf{v}$ and an offset m parameterizing a plane, the closest point can be computed as $m\mathbf{v}$. Conversely, given the closest point, which is itself a vector, the plane offset is provided by its magnitude while the plane normal corresponds to its normalized direction. Thus, we can incorporate our original orientation-constrained planes into the optimization simply by introducing two additional affine constraints for each point representing a plane, which constrain its position to reside along a 1D subspace defined by the plane's assigned principal direction.

Notice that we can use this approach in order to enforce additional interpenetration constraints between objects and the surrounding layout. By introducing additional convex

inequality constraints of the form,

$$\mathbf{v}_k^T(\mathbf{m}_{\text{point}} - \mathbf{m}_{\text{plane}}^k) \geq b_{\text{width}}, \tag{3.44}$$

where b_{width} corresponds to a bounding box width, we can ensure that a particular object's extent never intersects a layout plane. This constraint can even be made an equality in order to encode the fact that an object is explicitly supported by some plane. Moreover, either type of constraint can be included without violating the convexity of the original problem.

3.4 Experimental Results

The following set of reconstruction experiments were carried out using our initial approach assuming a Manhattan World, which is described in Section 3.1. The datasets used contain information collected in several different areas across our campus and have been made publicly available at (Pfrommer and Daniilidis, 2019). Data from three different sensor rigs was used for these experiments in order to demonstrate robustness. All of them have in common a custom stereo monochrome camera running at 20 Hz with a $97^\circ \times 81^\circ$ field of view (FOV) (Quigley et al., 2018). Together with a hardware synchronized inertial measurement unit (IMU), they provide data to drive the stereo VIO algorithm. The primary difference between rigs is the choice of depth sensor and its frame rate. Rig A (used for Areas 1,2, and 3) hosts a PMD Monstar time-of-flight depth sensor, which captures 352×287 resolution frames at 10 Hz and has a FOV of $100^\circ \times 85^\circ$. Rig B (used for Areas 4 and 5) features the same depth sensor, but run at 5Hz. Lastly, Rig C (used for Area 6) replaces the Monstar with an Orbbec Astra structured light camera, which runs at 30Hz and has a resolution of 640×400 and a FOV of $60^\circ \times 49.5^\circ$. All of our computations were carried out using an Intel i7-8700K CPU with 32GB of RAM.

With the exception of the threshold used for enumerating possible equivalences, all computations were carried out using the same set of optimization parameters agnostic to sensor

choice or environment. The entropy compass explored a radius of 0.3° around the estimate of angular displacement provided by the VIO system. The value of ϵ used to determine δ was empirically set to 2%. Finally, the threshold value of μ used to conclude equivalence between two segments was set to 30 centimeters. We crucially note however, that the use of so many user-defined parameters in these experiments is a significant drawback of our original formulation, which is why we have undertaken the effort to eliminate virtually all of those just described in our latest revisions to the approach described in Section 3.2. While it still remains in the latest implementation, the parameter μ is set to a more principled zero-threshold value of 0.001.

Figure 9 illustrates a composite of the reconstruction results in each environment using our convex approach, which we compare to reconstructions based on registering layout segments to the axis-aligned frame position they were observed in, as well as the results of the least-squares optimization described in Equation 3.5.

Different environments presented different challenges. For instance, Area 3 is rather large and contains many classrooms along its eastern wing. As a result, redundant candidate layout structures were generated as we encountered the same walls multiple times upon entering and exiting each classroom, which our optimization then had to consider. Area 6 features a modern architecture with many glass surfaces (embedded even in doors), large open areas, and exposed structural I-beams oriented at various angles. As a result, not only was the entropy analysis and layout segment detection confounded by the actual layout itself, but also by missing and corrupted depth measurements. Importantly, almost all of the examples involve situations where the robot needs to perform loop closures to account for situations where the same surface is encountered again after a significant interval of time. These loop closures are automatically detected and factored into the reconstruction as part of our procedure.

For each environment, Table 1 shows the number of equivalences considered, the number that were accepted, the number of layout structures in the original model and the number in

Table 1: Complexity results in each of the mapped environments

Area ID	1	2	3	4	5	6
Number of Equivalences Considered	151	2734	15631	454	1173	1704
Number of Equivalences Accepted	125	1431	10761	247	1092	1510
Number of Initial Layout Segments in Model	48	239	408	89	108	167
Number of Layout Structures After Analysis	14	34	40	28	25	16
Complexity Reduction %	70.8	85.8	90.2	68.5	76.9	90.4
Optimization Time (s)	0.12	2.02	50.67	0.23	0.79	0.59
Path Length (m)	53	200	249	69	113	67

Table 2: Drift in meters after each type of optimization

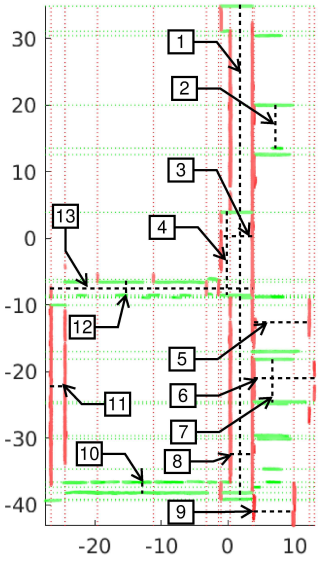
Optim.	Raw VIO	Entropy Compass	Least-Squares	Convex
Area 1	0.67	0.82	0.62	0.16
Area 2	2.36	3.06	1.58	0.39
Area 3	2.24	2.98	1.63	0.19
Area 4	0.46	0.35	0.48	0.09
Area 5	1.32	0.81	0.46	0.15
Area 6	0.73	1.07	0.88	1.11

the final simplified model. It also indicates the computational time required for the analysis and the length of the robot's trajectory. Note that the final optimized model contains far fewer layout planes than the original model.

We also mention that these reported computation times should be significantly improved with our latest implementation made publicly available at the following address `https://github.com/ashariati/occamsam`. In fact, the incorporation of the marginalization scheme outlined in Section 3.2.2 results in online update rates at about 10Hz on average.

Table 2 provides a quantitative analysis of the effects different types of optimization have on the trajectory drift. As the sensor rig is carried back to the starting location after each exploration, the values reported are the distances between the starting point and ending point of the trajectory after reconstruction. Note that in all cases but one, the convex optimization significantly reduced the drift in the reconstructed trajectory.

Figure 8 provides a quantitative analysis of the distance between selected surfaces in the recovered model compared to ground truth measurements of these distances taken with a



ID	GT	Model	Δ
1	73.15	74.06	0.91
2	6.48	6.47	0.01
3	4.72	4.72	0
4	12.70	12.46	0.24
5	8.48	8.34	0.14
6	9.01	9.08	0.07
7	6.34	6.38	0.04
8	3.35	3.32	0.03
9	6.10	6.00	0.10
10	1.47	1.59	0.12
11	2.10	2.10	0
12	2.32	2.36	0.04
13	30.00	30.26	0.26

Figure 8: Comparison of surface to surface distances against ground truth measurements collected with a laser range finder. All values in meters.

laser range finder in Area 3. The average reconstruction error in this set of measurements of 1.5%.

In an effort to better characterize the performance of our general case method, described in Section 3.2, we provide a comparative assessment against the Expectation Maximization approach described in (Bowman et al., 2017). We have chosen the latter approach for comparison as it stands as the most general and principled probabilistic approach to back-end data association for SLAM among the literature. Our experiment begins by generating 40 random simulations containing about 1000 keyframes and 80 landmarks belonging to one of 10 different unique landmark classes. We also note that all measurements are infused with synthetic noise. The primary difference we observe between both approaches is how the convergence time varies with the maximum number of landmark measurements made in a given keyframe. We vary the latter for each set of trials on the 40 simulations by explicitly pruning the set of measurements made at all keyframes such that their number does not exceed the fixed allotted amount. The results of our experiment are summarized in Figure 10. As we can see in Figure 10b, where the maximum number of measurements

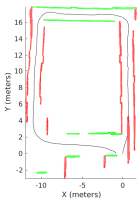
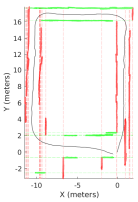
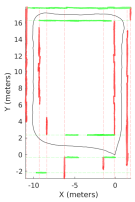
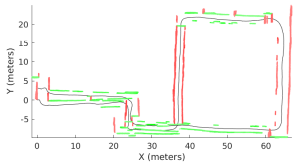
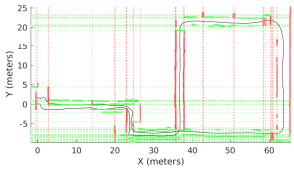
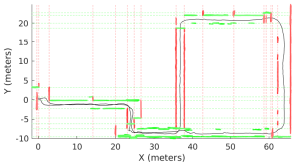
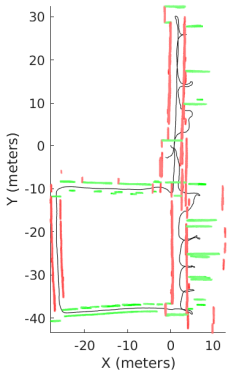
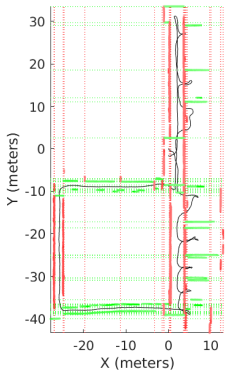
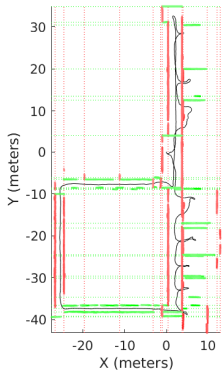
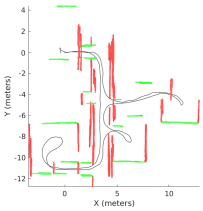
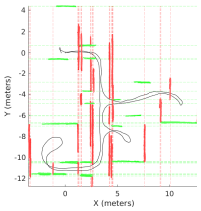
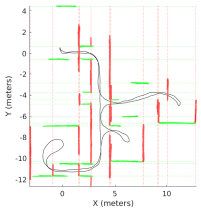
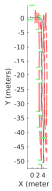
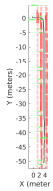
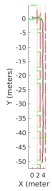
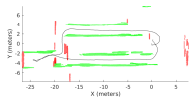
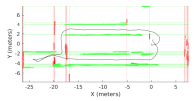
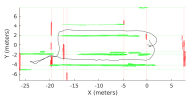
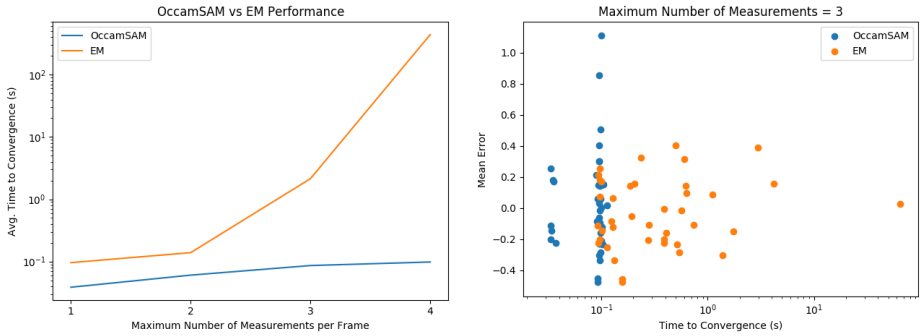
Area ID	Axis-Aligned Frame Registration	Least-Squares Optimization	Convex Optimization
Area 1			
Area 2			
Area 3			
Area 4			
Area 5			
Area 6			

Figure 9: Birds-eye view of the reconstruction results of our analysis in several Manhattan environments. Each column illustrates the effect of a different reconstruction method while each row corresponds to a different area. Red and green point clouds correspond to x and y-aligned layout segments, which reside on infinite layout planes denoted in each figure with red and green dotted lines. The black curve illustrates the sensor trajectory.



(a) (b)

Figure 10: Comparative results between our method, OccamSAM, and the Expectation-Maximization approach. Provided is a log-plot of the average time to convergence versus the maximum number of landmark measurements made in a given keyframe (a) and a log-plot of the mean error versus the time to convergence for that trial (b).

is capped at three, the accuracy of both methods is comparable across all trials. Although it would seem the EM approach is slower in most cases, the quality of the reconstruction is not correlated with the amount of time spent converging to a solution for either method. The real difference between the two methods comes to light as we vary the cap and observe that the convergence time of the EM approach begins to grow exponentially while that of OccamSAM remains fairly constant, as shown in Figure 10a.

We can better understand the reason behind this phenomena by closely examining the assumptions of the EM method. The EM SLAM method provides an approach to approximating the distribution of all possible data associations since any attempt to model it explicitly is quite intractable. However, even with the naive Bayes assumption that data associations across different keyframes are independent, their expectation step still requires a series of combinatorial sums. When used in many Semantic SLAM settings, the computational overhead remains low as one might expect to see at most two noteworthy objects in a given keyframe, but in general if this second assumption is violated, we can see that the method has difficulty scaling. On the other hand, OccamSAM allows for a comprehensive search over the entire space of data associations without any perceivable degradation in performance with the number of measurements.

3.5 Discussion

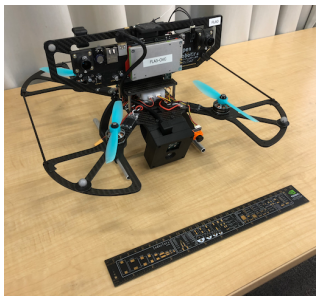
In summary, we have demonstrated an approach for generating compact planar reconstructions of indoor environments. In scenarios where a reasonable estimate for one's rotation can be inferred from the visible building structure, we can solve the full SLAM problem using convex optimization. Furthermore, our sparse objective enables us to explore the vast combinatorial space of potential data associations and loop closures, which results in a more accurate trajectory alongside a compact representation of the map. We validate our mapping procedure on a set of representative indoor environments using our original Manhattan assumption described in Section 3.1.

We have also further refined our approach, building upon the early success of our original system, by providing a generalization to a larger class of more complex indoor environments, achieving real-time performance, and computing robust reconstructions. This latest implementation, which is described in Section 3.2, has also been made available to the public in our open source release. We have even extended our model selective approach in order to begin addressing the full 3D SLAM problem with the ability to map 3D landmarks.

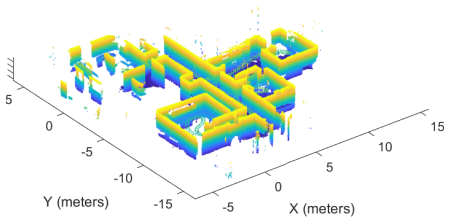
Chapter 4

Inferring Semantically Meaningful Building Models

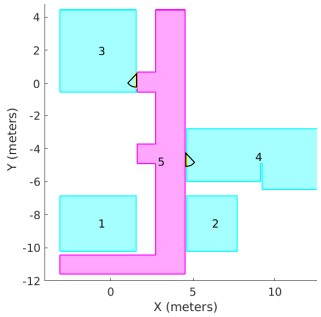
In this chapter we discuss how one might use the compact planar reconstructions provided by the structural SLAM systems described in Chapter 3, in order to estimate a complete building information model (BIM). Many current layout estimation pipelines focus on scene understanding from a single image or on generating BIMs only after the entire space has been surveyed. For an autonomous agent, such as the one shown in Figure 11, that must navigate through a previously unexplored environment, it is neither necessary nor desirable for it to have to observe every corner and crevasse of the environment in order to construct a water-tight model of the space. To this end, we provide an outline of two different algorithms which deliver comprehensive understanding of the general building structure by fusing information from multiple vantage points as the agent moves through the environment using only the typical sets of incomplete scans available to a robot over the course of its mission at any moment in time. Furthermore, our methods are designed to provide robust speculation as to the presence of boundaries and freespace in regions of the environment beyond the field of view. As we will see the compact planar representation provided by our structural SLAM system acts as a highly informative prior for inferring the nature of



(a) Quadrotor w/ Sensor Suite



(b) 3D Point Cloud Reconstruction



(c) Semantic Floor Plan

Figure 11: Our goal is to be able to automatically construct semantic layouts of indoor spaces based on the kinds of data that could be acquired from an autonomous robot like the one shown in (a). This system is equipped with a pair of stereo cameras, an IMU and a PMD depth camera. (b) Shows a small portion of the 3D point cloud that we can acquire by integrating information from the robots sensors (c) Shows the abstracted floor plan distilled from the 3D measurements that are acquired as the sensor suite is moved through the scene.

space in unexplored regions of the map. Ultimately, the intention behind synthesizing such models is to provide a semantic context for higher level planning tasks.

4.1 Door Detection

We begin first with the observation that within the context of buildings, doorways play a special role in scene understanding. That is, they signal a clear transition between two functionally disjoint spaces such as rooms and hallways. This motivates us to develop a scheme to automatically detect these features as they serve the more functional representation of our building models.

Given a layout plane we begin by accumulating all of the points associated with that plane as shown in Figure 12. Notice how the discontinuity of observations, which arises from incomplete and amodal perception, makes the problem of identifying concrete door boundaries more challenging than simply looking for negative space.

After aggregating each point set, we crop the resulting cloud at 2 meters which is close to the typical door height while the remaining points are projected to the line in order to construct a histogram of density as shown in Figure 12. We then compute a smoothed gradient of

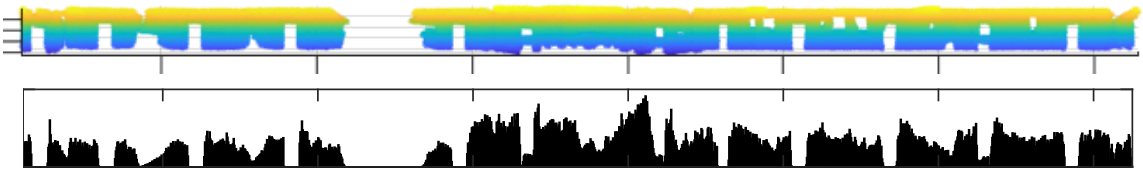


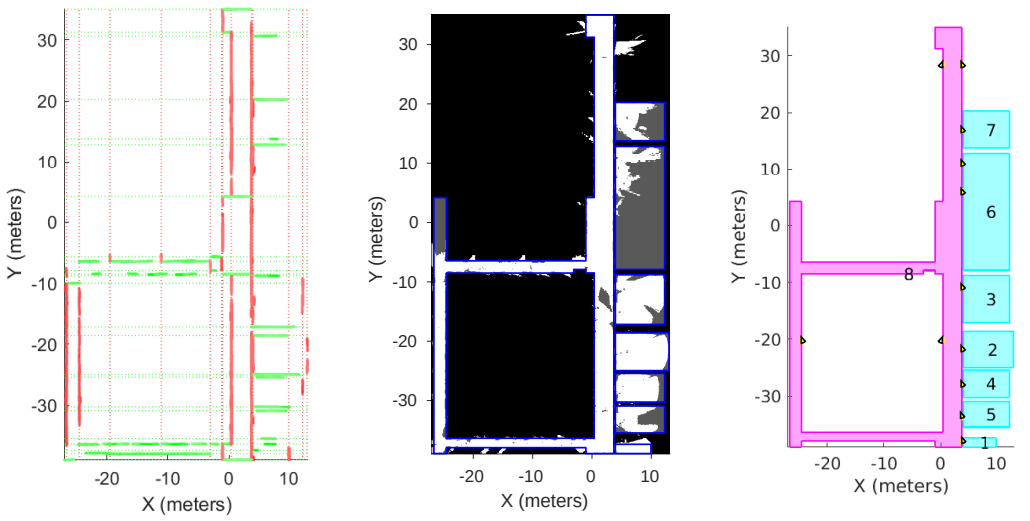
Figure 12: Result of merging the individual cloud segments associated with a particular layout plane (top). Histogram of projected points corresponding to the point cloud in Figure 12 cropped at 2 meters (bottom). The distance between ticks along the axis is 10 meters. Histogram bin counts range from 0 - 500.

the resulting signal and then convolve the result with a matched filter that is designed to detect 1 meter wide apertures. The existing system readily detects openings between 0.8 and 1.2 meters wide and can easily be extended to accommodate varying dimensions. Note the parent-child relationship between layout planes and door openings. Once the layout has been determined these doorways help to define the functional transitions between different spaces.

4.2 Layout Estimation

The two methods we present here for layout estimation act as the bedrock for our building modeling scheme. Recall that the output of our structural SLAM system is an optimized trajectory and the minimum set of layout planes necessary to explain all observed layout segments, where each layout plane is parameterized by its position and orientation class. Unfortunately, as is illustrated in Figure 13a, while a human may be able to readily discern each of the independent regions of the building given this planar representation alone, it is unclear how this process may be automated as a program since the true extent of individual walls and their meetings at corners are typically ill defined in the data. In addition, we note that a planar model alone does not provide any information with regards to which of the bounded regions consist of navigable freespace.

Although the approaches for layout modeling described in the sequel can readily be lifted to the general 3D case, for the sake of simplicity we describe the approach for modeling a single floor. The result is a 2D floor plan model of the building. Note however that the



(a) SL-LMS Input (b) Freespace Speculation (c) Semantic Floor Plan

Figure 13: A birds-eye perspective of the 3D reconstruction provided by our structural SLAM system is shown in (a). Red and green dotted lines indicate the position of different layout planes perpendicular to the x and y axis, respectively. Each red and green point cloud illustrates the portion of its corresponding layout plane which is observed. A generated floor plan outlined in blue overlaid on top of the occupancy grid is given in (b). Known free cells are colored white while unobserved cells speculated to be free based on the floor plan are colored gray. Occupied cells and unobserved cells outside of the domain of the floor plan are colored black. The final semantically colored floor plan with labeled region is shown in (c). Cyan regions correspond to rooms, while magenta regions regions correspond to corridors. Open doorways on the borders of each region are indicated.

system still analyzes a complete 3D point cloud and 3D trajectory in order to distill the aforementioned floor plan.

4.2.1 Greedy Cover Approach

This greedy approach to layout estimation is first described in our prior publication (Shariati et al., 2018). We begin by recognizing that regions of indoor freespace are typically enclosed by pairs of inward facing structures *i.e.* north-facing wall to south-facing wall and east-facing wall to west-facing wall. Therefore, given a set of axis-aligned layout planes as well as their observed orientation, we enumerate all possible north-south and east-west pairs,

$$\mathcal{P}_x = \mathcal{X}^+ \times \mathcal{X}^- \tag{4.1}$$

$$\mathcal{P}_y = \mathcal{Y}^+ \times \mathcal{Y}^- \tag{4.2}$$

where $\mathcal{X}^+$ and $\mathcal{X}^-$ are defined as the set of east-facing plane positions and west-facing plane positions respectively. Similarly, $\mathcal{Y}^+$ and $\mathcal{Y}^-$ are defined in the same way for north and south-facing planes. Using these intermediary sets, we then enumerate all the possible rectangles that could be used to explain the freespace

$$\mathcal{R} = \mathcal{P}_x \times \mathcal{P}_y. \tag{4.3}$$

This set however contains several different types of invalid rectangles including: those that have opposing faces which are outward facing; those whose length or width are too narrow (less than 1 meter for most indoor spaces); those which include portions of observed layout segments (projected to the ground plane after thresholding all points at a height greater than 2 meters) within their bounds; and those which include detected doorways within their bounds. Therefore, we prune the set of all such offending elements. It is interesting to note that this operation typically reduces the size of the original space of candidates by about 70 - 95%, which greatly improves the speed of our algorithm.

This approach to defining rectangular regions is similar to the scheme employed by Xiao and Furukawa (Xiao and Furukawa, 2014) but here we leverage the fact that we are pairing structural planes with infinite extent rather than incomplete wall segments. More specifically if we consider the example environment shown in Figure 13a the system considers pairs of the infinite dotted lines shown rather than just the solid segments where direct evidence is available. This approach allows the scheme to effectively speculate in regions that have not been observed yet.

After generating a voxel map reconstruction using the point cloud registration provided by our SLAM system, we can sample the map at a particular height in order to determine which cells each rectangle in $\mathcal{R}$ spans. We note that each voxel is 0.1 meters on side. At this point, we observe that the problem we are presented with can be phrased as a set

cover problem. We are given a universe F_h of n freespace cells in the 2D occupancy grid – generated by sampling from the 3D voxel map at a height h – each covered by at least one $R \in \mathcal{R}$, and a list of $R_1, \ldots, R_m \in \mathcal{R}$ rectangle subsets of F_h, each with its own weight defined as the total number of free, occupied, and unobserved cells in the grid it covers. What we would like is to select the a collection of rectangles, $\mathcal{C}$, of minimum total weight, whose union is equal to all of F_h. Minimizing this objective should, in principle, select those candidates which explain as much of the freespace as they can, while also yielding the simplest explanation of the space. The set cover problem is, of course, NP-Complete, however effective greedy solutions have been developed and we exploit one of these.

We construct the cover $\mathcal{C}$ of rectangles, by iteratively making the following greedy choice: select the rectangle R_i that minimizes

$$\frac{A_i}{|R_i \cap D|} \tag{4.4}$$

until no freespace voxels remain uncovered, where A_i denotes the sum of the number of free, occupied, and unobserved voxels within the span of R_i, and D is the set of remaining uncovered free voxels. If there happen to be two or more rectangles with the same ratio, we choose the one with the largest A_i. This algorithm has the interesting property that the cover selected has weight within a factor $O(\log d^*)$ of the optimal, where $d^* = \max_i |S_i|$ (Kleinberg and Tardos, 2006).

Given this cover, a floor plan can be generated by computing the union of all rectangles in $\mathcal{C}$. An example of such a floor plan can be seen in Figure 13b. Notice that our segment and doorway collision constraint on $\mathcal{R}$ results in the generation of functionally disjoint regions. These regions may also be given unique identifiers as illustrated in Figure 13c. It is important to mention that these regions are subject to one filtering criteria, which is that no region my have a ratio of total cells spanned to free cells spanned greater than 1000. This threshold may be tuned in order to limit the desired degree of risk in the speculation.

Based on which layout planes form the faces of each region, we can also reason as to which doorways act as transitions between pairs of adjacent regions given each region's well defined boundary. These doorways are highlighted in the semantically annotated version of the floor plan shown in Figure 13c.

Note that this optimization procedure seeks to find the simplest set of boxes that explains the available data which encourages the system to expand corridors and rooms since this allows it to explain larger regions with fewer primitives. In contrast, the optimization in (Xiao and Furukawa, 2014) was designed for situations where the space was completely scanned so the optimization penalizes primitives that include unexplored voxels.

We highlight the fact that while our method for cuboid generation relies on a Manhattan World assumption, the crux of our covering algorithm is agnostic to the specific shape of the provided primitives, since the notion of shape is abstracted away as simply a set spanning some elements. In other words, given some alternative method for enumerating any n-sided polyhedra, the same algorithm can be used to infer the layout extent in both the 2D and the 3D case.

4.2.2 Connected Components Approach

Here we discuss an alternative, more topological, approach to layout estimation than the one discussed previously. Though we stress that both methods are orthogonal to one another and could be used interchangeably.

We first begin with a description of a *cell complex* (Mura et al., 2014) and how one might be constructed given the planar representation provided by our structural SLAM system. Roughly speaking, a cell complex may be thought of as a non-recursive partitioning of space into a set of distinct cells. The result is something resembling that of an n-D grid, where each cell is now a convex shape instead of a cuboid, and each cell has as many neighbors as it has sides. An example of a simple 2D cell complex may be seen in Figure 14a. In order to construct this object, we first define the set of all planes aligned with the r^{th} principal

direction in the building as,

$$\mathcal{P}_r = \{v_r^x x + v_r^y x + v_r^z z - m_j^r = 0\}_{j=1}^{N_r}, \tag{4.5}$$

where each plane is simply parameterized by its normal direction $\mathbf{v}_r$ and its offset m^r along this axis. Furthermore, let the set of all planes in the world identified by our structural SLAM system be defined as

$$\mathcal{P} = \bigcup_{r=1}^{k} \mathcal{P}_r. \tag{4.6}$$

Given the set of planes $\mathcal{P}$ we can proceed with the algorithm provided in Algorithm 1 in order to obtain the set of cells $\mathcal{C}$ which define the cell complex. We begin in lines 2-3 by initializing the cell complex with the space spanned by the four corners defined by $u_{tr}, u_{tl}, u_{bl}, u_{br}$ which all reside along some predefined ground plane. We then proceed in sequence for each new plane. First, we partition the set of existing vertices into two sets depending on which side of the plane each corresponding point resides on in line 5. Then, the remainder of the procedure involves identifying each cell whose spanned vertices are partitioned by the cutting plane, introducing a new splitting edge according to the two points of intersection of the plane with the cells originally boundary, and subdividing the cell into two partitions. Note that since all cells remain convex throughout the procedure, based on the fact that they are each defined by a collection of half-spaces, we can be sure that if a cell is "hit" by a plane that the plane will intersect its boundary at exactly two points. Although we assume these two incident points occur along edges in line 7, this is without any loss of generality as if one or two of the points of intersection occur at an existing vertex, we simply refrain from introducing a new point and continue accordingly. With careful bookkeeping, an implementation of the algorithm just described can be made highly efficient. In addition, by generalizing the definition of a cell from a single convex polygon to that of a polyhedra defined by multiple polygonal faces, the algorithm can be readily extended to 3D.

Algorithm 1 Cell Complex Construction in 2D

procedure ConstructCellComplex($\mathcal{P}$)
  $\mathcal{V} \leftarrow \{u_{tr}, u_{tl}, u_{bl}, u_{br}\}$ ▷ Initialize vertex set
  $\mathcal{C} \leftarrow \{\{(u_{tr}, u_{tl}), (u_{tl}, u_{bl}), (u_{bl}, u_{br}), (u_{br}, u_{tr})\}\}$ ▷ Initialize cell set
  for all $p \in \mathcal{P}$ **do**
    Partition $\mathcal{V}$ into $\mathcal{V}^+$ and $\mathcal{V}^-$ with p
    for all $c \in \mathcal{C}$ with edges spanning sets $\mathcal{V}^+$ and $\mathcal{V}^-$ **do**
      Let $(u, v), (u', v') \in c$ be these two edges ▷ W.L.O.G.
      Let t and t' be the intersections of p with (u, v) and (u', v')
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{t, t'\}$
      $\mathcal{V}^+ \leftarrow \mathcal{V}^+ \cup \{t, t'\}$
      $\mathcal{V}^- \leftarrow \mathcal{V}^- \cup \{t, t'\}$
      $c' \leftarrow (c - \{(u, v), (u', v')\}) \cup \{(u, t), (t, v), (u', t'), (t', v')\}$
      $c^+ \leftarrow \{e = (u, v) \mid e \in c', u \in \mathcal{V}^+, v \in \mathcal{V}^+\} \cup (t, t')$
      $c^- \leftarrow \{e = (u, v) \mid e \in c', u \in \mathcal{V}^-, v \in \mathcal{V}^-\} \cup (t, t')$
      $\mathcal{C} \leftarrow (\mathcal{C} - c) \cup \{c^+, c^-\}$
    end for
  end for
  return $\mathcal{C}$
end procedure

This concludes the discussion of how we can transform the strictly planar representation provided by the structural SLAM system into a volumetric cell complex representation. However, in order to set the stage for our subsequent approach to layout speculation, we must first define two additional concepts.

The first concept is that of cell adjacency. Observe that any two cells in $\mathcal{C}$ that both contain the same boundary edge (u, v) are adjacent in this representation. If we collect all such pairs of cells into a set $\mathcal{A}$, we can construct a graph defined by $\mathcal{G} = (\mathcal{C}, \mathcal{A})$, where cells act as nodes in the graph G while their adjacency relations can be modeled with edges. We refer to this graph as a *graph complex*. Figure 14b provides a visualization of a graph complex superimposed onto its underlying cell complex.

The next concept we must define is that of *freespace evidence*. Recall in our previous layout estimation approach how we use voxels labeled as "free" by a ray-tracing algorithm as the entity to be explained by our covering algorithm. Here, we simply aim to group all such geometric objects that could be used to represent evidence for observed freespace under the

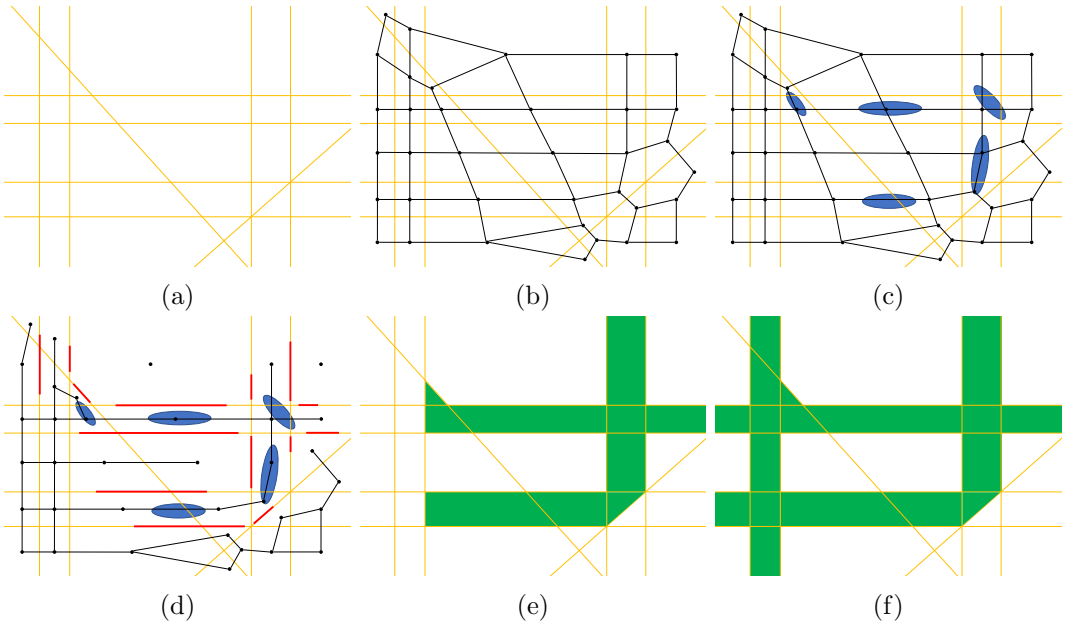


Figure 14: An example of a cell complex given a set of planes (a). Using implicit adjacency relations between cells, we can define a graph complex shown in (b). Before layout inference using a connected component analysis (e,f), we first insert freespace evidence (c) and then prune edges that cross an observed layout segment (d). The effects of using an increased speculation horizon of 2 is shown in (f).

same classification. For example, in this approach, we first compute covariance matrices for each detected floor and ceiling segment observed in each keyframe, project the corresponding ellipsoid to the ground plane, and then compute a convex approximation to the resulting ellipse's boundary. The resulting ellipse serves as a rough indicator of the area of freespace surrounding the robot at any given keyframe. By using a convex approximation, we can efficiently incorporate these shapes delineating freespace into the cell partitioning scheme in Algorithm 1 such that after the cell complex is constructed, some cells might contain fractional portions of the original freespace evidence.

We next describe our algorithm for layout speculation using this representation, which is provided in Algorithm 2. We also provide an illustration of the effects of each step of the algorithm against a simple example in Figure 14. In lines 2-3 the procedure begins with a set of preprocessing steps. We first mark all cells that contain freespace evidence. This step could also be augmented to take into account ratios of occupancy between the area occupied

Algorithm 2 Speculative Layout Estimation

1: **procedure** EstimateLayout($\mathcal{G} = (\mathcal{C}, \mathcal{A})$, speculationHorizon)
2: Mark all nodes $c \in \mathcal{C}$ that contain freespace evidence
3: Remove all edges $a \in \mathcal{A}$ whose shared cell boundary overlaps with any layout segment
4: Let $\mathcal{M} \subset \mathcal{G}'$ be the subgraph containing only marked nodes
5: Select $\mathcal{L} \subset \mathcal{G}'$ such that all nodes are within a distance of speculationHorizon of $\mathcal{M}$
6: **return** $\mathcal{L}$
7: **end procedure**

by the evidence and the area spanned by the cell itself. Next, we prune the set of edges of all adjacency relations for which the adjacent cells share a boundary which overlaps with any of the observed layout sections. In our implementation, checking this condition amounts to projecting the 3D layout segment point clouds associated with a particular plane to the induced boundary line and comparing the lengths of overlapping 1D intervals. Finally, using a slightly modified connected components procedure, we select the subgraph containing all marked nodes as well as those nodes which are within a specified distance of the original set of marked nodes. Note that the algorithm's level of speculative aggressiveness can be varied based on the magnitude of this parameter which takes the form of a positive integer. At the most conservative end of the spectrum, with a horizon of 0, our algorithm would return only the cells containing observed freespace evidence.

Though we have previously mentioned that this algorithm can be extended to 3D by generalizing the cell complex construction, notice that by instantiating a new 2D complex at every floor and simply sharing planes and freespace evidence across them, this procedure can be readily extended to 2.5D as well.

We draw the readers attention the fact that layout representation output by our algorithm is also entirely topological in nature. The union of all cells contained within a single connected component represents a disjoint functional space such as a room or hallway. Meanwhile, doorways, which now act as first-class citizens within this context, can be incorporated after the fact as special edges joining these otherwise disjoint connected components. Not only does this model provide a simplified geometric approximation to the building structure, but

it explicitly models the transitional continuity between each space.

4.3 Semantic Labeling

For any functional interpretation of space, it is important to understand what each region represents. In our classification scheme, we distinguish between two types of spaces: rooms and corridors. While this label space may not be comprehensive, we argue that it does capture the general purpose of most types of space – either the space itself acts as a transition, or the space is itself a terminal point where some particular event or action takes place. Observe however, that these categories, rooms in particular, can each be readily extended to include subcategories such as office, kitchen, etc.

The layout estimation algorithms described in the previous section produce a floor plan $\mathcal{L}$ comprised of k disjoint regions parameterized by sets of vertices. For each of the k regions we can compute several features to describe the particular space, including the area, perimeter and aspect ratio. Recognizing that the outer boundary of most rooms are typically close to square, the feature which yields the largest information gain between the two classes is the Turning distance (Arkin et al., 1991) between the region's outer boundary, with its perimeter normalized to 1, and the unit square. This quantity turns out to be quite useful as it implicitly captures the magnitude of various other attributes at once such as the number of sides and the aspect ratio.

Using these features, it is possible to use the following classifier to discriminate between the two types of regions:

$$h(x) = \begin{cases} \text{if } \texttt{perim}(x) < 60 \text{ and} & \textbf{room} \\ \texttt{turndist}(x, \texttt{square}) < 1, & \\ \text{else} & \textbf{corridor} \end{cases} \tag{4.7}$$

An example of a semantically labeled floor plan can be seen in Figure 13c. While this hand

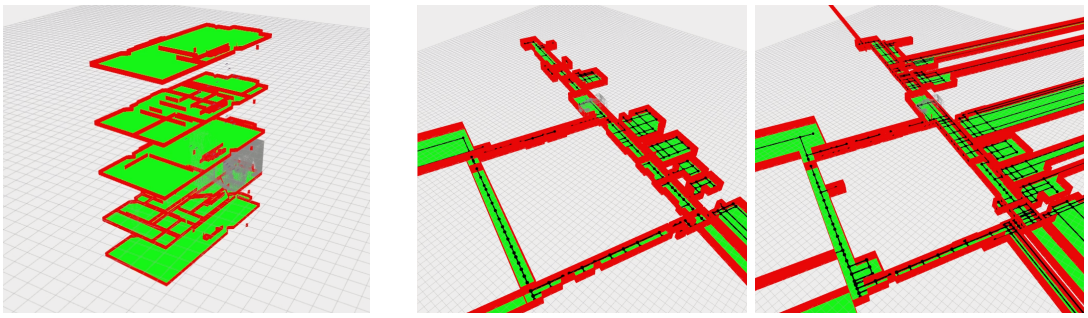
crafted classifier is quite simple, it does represent a useful baseline against which other, more sophisticated, schemes could be compared.

We feel it is important to emphasize that in this approach we are *not* performing semantic segmentation, but rather a semantic labeling of high-level regions in a water-tight model. While it may help provide a description of individual observations, semantic segmentation of a point cloud or an occupancy grid does not produce any abstraction or model that may be useful for higher-level reasoning.

4.4 Experimental Results

In order to evaluate the efficacy of our proposed methods for inferring semantically meaningful building models, we test them within a number of extended indoor environments. While we primarily utilize the GRASP MultiCam dataset (Pfrommer and Daniilidis, 2019), which consists of sequences of depth images, stereo images, and inertial measurements, we also rely on a supplemental dataset provided by Exyn Technologies, which consists of panoramic LIDAR scans with corresponding inertial measurements acquired by the robotic system shown previously in Figure 7.

Figure 13 and Figure 19 show the results of applying the interpretation scheme using the greedy approach to layout estimation in a batch mode on various datasets. In each of these cases the system was able to correctly infer the large scale building layout and partition the space into rooms and corridors. It was also able to correctly detect doorways which are indicated in each of the figures. These figures also compare the inferred freespace area with the freespace area that is actually observed to provide an indication of the systems ability to speculate about unexplored regions. The results show the system's ability to infer the presence of structure which is not directly observed. For instance, in Area 3, the algorithm is able to use the easternmost plane of Rooms 3, 4, and 5, in order to infer the presence of a back wall in Rooms 6 and 7, and cover the freespace observed in them. Also notice that the use of these rectangular primitives allows the system to approximate more complicated structure such as that seen in Room 4 of Area 4, which would have otherwise



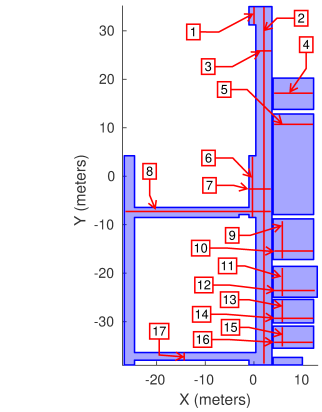
(a) (b)

Figure 15: A multilevel floorplan (a) constructed using our connected component approach to layout estimation over the course of an exploratory sequence provided by Exyn Technologies. For comparison against the greedy approach, floorplans produced using the same connected component method applied to GRASP MultiCam data collected in Area 3 (b) without a speculation horizion (left) and with a speculation horizon of 1 (right). Green surfaces correspond to freespace, while red surfaces correspond to boundaries. In (b), we also visualize the underlying graph complex.

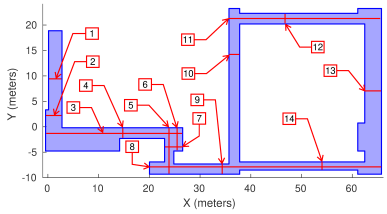
been lost in direct geometric model fitting schemes that would seek to approximate the entire space with a single cuboid. For a qualitative comparison against our connected components approach, in Figure 15b we provide the latter's best reconstruction estimate at the end of the exploratory sequence in Area 3. Observe that while both provide similar approximations to the space, the results of our connected component procedure can be varied depending on how aggressively the speculation parameter is set. We also demonstrate the system's ability to generate multiple floorplans for each level of a row home in Figure 15a. Note the incorporation of non-Manhattan walls in the final reconstruction.

In order to provide a more quantitative evaluation, we compare the dimensions of specific rooms and corridors within two inferred floorplans to measurements taken using a hand held laser range finder. Note that these floorplans are produced using the greedy cover approach to layout estimation. These results, which are presented in Figure 16, indicate that over all the dimensions that were considered the measurements and the predictions agreed on average within 2%.

Our last set of experiments involves running the cover-based algorithm in an online manner at regular intervals (10 - 20 seconds) during a set of sequences to provide an understanding



(a) Floor Plan A



(b) Floor Plan B

ID	Floor Plan A			Floor Plan B		
	GT	FP	Δ	GT	FP	Δ
1	3.62	3.75	0.13	2.72	2.66	0.06
2	73.15	73.91	0.76	3.23	3.25	0.02
3	3.34	3.32	0.02	27.13	27.04	0.09
4	9.03	8.31	0.72	2.17	2.10	0.07
5	8.46	8.31	0.15	9.15	9.14	0.01
6	12.60	12.73	0.13	4.61	4.57	0.04
7	4.72	4.72	0	3.59	3.58	0.01
8	29.99	30.34	0.35	45.04	45.72	0.68
9	8.41	8.38	0.04	2.02	2.02	0
10	8.48	8.31	0.17	2.10	2.10	0
11	6.33	6.37	0.04	30.01	30.06	0.05
12	9.02	9.06	0.04	2.00	2.08	0.08
13	4.80	4.84	0.04	3.30	3.31	0.01
14	8.44	8.31	0.13	1.68	1.6	0.08
15	4.56	4.56	0			
16	8.57	8.31	0.26			
17	1.47	1.58	0.11			
Avg. Δ	0.18			0.09		
% Δ.	1.98			1.31		

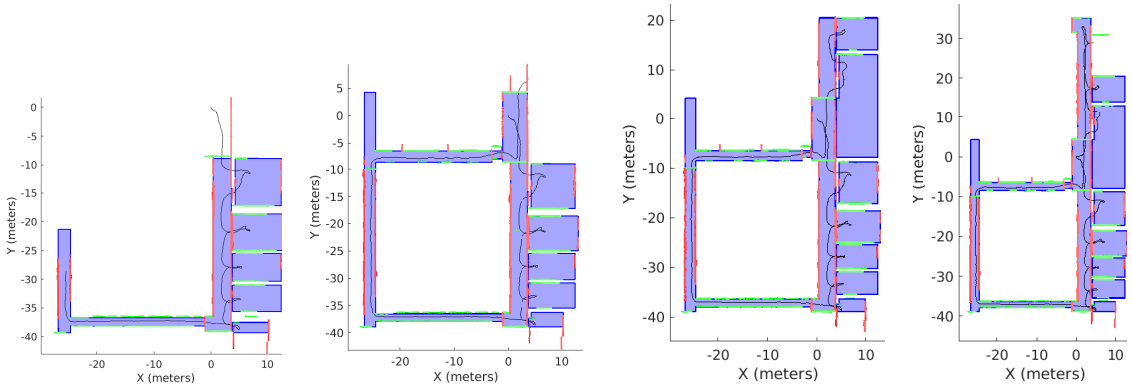
Figure 16: Floorplans A and B (left) annotated with locations of ground truth measurements. Differences between ground truth measurements and floor plan estimates are provided in table (right). All values are given in meters.

of how a robots' concept of the space would evolve as it moved through the environment. The procedure was carried out in two extended environments and the results are shown in Figures 17b and 18b. Although both spaces are of roughly the same dimension, 80×40 meters, the length of the exploratory path taken through these spaces as well as their respective topologies are quite different. Sample images taken in both environments can be seen in Figures 17a and 18a, which provide more context for the types of environments being explored.

The first environment is an academic building featuring vast hallways, large classrooms, and highly visible walls. These qualities naturally lead to simpler space, which leads to a faster convergence and a more accurate model being produced. The second environment is

an abandoned industrial laboratory and contains a larger number of densely packed interconnected rooms with more built in furniture which occludes the structural wall surfaces. This more complex structure results in fewer observations of the dominant structure, but a significant overall increase in the total number of planes detected, and as a result leads to a more challenging optimization. The sequence is also significantly longer than first (767 meters vs 247 meters). Nonetheless, despite these challenges the system is still able to extract the major structural features of the space and produce an estimate for the floor plan. Observe that as the exploration proceeds and the system learns about more structural planes it is able to use these to posit more accurate completions of the space. For example the system is able to apprehend the dimensions of neighboring rooms on a corridor by suggesting that they share some of the same structural walls even when those surfaces are not directly observed in each room since the optimization algorithm favors simple, regular explanations.

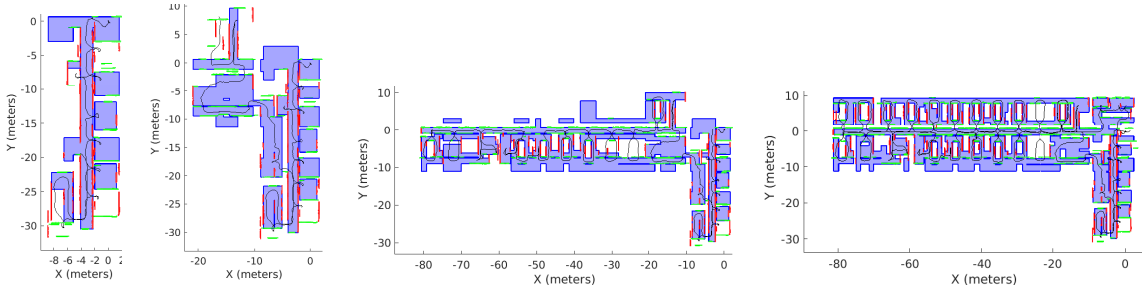
(a) Sample Images



(b) Estimated floor plan at $t = 1000$, 1400, 1600, and 2075 seconds.

Figure 17: Online estimation results for Building A. Total distance traveled = 247 meters.

(a) Sample Images



(b) Estimated floor plan at $t = 600$, 1000, 2400, and 3926 seconds.

Figure 18: Online estimation results for Building B. Total distance traveled = 767 meters.

Figure 19: Batch floorplan generation results in several indoor environments. The first column shows the input provided by our structural SLAM system. The second column illustrates the difference between occupied, free, and speculated freespace. The third column shows the semantic floor plan with doors, corridors, and rooms highlighted in yellow, magenta, and cyan respectively.

4.5 Discussion

In this chapter, we have described two algorithms that can be used to automatically extract plausible building models of indoor scenes based on the kinds of incomplete 3D data that an autonomous mobile robot could acquire. The extracted building model is designed to be useful for subsequent motion planning procedures since it produces water tight explanations of space that complete partially observed layout surfaces and infer likely regions of freespace. This ability to better understand the space from limited data allows the system to construct better motion plans with less data obviating the need to exhaustively explore each corner of the scene.

Both of described methods for speculative layout estimation exploit an underlying structural SLAM algorithm which provides them with estimates for the salient structural planes in the environment and construct volumetric explanations using those infinite planes as boundaries. This allows each system to suggest relationships between rooms that are not readily evident in the acquired data. While they share this key commonality, they also share several key differences.

The first algorithm leverages a connection between the layout estimation task and the set cover problem, reducing one to the other. This enables an effective optimization algorithm with provable performance guarantees with respect to how well the resulting solution approximates the space. Though its greedy nature leads to a simple linear time optimization procedure, the algorithm can have difficulty scaling over longer missions. This is due to the fact that the enumeration of primitives and the subsequent search through the set must be done from scratch at every timestep. However, in reality, over a sequence of spatiotemporal observations, changes to the model typically occur in highly local regions around the sensor. Another outstanding issue is how to enumerate other types of shape primitives other than cuboids. In the Manhattan case, it is relatively simple to enumerate all the pairs of planes necessary to form 6 sides of cube. This method however quickly becomes intractable for enumerating general convex polygonal shapes of arbitrary sidedness.

In an effort to address these limitations, we devise our second algorithm that exploits the implicit adjacency relationships within cell complexes. This formulation of the problem leads to a connected component analysis with constraints, which can also be solved in linear time. We also showcase the fact that the graphical nature of the underlying representation leads to a natural topological understanding of the space. An additional advantage of this approach is that the degree of speculation can be varied using a user-defined parameter. Though this method alleviates some of the concerns surrounding its predecessor, it introduces others. Chief among these is that those models produced by this algorithm may be slightly more complex – using more faces than is required to approximate the space – than those produced by the greedy method which explicitly prioritizes model simplicity. Furthermore, while a fully online implementation of the 3D generalization is feasible, it may require significant engineering.

The ability to accurately extract room layout structure is an important first step in scene interpretation. These results provide context which can be used to inform other semantic analysis operations such as detecting and positioning furniture and speculating about the function of different spaces. Ultimately, it helps the autonomous system to apprehend the scene at a higher level of abstraction and communicate more effectively with human interlocutors.

Chapter 5

Conclusion and Future Work

In conclusion, we believe that the set of methods presented in this work may act as viable solutions towards producing compact higher-level representations of indoor environments that may be useful for robot navigation and planning. Specifically, they are designed to fuse partial observations of the space collected over the course of exploration into a comprehensive building information model (BIM) of the entire space incrementally over time. Our structural SLAM procedure, described in Chapter 3, begins by providing an accurate estimate of the robot trajectory alongside the simplest planar model of the building which best explains the different planar segments corresponding to walls, floors, and ceilings. This planar model is then passed to our system for speculative building modeling, described in Chapter 4, which deduces freespace bounds, infers the presence of higher level semantic regions, and provides insights as to the nature of space beyond the explored horizons of the map.

In addition to providing experimental results that support the efficacy of our approach towards online construction of speculative BIMs, we also outline several theoretical extensions of our work for future investigations that may greatly expand the richness and utility of our synthesized models. Though it extends beyond the scope of this thesis, the advances

made in 3D learning in recent years begs the incorporation of detected 3D object models into our building modeling schemes. Not only would objects greatly expand the richness of the final product,but their identification within the scene could also greatly serve the effectiveness of the entire system. In the front end, methods for object detection may be used to more accurately separate layout from object clutter. Meanwhile, in the back-end, the presence or absence of different types of objects would likely serve as a much stronger signal when classifying various disjoint spaces than what is currently used. At the very least, the inclusion of water-tight object models embedded within the final reconstruction would better serve subsequent planning algorithms by providing information as to the different interactive elements within the environment. Fortunately, in Section 3.3.2, we have already described how one might intelligently incorporate 3D object detections within our model selective SLAM approach, such that their space of data associations and positions can be explored simultaneously alongside those of the accompanying layout planes in a way that also respects physical interpenetration constraints. We also note that these same methods for 3D object detection and modeling may also be adapted for more intelligent door detection as well.

Ultimately, while we assert that we have either addressed, or at least begun to address, most of the challenges associated with our problem statement, which we outlined in Chapter 1, with respect to our original goal, two major avenues of future work still remain.

The first includes the development of a standard benchmark for online freespace prediction. While our systems qualitatively appear to provide reasonable speculations when compared to what a human might estimate the space should look like, we still lack a method for explicitly quantifying this notion of accuracy. It is clear that some form of a binary classification scheme for evaluation is necessary. However, observe that the issue arises when establishing the ground truth signal. If we take for instance the ground truth BIM of the entire building for comparison, it is clear that predictions made earlier during the exploration may be quite inaccurate although they might still be reasonable guesses at the time. On the other hand,

if the ground truth is taken at some fixed time too near in the future, the system may receive too much credit for relatively trivial predictions. It is thus our goal to make this notion of predictive accuracy more concrete such that future methods may be developed against a well defined benchmark.

The second is that we still require an experiment demonstrating the effectiveness of speculative BIMs for high-performance motion planning into unobserved space as compared to alternative methods such as (Richter et al., 2018). While the latter addresses this challenge from the planner point-of-view, asserting that any attempt to directly model the distribution of maps to be computationally intractable due to the large dimensionality and inherent complexity of the space, we aim to challenge that claim by effectively identifying modes in the distribution using more advanced perception and mapping methods. Therefore, we propose an experiment which involves outfitting two equivalent high-performance robotic platforms with one of the two aforementioned software systems, and measuring the speed and efficiency with which they explore a set of complex indoor environments over the course of a given mission. Although we might expect the planning-based approach to be computationally cheaper and to achieve better performance when encountering scenes similar to those that it had seen during training, we expect our vision-based approach to enable the computation of much longer trajectories and more efficient global plans across a larger set of environments.

Bibliography

G. S. Administration. 3D-4D Building Information Modeling, 2003. URL https://www.gsa.gov/real-estate/design-construction/3d4d-building-information-modeling.

S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle Adjustment in the Large. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 29–42. Springer, Berlin, Heidelberg, 2010. ISBN 3642155510. doi: 10.1007/978-3-642-15552-9_3. URL http://link.springer.com/10.1007/978-3-642-15552-9{_}3.

S. Agarwal, V. Shree, and S. Chakravorty. RFM-SLAM: Exploiting relative feature measurements to separate orientation and position estimation in SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6307–6314. IEEE, may 2017. ISBN 978-1-5090-4633-1. doi: 10.1109/ICRA.2017.7989746. URL http://ieeexplore.ieee.org/document/7989746/.

S. Agarwal, K. S. Parunandi, and S. Chakravorty. Robust Pose-Graph SLAM Using Absolute Orientation Sensing. *IEEE Robotics and Automation Letters*, 4(2):981–988, apr 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2893436. URL https://ieeexplore.ieee.org/document/8613891/.

P. Amayo, P. Pinies, L. M. Paz, and P. Newman. A unified representation for application of architectural constraints in large-scale mapping. In *Proceedings of the IEEE International*

Conference on Robotics and Automation (ICRA), pages 1339–1345, Stockholm, Sweden, may 2016. IEEE. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487267. URL http://ieeexplore.ieee.org/document/7487267/.

V. Angladon, S. Gasparini, and V. Charvillat. Room Floor Plan Generation on a Project Tango Device. In *Proceedings of the International Conference on Multimedia Modeling*, pages 226–238. Springer, Cham, 2018. ISBN 9783319735993. doi: 10.1007/978-3-319-73600-6_20. URL http://link.springer.com/10.1007/978-3-319-73600-6{_}20.

E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216, mar 1991. ISSN 01628828. doi: 10.1109/34.75509. URL http://ieeexplore.ieee.org/document/75509/.

I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, Las Vegas, NV, USA, jun 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.170. URL http://ieeexplore.ieee.org/document/7780539/.

I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3D Scene Graph: A Structure for Unified Semantics, 3D Space, and Camera. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5664–5673, 2019. URL http://openaccess.thecvf.com/content_ICCV_2019/html/Armeni_3D_Scene_Graph_A_Structure_for_Unified_Semantics_3D_Space_ICCV_2019_paper.html.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, sep 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1678144. URL http://ieeexplore.ieee.org/document/1678144/.

S. Y. Bao and S. Savarese. Semantic structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2025–2032. IEEE, jun 2011. ISBN 978-1-4577-0394-2. doi: 10.1109/CVPR.2011.5995462. URL http://ieeexplore.ieee.org/document/5995462/.

J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally Optimal Consensus Set Maximization through Rotation Search. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 539–551. Springer, Berlin, Heidelberg, 2013. ISBN 9783642374432. doi: 10.1007/978-3-642-37444-9_42. URL http://link.springer.com/10.1007/978-3-642-37444-9{_}42.

P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. doi: 10.1109/34.121791.

M. Bosse, R. Rikoski, J. Leonard, and S. Teller. Vanishing points and three-dimensional lines from omni-directional video. *The Visual Computer*, 19(6):417–430, oct 2003. ISSN 0178-2789. doi: 10.1007/s00371-003-0205-3. URL http://link.springer.com/10.1007/s00371-003-0205-3.

S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, Singapore, Singapore, 2017. IEEE. ISBN 9781509046331. doi: 10.1109/ICRA.2017.7989203.

A. Budroni and J. Boehm. Automated 3D Reconstruction of Interiors from Point Clouds. *International Journal of Architectural Computing*, 8(1):55–73, 2010. ISSN 1478-0771. doi: 10.1260/1478-0771.8.1.55. URL http://journals.sagepub.com/doi/10.1260/1478-0771.8.1.55.

R. Cabral and Y. Furukawa. Piecewise planar and compact floorplan reconstruction from

images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 628–635, 2014. ISBN 9781479951178. doi: 10.1109/CVPR.2014.546.

C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics*, 32(6):1309–1332, dec 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2624754. URL `http://ieeexplore.ieee.org/document/7747236/`.

F. Camposeco and M. Pollefeys. Using vanishing points to improve visual-inertial odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5219–5225, 2015. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139926.

L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona. A fast and accurate approximation for planar pose graph optimization. *The International Journal of Robotics Research*, 33 (7):965–987, jun 2014a. ISSN 0278-3649. doi: 10.1177/0278364914523689. URL `http://journals.sagepub.com/doi/10.1177/0278364914523689`.

L. Carlone, A. Censi, and F. Dellaert. Selecting good measurements via l1 relaxation: A convex approach for robust estimation over graphs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2667–2674. IEEE, sep 2014b. ISBN 978-1-4799-6934-0. doi: 10.1109/IROS.2014.6942927. URL `http://ieeexplore.ieee.org/document/6942927/`.

L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4597–4604. IEEE, may 2015. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139836. URL `http://ieeexplore.ieee.org/document/7139836/`.

R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4102–4107. IEEE, apr 2007. ISBN 1-4244-0602-1. doi: 10.1109/ROBOT.2007.364109. URL `http://ieeexplore.ieee.org/document/4209727/`.

A.-L. Chauve, P. Labatut, and J.-P. Pons. Robust piecewise-planar 3D reconstruction and completion from large-scale unstructured point data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1261–1268. IEEE, jun 2010. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5539824. URL `http://ieeexplore.ieee.org/document/5539824/`.

X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3D Object Detection for Autonomous Driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2147–2156. IEEE, jun 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.236. URL `http://ieeexplore.ieee.org/document/7780605/`.

X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3D Object Detection Network for Autonomous Driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6526–6534. IEEE, jul 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.691. URL `http://ieeexplore.ieee.org/document/8100174/`.

S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565. IEEE, jun 2015. ISBN 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7299195. URL `http://ieeexplore.ieee.org/document/7299195/`.

J. Civera, D. Galvez-Lopez, L. Riazuelo, J. D. Tardos, and J. M. M. Montiel. Towards semantic SLAM using a monocular camera. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1277–1284. IEEE, sep 2011.

ISBN 978-1-61284-454-1. doi: 10.1109/IROS.2011.6048293. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6048293.

J. Coughlan and A. Yuille. Manhattan World: compass direction from a single image by Bayesian inference. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 941–947, 1999. ISBN 0-7695-0164-8. doi: 10.1109/ICCV.1999.790349. URL http://ieeexplore.ieee.org/document/790349/.

A. Criminisi, I. Reid, and A. Zisserman. Single View Metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. doi: 10.1023/A:1026598000963. URL https://www.cs.cmu.edu/{~}ph/869/papers/Criminisi99.pdf.

A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt. BundleFusion. *ACM Transactions on Graphics*, 36(3):1–18, may 2017. ISSN 07300301. doi: 10.1145/3054739. URL http://dl.acm.org/citation.cfm?doid=3072959.3054739http://dl.acm.org/citation.cfm?doid=3087678.3054739.

A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Niessner. ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587. IEEE, jun 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00481. URL https://ieeexplore.ieee.org/document/8578579/.

S. Dasgupta, K. Fang, K. Chen, and S. Savarese. DeLay: Robust Spatial Layout Estimation for Cluttered Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 616–624. IEEE, jun 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.73. URL http://ieeexplore.ieee.org/document/7780442/.

A. J. Davison and D. W. Murray. Simultaneous localization and map-building using active

vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002. ISSN 01628828. doi: 10.1109/TPAMI.2002.1017615.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (6):1052–1067, jun 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1049. URL `http://ieeexplore.ieee.org/document/4160954/`.

P. de la Puente and D. Rodriguez-Losada. Feature based graph-SLAM in structured environments. *Autonomous Robots*, 37(3):243–260, oct 2014. ISSN 0929-5593. doi: 10.1007/s10514-014-9386-z. URL `http://link.springer.com/10.1007/s10514-014-9386-z`.

P. de la Puente and D. Rodriguez-Losada. Feature based graph SLAM with high level representation using rectangles. *Robotics and Autonomous Systems*, 63(P1):80–88, jan 2015. ISSN 09218890. doi: 10.1016/j.robot.2014.09.006. URL `https://linkinghub.elsevier.com/retrieve/pii/S092188901400181X`.

E. Delage, H. Lee, and A. Y. Ng. A Dynamic Bayesian Network Model for Autonomous 3D Reconstruction from a Single Indoor Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2418–2428, New York, NY, USA, USA, 2006. IEEE. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.23. URL `http://ieeexplore.ieee.org/document/1641050/`.

Z. Deng and L. J. Latecki. Amodal Detection of 3D Objects: Inferring 3D Bounding Boxes from 2D Ones in RGB-Depth Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 398–406. IEEE, jul 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.50. URL `http://ieeexplore.ieee.org/document/8099533/`.

K. Doherty, D. Fourie, and J. Leonard. Multimodal Semantic SLAM with Probabilistic Data Association. In *Proceedings of the IEEE International Conference on Robotics*

and Automation (ICRA), pages 2419–2425. IEEE, may 2019. ISBN 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8794244. URL https://ieeexplore.ieee.org/document/8794244/.

Y. Du, Z. Liu, H. Basevi, A. Leonardis, W. T. Freeman, J. B. Tenenbaum, and J. Wu. Learning to Exploit Stability for 3D Scene Parsing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1726–1736. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper/7444-learning-to-exploit-stability-for-3d-scene-parsing.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, jun 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1638022. URL http://ieeexplore.ieee.org/document/1638022/.

M. Dzitsiuk, J. Sturm, R. Maier, L. Ma, and D. Cremers. De-noising, stabilizing and completing 3D reconstructions on-the-go using plane priors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3976–3983. IEEE, may 2017. ISBN 978-1-5090-4633-1. doi: 10.1109/ICRA.2017.7989457. URL http://ieeexplore.ieee.org/document/7989457/.

F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 3-D Mapping With an RGB-D Camera. *IEEE Transactions on Robotics*, 30(1):177–187, feb 2014. ISSN 1552-3098. doi: 10.1109/TRO.2013.2279412. URL http://ieeexplore.ieee.org/document/6594910/.

J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 834–849, Zurich, Switzerland, 2014. Springer, Cham. ISBN 9783319106045. doi: 10.1007/978-3-319-10605-2_54. URL http://link.springer.com/10.1007/978-3-319-10605-2{_}54.

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with

Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, sep 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2009.167. URL http://ieeexplore.ieee.org/document/5255236/.

C. Fernandez-Labrador, A. Perez-Yus, G. Lopez-Nicolas, and J. J. Guerrero. Layouts From Panoramic Images With Geometry and Deep Learning. *IEEE Robotics and Automation Letters*, 3(4):3153–3160, oct 2018. ISSN 2377-3766. doi: 10.1109/LRA.2018.2850532. URL https://ieeexplore.ieee.org/document/8395352/.

A. Flint, C. Mei, I. Reid, and D. Murray. Growing semantically meaningful models for visual SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 467–474, San Francisco, CA, USA, 2010. IEEE. ISBN 9781424469840. doi: 10.1109/CVPR.2010.5540176.

A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3D features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2228–2235. IEEE, nov 2011. ISBN 978-1-4577-1102-2. doi: 10.1109/ICCV.2011.6126501. URL http://ieeexplore.ieee.org/document/6126501/.

Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 80–87. IEEE, sep 2009a. ISBN 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459145. URL http://ieeexplore.ieee.org/document/5459145/.

Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1422–1429, Miami, FL, USA, 2009b. IEEE. ISBN 9781424439935. doi: 10.1109/CVPRW.2009.5206867.

D. Gálvez-López, M. Salas, J. D. Tardós, and J. Montiel. Real-time monocular object SLAM. In *Robotics and Autonomous Systems*, volume 75, pages 435–449. Elsevier B.V.,

jan 2016. ISBN 0921-8890. doi: 10.1016/j.robot.2015.08.009. URL https://linkinghub.elsevier.com/retrieve/pii/S0921889015001864.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE, jun 2014. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.81. URL http://ieeexplore.ieee.org/document/6909475/.

M. C. Graham, J. P. How, and D. E. Gustafson. Robust incremental SLAM with consistency-checking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 117–124. IEEE, sep 2015. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS.2015.7353363. URL http://ieeexplore.ieee.org/document/7353363/.

V. Guizilini, R. Senanayake, and F. Ramos. Dynamic Hilbert Maps: Real-Time Occupancy Predictions in Changing Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4091–4097. IEEE, may 2019. ISBN 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8793914. URL https://ieeexplore.ieee.org/document/8793914/.

R. Guo and D. Hoiem. Support Surface Prediction in Indoor Scenes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2144–2151. IEEE, dec 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.266. URL http://ieeexplore.ieee.org/document/6751377/.

A. Gupta, A. A. Efros, and M. Hebert. Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 482–496, 2010a. doi: 10.1007/978-3-642-15561-1_35. URL http://link.springer.com/10.1007/978-3-642-15561-1{_}35.

A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating Spatial Layout of Rooms using Volumetric Reasoning about Objects and Surfaces. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1288–1296. Curran Associates, Inc., 2010b. URL `http://papers.nips.cc/paper/4120-estimating-spatial-layout-of-rooms-using-volumetric-reasoning-about\protect\discretionary{\char\hyphenchar\font}{}{}objects-and-surfaces`.

S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 345–360. Springer, Cham, 2014. doi: 10.1007/978-3-319-10584-0_23. URL `http://link.springer.com/10.1007/978-3-319-10584-0{_}23`.

S. Gupta, P. Arbelaez, R. Girshick, and J. Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4731–4740, Boston, MA, USA, jun 2015. IEEE. ISBN 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7299105. URL `http://ieeexplore.ieee.org/document/7299105/`.

D. Hähnel, W. Burgard, and S. Thurn. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 44(1):15–27, 2003. ISSN 09218890. doi: 10.1016/S0921-8890(03)00007-1.

M. Halber and T. Funkhouser. Fine-to-Coarse Global Registration of RGB-D Scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6660–6669. IEEE, jul 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.705. URL `http://ieeexplore.ieee.org/document/8100188/`.

C. Harris and J. Pike. 3D positional integration from image sequences. *Image and Vision Computing*, 6(2):87–90, may 1988. ISSN 02628856. doi: 10.1016/0262-8856(88)90003-0. URL `http://linkinghub.elsevier.com/retrieve/pii/0262885688900030`.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, mar 2004. ISBN 9780521540513. doi: 10.1017/CBO9780511811685. URL https://www.cambridge.org/core/product/identifier/9780511811685/type/book.

K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. IEEE, oct 2017. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.322. URL http://ieeexplore.ieee.org/document/8237584/.

V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1849–1856, Kyoto, Japan, sep 2009. IEEE. ISBN 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459411. URL http://ieeexplore.ieee.org/document/5459411/.

V. Hedau, D. Hoiem, and D. Forsyth. Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 224–237, Crete, Greece, 2010. Springer, Berlin, Heidelberg. doi: 10.1007/978-3-642-15567-3_17. URL http://link.springer.com/10.1007/978-3-642-15567-3{_}17.

V. Hedau, D. Hoiem, and D. Forsyth. Recovering free space of indoor scenes from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2807–2814. IEEE, jun 2012. ISBN 978-1-4673-1228-8. doi: 10.1109/CVPR.2012.6248005. URL http://ieeexplore.ieee.org/document/6248005/.

P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, 31(5):647–663, 2012. ISSN 02783649. doi: 10.1177/0278364911434148.

D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *Proceedings of*

the IEEE International Conference on Computer Vision (ICCV), pages 654–661, Beijing, China, 2005a. IEEE. ISBN 0-7695-2334-X. doi: 10.1109/ICCV.2005.107. URL http://ieeexplore.ieee.org/document/1541316/.

D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*, 24(3):577, jul 2005b. ISSN 07300301. doi: 10.1145/1073204.1073232. URL http://portal.acm.org/citation.cfm?doid=1073204.1073232.

D. Hoiem, A. A. Efros, and M. Hebert. Recovering Surface Layout from an Image. *International Journal of Computer Vision*, 75(1):151–172, jul 2007. ISSN 0920-5691. doi: 10.1007/s11263-006-0031-y. URL http://link.springer.com/10.1007/s11263-006-0031-y.

D. Hoiem, A. A. Efros, and M. Hebert. Putting Objects in Perspective. *International Journal of Computer Vision*, 80(1):3–15, oct 2008. ISSN 0920-5691. doi: 10.1007/s11263-008-0137-5. URL http://link.springer.com/10.1007/s11263-008-0137-5.

M. Hosseinzadeh, Y. Latif, T. Pham, N. Suenderhauf, and I. Reid. Structure Aware SLAM using Quadrics and Planes. *CoRR*, abs/1804.0:1–22, apr 2018. URL http://arxiv.org/abs/1804.09111.

M. Hosseinzadeh, K. Li, Y. Latif, and I. Reid. Real-Time Monocular Object-Model Aware Sparse SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 7123–7129. IEEE, may 2019. ISBN 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8793728. URL http://arxiv.org/abs/1809.09149https://ieeexplore.ieee.org/document/8793728/.

M. Hsiao, E. Westman, G. Zhang, and M. Kaess. Keyframe-based dense planar SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5110–5117, Singapore, Singapore, 2017. IEEE. ISBN 9781509046331. doi: 10.1109/ICRA.2017.7989597.

M. Hsiao, E. Westman, and M. Kaess. Dense Planar-Inertial SLAM with Structural Constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6521–6528, Brisbane, QLD, Australia, 2018. IEEE. ISBN 9781538630808. doi: 10.1109/ICRA.2018.8461094.

S. Ikehata, H. Yang, and Y. Furukawa. Structured Indoor Modeling. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1323–1331. IEEE, dec 2015. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.156. URL `http://ieeexplore.ieee.org/document/7410513/`.

S. Ikehata, I. Boyadzhiev, Q. Shan, and Y. Furukawa. Panoramic Structure from Motion via Geometric Relationship Detection. *CoRR*, abs/1612.0, dec 2016. URL `http://arxiv.org/abs/1612.01256`.

H. Izadinia, Q. Shan, and S. M. Seitz. IM2CAD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2422–2431. IEEE, jul 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.260. URL `http://ieeexplore.ieee.org/document/8099743/`.

K. Joo, T.-H. Oh, J. Kim, and I. S. Kweon. Robust and Globally Optimal Manhattan Frame Estimation in Near Real Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):682–696, mar 2019. ISSN 0162-8828. doi: 10.1109/TPAMI.2018.2799944. URL `https://ieeexplore.ieee.org/document/8275042/`.

J. Jung, C. Stachniss, S. Ju, and J. Heo. Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM. *Advanced Engineering Informatics*, 38:811–825, oct 2018. ISSN 14740346. doi: 10.1016/j.aei.2018.10.007. URL `https://doi.org/10.1016/j.aei.2018.10.007https://linkinghub.elsevier.com/retrieve/pii/S1474034618300600`.

M. Kaess. Simultaneous localization and mapping with infinite planes. In *Proceedings of the*

IEEE International Conference on Robotics and Automation (ICRA), pages 4605–4611, Seattle, WA, USA, may 2015. IEEE. ISBN 978-1-4799-6923-4. doi: 10.1109/ICRA.2015.7139837. URL http://ieeexplore.ieee.org/document/7139837/.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, 2008. ISSN 15523098. doi: 10.1109/TRO.2008.2006706.

K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager. Uncertainty-Aware Occupancy Map Prediction Using Generative Networks for Robot Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5453–5459. IEEE, may 2019. ISBN 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8793500. URL https://ieeexplore.ieee.org/document/8793500/.

M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-Time 3D Reconstruction in Dynamic Scenes Using Point-Based Fusion. In *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, pages 1–8. IEEE, jun 2013. ISBN 978-0-7695-5067-1. doi: 10.1109/3DV.2013.9. URL http://ieeexplore.ieee.org/document/6599048/.

C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2100–2106, Tokyo, Japan, 2013. IEEE. ISBN 9781467363587. doi: 10.1109/IROS.2013.6696650.

B.-s. Kim, P. Kohli, and S. Savarese. 3D Scene Understanding by Voxel-CRF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1425–1432. IEEE, dec 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.180. URL http://ieeexplore.ieee.org/document/6751287/.

P. Kim, B. Coltin, and H. J. Kim. Linear RGB-D SLAM for Planar Environments. In

Proceedings of the European Conference on Computer Vision (ECCV), pages 350–366, Munich, Germany, 2018a. Springer, Cham. doi: 10.1007/978-3-030-01225-0_21. URL http://link.springer.com/10.1007/978-3-030-01225-0{_}21.

P. Kim, B. Coltin, and H. J. Kim. Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 7247–7253, 2018b. ISBN 9781538630808.

Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM Transactions on Graphics*, 31(6):1, nov 2012. ISSN 07300301. doi: 10.1145/2366145.2366157. URL http://dl.acm.org/citation.cfm?doid=2366145.2366157.

G. Klein and D. Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–10. IEEE, nov 2007. ISBN 978-1-4244-1749-0. doi: 10.1109/ISMAR.2007.4538852. URL http://ieeexplore.ieee.org/document/4538852/.

J. Kleinberg and E. Tardos. *Algorithm Design*. Pearson/Addison-Wesley, Boston, MA, USA, 2006.

R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613. IEEE, may 2011. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5979949. URL http://ieeexplore.ieee.org/document/5979949/.

A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3D reconstruction from monocular video. In *Proceedings of the European Conference on*

Computer Vision (ECCV), pages 703–718, 2014. ISBN 9783319105987. doi: 10.1007/978-3-319-10599-4_45.

J. Lahoud and B. Ghanem. 2D-Driven 3D Object Detection in RGB-D Images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4632–4640. IEEE, oct 2017. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.495. URL http://ieeexplore.ieee.org/document/8237757/.

Y. Latif, G. Huang, J. Leonard, and J. Neira. Sparse optimization for robust and efficient loop closing. *Robotics and Autonomous Systems*, 93:13–26, jul 2017. ISSN 09218890. doi: 10.1016/j.robot.2017.03.016. URL https://linkinghub.elsevier.com/retrieve/pii/S0921889015302281.

C.-Y. Lee, V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-End Room Layout Estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4875–4884. IEEE, oct 2017a. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.521. URL http://ieeexplore.ieee.org/document/8237783/.

D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2136–2143. IEEE, jun 2009. ISBN 978-1-4244-3992-8. doi: 10.1109/CVPRW.2009.5206872. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206872.

J.-K. Lee, J. Yea, M.-G. Park, and K.-J. Yoon. Joint Layout Estimation and Global Multi-view Registration for Indoor Reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 162–171, Venice, Italy, oct 2017b. IEEE. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.27. URL http://ieeexplore.ieee.org/document/8237289/.

T.-k. Lee, S. Lim, S. Lee, S. An, and S.-y. Oh. Indoor mapping using planes extracted from noisy RGB-D sensors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1727–1733. IEEE, oct 2012. ISBN 978-1-4673-1736-8. doi: 10.1109/IROS.2012.6385909. URL `http://ieeexplore.ieee.org/document/6385909/`.

H. Li, J. Yao, X. Lu, and J. Wu. Combining Points and Lines for Camera Pose Estimation and Optimization in Monocular Visual Odometry. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1289–1296, Vancouver, BC, Canada, 2017. IEEE. ISBN 9781538626825. doi: 10.1109/IROS.2017.8202304.

H. Li, J. Yao, J.-c. Bazin, X. Lu, Y. Xing, and K. Liu. A Monocular SLAM System Leveraging Structural Regularity in Manhattan World. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2518–2525, 2018. ISBN 9781538630808.

H. Li, Y. Xing, J. Zhao, J.-c. Bazin, Z. Liu, and Y.-h. Liu. Leveraging Structural Regularity of Atlanta World for Monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2412–2418. IEEE, may 2019. ISBN 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8793716. URL `https://ieeexplore.ieee.org/document/8793716/`.

C. Lin, C. Li, Y. Furukawa, and W. Wang. Floorplan Priors for Joint Camera Pose and Room Layout Estimation. *CoRR*, abs/1812.0, dec 2018. URL `http://arxiv.org/abs/1812.06677`.

D. Lin, S. Fidler, and R. Urtasun. Holistic Scene Understanding for 3D Object Detection with RGBD Cameras. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1417–1424. IEEE, dec 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.179. URL `http://ieeexplore.ieee.org/document/6751286/`.

C. Liu, J. Wu, and Y. Furukawa. FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 203–219. Springer, Cham, mar 2018a. doi: 10.1007/978-3-030-01231-1_13. URL http://link.springer.com/10.1007/978-3-030-01231-1{_}13.

X. Liu, Y. Zhao, and S. C. Zhu. Single-View 3D Scene Reconstruction and Parsing by Attribute Grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):710–725, 2018b. ISSN 01628828. doi: 10.1109/TPAMI.2017.2689007.

Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to Learn 3d Environment Models with Mobile Robots. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 329–336, 2001. URL https://dl.acm.org/citation.cfm?id=655822.

H. Longuet-higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981. doi: 10.1038/293133a0.

D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL http://link.springer.com/10.1023/B:VISI.0000029664.99615.94.

S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. Visual Place Recognition: A Survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. ISSN 15523098. doi: 10.1109/TRO.2015.2496823.

F. Lu and E. Milios. Globally Consistent Scan Matching For Environment Mapping. *Autonomous Robots*, 4(4):333–349, 1997. doi: 10.1023/A:1008854305733.

Y. Lu and D. Song. Visual Navigation Using Heterogeneous Landmarks and Unsupervised Geometric Constraints. *IEEE Transactions on Robotics*, 31(3):736–749, jun 2015.

ISSN 1552-3098. doi: 10.1109/TRO.2015.2424032. URL http://ieeexplore.ieee.org/document/7103351/.

Q. Luo, H. Ma, Y. Wang, L. Tang, and R. Xiong. 3D-SSD: Learning Hierarchical Features from RGB-D Images for Amodal 3D Object Detection. *CoRR*, abs/1711.0, nov 2017. URL http://arxiv.org/abs/1711.00238.

M. Luperto, V. Arcerito, and F. Amigoni. Predicting the Layout of Partially Observed Rooms from Grid Maps. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6898–6904. IEEE, may 2019. ISBN 978-1-5386-6027-0. doi: 10.1109/ICRA.2019.8793489. URL https://ieeexplore.ieee.org/document/8793489/.

L. Ma, C. Kerl, J. Stuckler, and D. Cremers. CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1285–1291. IEEE, may 2016. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487260. URL http://ieeexplore.ieee.org/document/7487260/.

D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, may 1979. ISSN 2053-9193. doi: 10.1098/rspb.1979.0029. URL http://www.royalsocietypublishing.org/doi/10.1098/rspb.1979.0029.

O. Mattausch, D. Panozzo, C. Mura, O. Sorkine-Hornung, and R. Pajarola. Object detection and classification from large-scale cluttered indoor scans. *Computer Graphics Forum*, 33 (2):11–21, may 2014. ISSN 01677055. doi: 10.1111/cgf.12286. URL http://doi.wiley.com/10.1111/cgf.12286.

J. McCormac, A. Handa, A. Davison, and S. Leutenegger. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In *Proceedings of the IEEE Inter-*

national Conference on Robotics and Automation (ICRA), pages 4628–4635, Singapore, Singapore, may 2017. IEEE. ISBN 978-1-5090-4633-1. doi: 10.1109/ICRA.2017.7989538. URL http://ieeexplore.ieee.org/document/7989538/.

J. Mccormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric Object-Level SLAM. In *IEEE International Conference on 3D Vision (3DV)*, pages 32–41. IEEE, sep 2018. ISBN 978-1-5386-8425-2. doi: 10.1109/3DV.2018.00015. URL https://ieeexplore.ieee.org/document/8490953/.

K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico, K. Karydis, N. Atanasov, G. Loianno, D. Scaramuzza, K. Daniilidis, C. J. Taylor, and V. Kumar. Fast, autonomous flight in GPS-denied and cluttered environments. *Journal of Field Robotics*, 35(1):101–120, 2018. ISSN 15564967. doi: 10.1002/rob.21774.

A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra. RAPter: Rebuilding Man-made Scenes with Regular Arrangements of Planes. *ACM Transactions on Graphics*, 34(4): 103:1–103:12, jul 2015. ISSN 07300301. doi: 10.1145/2766995. URL http://dl.acm.org/citation.cfm?doid=2809654.2766995.

R. Mur-Artal, J. M. Montiel, and J. D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. ISSN 15523098. doi: 10.1109/TRO.2015.2463671.

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(1):20–32, nov 2014. ISSN 00978493. doi: 10.1016/j.cag.2014.07.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0097849314000661.

C. Mura, O. Mattausch, and R. Pajarola. Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements. *Computer Graphics Forum*, 35(7):179–188,

oct 2016. ISSN 01677055. doi: 10.1111/cgf.13015. URL http://doi.wiley.com/10.1111/cgf.13015.

F. Nardi, B. Della Corte, and G. Grisetti. Unified representation and registration of heterogeneous sets of geometric primitives. *IEEE Robotics and Automation Letters*, 4(2):625–632, apr 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2891989. URL https://ieeexplore.ieee.org/document/8607088/.

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Basel, Switzerland, 2011. IEEE. ISBN 9781457721830. doi: 10.1109/ISMAR.2011.6092378.

V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis. Orthogonal SLAM: A step toward lightweight indoor autonomous navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5007–5012, Beijing, China, 2006. IEEE. ISBN 142440259X. doi: 10.1109/IROS.2006.282527.

V. Nguyen, A. Harati, and R. Siegwart. A lightweight SLAM algorithm using Orthogonal planes for indoor mobile robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 658–663, San Diego, CA, USA, 2007. IEEE. ISBN 1424409128. doi: 10.1109/IROS.2007.4399512.

L. Nicholson, M. Milford, and N. Sunderhauf. QuadricSLAM: Dual Quadrics From Object Detections as Landmarks in Object-Oriented SLAM. *IEEE Robotics and Automation Letters*, 4(1):1–8, jan 2019. ISSN 2377-3766. doi: 10.1109/LRA.2018.2866205. URL https://ieeexplore.ieee.org/document/8440105/.

M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics*, 32(6):1–11, nov 2013. ISSN

07300301. doi: 10.1145/2508363.2508374. URL http://dl.acm.org/citation.cfm?doid=2508363.2508374.

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 652–659. IEEE, 2004. ISBN 0-7695-2158-4. doi: 10.1109/CVPR.2004.1315094. URL http://ieeexplore.ieee.org/document/1315094/.

A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *Robotics and Autonomous Systems*, 56(11):915–926, 2008. ISSN 09218890. doi: 10.1016/j.robot.2008.08.001. URL 10.1016/j.robot.2008.08.001.

S. T. O'Callaghan and F. T. Ramos. Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62, jan 2012. ISSN 0278-3649. doi: 10.1177/0278364911421039. URL http://journals.sagepub.com/doi/10.1177/0278364911421039.

S. Ochmann, R. Vock, R. Wessel, and R. Klein. Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics*, 54:94–103, feb 2016. ISSN 00978493. doi: 10.1016/j.cag.2015.07.008. URL https://linkinghub.elsevier.com/retrieve/pii/S0097849315001119.

S. Oesau, F. Lafarge, and P. Alliez. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90:68–82, apr 2014. ISSN 09242716. doi: 10.1016/j.isprsjprs.2014.02.004. URL http://dx.doi.org/10.1016/j.isprsjprs.2014.02.004https://linkinghub.elsevier.com/retrieve/pii/S0924271614000410.

K. Ok, K. Liu, K. Frey, J. P. How, and N. Roy. Robust Object-based SLAM for High-speed Autonomous Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 669–675. IEEE, may 2019. ISBN 978-1-

5386-6027-0. doi: 10.1109/ICRA.2019.8794344. URL https://ieeexplore.ieee.org/document/8794344/.

B. Okorn, X. Xiong, B. Akinci, and D. Huber. Toward Automated Modeling of Floor Plans. In *Symposium on 3D Data Processing, Visualization and Transmission*, 2010.

B. Pfrommer and K. Daniilidis. The GRASP MultiCam Data Set, 2019. URL https://daniilidis-group.github.io/grasp{_}multicam/.

A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer. PL-SLAM: Real-time monocular visual SLAM with points and lines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4503–4508, Singapore, Singapore, 2017. IEEE. ISBN 9781509046331. doi: 10.1109/ICRA.2017.7989522.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85. IEEE, jul 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.16. URL http://ieeexplore.ieee.org/document/8099499/.

C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, Salt Lake City, UT, USA, jun 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00102. URL https://ieeexplore.ieee.org/document/8578200/.

M. Quigley, K. Mohta, S. S. Shivakumar, M. Watterson, Y. Mulgaonkar, M. Arguedas, K. Sun, S. Liu, B. Pfrommer, V. Kumar, and C. J. Taylor. The Open Vision Computer: An Integrated Sensing and Compute System for Mobile Robots. *CoRR*, abs/1809.0, sep 2018. URL http://arxiv.org/abs/1809.07674.

F. Ramos and L. Ott. Hilbert maps: Scalable continuous occupancy mapping with stochastic

gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, dec 2016. ISSN 0278-3649. doi: 10.1177/0278364916684382. URL http://journals.sagepub.com/doi/10.1177/0278364916684382.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE, jun 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.91. URL http://ieeexplore.ieee.org/document/7780460/.

Z. Ren and E. B. Sudderth. Three-Dimensional Object Detection and Layout Prediction Using Clouds of Oriented Gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1525–1533. IEEE, jun 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.169. URL http://ieeexplore.ieee.org/document/7780538/.

C. Richter, W. Vega-Brown, and N. Roy. Bayesian Learning for Safe High-Speed Navigation in Unknown Environments. In *Robotics Research*, pages 325–341. Springer, Cham, 2018. ISBN 978-3-319-60915-7. doi: 10.1007/978-3-319-60916-4_19. URL http://link.springer.com/10.1007/978-3-319-60916-4{_}19.

R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. SLAM++: Simultaneous Localisation and Mapping at the Level of Objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1352–1359. IEEE, jun 2013. ISBN 978-0-7695-4989-7. doi: 10.1109/CVPR.2013.178. URL http://ieeexplore.ieee.org/document/6619022/.

R. F. Salas-Moreno, B. Glocker, P. H. J. Kelly, and A. J. Davison. Dense planar SLAM. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 367–368. IEEE, 2014. ISBN 978-1-4799-6184-9. doi: 10.1109/ISMAR.2014.6948492. URL http://ieeexplore.ieee.org/document/6948492/.

V. Sanchez and A. Zakhor. Planar 3D modeling of building interiors from point cloud data. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1777–1780. IEEE, sep 2012. ISBN 978-1-4673-2533-2. doi: 10.1109/ICIP.2012.6467225. URL `http://ieeexplore.ieee.org/document/6467225/`.

A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, may 2009. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.132. URL `http://ieeexplore.ieee.org/document/4531745/`.

G. Schindler and F. Dellaert. Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–209. IEEE, 2004. ISBN 0-7695-2158-4. doi: 10.1109/CVPR.2004.1315033. URL `http://ieeexplore.ieee.org/document/1315033/`.

A. G. Schwing and R. Urtasun. Efficient exact inference for 3D indoor scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–313, Florence, Italy, 2012. Springer, Berlin, Heidelberg. ISBN 9783642337826. doi: 10.1007/978-3-642-33783-3_22.

A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3D indoor scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2815–2822, Providence, RI, USA, 2012. IEEE. ISBN 9781467312264. doi: 10.1109/CVPR.2012.6248006.

A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 353–360, 2013. ISBN 9781479928392. doi: 10.1109/ICCV.2013.51.

T. Shakunaga. 3-D corridor modeling from a single view under natural lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):293–298, 1992. ISSN 01628828. doi: 10.1109/34.121796. URL http://ieeexplore.ieee.org/document/121796/.

A. Shariati, B. Pfrommer, and C. J. Taylor. Predictive and Semantic Layout Estimation for Robotic Applications in Manhattan Worlds. *CoRR*, abs/1811.0, nov 2018. URL http://arxiv.org/abs/1811.07442.

A. Shariati, B. Pfrommer, and C. J. Taylor. Simultaneous Localization and Layout Model Selection in Manhattan Worlds. *IEEE Robotics and Automation Letters*, 4(2):950–957, apr 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2893417. URL https://ieeexplore.ieee.org/document/8613887/.

M. Simon, S. Milz, K. Amende, and H.-M. Gross. Complex-YOLO: An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 197–209. Springer, Cham, Cham, 2018. ISBN 978-3-030-11009-3. doi: 10.1007/978-3-030-11009-3_11. URL http://link.springer.com/10.1007/978-3-030-11009-3{_}11.

S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1881–1888. IEEE, sep 2009. ISBN 978-1-4244-4420-5. doi: 10.1109/ICCV.2009.5459417. URL http://ieeexplore.ieee.org/document/5459417/.

P. Smith, I. Reid, and A. J. Davison. Real-Time Monocular SLAM with Straight Lines. In *Procedings of the British Machine Vision Conference*, pages 3.1–3.10. British Machine Vision Association, 2006. ISBN 1-901725-32-4. doi: 10.5244/C.20.3. URL http://www.bmva.org/bmvc/2006/papers/162.html.

R. C. Smith and P. Cheeseman. On the representation and estimation of spatial un-

certainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986. doi: 10.1177/027836498600500404. URL https://www.frc.ri.cmu.edu/{~}hpm/project.archive/reference.file/Smith{&}Cheeseman.pdf.

N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring Photo Collections in 3D. *ACM Transactions on Graphics*, 25(3):835–846, 2006. ISSN 07300301. doi: 10.1145/1141911.1141964. URL http://doi.acm.org/10.1145/1141911.1141964{%}5Cnhttp://portal.acm.org/citation.cfm?doid=1141911.1141964.

S. Song and J. Xiao. Sliding Shapes for 3D Object Detection in Depth Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 634–651. Springer, Cham, 2014. ISBN 978-1-4673-8851-1. doi: 10.1007/978-3-319-10599-4_41. URL http://link.springer.com/10.1007/978-3-319-10599-4{_}41.

S. Song and J. Xiao. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 808–816. IEEE, jun 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.94. URL http://ieeexplore.ieee.org/document/7780463/.

S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic Scene Completion from a Single Depth Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 190–198, Honolulu, HI, USA, jul 2017. IEEE. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.28. URL http://ieeexplore.ieee.org/document/8099511/.

J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher. Real-time manhattan world rotation estimation in 3D. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1913–1920. IEEE, sep 2015a. ISBN 978-1-4799-9994-1. doi: 10.1109/IROS.2015.7353628. URL http://ieeexplore.ieee.org/document/7353628/.

J. Straub, J. Chang, O. Freifeld, and J. Fisher III. A Dirichlet Process Mixture Model for Spherical Data. *Journal of Machine Learning Research*, 38:930–938, 2015b. URL http://proceedings.mlr.press/v38/straub15.html.

J. Straub, R. Cabezas, J. Leonard, and J. W. Fisher. Direction-Aware Semi-Dense SLAM. *CoRR*, abs/1709.0, sep 2017. doi: 10.1007/978-1-4419-0284-9_7. URL http://arxiv.org/abs/1709.05774.

K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. *IEEE Robotics and Automation Letters*, 3(2):965–972, apr 2018. ISSN 2377-3766. doi: 10.1109/LRA.2018.2793349. URL http://ieeexplore.ieee.org/document/8258858/.

N. Sunderhauf and P. Protzel. Switchable constraints for robust pose graph SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1879–1884. IEEE, oct 2012. ISBN 978-1-4673-1736-8. doi: 10.1109/IROS.2012.6385590. URL http://ieeexplore.ieee.org/document/6385590/.

N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps with object-oriented semantic mapping. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5079–5085. IEEE, 2017. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8206392. URL http://ieeexplore.ieee.org/document/8206392/.

Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. Point-plane SLAM for hand-held 3D sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 5182–5189. IEEE, may 2013. ISBN 978-1-4673-5643-5. doi: 10.1109/ICRA.2013.6631318. URL http://ieeexplore.ieee.org/document/6631318/.

P. Tang, D. Huber, B. Akinci, R. Lipman, and A. Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related

techniques. *Automation in Construction*, 19(7):829–843, nov 2010. ISSN 09265805. doi: 10.1016/j.autcon.2010.06.007. URL http://dx.doi.org/10.1016/j.autcon.2010.06.007https://linkinghub.elsevier.com/retrieve/pii/S0926580510000907.

C. Taylor and A. Cowley. Parsing Indoor Scenes Using RGB-D Imagery. In *Robotics: Science and Systems*, pages 401–408. Robotics: Science and Systems Foundation, jul 2012. ISBN 9780262519687. doi: 10.15607/RSS.2012.VIII.051. URL http://www.roboticsproceedings.org/rss08/p51.pdf.

B. Tekin, S. N. Sinha, and P. Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 292–301. IEEE, jun 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00038. URL https://ieeexplore.ieee.org/document/8578136/.

S. Thrun, C. Martin, Y. Liu, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A Real-Time Expectation Maximization Algorithm for Acquiring Multi-Planar Maps of Indoor Environments with Mobile Robots. *IEEE Transactions on Robotics and Automation*, 20(3):433 – 443, 2004. doi: 10.1109/TRA.2004.825520. URL http://www.cs.cmu.edu/{~}deepay/mywww/papers/ieeeTORA03-realtime.pdf.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT Press, Cambridge, MA, USA, 2005. ISBN 0262201623.

R. Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society*, 73(3):273–282, jun 2011. ISSN 13697412. doi: 10.1111/j.1467-9868.2011.00771.x. URL http://doi.wiley.com/10.1111/j.1467-9868.2011.00771.x.

A. J. B. Trevor, J. G. Rogers, and H. I. Christensen. Planar surface SLAM with 3D and 2D sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3041–3048, Saint Paul, MN, USA, may 2012. IEEE. ISBN

978-1-4673-1405-3. doi: 10.1109/ICRA.2012.6225287. URL http://ieeexplore.ieee.org/document/6225287/.

G. Tsai, Changhai Xu, Jingen Liu, and B. Kuipers. Real-time indoor scene understanding using Bayesian filtering with motion cues. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 121–128. IEEE, nov 2011. ISBN 978-1-4577-1102-2. doi: 10.1109/ICCV.2011.6126233. URL http://ieeexplore.ieee.org/document/6126233/.

S. Tulsiani, S. Gupta, D. Fouhey, A. A. Efrosefros, and J. Malik. Factoring Shape, Pose, and Layout from the 2D Image of a 3D Scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 302–310, Salt Lake City, UT, jun 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00039. URL https://ieeexplore.ieee.org/document/8578137/.

E. Turner and A. Zakhor. Floor Plan Generation and Room Labeling of Indoor Environments from Laser Range Data. In *Proceedings of the IEEE International Conference on Computer Graphics Theory and Applications*, pages 22–33, Lisbon, Portugal, 2014. IEEE. ISBN 978-989-758-002-4. doi: 10.5220/0004680300220033. URL https://ieeexplore.ieee.org/document/7296026.

C. A. Vanegas, D. G. Aliaga, and B. Benes. Automatic Extraction of Manhattan-World Building Masses from 3D Laser Range Scans. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1627–1637, oct 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.30. URL http://ieeexplore.ieee.org/document/6143940/.

Velodyne. HDL-64E, 2020. URL https://velodynelidar.com/hdl-64e.html.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*

(CVPR), pages 511–518. IEEE Comput. Soc, 2001. ISBN 0-7695-1272-0. doi: 10.1109/CVPR.2001.990517. URL http://ieeexplore.ieee.org/document/990517/.

W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2569–2578. IEEE, 2018. ISBN 9781538664209. doi: 10.1109/CVPR.2018.00272.

T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, apr 2015a. ISSN 0278-3649. doi: 10.1177/0278364914551008. URL http://journals.sagepub.com/doi/10.1177/0278364914551008.

T. Whelan, S. Leutenegger, R. F. Salas-moreno, B. Glocker, and A. J. Davison. ElasticFusion : Dense SLAM Without A Pose Graph. In *Robotics: Science and Systems*, Rome, Italy, 2015b.

J. Xiao and Y. Furukawa. Reconstructing the World's Museums. *International Journal of Computer Vision*, 110(3):243–258, dec 2014. ISSN 0920-5691. doi: 10.1007/s11263-014-0711-y. URL http://link.springer.com/10.1007/s11263-014-0711-y.

X. Xiong, A. Adan, B. Akinci, and D. Huber. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, 31:325–337, 2013. ISSN 09265805. doi: 10.1016/j.autcon.2012.10.006. URL http://dx.doi.org/10.1016/j.autcon.2012.10.006.

D. Xu, D. Anguelov, and A. Jain. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253. IEEE, jun 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00033. URL https://ieeexplore.ieee.org/document/8578131/.

S. Yang and S. Scherer. Monocular Object and Plane SLAM in Structured Environments. *IEEE Robotics and Automation Letters*, 4(4):3145–3152, oct 2019. ISSN 2377-3766. doi: 10.1109/LRA.2019.2924848. URL https://ieeexplore.ieee.org/document/8744612/.

S. Yang, D. Maturana, and S. Scherer. Real-time 3D scene layout from a single image using Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2183–2189, Stockholm, Sweden, may 2016a. IEEE. ISBN 978-1-4673-8026-3. doi: 10.1109/ICRA.2016.7487368. URL http://ieeexplore.ieee.org/document/7487368/.

S. Yang, Y. Song, M. Kaess, and S. Scherer. Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1222–1229, Daejeon, South Korea, oct 2016b. IEEE. ISBN 978-1-5090-3762-9. doi: 10.1109/IROS.2016.7759204. URL http://ieeexplore.ieee.org/document/7759204/.

J. Zhang and S. Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Robotics: Science and Systems*, page 9, Berkeley, CA, USA, 2014. ISBN 9780992374709. doi: 10.15607/RSS.2014.X.007. URL http://www.roboticsproceedings.org/rss10/p07.pdf.

J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3D Layout of Indoor Scenes and Its Clutter from Depth Sensors. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1273–1280. IEEE, dec 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.161. URL http://ieeexplore.ieee.org/document/6751268/.

Y. Zhang, S. Song, P. Tan, and J. Xiao. PanoContext: A Whole-Room 3D Context Model for Panoramic Scene Understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 668–686. Springer, Cham, 2014. doi: 10.1007/978-3-319-10599-4_43. URL http://link.springer.com/10.1007/978-3-319-10599-4{_}43.

Y. Zhang, W. Xu, Y. Tong, and K. Zhou. Online Structure Analysis for Real-Time Indoor Scene Reconstruction. *ACM Transactions on Graphics*, 34(5):1–13, 2015. ISSN 07300301. doi: 10.1145/2768821.

S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison. SceneCode: Monocular Dense Semantic Reconstruction using Learned Encoded Scene Representations. *CoRR*, abs/1903.0, mar 2019. URL `http://arxiv.org/abs/1903.06482`.

H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu. StructSLAM: Visual SLAM with building structure lines. *IEEE Transactions on Vehicular Technology*, 64(4):1364–1375, 2015. ISSN 00189545. doi: 10.1109/TVT.2015.2388780.

Y. Zhou and O. Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499. IEEE, jun 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00472. URL `https://ieeexplore.ieee.org/document/8578570/`.

C. Zou, A. Colburn, Q. Shan, and D. Hoiem. LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2051–2059. IEEE, jun 2018. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00219. URL `https://ieeexplore.ieee.org/document/8578317/`.

C. Zou, R. Guo, Z. Li, and D. Hoiem. Complete 3D Scene Parsing from an RGBD Image. *International Journal of Computer Vision*, 127(2):143–162, feb 2019. ISSN 0920-5691. doi: 10.1007/s11263-018-1133-z. URL `http://link.springer.com/10.1007/s11263-018-1133-z`.