UNICA

UNIVERSITÀ
DEGLI STUDI
DI CAGLIARI

Ph.D. DEGREE IN
MATHEMATICS AND COMPUTER SCIENCE

Cycle XXXVI

TITLE OF THE Ph.D. THESIS
Data-driven depth and 3D architectural layout estimation of an interior environment from monocular panoramic input

Scientific Disciplinary Sector(s)
INF/01 INFORMATICA

Ph.D. Student : Eva Almansa
Supervisor : Prof. Riccardo Scateni (UniCa)
Co-Supervisor : Dr. Enrico Gobbetti (CRS4)

Final Exam Academic Year (2022-2023)
Thesis Defence: February 2024 Session

Abstract

In recent years, there has been significant research interest in the automatic 3D reconstruction and modeling of indoor scenes from capture data, giving rise to an emerging sub-field within 3D reconstruction. The primary goal is to convert an input source, which represents a sample of a real-world indoor environment, into a model that may encompass geometric, structural, and/or visual abstractions.

Within the scope of this thesis, the focus has been on the extraction of geometric information from a single panoramic image, either by using only visual data or aided by very sparse registered depth information. This particular setup has attracted a lot of interest in recent years, since 360° images offer rapid and comprehensive single-image coverage and they are supported by a wide range of professional and consumer capture devices, which makes the data acquisition process both efficient and cost-effective. On the other hand, despite the 360° coverage, inferring a comprehensive model from mostly visual input in preseence of noise, missing data, and clutter remains very challenging. Thus, my research has focused on finding clever ways to exploit prior information, in the form of architectural priors and data-driven priors derived from large sets of examples, to design end-to-end deep learning solutions to solve well-defined fundamental tasks in the structured reconstruction pipeline. The tasks on which I have focused are, in particular, depth estimation from a single 360° image, depth completion from a single 360° image enriched with sparse depth measurements, and 3D architectural layout estimation from a single 360° image. While the first two problems produce pixel-wise input in terms of a dense depth map, the latter consists in the reconstruction, from the image of the furnished room, of a simplified model of the 3D shape of the bounding permanent surfaces of a room.

As a first contribution towards reconstructing indoor information from purely visual data, I introduced a novel deep neural network to estimate a depth map from a single monocular indoor panorama. The network directly works on the equirectangular projection, exploiting the properties of indoor 360-degree images. Starting

from the fact that gravity plays an important role in the design and construction of man-made indoor scenes, the network compactly encodes the scene into vertical spherical slices, and exploits long- and short-term relationships among slices to recover an equirectangular depth map directly from an equirectangular RGB image.

My second contribution expands this approach to the common situation in which we receive as input a single equirectangular image registered with a sparse depth map, as provided by a variety of common capture setups. In this approach, depth is inferred by an efficient and lightweight single-branch network, which employs a dynamic gating system to process together dense visual data and sparse geometric data. Furthermore, a new augmentation strategy makes the model robust to different types of sparsity, including those generated by various structured light sensors and LiDAR setups.

While the two preceding contribution focus on the estimation of per-pixel geometric information, my third contribution has tackled the problem of recovering the 3D shape of the bounding permanent surfaces of a room from a single panoramic image. The method also exploits gravity-alighted features, but within a significantly different setup, dictacted by the fact that not only we need to separate walls, ceilings, and floor, but we need to recover the plausible shape of invisible areas. The proposed approach, differently from prior state-of-the-art methods, fully addresses the problem in 3D, significantly expanding the reconstruction space. In particular, a graph convolutional network directly infers the room structure as a 3D mesh by progressively deforming a graph-encoded tessellated sphere mapped to the spherical panorama, leveraging perceptual features extracted from the input image. Gravity-aligned features are actively incorporated in the graph in a projection layer that exploits the recent concept of multi head self-attention, and specialized losses guide towards plausible solutions even in presence of massive clutter and occlusions.

The benchmarks on publicly available data show that all three methods are on par or better with respect to the state-of-the-art.

Keywords: Visual Computing, Computer Vision, Computer Graphics, Spherical Capture, Omnidirectional Capture, Panoramic Capture, Equirectangular Projection, 3D Reconstruction, Indoor Environment, Monocular Vision, Depth Estimation, Depth Completion, 3D Layout Estimation.

Acknowledgments

Words cannot express my gratitude to my supervisor **Dr. Enrico Gobbetti**, Director of Visual and Data-intensive Computing at CRS4, for his guidance, for his invaluable patience, for sharing a small part of his wisdom with me, and above all for having given me this opportunity to work in a research center at CRS4 (with very competent colleagues) in the framework of a research and training network involving many other entities scattered around Europe, study my long-pursued PhD, and, also, live in a wonderful island, Sardinia. I am deeply indebted to my co-supervisor at CRS4 **Dr. Giovanni Pintore** for his wonderful guidance, for his total availability when I needed it, for all his advice on a subject that he masters at the highest possible level, and for making sure I did not get lost during this journey.

I would like to also express my deepest gratitude to my academic supervisor **Prof. Riccardo Scateni** for his guidance, for his patience and for giving this opportunity of studying a PhD at the University of Cagliari. Additionally, this endeavor would not have been possible without the generous support from the Marie Sk lodowska-Curie Fellowship by European Union's H2020 research and innovation program grant 813170, who financed my research.

I am also grateful to my colleagues at **CRS4**, especially to the Visual and Data-intensive Computing (ViDiC) group for their expert knowledge and for sharing passions; the administrative department for their helping and patience in bureaucratic stuff; and Unitary Union Representation (RSU), for their constant support of workers' rights. Special thanks to **Katia Brigaglia** for her help with bureaucratic issues, professionalism and, above all, patience, **Fabio Bettio** for his technical support, generosity and kindness (makes excellent Genoese pesto!), **Moonisa Ahsan** for her kindness, **Fabio Marton, Francesco Versaci, Vittorio Meloni, Marco Cogoni, Mauro del Rio, Simone Leo, Alessandro Sulis, Ruggero Pintus, Matteo Vocale, Antonio Zorcolo, Massimo Gaggero, Francesca Frexia, Giovanni Busonera**, and more, for their lunchtime chats, their moral support, and the many coffees that have not let me sleep.

I am also thankful to **Prof. Marco Agus**, now at HBKU, Qatar, for his professionalism and kindness and his collaboration on some of our research efforts.

I would like to extend my sincere thanks to the **EVOCATION team**, composed of Early-stage researchers like me, support personnel, and supervisors, for sharing fresh passions and becoming friends.

Thanks should also go to all the people who have crossed my path professionally and who impacted and inspired me, specially **Prof. Paco Herrera, Prof. Luciano Sánchez, Prof. Jesús Chamorro, Alfredo Rivela and Pilar Holgado**. I am also grateful to all the professors of both the bachelor's and master's degrees who have made it possible for me to be here today, each and every one of them has contributed something to the researcher/professional that I am today.

Lastly, I would like to acknowledge to my family, especially **my parents (Pepe and Isa), spouse, sister (Marta), my sister's child (Konstantin and Elisabeth), aunts/uncles, cousins, my new in-laws, and friends**. Their belief has kept my spirits and motivation high throughout this entire process. Mainly, I would like to recognize the unconditional support that encouraged me, to my parents and, above all, the moral/daily support from my husband, **Andrea**. To my special friends, **Salva, Mari Carmen, Maica, Pedro, Ana, Carla, Mamen, María, Lidia, Maika, Alicia, Ximena, Nacho, Jesús, Carlos, Raquel, Samu, Eugenio, Paco, Hans**, and a long etcetera for everyone who has crossed my life in a positive way and taken care of me even from a distance.

Eva Almansa
Cagliari, Italy
30th October — 2023.

Contents

List of Figures

List of Tables

Preface

This thesis represents a summary of the work done from 2020 to 2023 at the Visual and Data-intensive Computing **(ViDiC)** group of **CRS4** (Center for Advanced Studies, Research and Development in Sardinia) under the direction and supervision of **Dr. Enrico Gobbetti**, in close collaboration with **Dr. Giovanni Pintore**. I really want to warmly thank both of them for the opportunity to be part of the team and for all the great support that I have received. Also, my gratitude to **Prof. Riccardo Scateni** for his constant motivation and academic support from University of Cagliari. This was, indeed, a rewarding experience both scientifically and personally.

The scientific work in this thesis has been performed mostly within the international framework of **EVOCATION** (Advanced Visual and Geometric Computing for 3D Capture, Display, and Fabrication) project, a leading European-wide doctoral Collegium for research in Advanced Visual and Geometric Computing for 3D Capture, Display, and Fabrication supported by European Union's H2020 research and innovation program grant 813170 (October 2018-May 2023). The consortium participants are the University of Rostock (UNIRO), the Center for Research, Development and Advanced Studies in Sardinia (CRS4), the University of Zurich (UZH), the Italian National Research Council (CNR), the Technical University of Vienna (TUW), Fraunhofer IGD (FHG-IGD), and the two companies Holografika (HOLO) and GEXCEL.

The objective of the EVOCATION research network was, on one hand, to equip the enrolled Early-Stage Researchers (ESR) with the right combination of research-related and transferable competencies, and, on the other hand, to foster, by sci-

entific exchange and collaboration, the development of new technologies and knowledge around four interconnected interdisciplinary themes:

- **Innovation in visual and geometric acquisition and processing**. The focus, here, was on two separate challenging use cases, that led to well-defined research lines. The first one was dedicated to scalable mass-digitization of shape and appearance of large collections of 3D objects with complex materials, with special emphasis on cultural heritage objects. The second one was concerning the introduction of solutions for fast mobile capture of large environments and for creating semantically-rich representations, with a particular focus on complex indoor environments.
- **Innovation in interactive data-intensive visualization**. In this context, the project studied solutions both to enable the exploration of massive data at interactive rates, and to provide useful navigation tools supporting the exploration of complex acquired objects with semantically-rich annotations, beyond pure raw-data inspection, with a special focus on flat, but visually reach objects (e.g., paintings and bas-reliefs).
- **Innovation in computational fabrication**. This research line concerned fabrication and 3D printing technologies, both to expand the design space and to ensure a higher quality reproduction of acquired models.
- **Innovation in display systems**. The goal, here, was to improve visual replication and understanding of 3D data and associated information through novel high-bandwidth display environments, including high-density ubiquitous displays, large high-resolution displays (LHDs), novel multi-user computational 3D displays capable of fully matching human perceptual capabilities (light field displays), and multi-display environments.

More details on the project are available at the project web site (www.evocation.eu).

As an ESR and Marie Sklodowska-Curie Fellow in the project, my research trajectory focused mostly on the first research theme, and more precisely on the automatic 3D reconstruction of indoor environments from panoramic images.

With this fellowship, I was also enrolled as PhD Student in the Computer Science Program at the Department of Mathematics and Computer Science at the University of Cagliari under the kind tutoring of Prof. Riccardo Scateni.

My topic was inserted in a specific research project under the first research theme, devoted to ***"Scalable Reconstruction and Exploration of Complex Indoor Environments"***, where the goal is to study techniques to apply prior knowledge for the automatic extraction of structured representation of interior environments from

incomplete and noisy sampled data. For my thesis, I specifically focused on inferring a maximum amount of information from a single omnidirectional image, using only visual data, eventually enriched with very sparse depth information.

The work described in this thesis received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sk lodowska-Curie Actions Innovative Training Network (MCSA-ITN) grant agreement *No 813170*, as well as from Sardinian Regional Authorities for projects connected to CRS4 Visual and Data-intensive Computing activities.

Eva Almansa
Cagliari, Italy
November — 2023.

Chapter 1

Introduction

The automatic 3D reconstruction and modeling of indoor scenes has attracted a lot of research in recent years, making it an emerging well-defined sub-field of 3D reconstruction. The aim is to transform an input source containing a sample of a real-world interior environment into a compact structured model containing geometric, structural, and/or visual abstractions. In this thesis, I have concentrated on extracting information from panoramic images, since they provide the quickest and most complete single-image coverage and are supported by a wide variety of professional and consumer capture devices that make acquisition fast and cost-effective. This chapter outlines the scientific motivation behind this research, provides a brief summary of research achievements, and presents the organization of this thesis.

1.1 Background and motivation

The automated reconstruction of 3D models from acquired data (e.g., images or depth measurements) has been one of the central topics in computer graphics and computer vision for decades. The growth of this field can be attributed to the simultaneous alignment of scientific, technological, and market developments. These developments now align with the widespread accessibility and increasing affordability of high-quality visual and 3D sensors, which are coupled with expanded opportunities for large-scale data processing.

In this context, the automatic reconstruction of indoor environments is gaining wide attention. As detailed in a well-established survey [1], the focus has been on the

creation of specialized techniques for very common and very structured multi-room environments, such as residential, office, or public buildings. This is because the construction, management, and analysis of those buildings is common in diverse fields such as architecture, civil engineering, digital mapping, urban geography, and real estate [2]. Commercial solutions in these areas range from virtual tours creators (e.g., *3DVista* [3]), to systems that support the construction process (e.g., *StructionSite* [4] or *Reconstruct* [5]), to general solution for sharing and exploring structured models (e.g., *Matterport* [6]).

In this sub-field of general 3D reconstruction, 3D representation of an interior scene must be inferred from a collection of measurements that sample its shape and/or appearance, exploiting and/or combining sensing technologies ranging from passive methods, such as single- and multi-view image capture setups, to active methods, such as depth cameras, optical laser-based range scanners, structured-light scanners, and LiDAR scanners [7, 1].

Within the EVOCATION project (MCSA-ITN grant agreement 813170), in which I have carried out my dissertation work, the research team has extensively analyzed the research domain in a survey published in Computer Graphics forum [1], and has illustrated the main techniques in a SIGGRAPH Course [8] and a CVPR Course [9]. Since these works have been the start of my journey into 3D reconstruction of indoor environments, and have also become well-established surveys in the research community, I will frequently refer to those summaries for an extended view of the domain that goes beyond the scope of this thesis.

All 3D indoor reconstruction techniques aim to transform an input source containing a sample of a real-world interior environment into a compact structured model containing geometric and/or information at an application-specific level of abstractions. Since many variations exist, the first points to be defined are, therefore, the targeted input and output of this research.

The input data can be obtained from a variety of sensors. Visual input (e.g., photographic images) has attracted a lot of interest, due to the abundance of means to acquire it, the ease of capture, and its low cost. A single perspective image, however, provides a very narrow view, and capturing multiple images complicates capture and requires multi-image registration. For this reason, in recent years, 360° capture has emerged as a very appealing solution, since it provides the quickest and most complete single-image coverage and is supported by a wide variety of professional and consumer capture devices that make acquisition fast and cost-effective. While pure 360° visual input is, possibly, the most widespread capture method, (semi-)professional indoor capture techniques have also witnessed the emergence of synchronized depth and visual 360° capture devices, that augment

dense image data with sparse depth information. Such solutions, for instance, have widespread use in the real-estate domain [1].

Extracting geometric and/or structural information of an interior model from a single 360° image, eventually augmented with sparse data, has also attracted a lot of research in recent years and has lots of practical applications that require different pipelines, as discussed in Pintore et al. [1]. Two fundamental problems that have emerged in this context, and form basic building blocks in most, if not all, reconstruction pipelines are *depth estimation*, which consists to augment the input visual representation with per-pixel data consisting in the distance of the visible pixel from the viewer, and *3D architectural layout estimation*, which consists in inferring from the image of a furnished room the 3D layout surface determined by joining the walls, ceilings, and floor that bound the imaged room's interior [1].

Despite the wide context provided by a spherical panorama, without prior assumptions, these fundamental reconstruction problems remain, however, ill-posed, since an infinite number of solutions may exist that fit the under-sampled or partially missing data provided by a single image, even if enriched with a few depth measurements. For this reason, very specific geometric priors have been proposed in the past for structural and geometric recovery in indoor environments (see Chapter 2). These solutions, however, are typically very restrictive in terms of supported room shapes, and also rely on the ability to extract specific visual features in the images (e.g., corners or edges), which may be difficult in the typical indoor environments dominated by large featureless walls and big occluded areas due to furniture. In recent years, data-driven solutions that discover hidden relations from large data collections have shown that many priors imposed by pure geometric reasoning approaches can be relaxed [7, 1].

Considering all of the above, the research comprising this thesis has been focused on deep learning solutions based on monocular panoramic image analysis for the reconstruction and representation of indoor environments, either using it standalone or eventually combining it with sparse geometric information. The main hypothesis under which this thesis is performed is that selected capture and architectural priors can be effectively combined with data-driven solutions to create indoor reconstruction techniques that outperform specific indoor reconstruction methods based on geometric reasoning, as well as generic data-driven 3D reconstruction solutions that are not indoor-specific.

1.2 Objectives

Based on aforementioned considerations, further expanded in Chapter 2, I set as a goal of this thesis to advance the state-of-the-art in the reconstruction from panoramic images by answering the following questions:

1. **How to better associate pixel-wise geometric information to a single panoramic image of an interior model.**
   Depth estimation from a single image is a classic problem in computer vision and many solutions exists. However, the aim is to study ways to exploit the peculiar characteristics of the data source (a single panorama) and of the imaged environment (an interior one, e.g., a room). By exploiting priors stemming from this restriction, and combining them with data-driven priors that can be learned from large collections of data, the expectation is to obtain deep-learning-based solutions that outperform generic ones. Network structure, loss functions, and training methods will be the targets of the research discussed in this thesis.

2. **How to improve the previous approach in the presence of limited geometric information.**
   Since purely visual capture is inherently ambiguous, many efforts have been devoted to solutions that also exploit capture devices that provide synchronized high-resolution depth and color data. Due to the limitations of these devices, however, the input geometric information is typically much sparser than the visual input. For this reason, many solutions to depth estimation and infilling problems have been presented (see 2). The goal of this research line is to push the boundary by exploiting priors that are typical of indoor environments. By doing that, we expect to improve the performance of methods that perform depth estimation on general environments, as well as of methods that perform infilling of small holes taking into account the characteristics of the neighborhood.

3. **How to extract layout information from a single panorama.**
   The goal, here, is to go beyond the simple extraction of per-pixel depth, transforming a single image of a furnished room into the 3D layout surface determined by joining the walls, ceilings, and floor that bound the room's interior. The problem is a fundamental one for many applications, for instance as a building block to produce building information models, and is very challenging, due to the intrinsic characteristics of indoor environments, where furniture and other indoor elements mask large areas of the structures of interest, and concave room shapes generate vast amounts of self-occlusions.

The aim, here, will be to extend the deep-learning solutions developed for depth estimation to layout estimation. It is expected that the different nature of the problem will lead to different data representations and network structures. On the other hand, the expectation is that the knowledge gained on indoor-specific priors for depth estimation could also provide a guide in this context.

1.3 Achievements

The solutions proposed in this thesis have achieved to improve the state-of-the-art in all the three identified research lines. Novel deep-learning methods have been thus proposed for depth inference, depth densification, and layout estimation.

My main results and contributions to the state of the art are the following:

- **An innovative end-to-end technique for deep dense depth estimation from a single indoor panorama** (Chapter 3). The method, introduced in a CVPR 2021 contribution [10], predicts the depth map starting from a single indoor 360° image. Since gravity plays an important role in the design and construction of interior environments, world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. The proposed solution, therefore, leverages the peculiar characteristics of gravity-aligned images of indoor environments in the network design. My prime contribution was to the methodology, implementation, testing, and validation of the developed method. In particular, I participated in the discussions that led to the introduction of the methods, contributed to their implementation, and ran the tests of the methods and competitors' implementation, generating the results and analyzing them.
- **A novel approach for deep panoramic depth prediction and completion for Indoor Scenes** (Chapter 4). The method, published as an article in the Computational Visual Media journal [11], with myself as a joint first author, expands over the previous approach by also exploiting optional sparse depth information, without any assumption on the sparsity pattern. The end-to-end deep learning solution to jointly perform real-time dense depth prediction and completion from single-shot indoor 360° captures. The input is a single equirectangular image registered with a sparse depth map, as provided by a variety of common capture setups. This method, the first to work directly on equirectangular images of indoor environments, introduces several specific novelties, including a dynamic gating system to process together dense visual data and sparse geometric data and a new augmentation strategy that

increases robustness to different types of sparsity, including those generated by various structured light sensors and LiDAR setups. My contribution to this work was major and concerned with the conceptualization, methodology, implementation, testing, and validation of the developed method. In particular, I jointly invented the method with Giovanni Pintore, implemented major portions of the system, created code for generating synthetic datasets and using them for training and testing, created code for integrating real-world data acquired with a mobile scanner, ran the tests of the methods and competitors' implementation, generating the results and analyzing them. I consider this contribution the primary one in this thesis. It is also interesting to note that, beyond solving the sparse-to-dense problem, the proposed network design is also suitable for pure depth estimation.

- **An innovative solution for 3D reconstruction of an indoor layout from a single omnidirectional image** (Chapter 5). The method, presented at SIGGRAPH Asia 2021 and published in ACM Transactions on graphics [12] targets the recovery of the 3D shape of the bounding permanent surfaces of a room from a single panoramic image, using a graph-convolutional network capable to infer a tessellated bounding 3D surface from a single 360-degree image. Differently from prior solutions, the problem is fully addressed in 3D, significantly expanding the reconstruction space of competing solutions comprising the prior state-of-the-art. My prime contribution was to the conceptualization, methodology, implementation, testing, and validation of the developed method. For this work, in particular, I participated in the discussions that led to the introduction of the methods, contributed to the creation of testing datasets, and ran the tests of the methods and competitors' implementation, generating the results and analyzing them.

In addition, during the course of my thesis, I have also contributed to an additional work [13], that I have not included in the thesis since I have only contributed to the validation of the approach by performing tests on standard benchmarks and user-captured data. The work introduces a novel light-weight end-to-end deep network that, from an input 360° image of a furnished indoor space automatically returns an omnidirectional photorealistic view and architecturally plausible depth of the same scene emptied of all clutter. While my contribution to that work was limited, it shows an important direction for future work, i.e., mixing the per-pixel and layout extraction problems in solutions that also strive to synthesize visual data.

1.4 Organization

This thesis is based on the results that I have published in EVOCATION project deliverables [14, 15, 16], articles [12, 11], and conference proceedings [10]. I have organized them in order to show in a natural and coherent order all the outcomes obtained. Following is a brief overview of each chapter:

- **Chapter 1** (this chapter) introduces the topic and motivation for this Ph.D. dissertation, describes my objectives, and summarizes my results.
- **Chapter 2** provides a general background for the thesis, providing a wider view of previous approaches.
- **Chapter 3** describes the technique I have introduced for inferring depth from a single panoramic image using and end-to-end deep-learning solution;
- **Chapter 4** describes how additional sparse depth information can be exploited to significantly improve depth reconstruction, while remaining within end-to-end deep learning techniques and without making assumptions on specific sparsity patterns;
- **Chapter 5** illustrates how architectural and data-driven priors can be exploited to infer plausible 3D layout information from a single panoramic image;
- **Chapter 6** provides a conclusion and short summary of the achievements, a critical discussion of the results obtained and of how they advance the state-of-the-art, as well as some reflections on future lines of work.

Chapter 2

General background

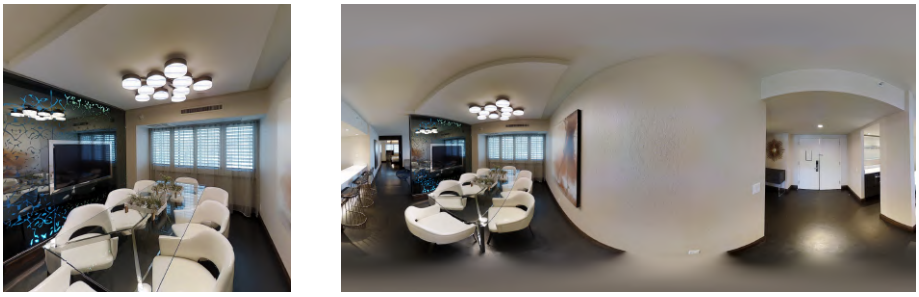
Before presenting the thesis contribution, I provide here relevant background information on panoramic capture, on the targeted reconstruction problems, and on the priors that are typically employed to cope with noise and ambiguities, also covering publicly available panoramic indoor datasets that can serve to define data-driven priors. I will the provide a brief survey of the state-of-the-art on the specific targeted tasks, i.e., depth estimation, depth completion, and layout estimation, and conclude with the identification of open problems and of the hypotheses behind the solutions that will be detailed in the forthcoming chapters.

2.1 Introduction

Reconstruction of interior structures is a well-defined topic which has attracted significant interest recently. In this field, the aim is to extract information from an input source to convert it into a representation of the imaged models that optimizes certain application-specific characteristics. The field is very vast, and I refer the reader to established surveys for a general introduction and coverage of the state-of-the-art [1]. In this thesis, as discussed in Chapter 1, I focus on monocular 360° input, and tackle the three fundamental problems of depth estimation, depth completion in presence of sparse depth information, and 3D architectural layout information.

Before presenting in the next chapters the methods and results obtained on these tasks, I provide here relevant background information and motivation for the direction taken. First, I will briefly introduce methods and tools for panoramic capture

and panoramic image representation, covering both the pure visual case (Sec. 2.2) and the presence of extra depth (Sec. 2.3). Then, I will characterize the depth estimation and completion problems and differentiate them from the layout estimation problems (Sec. 2.4), before introducing the priors that are typically employed to solve make them tractable (Sec. 2.5), introducing both geometric and data-driven ones. Since the work will concentrate on methods that learn from large sets of examples, I will briefly introduce the main concepts behind deep learning solutions (Sec. 2.6), before analyzing the state-of-the-art in the areas covered by this thesis (Sec. 2.7). I will then summarize the characteristics of the available annotated public datasets that can serve to train, validate and test data-driven solutions (Sec. 2.8).



(a) ***Perspective image*** **(b)** ***Equirectangular image***

Figure 2.1: ***Comparing conventional perspective capture with 360° capture.*** *Fig. 2.1a shows a perspective image that is a transformation from the equirectangular image, Fig. 2.1b. Both images are from the Matterport3D dataset [6].*

2.2 Omnidirectional image capture

A wide variety of solutions exists for capturing 3D information on indoor environments, ranging from mobile laser scanners to active depth sensors [1]. Among the many possibilities, purely image-based methods are very important, not only because cameras provide a very widespread, practical, and affordable solution, but also because visual information is paramount for a variety of applications, ranging from navigation, location awareness, as-built-, and existing-condition reconstructions [17]. For this reason, many efforts have been devoted to exploit captured visual information, either alone or in conjunction with some registered depth information (Sec. 2.3).

Pure visual capture and processing is one of the most well-researched topics. Using a classic camera with a limited field-of-view, however, does not provide enough information for achieving plausible full-room reconstruction, and forces users to

employ multi-view methods, that increase capture efforts [18]. Moreover, classic approaches based on multi-view stereo [19] and structure-from-motion (SfM) [20] do not, by themselves, provide complete solutions in indoor environments, due to the abundanced of clutter and non-cooperative textureless and reflective surfaces that make feature detection, triangulation, and surface reconstruction difficult in interior environments [1].

Figure 2.2: ***Matterport High Resolution 360° Cameras.*** *These 360° cameras are fast and affordable to capture small to medium spaces in 3D using Matterport (this picture belongs to here matterport.com/cameras/360-cameras).*

For these reasons, in recent years, 360° capture, also known as *panoramic*, *spherical*, or *omnidirectional* capture, has attracted a lot of attention, since it provides the quickest and complete single-shot coverage [21, 1, 22]. Fig. 2.1 provides a comparison between a perspective and panoramic view of an indoor environments.

While a panoramic image with a 360° horizontal viewing angle and 180° vertical viewing angle can be obtained by many means, including stitching of a sequence of photos captured with a mobile phone [23], modern commodity spherical cameras have become very widespread and increasingly popular in many application fields [24]. Fig. 2.2 shows, for instance, a set of 360° cameras that are usable with the popular Matterport industrial interior capture, reconstruction, and touring system.

With such cameras, with a single click, a user obtains a full-view image with the same efforts needed to take a single regular photo, since the processing (e.g., stitching of multiple fish-eye views) is performed fully internally before delivering the output. The captured content has the benefits that it has a full-view, capturing the light intensity of the entire surrounding environment in a single-shot and at (approximately) the same instant. The camera design and processing methods typically ensure, also, that there is a single (effective) center of projection, and that uniform resolution is maintained in the horizontal direction, which is difficult to achieve with the stitching of multiple casually captured images [25, 26]. From the

processing point of view, the spherical camera can be modeled as a unit sphere with no intrinsic parameters, and the capture is thus determined fully by the extrinsic parameters [21].

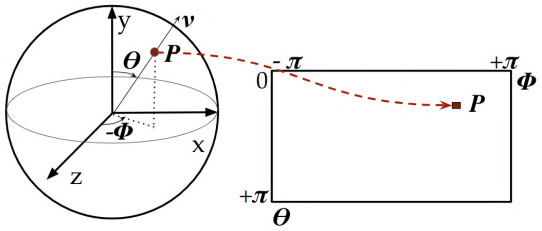
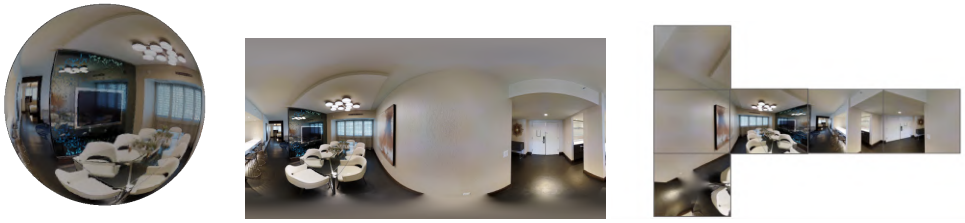


Figure 2.3: ***Equirectangular mapping.*** *A spherical projection by a 360° camera is directly transformed to a 2D equirectangular projection. The intensity value from the point P of the spherical representation, where $\theta \in [0, 2\pi)$ and $\phi \in [0, \pi)$, is mapped to an integer pixel position (x, y) of a $width(w) \times hight(h)$ equirectangular image where $x = \frac{\theta w}{2\pi}, y = \frac{\phi h}{\pi}$.*



(a) ***Spherical capture*** **(b)** ***Equirectangular projection*** **(c)** ***Cube-map transformation***

Figure 2.4: ***Omnidirectional image representations.*** *Fig. 2.4a shows a spherical image, Fig. 2.4b its equirectangular projection and Fig. 2.4c its cube-map projection. The original image is from the Matterport3D dataset [6].*

Nevertheless, the sphere is not isomorphic to a plane, and representing the capture as an image typically involves a mapping transformation. While some cameras provide access to the original unstitched images, that provide the highest resolution capture, the most common approach, that has become a de-facto standard in indoor capture and processing, is to extract from the device an equirectangular projection sampled into a regular rectangular 2D grid [27], obtaining what is often called a *full panoramic image* (Fig. 2.3).

Other projections can also be used to mitigate spherical distortion. For example, the cube-map projection (i.e., projecting around the sphere a 90° vertical and 90° horizontal FOV to each face of the six faces of the cube) (Fig. 2.4) is often used as a representation for image viewing, e.g., in WebXR environments or popular

streaming viewer (e.g., YouTube 360 video format - YouTube 360 video format (see paulbourke.net/panorama/youtubeformat/). Other popular formats, include tangent image projections [28, 27], are covered in a recent survey [21].

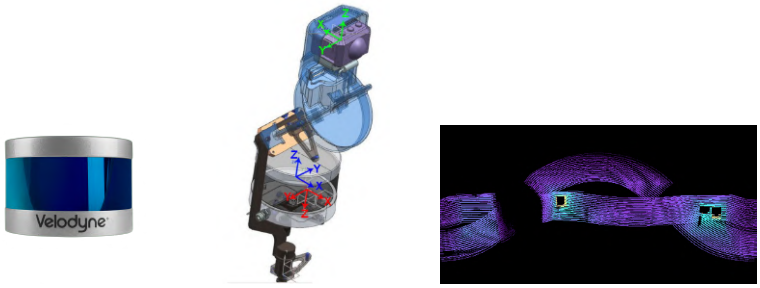
Since equirectangular images are device-independent and supported by most, if not all, the devices, we set as a goal in this thesis to provide solutions that directly take as input an equirectangular image, without conversion to other intermediate formats, and, where relevant, also provides output (e.g., depth) in an equirectangular format. It should be noted that, in contrast to other formats such as cube maps, the equirectangular representation provides full continuity along the horizontal direction, and that the reduction in resolution and singularity at the poles are not that relevant in gravity-aligned indoor capture, as the low-res/discontinuous are occurring directly above or below the capturing camera, in areas that are typically masked by the capture device (floor) or presenting not important information (ceiling). As we will see, relying on gravity-aligned capture will be a fundamental aspect of the introduced approach, that will be exploited for the design of all solutions, and that is guaranteed by most capture protocols.

(a) *Structured-light sensor* (b) *Matterport Pro 3D camera* (c) *Depth map*

Figure 2.5: ***Structured light scanners.*** *Structured light scanners use trigonometric triangulation by a projector to display a series of linear patterns onto an object. Then, by analyzing the distortions of these lines or dots is determined the depth, Fig. 2.5a. Although, the captures can have some artifacts such as lots of missing areas when has a large depth as Fig. 2.5c shows, which is a depth map captured by structured-light sensor (Matterport Pro 3D camera, Fig. 2.5b).*

2.3 Augmenting single-shot panoramas with depth information

Since visual input alone, especially in the monocular case, is inherently ambiguous, combining active scanners with passive cameras to acquire jointly shape and color has been studied for a long time [29, 30]. This topic has gained increased attention, recently, due to the numerous affordable solutions that are emerging both in the



(a) *Velodyne VLP-16* (b) *Heron LiDAR* (c) *Sparse depth captured by a Heron LiDAR*

Figure 2.6: ***LiDAR scanner.*** *Here is shown two LiDAR scans (Light Detection And Ranging). Fig. 2.6a which has, in general, the following specifications: 16 beams/lasers, a full 360° horizontal FOV and 30° vertical FOV. Fig. 2.6b is a Heron LiDAR which has 2 Velodyne VLP-16, one on top of the other and oriented 45° down, this Heron LiDAR scan is from $GEXCEL$ company (gexcel.it/it) which is explained in more detailed in Chapter 4. As an example, the sparse depth captured by a Heron LiDAR is shown in Fig. 2.6c, having two groups of 16 beams/lasers that each Velodyne VLP-16 has captured. Each sparse scan takes about 300 $milliseconds$ and produces about 16% of pixels with valid depth.*

professional field (e.g., backpacks [31]) and consumer markets (e.g., consumer RGB-D cameras [32]). RGB-D cameras are compact systems that (virtually) acquire RGB images along with per-pixel depth information, while scanner solutions typically employ separate geometric and visual capture subsystems (e.g., LiDAR and RGB camera) that are later synchronized and merged together. In this context, structures can be recovered from data fusion [33, 34]. Several solutions are specifically designed for indoor captures [31], since outside captures often have a too high depth range for several active methods or highly illuminated environment [35, 36]. While the majority of works are focused on small-FOV perspective poses [37] or planar projections for outdoor acquisitions[38, 39], in this thesis we only discuss the devices that can enrich omnidirectional images with some depth information.

Available solutions include combining (panoramic) cameras with structured-light sensors (e.g., Fig. 2.5) or LiDAR (Light Detection And Ranging) scanners (e.g., Fig. 2.6 and Fig. 2.7), that both can provide, as output of the capture process, an equirectangular depth image aligned with the color image. However, in both cases, the amount of depth information that can be recovered with each captured color image is very limited. For instance structured-light sensors are at lower resolution than comparable visual cameras, are very sensitive to illumination variations, and suffer from short ranging distance. Longer ranging LiDAR sensors are more robust and accurate, but can only provide extremely sparse measurements at real-time rates [30], typically only on a few stripes. Sparsity patterns of the depth signal, moreover, are

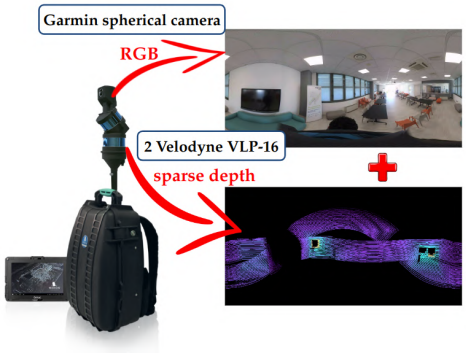


Figure 2.7: ***A mobile backpacked RGB+LiDAR acquisition system.*** *This mobile backpacked LiDAR acquisition system is equipped with a Garmin spherical camera on the top for the RGB panoramic capture and, below, it has a Heron LiDAR (i.e., 2 Velodyne VLP-16, see Fig. 2.6) for the sparse panoramic depth capture. This mobile backpacked system is a product that belongs to GEXCEL company (gexcel.it/it) and data generated by this have been used in this thesis (Chapter 4).*

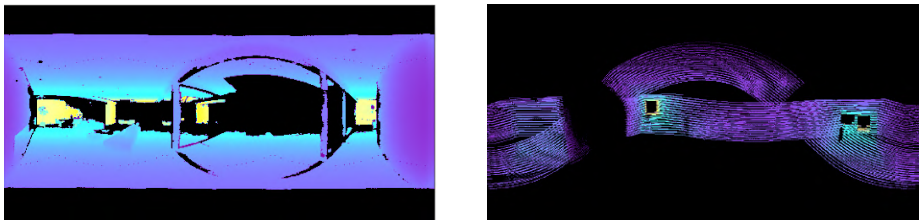
very different depending on the depth sensing technology (see Fig. 2.8).

We therefore set as a goal for this thesis to evaluate how we can exploit the higher quality visual signal to improve quality of the depth signal that is coming from the depth sensors, in a way that is robust to density pattern variations.

2.4 Targeted indoor reconstruction problems

The goal of any 3D indoor reconstruction pipeline is to transform the input source into a problem-specific representation that contains geometric and/or structural information on the scene. We have seen that the input, in this thesis, is a single panoramic image, represented in equirectangular format, eventually enriched with a second aligned equirectangular image that contains depth information for some of the pixels. The expected output depends on the specific targeted problem, that is the extraction of a dense equirectangular depth map (with or without the support of sparse depth information) or of an architectural 3D layout. Both problems can be interpreted as ill-posed inverse problems, since, due to the presence of outliers, noise, and missing data many plausible reconstruction can produce an indoor environment fully compatible with the measurements. For these reasons, the research community has proposed many solutions [40, 1], that all rely on the introduction and exploitation of prior knowledge (Sec. 2.5).

The nature of these problems is similar, in the sense that the input and the targeted environment are the same, there are also some important differences. First of all,



(a) *Depth map by a Matterport camera*

(b) *Sparse depth captured by a Heron LiDAR*

Figure 2.8: ***Different kinds of sparse depth.*** *Fig. 2.8a is a depth map captured by structured-light sensors (Matterport Pro 3D camera), has lots of missing areas when rooms are large, surfaces are shiny or thin, and strong lighting is abundant. Fig. 2.8b is a depth map captured by a LiDAR setup (two Velodyne VPN-16 shifted of the vertical direction with different direction) has lots of valid information but only for a few stripes, where obtains horizontal 360° depth information but still has narrow vertical FOV. In both captures are represented in black color the holes/missing area.*

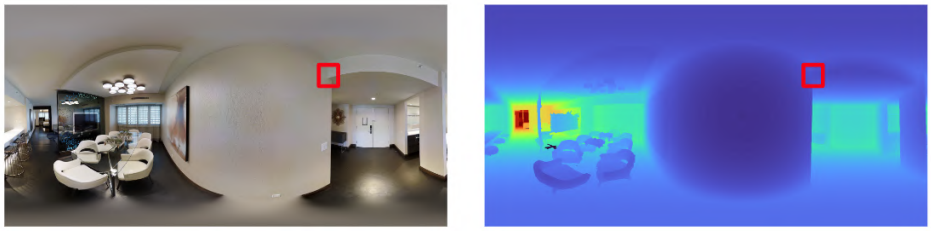


Figure 2.9: ***Pixel by pixel depth estimation.*** *Here is shown an equirectangular image (the image on the left), its registered depth map (the depth map on the right), and into both captures are represented a red box pointing out one pixel depth estimated from the RGB image. The sample is from the Matterport3D dataset [6].*

dense depth reconstruction and depth completion must produce *per pixel* information that is associated with the corresponding color (and eventually sparse depth) information present at the same pixel. Layout estimation, by contrast, requires to further parse the imaged space into the structural elements that bound its geometry [1] (e.g., floor, ceiling, walls, etc.). This task is very challenging, due to the intrinsic characteristics of indoor environments, where furniture and other indoor elements mask large areas of the structures of interest, and room shapes generate vast amounts of self-occlusions (see Fig. 2.10). Thus, 3D layout reconstruction is more complex than depth estimation, since it does not simply assign a depth to each visible pixel, but must extrapolate large portions of the invisible structure, which can be occluded not only by objects but by the structure itself, leading to multiple intersections per view ray (see Fig. 2.11). For this reason, we cannot expect to encode the output of 3D architectural layout estimation into a single value per pixel, but we must devise a representation that is simple enough to be extracted

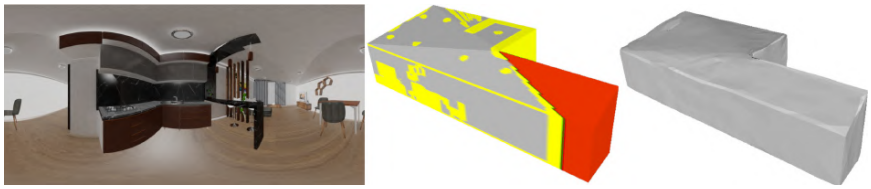


Figure 2.10: ***Types of occlusions in interiors.*** *Here is shown an equirectangular image (first figure on the left) with its layout representation (the others two figures). The layout representation is the room's interior bounded by the walls, ceilings, and floor. The colored layout is a room that has occlusions from walls (red) or from furniture (yellow).*

from the very partial information we have as input, while expressive enough to represent important classes of indoor environments.

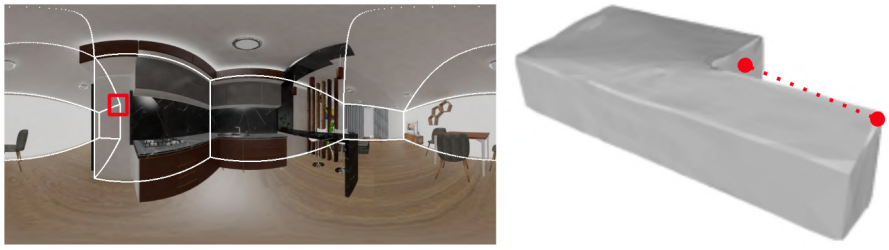


Figure 2.11: ***Dealing with occlusions.*** *Here is shown an equirectangular image, its layout representation (a room's interior delimited by walls, ceiling and floor), and pointing out one pixel from the RGB image that has to estimate the shape by occluded structure itself, being multiple intersections and thus multiple values, one for each intersected wall.*

2.5 Prior knowledge

Extracting geometric information from monocular input, even with the full context provided by 360° capture, is inherently ambiguous and is particularly complex in indoor settings characterized by large texture-less surfaces, abundance of clutter, and severe occlusions [1]. Thus, indoor reconstruction requires very wide context information and must exploit very specific geometric priors for structural recovery [1].

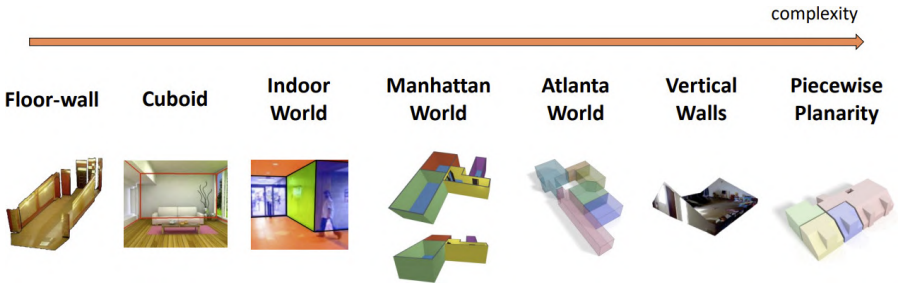


Figure 2.12: ***Architectural priors.*** *A list of architectural priors used in 3D reconstruction, in order of complexity (image courtesy of Pintore et al., CVPR 2023 [9]).*

Fig. 2.12 summarizes, in order of complexity, the most commonly geometric priors used in indoor for surface reconstruction. They include *Floor-Wall* (FW) [41], composed by a single flat floor and straight vertical walls; *cuboid* (CB) [42], being a single room of cuboid shape; *Indoor World Model* (IWM) [43], with a single horizontal floor, a single horizontal ceiling, and vertical walls that meet at right angles; *Manhattan World* (MW) [44], an IWM without the restriction of a single floor and celing; *Atlanta World* (AW) [45], similar to MW, without the restriction of walls connecting at right angles; *Vertical Walls*, and Atlanta-World model with possibly sloped ceilings and floors [45], and piecewise planarity, that simply bounds the interior with large planar surfaces [46].

Relying on architectural priors makes it possible to reduce the solution space, making reconstruction more tractable. For instance, methods based on the IWM assumption [47] can rely on finding and extruding a 2D floor plan, whose walls are forced to be aligned with one of the two principal directions. This makes it possible to derive solutions that detect simple structures by simply looking for a limited number of corners [47, 48] (Fig. 2.13). On the other hand, this sort of approach has also several important limitations. First of all, methods that only employ geometric reasoning based on the matching of features detected in images with possible reconstructions compatible with the prior are heavily dependent on the number

and quality of detected features, and only the most restrictive priors (e.g., those based on MW assumptions) are robust enough to cope with the typical amount of missing/inconsistently-detected features that may occur in typical panoramic images of furnished rooms [1]. These priors, however, are representative only for a restricted class of rooms, since, for instance, non-orthogonal walls are not uncommon. Moreover, while solely relying on geometry reasoning can produce solutions for the layout detection problem, it can only serve as a support for the depth estimation problem, since non-permanent structures do not typically follow strict arrangements that can be modeled with simple rules.

Recent research trends have shown that data-driven solutions that discover hidden relations from large data collections, and in particular those based on deep learning, have shown that many priors imposed by pure geometric reasoning approaches can be relaxed [1]. I will also pursue this research line in this thesis, where relaxed geometry priors will be used not as as a basis for geometry reasoning based on detected features, but to drive the design of effective networks and training structures.

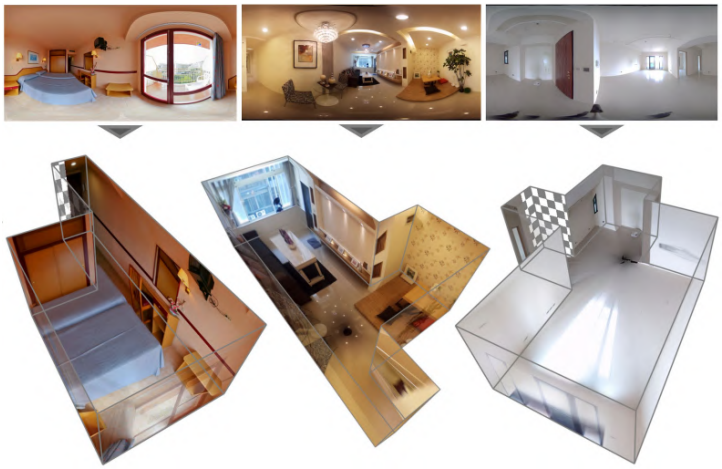


Figure 2.13: ***Manhattan Room Layout Reconstruction from a Single 360° image.*** *In this comparative study introduce the prior MW used previously in perspective view to full-panoramic view. This figure belongs to [49].*

In particular, one prior that will be consistently used throughout this thesis is the assumption that capture of the scene through an equirectangular image is aligned to the gravity vector (i.e., camera is placed on an horizontal-ground plane, see Fig. 2.14). Gravity-aligned capture is a very common setup, and all the public 3D indoor datasets (Sec. 2.8) commonly used for training and testing reconstruction solutions appear to have very small orientation deviations. Even in cases where

this assumption is not verified at capture time, several orthogonal solutions exist to gravity-rectify images in a pre-processing step (e.g., [50, 51, 52]), simplifying the practical application of gravity-oriented methods. Thus, it is rational to assume that gravity-aligned processing of images can directly exploit gravity-aligned world-space features [52].

Moreover, all the designs presented will take into account that gravity plays an important role in the design and construction of interior environments, and, thus, world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. For instance vertical and horizontal lines with different characteristics appear both in boundary surfaces (e.g., walls, floor) and in furniture (tables, desks, ...). This will allow us to design networks that perform different operations along the vertical and horizontal directions. In particular, vertical lines are very common in the scene, and are practically not deformed in the projection while the horizontal ones are more so. Moreover, most 360° capture setups have a much more regular coverage along the horizontal than on the the vertical direction because of masking effects [6]. Because of these characteristics, we expect that it will be possible, for each scene region along the dominant vertical direction, to find specific relations to the others by both short-term and long-term spatial dependencies that encode construction constraints typical of certain scene characteristics (e.g., symmetries, spacing, and so on) [53, 54, 45].

Figure 2.14: ***Equirectangular image aligned to the gravity vector.*** *Camera is aligned with an horizontal-ground plane.*

2.6 Basic components of a deep learning solution

Research in structured interior reconstruction has focused its efforts on building models based on data-driven approaches by applying inherent concepts of indoors to guide the transformation of an input using deep learning architectures to achieve the desired target. A definition complete of what is deep learning is beyond the

scope of this thesis, although a few strokes of the most interesting concepts are mentioned, to go into more detail, I refer the reader to a classic book [55] for a wide coverage. I only summarize here the main components.

Basically, deep learning is a specific type of machine learning, which a machine learning algorithm is able to learn from data. One definition of learning by Alpaydin [56], is: *"A computer program is said to learn from experience* E *with respect to some class of tasks* T *and performance measure* P, *if its performance at tasks in* T, *as measured by* P, *improves with experience* E". The meaning of these concepts, in terms of a machine learning system, are, *tasks* described how it should process an example (e.g., regression, task that given an input has to predict a number - for us it will be depth estimation or layout estimation); *performance* evaluates the abilities of the algorithm (e.g., accuracy which measures the correct output of the model directly related to the total number of examples); and *experience* by the information that can extract during the learning process (e.g., supervised learning process which the target is known and labeled, otherwise it is, for instance, unsupervised learning process).

Most of the deep learning algorithms follow a basic structure which is combining the specification of a dataset, a cost function, an optimization procedure, and a model. In our context, the model will be a neural network, whose behavior is determined by the set of parameters (weights and biases) that shape the mapping from the model input (for us equirectangular images that store color and eventually sparse depth) and the expected network output (for us the dense depth map for the depth estimation and completion problems, or the room boundary representation for the layout estimation problem).

Generally, the whole dataset (i.e., a collection of examples) is composed by three sub-sets, training, validation, and test sets; having the assumptions that each example, from each sub-set, is independent and identically distributed [55]. Thus, training/validation sets are used during the training process which trains a model by measuring a training-set (i.e., by training error), while validation-set is used to evaluate the performance of the model after each iteration. The test-set, in contrast, is used during the generalization process of unobserved inputs, i.e., this process is for applying a model on previously unseen inputs, different than those on which the model was trained, computing what is called the generalization error [55].

During the training process an optimization is applied by altering the input, which is the task of either minimizing or maximizing some function. That function is called the objective function, or criterion. The objective function is also called cost/loss/error function when is minimizing it (e.g., in linear regression, one cost function could be to compute the mean squared error between the prediction

and the target). Performing the learning process is caused by the cost function which usually includes at least one term and, also, may includes additional terms, such as regularization terms (e.g., adding to the linear regression cost function a weight decay, also known as L^2 regularization or ridge regression). A definition of regularization by Goodfellow et al. [55]: *"Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error"*.

Most regularization strategies are based on regularization estimators by trading increased bias for reduced variance, i.e., low variance without increasing enormously bias based on constraints and penalties, limiting the capacity of a model by norm penalties. In fact, one of the major research efforts is to find an effective balance between optimization and regularization strategy [55].

In this context, most approaches related to structured interior reconstruction have focused on an optimization process, measuring just the training process, in order to analyze novel ways of inquiring the relevant information from the input as well as identify a balance between optimization and regularization; considering that these models have to manage a huge amount of data to learn an output which causes a fairly high training time even though with GPU-accelerated computation.

2.7 Related work and proposed advances

The general concepts of deep learning have been already applied to the three fundamental tasks in computer vision tackled in this thesis, which are depth estimation, sparse to dense depth estimation, and reconstruction of the boundary surface of a room, all of them from a single omnidirectional image taken inside it. In the following, I discuss the approaches that are most closely related to the solutions that I introduced in this work. I refer the reader to recent surveys for a general coverage of 3D reconstruction in interior environments [1, 30, 40, 21].

2.7.1 Depth estimation from perspective images

Before discussing the works directly applied to omnidirectional images, that are the focus of this thesis, I briefly summarize earlier works on perspective images, since they have predated works specific to panoramic capture, introducing many components that have later been adapted to the task, and have also been used directly in panoramic settings by splitting a panoramic image into multiple perspective views.

Learning-based monocular depth estimation was introduced over a decade ago (e.g., Make3D [57]), becoming, thus, in a fundamental task in computer vision.

While early solutions used various combinations of feature detection, matching, and geometric reasoning, in recent years, a large body of deep learning methods are being proposed for handling this traditional ill-posed problem under less restrictive constraints [58]. This emergence deep learning as well as the availability of large-scale 3D datasets, has contributed to significant performance improvements.

Eigen et al. [59] were the first to use CNNs for regressing dense depth maps from a single image in a two-scale architecture, where the first stage—based on the *AlexNet* feature encoder—produces a coarse output and the second stage refines the prediction. Their work was later extended to additionally predict normals and labels with a deeper and more discriminative model, based on *VGG* features encoder, and a three-scale architecture for further refinement [60].

Laina et al. [61], instead, combined *ResNet* [62] with an up-projection module for upsampling. They also proposed the reverse Huber [63] loss to improve depth estimation. This baseline, named *FCRN*, has become of common use even in the case of panoramic images.

Lee et al. [64], instead, predicted depth from several cropped images combined in the Fourier domain. Conditional Random Fields (CRF) are also often exploited to refine prediction [65, 66, 67, 68].

Fu et al. [69] use dilated convolution to increase the receptive field and apply the ordinal regression loss to preserve the spatial relation among neighboring classes. Unsupervised training for depth estimation is instead performed using photometric loss [18, 70].

However, it has been shown that, without specific adaptations, the direct application of these solutions to 360° depth estimation of indoor environments produces sub-optimal results [71]. For this reason, research has started focusing on methods to exploit the wide geometric context present in omnidirectional images.

2.7.2 Depth estimation from a single omnidirectional image

One of the main limitation of single-image methods lies, in fact, in the restricted field of view (FOV) of conventional perspective images, which inevitably results in a limited geometric context [72]. With the emergence of consumer-level 360° cameras, a wide indoor context can now be captured with one or at least few shots. As a result, much of the research on reconstruction of indoors from sparse imagery is now focused in this direction, even for directly recovering the room layout under specific conditions [73, 74, 54, 45].

In the specific case of depth estimation, a first approach is to convert an omni-

directional image into a cube-map [75], both to deal with the distortion of an equirectangular projection and to take advantage of the consolidated monocular estimation techniques mentioned above.

To make the network aware of the distortion, spherical convolution methods have been also proposed [76, 77, 78, 79]. Tateno et al. [77], for example, demonstrated the effectiveness of distortion-aware convolution, compared to standard convolution, to improve depth estimation and segmentation on panoramic images.

Following this trend, Zioulis et al. [71] adopted the spherical layers of Su et al. [76] for depth estimation in the indoor environment, and proposed a large-scale synthetic dataset consisting of 22,096 re-rendered images from four existing datasets [80]. Wang et al. [81] proposed, instead, a two-branch network, respectively for the equirectangular and the cube-map projection, based on a distortion-aware encoder [71] and the FCRN decoder [61].

Recently, several orthogonal works [82, 83, 84, 85] have exploited the correlation among depth, room layout, and semantics to improve prediction. Such promising solutions require much additional input for training (e.g., annotated room layout, normal maps and semantic segmentation), and exploit a depth estimation baseline based on one of the above-cited approaches. All the above methods bring back the spherical projection to a standard projection to apply encoding-decoding schemes designed for conventional images (e.g., FCRN [61]), while this thesis introduces a scheme designed for equirectangular projections of indoor scenes (Chapter 3).

2.7.3 Depth estimation from a single omnidirectional image with associated sparse depth

Sparse-to-dense depth completion with the support of a guiding RGB image has been the focus of much research [30]. The majority of works focus, however, on small-FOV perspective poses [37] or planar projections for outdoor acquisitions[38, 39]. In this thesis, I only discuss the approaches that can be directly applied or easily adapted to panoramic indoor environments.

In order to upsample and complete a sparse depth signal, generic scene methods that rely on registered RGB-based appearance as guidance either devise custom convolutions and propagate confidence to consecutive layers [86], or use content-dependent and spatially-variant guiding convolutions [87]. Alternative sources of information that are exploited for depth completion may also include confidence masks and object cues [88]. Cross-guidance between the RGB and depth encoders [89] has also been used. Moreover, to avoid the depth mixing typically

induced by the standard MSE loss, a binned depth representation trained using a cross-entropy loss has been shown to be beneficial [90].

Recently, BIPS [91] proposes a bi-modal (RGB-D) panorama synthesis framework to jointly synthesize panoramic RGB and depth. Similar to the work presented in this thesis (Chapter 4), BIPS considers different kinds of sparsity patterns in depth input. However, the goal of BIPS is to provide realistic image inpainting and a complete 3D model for many applications (i.e., including layout), jointly synthesizing color and depth from partial input, rather than focusing on depth prediction and completion.

Even though deep learning has been widely used for inpainting of indoor scenes, extensions of those networks to color guided depth completion are still uncommon [39]. One of the main reasons is that large-scale training sets are not readily available for captured indoor RGB-D images paired with dense depth images. As a result, most methods for depth estimation have been classically trained and evaluated only for pixels that are captured by commodity RGB-D cameras [92]. From this data, they can, at-best, learn to reproduce observed depths, but not complete depths that are unobserved, which in indoors have significantly different characteristics. To address this issue, Zhang et al. [93] introduced a new dataset based on the large-scale Matterport3D [6], which provides 105k RGB-D images aligned with dense depth images computed from multi-view reconstructions in 72 real-world environments, and proposed a hybrid solution to estimate surface normals and solve for indoor depth via a final global optimization. The method, however, has speed limitations and does not scale for different kinds of sparsity (see Sec. 4.1).

More recently, pure deep-learning solutions have been proposed for color guided depth completion. Cheng et al. [94] proposed an approach in which a low-FOV dense depth camera is registered with an omnidirectional camera, and the dense depth from the limited FOV is extended to the rest of the recorded omnidirectional image through a convolutional network. This thesis tackles, instead, the more general problem of omnidirectional sparse-to-dense depth estimation without any region in which a dense estimation is provided. This problem is tackled by several recent works. Huang et al. [95] exploited an inpainting self-attention network [96] to generate the dense depth map and a dedicated U-Net [97] to preserve depth boundary information. Skip connections [97] are also used in their method to adapt the generic inpainting network to the specific depth prediction task and to better recover fine-grained details. In this thesis, I will propose to handle more general sampling patterns inside a much faster solution (Chapter 4).

Park et al. [98] proposed an interactive Non-Local Spatial Propagation Network (NLSPN) that predicts non-local neighbors for each pixel and then aggregates rele-

vant information using the spatially-varying affinities. To maximize the utility from the sparse source, Huang et al. [99] proposed a Sparse Signal Superdensity (S3) framework, tested for stereo sparse-guidance, for which expands the depth value from sparse cues while estimating the confidence of expanded region. Specifically targeted for guided monocular depth completion, Guizilini et al. [100] introduced Sparse Auxiliary Networks (SANs) to process the sparse signal separately from the dense RGB signal. Their pipeline consists of two parallel branches for the two signals, connected at encoder and decoder level by direct feature fusion.

With a similar decoupled design, Liu et al. [101] advance the pure depth prediction network RectNet [71] to support a SLAM-based reconstruction system where the scattered data are SLAM-SfM features. The method, however, costs 311 GFLOPs for a 512×256 image, while the solution presented in Chapter 4) takes 38.2 GFLOPs for a 1024×512 image.

These recent purely data-driven methods achieve state-of-the-art performance mainly on perspective views and at the cost of a significant computational cost (see Tab. 4.1). In this thesis, I propose, instead, a much leaner indoor solution for panoramic images, showing how the proposed design can cope with a variety of dense sampling patterns and density and can achieve high accuracy even without any fine-tuning after a training on synthetic data (Chapter 4).

2.7.4 3D layout estimation from a single omnidirectional image

Since man-made interiors often follow very strict rules, as discussed in Sec. 2.5, early methods used geometric reasoning to match image features to simple constrained 3D models. In particular, most methods target variants of the Manhattan World model (MWM: horizontal floors and ceilings, vertical walls meeting at right angles) [102], such as the Indoor World model (IWM: MWM with single horizontal ceiling and floor) [103] or the Atlanta World model (AWM: vertical walls with single horizontal ceiling and floor) [45]. In this context, Hedau et al. [42] successfully analyzed the labeling of pixels under a *cuboid* prior, while Lee et al. [47] exploited the IWM to infer 3D structures by analyzing detected corners.

Zhang et al. [72] were among the first to exploit $360°$ captures to overcome the limitation in contextual information present in regular field-of-view (FOV) shots. They proposed a whole-room 3D context model mapping a full-view panorama to a 3D cuboid model of the room through *Orientation Maps* (OM) [47] for the top part and a *geometric context* (GC) analysis for the bottom part [104]. Xu et al. [105] extended this approach to the IWM. Yang et al. [106], instead, proposed to infer a MWM room shape from a collection of partially oriented super-pixel facets

and line segments. A wide variety of follow-ups used similar approaches [1]. The effectiveness of these geometric reasoning methods is, however, heavily dependent on the count and quality of extracted features (e.g., corners, edges or flat patches). More and more research is thus now focusing on data-driven approaches [49].

Recently, several data-driven solutions have shown the capability to infer depth from a single interior image [81, 107, 10]. While these methods have been shown to cope with large amounts of clutter, they cannot produce seamless 3D boundary surfaces in case of self-occlusions, since they can only generate a single 3D position per view ray. For this reason, layout-specific approaches are being actively researched.

As noted by Zou et al. [102], most current data-driven layout reconstruction methods basically share the same pipeline: a MWM pre-processing (e.g., based on the approach of Zhang et al. [72]), the prediction of layout elements in image space and a post-processing for fitting a regularized 3D model to the predicted 2D elements.

Prominent examples are *LayoutNet* [73], which predicts the corner probability map and boundary map directly from a panorama and *HorizonNet* [54], which simplifies the layout as three 1D vectors that encode, at each image column, the positions of floor-wall and ceiling-wall boundaries, and the existence of wall-wall boundary. The 2D layout is then obtained by fitting MWM segments on the estimated corner positions. *DuLaNet* [74], instead, fuses features in the original panoramic view and in a ceiling-plane projection, to output a floor plan probability map, which is transformed to a 2D floor plan by a MWM regularization. Several recent extensions have further improved the performance of the *HorizonNet* baseline. In particular Led^2Net [103], which currently has the best performance in various benchmarks, augments the representation with the rendered depth maps of the panorama horizon, recovering IWM environments. Moreover, several recent methods exploit the correlation of depth, layout, and semantics to improve their joint prediction. In particular, Zeng et al. [85] exploit layout, full depth and semantic information to estimate a layout depth map for fitting an IWM layout. Typically, these methods require heavy pre-processing, such as detection of main Manhattan-world directions from vanishing lines analysis [49, 72, 47] and related image warping, or complex layout post-processing, such as Manhattan-world regularization of detected features [73, 54, 74]. *AtlantaNet* [45] removed these constraints by requiring that input images are roughly aligned with the gravity vector, and predicting the room layout under the less constrained AWM by combining two scaled projections of the spherical image, respectively on the horizontal floor and ceiling planes. Gravity-alignment capture, also exploited in this work, is a very common setup, and, as demonstrated by prior works [10, 107], all the public 3D indoor datasets commonly used for training and testing reconstruction solutions, both synthetic [80, 108] and

real [109, 6], appear to have very small orientation deviations. Even in cases where this assumption is not verified at capture time, several orthogonal solutions exist to gravity-align images at a low cost in a pre-processing step (e.g., [50, 110, 52]), simplifying the practical application of gravity-oriented methods.

The restriction to very constraining priors (MWM, IWM, or AWM) makes it possible to employ various forms of projections and simplifications, but limits the class of models that can be inferred and makes the inference less robust in case of major occlusions, which require full 3D reasoning to be resolved [111].

Differently from prior solutions, in this thesis, I will show how to infer a watertight 3D mesh from the panoramic image using a 3D approach (Chapter 5). This solution has been the subject of recent data-driven 3D object reconstruction methods [112, 113, 114] but has not been applied to the interior reconstruction realm, which bears very significant differences with respect to object reconstruction. In particular, object reconstruction methods assume an external perspective view of an uncluttered object, while this thesis target an interior full panoramic view of a cluttered environment. We must thus learn to separate clutter from structure and we cannot rely on simple projections to associate multi-scale image features to vertices, but we must learn to select local and non-local features depending on context. Moreover, we must take into account the peculiar shape of typical indoor structures, made of few large connected surface components. This has led to novel contributions in terms of network structure and loss functions (Chapter 5).

2.8 Available large data collections

Data-driven solutions must exploit large collections of data to learn hidden relations as well as to test the effectiveness of reconstruction. A remarkable number of freely available datasets containing indoor scenes have been published in the recent years for the purpose of comparing and/or training learning-based solutions. Many of them have been acquired with RGB-D scanners, due to the flexibility and low cost of this solution, being collected on these detailed established surveys [21, 1, 22] of which just the most used for benchmarking and also others recently published are mentioned as example in this Chapter.

In the following, I summarize the characteristics of major publicly available panoramic datasets. Tab. 2.1 and Tab. 2.2 show a simplified information of each one, while Tab. 2.3 lists all the published datasets used in this thesis.

- **Matterport3D Dataset** [6]: A large-scale dataset which provides 10,800 panoramic views RGB-D images from 194,400 RGB-D images of 90 building-

scale scenes. Annotations are provided with surface reconstructions, camera poses, and 2D and 3D semantic segmentations.

- **Stanford2D-3D-S Dataset** [109]: The dataset is collected in 6 large-scale indoor areas that originate from 3 different buildings of mainly educational and office use, captured by using the same Matterport system of the Matterport3D dataset [1]. The dataset contains over 70,000 RGB images, along with the corresponding depths, surface normals, semantic annotations, global XYZ images (all in forms of both regular and 360° equirectangular images) as well as camera information. It also includes registered raw and semantically annotated 3D meshes and point clouds.

- **360D Database** [80]: This database offers a synthetic benchmark. It contains 35,977 panoramas rendered by path-tracing scenes from two synthetic datasets (*SunCG* and *SceneNet*) and two realistic datasets (*Stanford2D3D* and *Matterport3D*). In this case, we adopted the splitting provided by Zioulis et al. [71]. The original *SunCG* data is no longer available for downloading due to legal reasons.

- **Structured3D Dataset** [108]: A large-scale photo-realistic synthetic dataset, containing $3.5K$ house designs created by professional designers with a variety of ground truth 3D structure annotations, including $21,000$ photo-realistic full-panoramic (i.e., 1024×512 equirectangular format) indoor scenes.

- **CRS4/ViC Research Datasets** [115]: Datasets that contain high-resolution equirectangualar panoramas convering 360×180 full-view for a variety of real-world cluttered indoor scenes. The scenes include multi-room environments, sloped ceilings, walls not aligned on rectangular coordinate system, and more challenge features. Also, the height of the camera is provided, being 170 cm from most datasets.

- **SUN360 Database** [116]: This dataset contains 80 categories and 67,583 panoramas, all of which have a resolution of 9104×4552 pixels and cover a full $360^\circ \times 180^\circ$ visual angle using equirectangular projection. To build the core of the dataset, the authors downloaded a massive amount of high-resolution panorama images from the Internet, and manually labeled them into different place categories.

- **PanoContext Dataset** [72]: This dataset contains 700 full-view panoramas for home environments from SUN360 database [116], including 418 bedrooms and 282 living rooms. Being the data manually annotated. They provide a tool which renders panoramic images and annotates several objects and its

3D bounding box, being all of the objects standing on the ground, sitting on another object, or attaching to a wall (i.e., none are floating).

- **Zhang et al.** [93] **Dataset**: Introduced a new dataset based on the large-scale Matterport3D [6], which provides 105k RGB-D images aligned with dense depth images computed from multi-view reconstructions in 72 real-world environments. This dataset contains 117,516 RGB-D images with rendered completions, which we split into a training set with 105,432 images and a test set with 12,084 images.

- **Zillow Indoor Dataset (ZInD)** [93]: A large-scale indoor dataset with 71,474 panoramas from 1,524 real unfurnished homes. ZInD provides annotations of 3D room layouts, 2D and 3D floor plans, panorama location in the floor plan, and locations of windows and doors. One particular characteristic of this dataset is about the room layout data which follows a real-world distribution not being just, as the mostly publicly available datasets, cuboid or Manhattan layouts.

- **Replica Dataset** [117]: A dataset of 18 highly photo-realistic 3D indoor scene reconstructions at room and building scale. Each scene consists of a dense mesh, high-resolution high dynamic-range (HDR) textures, per-primitive semantic class and instance information, and planar mirror and glass reflectors. Those scenes can be rendered within AI Habitat [118], specially on the AI Habitat Sim [119] which is a high-performance physics-enabled 3D simulator that achieves several thousand frames per second (FPS).

- **PNVS Dataset** [120]: A large-scale photo-realistic dataset upon Structured3D dataset [108]. It is a stereo dataset that provides two type of camera translations between a source camera position and its target camera position, getting an easy set and hard set. The easy set contains target panoramas with small camera translation between 0.2-0.3 meters, including 13,080 training images and 1,791 testing images. The hard set contains target panoramas with large camera translations between 1.0-2.0 meters, including 17,661 training images and 2,279 testing images.

- **Rey-Area et al.** [28] **Database**: A large-scale database based on Matterport3D Dataset [6] and Replica Dataset [117]. From Matterport3D dataset, they estimated the poses for the real skybox images relative to the mesh using 360° structure-from-motion [121], applying to a mixture of real and rendered skybox images at known camera positions. Then, using the estimated camera poses and the provided scene mesh, rendered ground-truth depth maps with pixel accuracy. Besides, they rendered 10 images and its registered

depth maps for each of the 13 rooms from Replica Dataset [117], generating random poses using the Replica360 renderer [122]. Thus, they provide two datasets, a Matterport3D 360° dataset that consists of 9,684 RGB-D pairs with a resolution 2048×1024; and a Replica 360° 2K/4K that consists of 130 RGB-D pairs rendered at 2048×1024 and 4096×2048.

- **AtlantaLayout Dataset** [45]: Contains rooms with curved walls or meeting at non-right angles, dubbed Atlanta World (AW).
- **Pano3DLayout Dataset** [11]: A synthetic dataset that contains 106 more complex environments, not included in previous benchmarks, such as, for example, scenes with sloped or stepped ceilings, domes, and interconnections of different rooms.
- **Indoor3Dmapping Dataset** [123]: A dataset from a real LiDAR RGB-D acquisition (i.e., mobile device with 2 Velodyne VLP-16 and a registered Garmin spherical camera) and a ground truth dense depth acquisition through a $FaroFocus3DX330TLS$. It is acquired in a multi-floor and multi-room environment, providing equirectangular image projections aligned with dense ground truth and sparse depth maps. Each sparse scan produces about 16% of pixels with valid depth.

In Chapter 3, depth estimation from an omnidirectional image, I will report results obtained on four publicly available datasets [109, 6, 80, 108] to facilitate comparison. These benchmarks were also adopted by the recent state-of-the-art works [61, 71, 81] comparable with the method discussed in this thesis. Additionally, I present the performance of the introduced method on the recent Structured3D synthetic dataset [108] to support ablation and gravity-alignment robustness studies (i.e., analyzing the performance by removing certain components to understand the contribution of the component to the overall system).

In Chapter 4, sparse-to-dense estimation from RGB and sparse intput, in order to cover a large variety of use cases, I created a novel dataset leveraging on synthetic data generated by sampling the large-scale Structured3D [108] photo-realistic synthetic dataset. The main advantage of such a synthetic dataset is that it provides a fully accurate dense ground-truth for color and depth, which is not available with other common large-scale datasets, such as Matterport3D [6] or Stanford2D-3D-S [109], whose completeness, even if based on multi-view, is still limited by visibility and sensor limitations. It is also possible, from synthetic data, to simulate a variety of sensors.

In Chapter 5, layout estimation from a single panoramic image, in order to provide a comparison with state-of-the-art work, I analyze results on standard publicly

Table 2.1: ***Publicly available panoramic datasets.*** *Each dataset/database has a particular visual data (i.e., at least containing purely visual data); being a real/synthetic source (Source column); capturing by a camera, manually modeling or rendering from other dataset (Camera column); having a number of samples (#Images column); what layout distribution (when it has layouts) (Distribution column); and its annotated information (Annotations column).*

Dataset	Source	Camera	# Images	Distribution	Annotations
Matterport3D [6]	Real	Matterport Pro 3D	10,800	Real-World	Surface reconstructions, 2D/3D semantics, depths, camera poses
Stanford2D-3D-S [109]	Real	Matterport Pro 3D	70,000	-	Surface reconstructions, depths, surface normals, 2D/3D semantics, camera poses
Structured3D [108]	Synth	Manual Modeling	21,000	Cuboids, MWM	Two light conditions, three clutters setups, layouts, depths, normals, albedo, instances, semantics, camera poses
CRS4/ViC [115]	Real	Tripod	191	cuboids, MWM, non-MWM	Layouts
PanoContext [72]	Real	Renderings from [116]	700	Real-World	Layout
Zhang et al. [93]	Real	Renderings from [6]	10,800	-	Depth
ZInD [93]	Real	Ricoh Theta (V and Z1)	71,474	Real-world	Layout, 2D/3D floor plans, windows/doors poses, camera poses
PNVS [120]	Synth	Renderings from [80]	34811	Cuboids, MWM	Stereo images, source depths, source layouts, stereo camera poses
AtlantaLayout [45]	-	-	-	Real-World	AW layouts
Pano3DLayout [11]	-	-	106	Syth	Depth, non-MWM Layouts
Indoor3Dmapping [123]	Real	Mobile device (2 Velodyne VLP-16 and Garmin spherical camera)	-	Real-World	Sparse depths, depths
360D [80]	Real/Synth	Renderings from [6, 109]	35,977	Real-World	Depth normals
SUN360 [116]	Real	Manual Modeling	67,583	Real-World	Layout
Rey-Area et al. [28]	Real/Synth	Renderings from [6, 117]	9,684/130	Real-World	Depth, camera poses

Table 2.2: *Publicly available scene datasets.* *These datasets provide scene descriptions, from which a rendering framework can generate the information required, for instance, panoramic image and its registered depth.*

Scene Dataset	Source	# Scenes	How to render
Replica [117]	Photo-realistic	18	AI Habitat Sim [119], Replica360 renderer [122]

Table 2.3: *Publicly available panoramic used in this thesis.* *I also mention what split of the dataset is consider in this work.*

Dataset	Splitting
Matterport3D [6]	Wang et al. [81]
Stanford2D-3D-S [109]	Wang et al. [81]
Structured3D [108]	Zheng et al. [108]
Zhang et al. [93]	Pintore et al. [45]
Pano3DLayout [11]	Pintore et al. [11]
Indoor3Dmapping [123]	Pintore et al. [11]
360D [80]	Zioulis et al. [71]

available datasets [72, 109, 49, 108], containing thousands of indoor scenes and commonly adopted for benchmarking 3D layout recovery [54, 45, 107, 103, 85]. However, due to the focus of prior works, these benchmarks mostly consisted of MWM structures [102]. Since the method introduced in this dissertation is more general, the testing set has been extended with the publicly available *AtlantaLayout* [45] dataset, which also contains rooms with curved walls or meeting at non-right angles. In addition, we prepared a specific dataset, called *Pano3DLayout*, containing more complex environments, not included in previous benchmarks, such as, for example, scenes with sloped or stepped ceilings, domes, and interconnections of different rooms.

2.9 Wrap-up

Considering all of the above, I have focused the research on deep learning solutions based on panoramic image analysis for the reconstruction and representation of indoor environments, either using it standalone or eventually combining it with sparse geometric information. The main challenge is that we have to reconstruct such a model from very partial input, be it images alone or with sparse depth measurements, with lots of noise, holes, and clutter. Thus, this thesis is focused

on finding clever ways to rapidly inferring geometry or layout from corrupted and minimal input (i.e., one image per room), without requiring users to do more than a single acquisition or to manually edit models. Specially, this thesis is focused on depth estimation from a single panoramic image (Chapter 3), sparse-to-dense estimationfrom a single panoramic image and its registered sparse depth map (Chapter 4) , and layout estimation from a single panoramic image (Chapter 5). All of these solutions will be designed as end-to-end networks, converting equirectangular input to the desired output. All the networks will be trained through a supervision learning process, exploiting large amounts of data on which the ground truth desired output is known.

2.10 Bibliographic notes

Several portions of this chapter have been taken from my contribution published in EVOCATION project deliverables [14, 15, 16], that I have later expanded in this thesis. These portion include the definitions of the problems and references to benchmark datasets. The survey of related work is adapted from the related work sections of the articles that I have publised in journals [12, 11] and conference proceedings [10].

Chapter 3

Deep estimation of dense depth information of an interior environment from a single omnidirectional image

As a first contribution towards reconstructing indoor information from purely visual data, I introduce in this chapter a novel deep neural network to estimate a depth map from a single monocular indoor panorama. The network directly works on the equirectangular projection, exploiting the properties of indoor 360-degree images. Starting from the fact that gravity plays an important role in the design and construction of man-made indoor scenes, we propose a compact representation of the scene into vertical slices of the sphere, and we exploit long- and short-term relationships among slices to recover the equirectangular depth map. Our design makes it possible to maintain high-resolution information in the extracted features even with a deep network. The experimental results demonstrate that our method outperforms current state-of-the-art solutions in prediction accuracy, particularly for real-world data.

3.1 Introduction

Understanding the 3D layout of an indoor scene from images is a crucial task in many domains [124, 1, 8]. Fast depth estimation from single images is a fundamental

sub-problem, as associating metric information to visual data is paramount for a variety of applications, including mobile Augmented Reality platforms, indoor mapping, autonomous navigation, 3D reconstruction, and scene understanding.

Since estimation of depth from single images is inherently ambiguous, all solutions must rely on prior information to guide reconstruction towards plausible architectural shapes that fit the input. In this context, we have recently seen an extraordinary development of data-driven methods that learn these priors from example data.

Early approaches were designed for a camera with a conventional limited field-of-view (FoV) (e.g., FCRN [61]). In recent years, however, 360° capture has emerged as a very appealing solution, since it provides the quickest and most complete single-image coverage and is supported by a wide variety of professional and consumer capture devices that make acquisition fast and cost-effective [125]. Since adapting monocular depth estimation models designed for traditional images to 360° depth estimation has been shown to produce sub-optimal results [71], specific 360° solutions have been recently introduced. In this context, many recent works [77, 71, 126] have adapted perspective depth estimation methods to omnidirectional imagery by proposing various types of distortion-aware convolution filters. However, few of them have explored the large-FoV nature provided by 360° images, which can provide, in one shot, the full-geometric context of an indoor scene [72].

In this work, we introduce a novel deep neural network solution, called *SliceNet*, which predicts the depth map of an indoor 360° image leveraging the characteristics of a gravity-aligned equirectangular projection of an interior scene. Since gravity plays an important role in the design and construction of interior environments, world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. Our network design starts from the assumption that capture of the scene through an equirectangular image is aligned to the gravity vector (i.e., camera is placed on an horizontal-ground plane), too, and, thus, it is rational to assume that gravity-aligned processing of images can directly exploit gravity-aligned world-space features [52]. In our network, an input equirectangular image is partitioned into vertical *slices* by performing a contractive encoding to reduce the input tensor only along the vertical direction, resulting in a compact and flattened sequence of slices made of a set of features. To preserve global information, we perform slicing over four different resolution levels, concatenating the result at the end (Sec. 3.2). This sequential representation enables the use of a convolutional long short-term memory (LSTM) network [127] to recover, with low computational overhead, long- and short-term spatial relationships among slices. Decoding proceeds symmetrically with respect to encoding, thereby increasing only

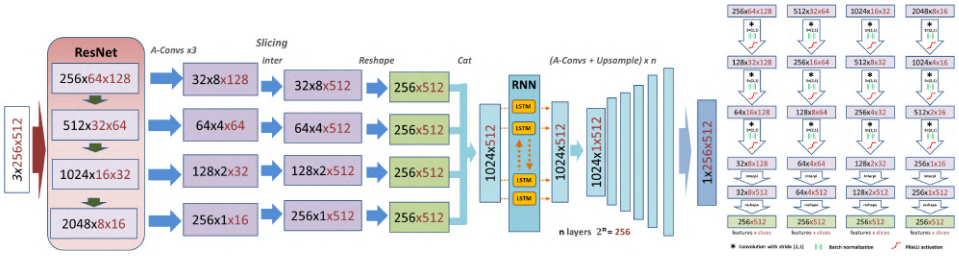
the vertical resolution of the feature map, until the target resolution is reached (Fig. 3.1a).

Our contributions are summarized as follows:

- We introduce a slice-based representation of an omnidirectional image that directly exploits the characteristics of the equirectangular projection of an indoor scene, without the need for distortion-aware convolution and transformation [71, 81], multi-branch architectures [81, 84] or additional information and priors [84]. Our representation based on vertical slices is very robust, as demonstrated by the important advantage in performance achieved in real-world cases (e.g., Stanford2D3D [109] and Matterport3D [6]), where a large area around the poles of the panorama is not acquired by the instrument (see Sec. 3.4.3 for details).
- We specialize and refine feature flattening, which has proven to be effective to regress one-dimensional tensors [54], for bi-dimensional depth encoding. In particular, we introduce an asymmetric contraction of the input tensor based on vertical slicing at different resolutions, so that the resulting feature map is flattened along a single direction (in our case, the sphere horizon), and we merge slices at different resolutions, so as to exploit deeper levels with larger receptive fields to capture global information, while at the same time exploiting higher resolution layers to preserve high-frequency details (Sec. 3.2). Our ablation study (Sec. 3.4.4) demonstrates the advantages of our approach.
- We introduce, for depth estimation from a single image, a LSTM multi-layer module to effectively recover long and short term spatial relationships between slices in the presence of a large number of features per slice due to the concatenation of multiscale representations. With this architectural choice, the decoder is simple and follows the same multi-layer scheme of the encoder with a vertical upsampling rather than a vertical reduction. We do not need, in particular, the chaining of up-projection blocks [62], making it easier to scale the method to different input resolutions. The ablation study (Sec. 3.4.4) confirms the benefits of the method by comparing different decoder configurations with or without LSTM and chaining up-projection blocks.

We tested our network on both synthetic and real datasets [109, 6, 71, 80, 108]. Our experimental results (Sec. 3.4) demonstrate that our method outperforms current state-of-the-art methods [61, 71, 81] in prediction accuracy, especially when working on real-world scenes. Exploiting gravity alignment leads to an efficient

network structure, without significant limitations on the applicability of the approach. As mentioned, gravity-aligned capture is a very common setup, and, as determined by our tests, Sec. 3.4.4, all the public 3D indoor datasets commonly used for training and testing reconstruction solutions, both synthetic [80, 108] and real [109, 6], appear to have very small orientation deviations. Even in cases where this assumption is not verified at capture time, several orthogonal solutions exist to gravity-rectify images in a pre-processing step (e.g., [50, 51, 52]), simplifying the practical application of gravity-oriented methods. Moreover, as demonstrated by our ablation study (Sec. 3.4.4), our method is robust to small variations of the inclination.



(a) (b)

Figure 3.1: ***Network architecture.*** *Our architecture is scalable with respect to the input resolution. In Fig. 3.1a, to simplify comparison with other methods, we show an example with an input image having size* $3 \times 256 \times 512$*. A* ResNet50 *encoder [62] extracts four layers at different resolutions. From each resolution layer we obtain a sliced feature map of* 256×512 *(purple blocks in Fig 3.1a, details in Fig. 3.1b). By concatenating the resulting four layers we obtain a single bottleneck with 512 slices and 1024 features, which is refined using a RNN scheme (cyan blocks). The decoder proceeds symmetrically, producing a depth map at the same input image resolution.*

3.2 Network architecture

Almost all CNNs for this task follow an encoder-decoder architecture [61]. Such a structure contains a contractive encoding part that progressively decreases the input image resolution through a series of convolutions and pooling operations, giving higher-level neurons large receptive fields, thus capturing more global information. As the target depth map is a high resolution image, the decoder regresses to the desired output by upscaling this representation. Our work introduces several important novelties in this structure.

Figure 3.1a illustrates the structure of our network for a 256×512 input. Note that our architecture is scalable with respect to the input resolution. In Sec.3.4 we provide results with the same input sizes adopted by recent state-of-the-art

methods [61, 71, 81], including 512×1024 resolution.

The first part of our network is devoted to extracting relevant low/mid/high-level features from the input tensor. To do that, we exploit *ResNet-50*, a deep neural network that supports, through a residual learning framework, the training of very deep networks without degradation problems [62]. Differently from other approaches [61, 71, 81], we exploit not only the deepest layer of ResNet, but the last four layers, processing them in parallel, in order to build a multi-resolution spatial representation, discussed in detail below. Following our gravity-aligned model, we recover from these 4 layers (Fig. 3.1a, red), 4 representative slice layers (Fig. 3.1a, green), having all the same size of 256×512 (i.e., 256 features for 512 slices). Figure 3.1b illustrates how we produce the sliced representation from the ResNet layer. First, we reduce the vertical dimension by a factor of 8 through an asymmetric convolution module with stride $(2,1)$ (*A-Conv*), applied 3 times, contains a 2D convolution, a batch normalization module and a Parametric Rectified Linear Unit [128] $\mathrm{PReLU}(x) := \max(0,x) + a * \min(0,x)$, where a is the coefficient of leakage learned during training. We selected PReLU instead of commonly adopted ReLU and Leaky-ReLU to minimize the vanishing gradient problems that are common in depth estimation. This kind of adaptive activation leads to convergence even on datasets with very different characteristics (e.g., real-world acquisition with missing parts or synthetic rendering whih high levels of noise). Sliced encoding is then completed by horizontally interpolating each feature map to have the same number of slices (i.e., 512), and by vertically reshaping the features to the target size (i.e., 256).

Finally, the four layers are concatenated in a single sequence (i.e., 1024×512), obtaining 1024 features for each of the 512 vertical slices of the input sphere. In this way, we obtain a bottleneck representation that exploits deeper levels with larger receptive fields to capture global information, and higher resolution layers to preserve high-frequency details.

It should be noted that both indoor scenes and equirectangular projections have particular properties that we exploit in our design. For example, vertical lines are very common in the scene, and are practically not deformed in the projection while the horizontal ones are more so. Because of these characteristics, we expect each slice sequence along the dominant vertical direction be related to the others by both short-term and long-term spatial dependencies [53, 54, 45]. Thus, we start our decoder by feeding such a sequence to a RNN multi-layer block [127]. In our case, we use a bi-directional LSTM (long-short term memory) having 512 hidden layers, which outputs a timestep of size 2×512 for each of the 512 slices, so that the final output is a feature map having the same size of the RNN block input,

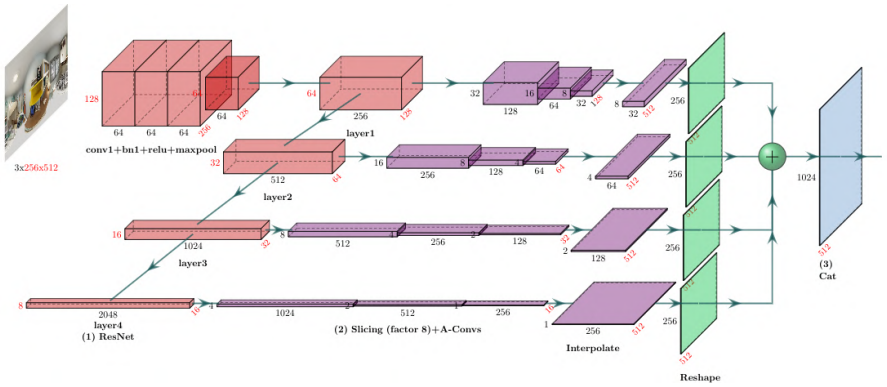
i.e., 1024×512. Once reshaped to $1024 \times 1 \times 512$, this flattened representation can be upsampled to the desidered output size (i.e., $1 \times 256 \times 512$) by following steps symmetrical to those used for encoding reduction. Actually, thanks to the flattened encoding and RNN features refinement, our network does not require the chaining of skipping up-projection blocks for upsampling, such as FCRN [61], also common in other recent works [81]. Our decoder, instead, consists of n layers, where for each layer we perform an upsampling of a factor of two of the height only, followed by a convolutional module *A-Conv* identical to that of the reduction phase (2D convolution and PReLU activation), but with stride (1,1). In the example of Fig. 3.1a, the decoder consists of $n = 8$ layers, in order to achieve the targeted vertical resolution (i.e., $2^n = 256$), and the resulting map is a tensor of $1 \times 256 \times 512$ representing the depth prediction for each of the input pixels. We also tested different upsampling modules adapted to our data encoding, (e.g., FCRN [61]) but experiencing lower performance, given our particular slice-based model. Numerical details are exposed in the ablation study in Sec. 3.4.4.

3.2.1 Detailed network architecture description

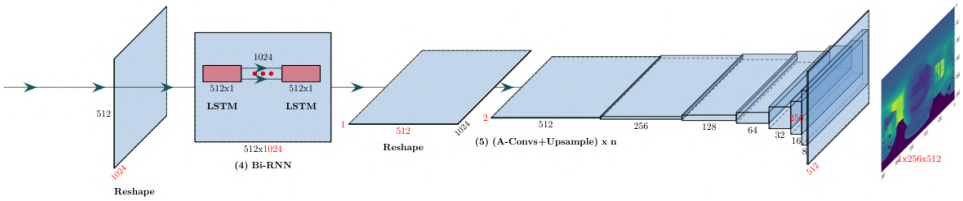
Fig. 3.2 provides details on all the individual network components and is aimed to complement the general description provided in the paper. Our deep convolutional neural network (CNN) architecture takes as input an equirectangular RGB image and outputs a registered depth image at the same resolution of the input. The detailed structure of the network is illustrated in Fig. 3.2. The network uses an encoder/decoder structure. The encoder is presented in Fig. 3.2a, while the decoder is presented in Fig. 3.2b.

The first 8 layers of the network consist of a standard ResNet encoder (Fig. 3.2a). The results presented in the paper are obtained with a ResNet50, but we verified that very good performances can also be obtained and with ResNet18 and ResNet34, with a considerable increase in terms of speed. The last 4 levels of the encoder are sliced, keeping the horizontal dimension unchanged and compressing the vertical one. This way, we accumulate a series of features associated with each element of the horizontal dimension (i.e., a slice). In order to merge the features, coming from different resolution levels and associated to the same slice, we interpolate the 4 maps so that they have the same horizontal dimension (i.e., 512). We then reshape and concatenate the 4 maps so as to obtain a single-sequential bottleneck (i.e., 1024×512).

The decoder (Fig. 3.2b) exploits a bi-directional LSTM with 512 hidden layers, which outputs a time-step of size 2×512 for each of the 512 slices. So, that the final output of this block is a feature map having the same size of the RNN block input,



(a) *Encoder*



(b) *Decoder*

Figure 3.2: ***Detailed illustration of the SliceNet architecture.*** *This illustration complements the architectural view provided in the paper. The network uses an encoder/decoder structure. The encoder is presented in Fig. 3.2a, while the decoder is presented in Fig. 3.2b. The last 4 levels of the encoder are sliced, keeping the horizontal dimension unchanged and compressing the vertical one (Fig. 3.2a). From the resulting sliced sequence* $(1024 \times 1 \times 512)$*, we recover long and short term information through a LSTM module (Fig. 3.2b). The final depth map is recovered by following steps symmetrical to those used for encoding reduction.*

i.e., 1024×512. Once reshaped to $1024 \times 1 \times 512$, this flattened representation is upsampled to the desired output size (i.e., $1 \times 256 \times 512$) by following steps symmetrical to those used for encoding reduction.

3.3 Loss function and training strategy

Similarly to other recent state-of-the-art solutions (e.g., BiFuse [81]), we build our objective function on top of the robust *Adaptive Reverse Huber Loss (BerHu)* [63]:

$$B_c(e) := \begin{cases} |e| & |e| \leq c \\ \frac{e^2+c^2}{2c} & |e| > c \end{cases} \tag{3.1}$$

where e is the error term and the parameter c determines where to switch from L1 to L2. In order to set the c value adaptively, we follow the same approach of Laina

et al. [61], so that c is set, in every gradient step, to 20% of the maximal error of the current batch. When applied to the depth maps, $e = D_{ij} - D^*_{ij}$ at each pixel (i, j), where D and D^* are, respectively, the predicted and the ground-truth depth maps. Since one of the typical problems encountered in predicting depths using convolutional networks is the loss of small details [61, 71], which is particularly noticeable when dealing with higher resolution images, we introduce an additional term by applying *BerHu* also to the gradient components obtained by convolving the maps with Sobel filters of width 3 to approximate the horizontal derivatives $\nabla_x D$ and $\nabla_x D^*$ and the vertical ones $\nabla_y D$ and $\nabla_y D^*$. Consequently, the full loss function L that guides our training is:

$$\begin{aligned} L_{c_1,c_2}(D, D^*) = & B_{c_1}(D - D^*) + \\ & B_{c_2}(\nabla_x D - \nabla_x D^*) + \\ & B_{c_2}(\nabla_y D - \nabla_y D^*) \end{aligned} \tag{3.2}$$

With a little abuse of notation, we intend the application of the function to the map as the sum of results on each individual pixel. The parameter c that determines the shape of each function B_c is computed at each batch independently for the depth term (c_1) and the two gradient terms (i.e., c_2 is independent from c_1 and shared for the x and y gradient terms). Moreover, in order to gracefully handle large areas with missing samples common in real-world data (e.g., the upper and lower parts of the hemisphere are not sampled by the instrument, as in Matterport [6]), we take the common approach [71] of ignoring errors on missing areas with a per-pixel binary mask.

In all experiments, we obtain the best performance when training with the loss in Eq. 3.2, even compared to other robust solutions [71], experiencing a noticeable difference when training and comparing with real-world datasets [109, 6], which contain noticeable amounts of noise. The gradient-based component improves image sharpening, as shown in the comparison presented in Sec. 3.4.4 and Fig. 3.6.

3.4 Implementation and results

Our approach is implemented using PyTorch 1.5.1 and has been tested on a large variety of indoor scenes. Source code and models will be made available to the public.

3.4.1 Datasets

In this paper, we report results obtained on four publicly available datasets [109, 6, 80, 108] to facilitate comparison. These benchmarks were also adopted by the

Table 3.1: ***Quantitative performance on real and virtual world datasets.*** *We show our performance evaluated on standard metrics and compared to the recent state-of-the-art approaches. In all cases our approach outperforms the competition.*

Dataset	Method	MRE	MAE	RMSE	RMSE log	δ_1	δ_2	δ_3
Stanford2D3D	FCRN [61]	0.1837	0.3428	0.5774	0.1100	0.7230	0.9207	0.9731
	OmniDepth [71]	0.1996	0.3743	0.6152	0.1212	0.6877	0.8891	0.9578
	BiFuse [81]	0.1209	0.2343	0.4142	0.0787	0.8660	0.9580	0.9860
	Our	**0.0744**	**0.1048**	**0.1214**	**0.0207**	**0.9031**	**0.9723**	**0.9894**
Matterport3D	FCRN [61]	0.2409	0.4008	0.6704	0.1244	0.7703	0.9174	0.9617
	OmniDepth [71]	0.2901	0.4838	0.7643	0.1450	0.6830	0.8794	0.9429
	BiFuse [81]	0.2048	0.3470	0.6259	0.1134	0.8452	0.9319	0.9632
	Our	**0.1764**	**0.3296**	**0.6133**	**0.1045**	**0.8716**	**0.9483**	**0.9716**
360D	FCRN [61]	0.0699	0.1381	0.2833	0.0473	0.9532	0.9905	0.9966
	OmniDepth [71]	0.0931	0.1706	0.3171	0.0725	0.9092	0.9702	0.9851
	BiFuse [81]	0.0615	0.1143	0.2440	0.0428	0.9699	0.9927	**0.9969**
	Our	**0.0467**	**0.1134**	**0.1323**	**0.0212**	**0.9788**	**0.9952**	**0.9969**

recent state-of-the-art works [61, 71, 81] comparable with our method. *Matterport3D* [6] and *Stanford2D-3D-S* [109] act as real-world examples. Similarly to Wang et al. [81], we used their official splitting and a resolution of 512×1024. *360D* [80] offers instead a synthetic benchmark. It contains 35,977 panoramas rendered by path-tracing scenes from two synthetic datasets (*SunCG* and *SceneNet*) and two realistic datasets (*Stanford2D3D* and *Matterport3D*). In this case, we adopted the splitting provided by Zioulis et al. [71] and a resolution of 256×512, which is a common baseline for many approaches [61, 71, 81]. At the time of this writing, the original *SunCG* data is no longer available for downloading due to legal reasons. Additionally, we present our performance on the recent *Structured3D* synthetic dataset [108] to support ablation and gravity-alignment robustness studies (Sec. 3.4.4).

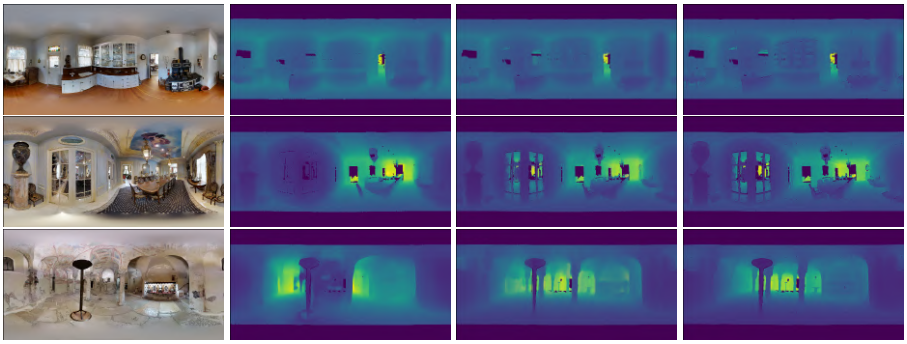
3.4.2 Experimental setup and timing performance

We trained the network using the Adam optimizer [129] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, on four NVIDIA RTX 2080Ti GPUs (11GB VRAM) with a batch size of 8 and a learning rate of 0.0001 for real-world data and 0.0003 for synthetic data. We adopt the specific panoramic data augmentation proposed by Sun et al. [54]. With the given setup, starting from default weight initialization, the best valid epoch was around 60 for real-world data and 90 for synthetic data. The average training speed is about $55ms/img$ for a 256×512 input image and $117ms/img$ for a 512×1024 image. Single-GPU inference time is $74ms$ (13 fps) for a 1024×512 image and $44ms$ (23 fps) for a 512×256 input image, showing that our method can be integrated in

interactive settings. It is important to note, in terms of computational complexity, that the best competing method, BiFuse [81], has 253M parameters and multi-branching, while our much simpler architecture has only 75M parameters, also leading to reduced inference time (e.g., $74ms$ vs. $616ms$ for a 1024×512 image). Additional details are provided in Sec. 3.4.4.

3.4.3 Quantitative and qualitative evaluation

We evaluated our method with the same error metrics used in prior depth estimation works [61, 71, 81]: mean absolute error (MAE), mean relative error (MRE), root mean square error of linear measures (RMSE), root mean square error of log measures (RMSE log scale invariant), and three relative accuracy measures δ_1, δ_2 and δ_3, defined, for an accuracy δ_n, as the fraction of pixels where the relative error is within a threshold of 1.25^n. Tab. 3.1 illustrates our quantitative results, in comparison with the most recent state-of-the-art works for which source code or numerical performance on the same data is available and using consistent training and testing setups. We compare with OmniDepth [71] (i.e., *RectNet*), BiFuse [81], as well as FCRN [61], which is the baseline of many current approaches (e.g., BiFuse [81]). Our method outperforms the others in terms of accuracy for all metrics, more markedly in cases of real data (Matterport3D and Stanford2D-3D-S in Tab. 3.1). In the case of synthetic data (360D in Tab. 3.1), our method also improves over other approaches, although here differences are smaller, due to the fact that virtual renderings guarantee uniform 2D sampling and very few discontinuities [71] (except, for example, for occlusions), to the benefit of methods based on symmetrical 2D reduction and expansion. Figures 3.3, 3.4, and 3.5 illustrate qualitative results on real and synthetic data. Figure 3.3 shows our prediction (Fig. 3.3c) on real-world RGB images (Fig. 3.3a) taken from Matterport3D[6], compared to ground truth (Fig. 3.3d) and BiFuse [81], for which the pre-trained model on Matterport3D was available. As we can see, our method finds a more accurate depth even in areas with smaller and repetitive structural details (first row of Fig. 3.3), in the case of large environments (second row of Fig. 3.3), and also for non-Manhattan-World but regular environments, as in the case of arched vaults (third row of Fig. 3.3). Figure 3.4 shows qualitative results on *360D* synthetic data [80], compared with the dataset creators' method [71]. The highlighted details illustrate qualitative differences. In particular, our method can infer a detailed reconstruction for typical man-made objects (Fig. 3.4, first row), even if they are far away (Fig. 3.4, second and third rows),



(a) *RGB* **(b)** *BiFuse [81]* **(c)** *Our* **(d)** *Ground truth*

Figure 3.3: ***Qualitative comparison on real-world datasets.*** *Depth maps are inferred from real-world captured RGB data (Matterport3D [6]). The first column is the input RGB image (Fig. 3.3a), the second one is the depth estimated by BiFuse [81] (Fig. 3.3b), the third one is the depth estimated by our method (Fig. 3.3c), and the fourth one is the ground-truth depth acquired by the instrument (Fig. 3.3d). Black pixels are missing samples in the ground-truth depth. All methods have been compared using the same original datasets and setting, without any further pre-process or alignment step.*

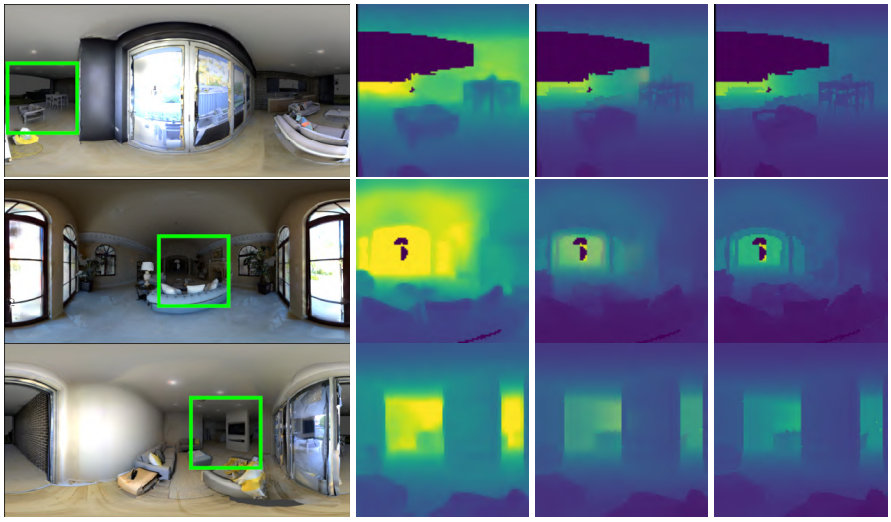
3.4.4 Ablation and gravity alignment study

We present in this section the model ablation and computational costs (Tab. 3.2), and specific experiments showing the effectiveness of using the gravity-alignment prior (Tab. 3.3).

Table 3.2: ***Ablation study.*** *The ablation study, performed on the* Structured3D *dataset[108], demonstrates how our proposed designs improve the accuracy of prediction. Results show only comparable-stable cases that actually increase it. We show in the last row the full architecture setup. PReLU activation provides identical benefits for each configuration in terms of convergence.*

ResNet-50	Slicing	LSTM	Asym	Grad	Params	MRE	MAE	RMSE	RMSE log	δ_1	δ_2	δ_3
23.5M	24.8M (last 1)	-	6.3M	No	54.6M	0.4712	0.5520	0.1596	0.0341	0.6845	0.8684	0.8824
23.5M	33M (last 4)	-	6.3M	No	62.8M	0.2990	0.5014	0.0775	0.0154	0.7045	0.8784	0.9124
23.5M	24.8M (last 1)	12.5M	6.3M	No	67.1M	0.2988	0.4814	0.0750	0.0149	0.7702	0.8892	0.9222
23.5M	33M (last 4)	12.5M	6.3M	No	75.3M	0.0147	0.1223	0.0558	0.0102	0.8854	0.9376	0.9492
23.5M	**33M (last 4)**	**12.5M**	**6.3M**	**Yes**	**75.3M**	**0.0147**	**0.1180**	**0.0549**	**0.0109**	**0.9085**	**0.9451**	**0.9502**

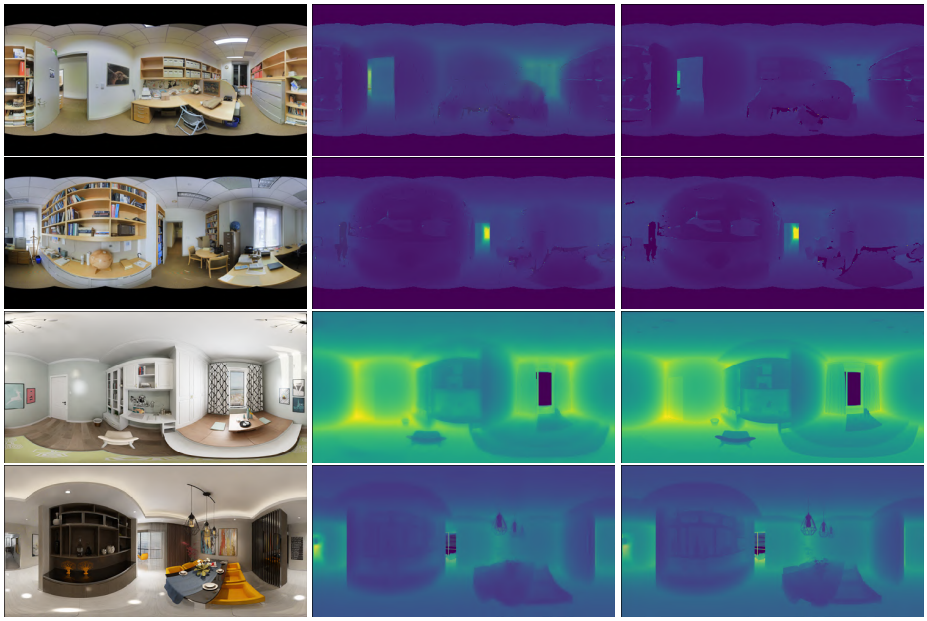
Ablation study and complexity. Our ablation experiments are presented in Tab. 3.2. To test the key components of the approach, we use results obtained with Structured3D [108], a synthetic dataset containing over 21,000 rendered rooms, that include, among other features, uniformly sampled color and very accurate depth panoramas. This very recent dataset has not yet been adopted by comparable works (Sec. 3.4.3), but provides an additional valuable benchmark for our method.



(a) *RGB* **(b)** *OmniDepth* **(c)** *Our* **(d)** *Ground truth*

Figure 3.4: ***Qualitative comparison on synthetic datasets.*** *Depth maps are inferred from synthetic data (360D [80]). We show in the first column the rendered RGB image (Fig. 3.3a), the estimated depth by OmniDepth [71] (Fig. 3.4b), by our method (Fig. 3.4c) and the rendered ground-truth depth (Fig. 3.3d). Black pixels are invalid pixels not rendered by the raytracer.*

The design variations discussed in the ablation study are those that consistently match decoder and encoder solution within our specific architecture and that better characterize our approach. Since our network has a simple single-branch structure, the computational cost of the model is directly related to the number of parameters of the model and its components. We thus illustrate the computational complexity of our method by presenting our network partitioned into macro blocks with their respective parameters: the *ResNet-50* features encoder block, the *Slicing* block (featuring slicing and asymmetric dimensional reduction), the *LSTM* block and the *Asym* asymmetric upsample block. We also show the overall number of parameters for each setup (i.e., *Params*). For each block, the number of parameters needed is independent of the input image resolution, except for the LSTM block and the last upsampling, where the value indicated (i.e., 12.5M) is relative to the 256×512 resolution, which would be 16.8M for 512×1024. The results in Tab. 3.2 show the improvements obtained when using the last 4 ResNet layers, compared to using only the last one, in the *Slicing* block. Results at row 3 and 4, instead, show the benefits of adopting LSTM bottleneck-features refinement, which are appreciable already using only one ResNet output level, and become very consistent on the full pipeline. In addition, we present a comparison on whether or not to use the gradient component in the loss function, which mainly affects the sharpening of



(a) *RGB* **(b)** *Prediction* **(c)** *Ground truth*

Figure 3.5: ***Qualitative performance.*** *We present additional qualitative performance on Stanford2D3D [109] and Structured3D [108].*

recovered depth details. Figure 3.6 shows a qualitative comparison between our model trained without or with the gradient loss. Many details typical of indoor environments (i.e., wall corners, objects with repetitive patterns), are lost without the contribution of the gradient component, even if from the point of view of the average numerical error the difference seems small. Since using the gradient, as for the PReLU activation (Sec. 3.2), provides identical benefits with every configuration, we expose the gradient contribution only for the last configuration. In particular, PReLU does not directly affect the best performance obtainable on single datasets but, instead, the ability to efficiently converge on both real and synthetic datasets. As an example, similar performances can be obtained using ELU without batch normalization on the synthetic OmniDepth dataset [80], but the same model would need batch normalization to work with Matterport3D [6], as also discussed in previous works [71, 81]. As shown in Tab. 3.2, each block adds a low and reasonable cost to the model, having as a counterpart a substantial increase in performance. In terms of computational cost, a standard decoder for equirectangular image based on FCRN [61], like the one adopted by BiFuse [81], needs about 38M of parameters, while the sum of our LSTM module (12.5M) and our actual decoder (6.3M) reaches

18.8M of parameters in total.

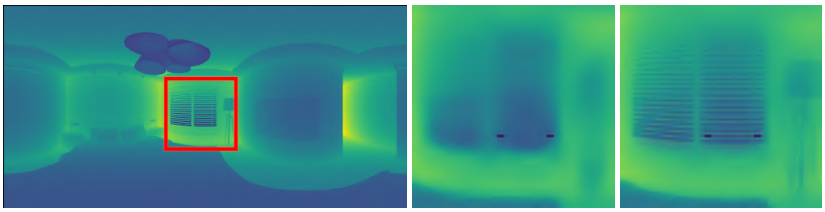
Table 3.3: ***Gravity alignment study.*** *We test the robustness of our method to horizontal ground plane misalignment on Structured3D [108] and Matterport3D [6].*

		MRE	MAE	RMSE	RMSE log	δ_1
Structured3D Our	$0°$	0.0147	0.1180	0.0549	0.1012	0.9085
	$\pm 2°$	0.0217	0.1393	0.0658	0.1368	0.8776
	$\pm 5°$	0.0263	0.1601	0.0714	0.1430	0.8527
Matterport3D Our	$0°$	0.1764	0.3296	0.6133	0.1045	0.8716
	$\pm 2°$	0.2645	0.4205	0.7026	0.1334	0.7256
	$\pm 5°$	0.3032	0.4806	0.7720	0.1482	0.6879
Matterport3D BiFuse [81]	$0°$	0.2048	0.3470	0.6259	0.1134	0.8452
	$\pm 2°$	0.3888	0.5378	0.9805	0.1852	0.6144
	$\pm 5°$	0.4905	0.6899	1.0225	0.2250	0.5440

Gravity evaluation of benchmark datasets. Our method assumes that the camera tripod is placed on a horizontal plane [52], which is common practice for capturing an indoor scene. We verified such feature on the four common publicly available datasets adopted above. All synthetic datasets [80, 108] are perfectly aligned by design. For real-world datasets [109, 6], we exploited the alignment pipeline of Zou et al. [73] to evaluate the misalignment with the ground plane. We found that the average misalignment with respect to the gravity vector of the Stanford2D3D [109] dataset is about 0.36 degrees, while the average misalignment of the Matterport3D [6] dataset is about 0.61 degrees (full statistics about gravity misalignment in Sec. 3.4.6).

Robustness to gravity misalignment. Even if our method assumes to work with gravity-aligned scenes, we do not require perfect alignment, as demonstrated by our consistent results with the mentioned real-world datasets (Tab. 3.1). Moreover, we verified that the model, trained on the original aligned data, is robust to alignment errors, even larger than those appearing in practice. To test the behavior of our method in the presence of wider inclination errors, we exploit the Structured3D synthetic [108] dataset (such that the baseline is surely aligned to the ground plane) and Matterport3D [6] as real-world dataset. Starting from their initial baseline, we generate two new testing sets by randomly rotating the up vector of the camera, simulating a much wider misalignment to gravity — i.e., $\pm 2°$ and $\pm 5°$ maximum inclination error, as reported in Tab. 3.3. $\pm 2°$ can be considered as a reliable error bound for a manual alignment without any correction, while $\pm 5°$ is a deliberately wide range (additional tests are presented in Sec. 3.4.6). Results in Tab. 3.3 show that our method produces reliable predictions even with significant camera misalignment. Performance on the Structured3D dataset reaches good accuracy in

all cases and low error values still competitive with state-of-art results. E.g., δ_1 is above 0.9 for the aligned case and degrades by only 0.03 for the moderate misalignment error of $\pm 2^\circ$ and 0.06 for the large misalignment error of $\pm 5^\circ$. The degradation obtained for Matterport3D are larger, but, by comparing the results with those in Tab. 3.1, we note that the results of our method on a dataset with $\pm 2^\circ$ error are still aligned with some of the state-of-the-art results obtained by other methods on perfectly aligned datasets. Moreover, we also present here the results obtained with BiFuse [81], for which the pre-trained model was available with the same training set, showing a much larger degradation in performance for non-gravity aligned data. This comparison shows how gravity alignment is also a fundamental assumption for other methods. It should be noted that these large errors can be avoided in practice by imposing capture constraints or performing a gravity-alignment pre-processing.

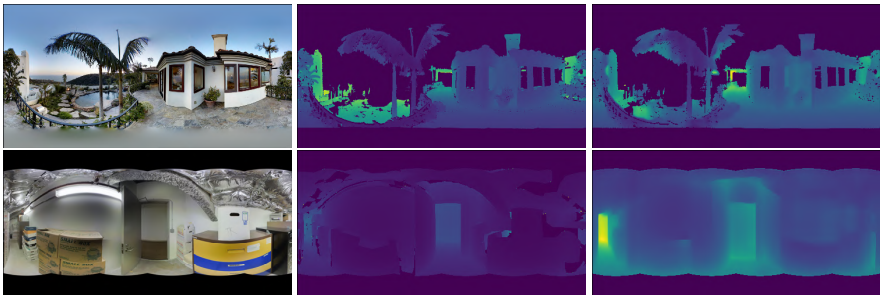


(a) *Ground truth* **(b)** *No grad.* **(c)** *With grad.*

Figure 3.6: ***Loss function qualitative comparison.*** *Example of qualitative effects depending on gradient loss (Sec. 3.3).*

3.4.5 Special cases and limits

In our experiments, we have verified that our model returns consistent results with all the man-made environments present in the tested datasets [109, 6, 80, 108], including scenes that can be defined as almost-outdoor (first row of Fig. 3.7). However, the quantitative and detailed performances depend on the ground truth data adopted, which in the case of depth often have masked parts due to lack of data from the sensor or unresolved ambiguities, such as reflections and fatal occlusions. In the second row of Fig. 3.7, we show one of these examples, that is one of the worst cases in our testes. Here the ground truth depth has numerous discontinuities and missing samples due to reflections, which are not easily predictable by our model. A large part of the structure is hidden by the insulating material.



(a) *Input* **(b)** *Ground truth* **(c)** *Output*

Figure 3.7: ***Special cases.*** *First row: results on almost-outdoor environment. Second row: one of the worst cases in our tests.*

3.4.6 Detailed gravity-alignment study

We provide a detailed gravity-alignment study that shows that available benchmark datasets are all well-aligned with respect to the gravity vector and that our method is robust to small gravity misalignments. These additional results show that our method can be directly applied in practice, even without recurring to pre-processing [52].

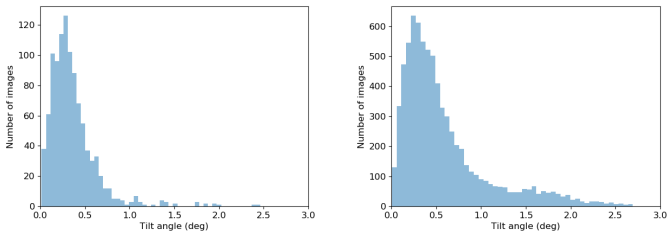
Our approach starts from the assumption that gravity plays an important role in the design and construction of interior environments, and that world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. Based on this fact, we strive to exploit gravity-aligned world-space features by performing a gravity-aligned processing of images. This assumes that input equirectangular images are aligned to the gravity vector. While this assumption could be managed by gravity-aligning images before our pipeline, it is rational to assume that, in most cases, captured images already meet these constraints. To verify this fact, we performed a study of gravity-alignment of available datasets, and verified the robustness of our method to small misalignment.

Gravity-alignment evaluation of benchmark datasets. All the commonly available synthetic datasets [80, 108] are perfectly aligned by design, and they thus perfectly meet the constraint.

The study, thus, focuses on real-world capture. A common practice for capturing an indoor scene is to place the camera on a tripod placed on a horizontal plane [52]. This capture method is in fact adopted in all the datasets available for benchmarking and also adopted in our work and the compared state-of-the-art methods [61, 71, 81].

For real-world datasets [109, 6] we exploited the alignment pipeline of Zou et al. [73] to evaluate the misalignment with the ground plane (see Fig. 3.8).

In our experiments we found that the average inclination, with respect to the gravity vector, is 0.36 degrees for the Stanford2D3D [109] dataset, while the average misalignment of the Matterport3D [6] dataset is 0.61 degrees.



(a) $Stanford2D3D$ **(b)** $Matterport3D$

Figure 3.8: ***Real-world datasets vertical misalignment.*** *The average inclination with respect to the gravity vector of the Stanford2D3D [109] dataset is about 0.36 degrees, while the average misalignment of the Matterport3D [6] dataset is about 0.61 degrees. Outliers are mainly due to inaccurate line detection and classification of the alignment tool [47].*

Indeed these values are really minimal, also considering that a significant part of the angular error is due to low accuracy detecting lines and estimating dominant direction by the automatic alignment tool [47]. We can, therefore conclude that available datasets all have a sub-degree accuracy with respect to gravity alignment.

Robustness to gravity misalignment. Even if our method assumes to work with gravity-aligned scenes, we do not necessary require a perfect alignment. In addition to the results and comparison already presented in the paper, we show, for completeness, the results obtained by introducing various degrees of error in the alignment ($0°$, $\pm 2°$, $\pm 5°$). We also performed a test, combining both *training* and *testing* of Structured3D [108] with and without alignment to the ground plane.

Results in Tab. 3.4 demonstrate the consistency of our model and effectiveness of our assumption, where the best performances are obtained the more the images are aligned with the ground plane, while the results do not improve even if a specific training is done on distorted data in order to find a better fit on the inclined images. Moreover, the method appears fairly robust to small alignment errors ($\leq \pm 2°$), and degrades as soon as input images are severely misaligned.

In other words, the effectiveness of the network is not given by the specific fitting of the training data with the expected result but by the consistency of the scene

Train incl.	Test incl.	MRE	MAE	RMSE	RMSE log	δ_1
$0°$	$0°$	0.0147	0.1180	0.0549	0.1012	0.9085
$0°$	$\pm 2°$	0.0217	0.1393	0.0658	0.1368	0.8776
$0°$	$\pm 5°$	0.0263	0.1601	0.0714	0.1430	0.8527
$\pm 2°$	$0°$	0.0238	0.1516	0.0632	0.1288	0.8672
$\pm 2°$	$\pm 2°$	0.0250	0.1589	0.0716	0.1434	0.8464
$\pm 2°$	$\pm 5°$	0.0281	0.1716	0.0743	0.1501	0.8310
$\pm 5°$	$0°$	0.0231	0.1530	0.0648	0.1245	0.8638
$\pm 5°$	$\pm 2°$	0.0250	0.1613	0.0721	0.1388	0.8438
$\pm 5°$	$\pm 5°$	0.02758	0.1697	0.0735	0.01422	0.8362

Table 3.4: ***Performance when training with misaligned images.*** *We show, for completeness, the results obtained by combining both training and testing with and without alignment to the ground plane on the Structured3D dataset [108].*

with our specific network architecture.

3.5 Conclusions

This chapter has introduced a novel deep neural network capable to rapidly estimate a depth map from a single monocular indoor panorama. The presented design exploits gravity-aligned features, characterizing man-made interior environments through a compact representation of the scene into vertical spherical *slices*. We exploit long- and short-term relationships among slices to recover the equirectangular depth map, and maintain high-resolution information in the extracted features within a deep network. Our experimental results demonstrate that our method outperforms current state-of-the-art solutions in prediction accuracy, particularly in the case of real-world data with noise and missing data.

While the current method targets monocular reconstruction, we plan to extend it to multi-view in the context of structured 3D reconstruction of indoor environments. We are also looking at integrating it with interactive solutions, where we plan to use real-time depth estimation for volume and surface computation in AR settings. Moreover, while the approach was designed for indoor scenes, gravity alignment of features occurs also in other settings, especially man-made ones. We thus envision an extension of our approach to outdoor environments, in particular urban scenes.

3.6 Bibliographic notes

Most of the content of this chapter was presented in the CVPR 2021 contribution [10] that I have co-authored, and for which I have significantly contributed to the methodology, implementation, testing, and validation of the method, as detailed in Chapter 1. The work has been very well received by the computer vision community (e.g., 59 citations on Google Scholar at the time of this writing). Various works have used our results as baseline to use for demonstrating advancements for the state-of-art (e.g., [130, 131]), and several of them have proposed follow-ups and derivations e.g., [132]). In particular, SliceNet [10] and HoHoNet [107] are discussed in the recent survey by Gao et al. [133] as the methods that introduced the squeezing of the extracted feature maps into a horizontal 1D representation due to the assumption that indoor panoramas are aligned to the gravity vector, followed by the recovering of the dense depth map predictions in the equirectangular projection. Among the various follow-ups, several works [132, 134] have noted that our original proposal of the slicing method ignores the latitudinal distortion property and thus is not suited to accurately predicting the depth near the poles. For this reason, Zheng et al. [134] proposed to perform bi-directional compression taking into account spherical distortion, while PanoFormer [132] proposed a transformer-based architecture that exploits tangent patches from spherical domain.

Chapter 4

Exploiting data fusion for deep panoramic depth prediction and completion for indoor scenes

While the previous chapter focused on purely visual data, here the focus is on the common situation in which we receive as input a single equirectangular image registered with a sparse depth map, as provided by a variety of common capture setups. The goal is to jointly exploit the dense visual channel and the sparse depth to infer a dense depth map. For that purpose, an efficient data-driven solution is introduced. Depth is inferred by a lightweight single-branch network, which employs a dynamic gating system to process together dense visual data and sparse geometric data. The design exploits the characteristics of typical man-made environments to efficiently compress multi-resolution features and find short- and long-range relations among scene parts. Furthermore, the training process introduces a new augmentation strategy to make the model robust to different types of sparsity, including those generated by various structured light sensors and LiDAR setups. The experimental results demonstrate that the presented method provides interactive performance and outperforms state-of-the-art solutions in computational efficiency, adaptivity to variable depth sparsity patterns, and prediction accuracy for challenging indoor data, even when trained solely on synthetic data without any fine tuning.

4.1 Introduction

Integrated visual and depth capture of indoor environments is a key enabling component for a wide range of applications, including autonomous navigation, mobile augmented reality, indoor mapping, and 3D reconstruction. In most situations, synchronized high-resolution depth and color data for the widest possible coverage around the viewer should be fed with low latency to further processing and analysis modules [124, 1].

Depth estimation is a fundamental problem for which a variety of active and passive solutions have been proposed over the past decades. While classic approaches exploit the correlation among multiple views, acquired simultaneously (e.g., stereo) or over time (e.g., video), single-shot capture and depth estimation has also attracted a lot of attention, since it ensures the lowest latency, reduces system hardware and synchronization burden, and offers basic building blocks for multi-view methods [40, 58].

As current 360° cameras offer viable low-cost and energy-efficient solutions for omnidirectional single-shot indoor capture [24], many research efforts are currently being focused on generating 3D from panoramic images. However, even with the full context provided by 360° capture, depth generation from monocular input remains inherently ambiguous, and is particularly complex in indoor settings characterized by large texture-less surfaces, abundance of clutter, and severe occlusions [1]. Despite the very significant recent advances in this field, especially with emerging deep-learning solutions that exploit hidden relations discovered in large data collections [81, 107, 10], monocular depth estimation remains extremely challenging.

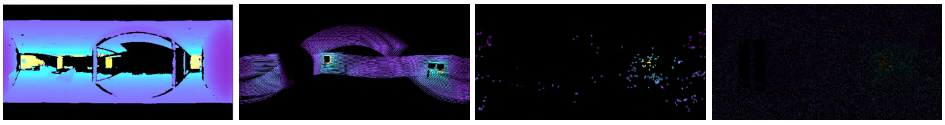


Figure 4.1: ***Different kinds of sparse depth.*** *First image (from the left): depth map captured by structured-light sensors (Matterport Pro 3D camera) has lots of missing areas when rooms are large, surfaces are shiny or thin, and strong lighting is abundant. Second image: a depth map captured by a LiDAR setup (two Velodyne VPN-16 shifted of the vertical direction with different direction) has lots of valid information but only for a few stripes. Third image: depth information may also come from triangulated features in purely image-based pipelines; indoor environments, however, have lots of flat texture-less surfaces, and reliable features, here detected from SIFT, may be very sparse. Fourth image: a typical input from low-cost structured light sensors with sparse measurements only for a small subset of the captured camera pixels; for synthetic training, a typical approach is to use a Bernoulli distribution to sparsify inputs [135].*

Depth can also be measured with depth-sensing devices. Current depth sensors exhibit, however, speed, cost, and resolution limitations that hamper their direct usability for full-frame dense 360° capture in interior scenes. In particular, stereo cameras require large baselines and tend to fail in texture-less indoor environments, structured-light sensors are at lower resolution than comparable visual cameras, are very sensitive to illumination variations, and suffer from short ranging distance. Longer ranging LiDAR sensors are more robust and accurate, but can only provide extremely sparse measurements at real-time rates [30]. Fig. 4.1 shows typical depth information provided by different low-latency techniques.

In view of these limitations, many research efforts have been devoted to exploit the coarse information coming from depth sensing to improve the performance of depth prediction from RGB [30]. Sparse depth input, in particular, has shown to be very useful to provide supervision at training time to pipelines that infer depth from visual data [59, 69, 136, 83]. More and more often, it is used at inference time [90, 39] for guided and non-guided depth completion [30]. However, the sparse output from various kinds of sensors imposes fundamental challenges on machine learning methods, since data relevance is not uniform and further processing is required to either reconstruct or ignore missing regions [99].

Because of this imbalance, depth prediction from dense RGB input and depth completion from sparse depth input have often been treated separately, and solved with different methods [98, 137, 138]. The few state-of-the-art solutions that try to jointly tackle completion and prediction target outdoor planar [139] or small field-of-view (FOV) perspective [37] projections, and are not efficiently applicable to 360° indoor capture (Sec. 2.7 and Sec. 4.4).

In this work, we introduce an end-to-end deep learning solution to jointly perform real-time dense depth prediction and completion from single-shot indoor 360° captures. This method, the first to work directly on equirectangular images of indoor environments, combines and extends state-of-the-art end-to-end deep learning solutions, introducing several specific novelties. Our input is a single equirectangular image registered with a sparse depth map, as provided by a variety of common capture setups. We do not make assumptions on the sparsity structure of the input depth, which can range from the few dense stripes produced by LiDAR solutions to the regular and irregular point sampling produced by other active and passive vision-based approaches. We expect, however, the images to be approximately gravity-aligned, as in all common datasets available [6, 109, 108, 93, 117, 80]. This condition is a de-facto standard for practically all indoor static and mobile acquisition setups, as they are equipped with automatic georeferencing and alignment systems [50, 110, 52, 107, 10]. It is worth noting that we can accommodate

for large tolerances in gravity alignment. In our results (Sec. 4.4), we demonstrate how our system even works in the case of a backpacked LiDAR acquisition system with variable vertical tilt.

Assuming a rough gravity alignment allows us to optimize our network design.

The network is constituted by a single-branch encoder-decoder, which jointly processes dense visual data and sparse geometric data in an efficient way. The initial residual encoder takes as input simultaneously 4 channels (i.e., RGB + sparse depth), and, through a gating system, returns fused visual and geometric features at different resolutions. Such features are efficiently compressed and flattened in an asymmetric way, by exploiting the intrinsic characteristics of gravity-aligned equirectangular projections of indoor scenes [10, 54]. In fact, since gravity plays an important role in the design and construction of interior environments, vertical and horizontal features have different characteristics in most, if not all, man-made environments. Moreover, most 360° capture setups have a much more regular coverage along the horizontal than on the the vertical direction because of masking effects [6]. As a result, we can exploit this anisotropy by compressing more on the vertical than on the horizontal direction. The resulting flattened features are refined through a lightweight self-attention module [140], which, acting as a bottleneck, exploits the wide context provided by omnidirectional capture in order to find the short- and long-range relations between parts of the scene which are typical of man-made environment. Decoding proceeds symmetrically to the encoder, but without need for gating, to reach the final output resolution.

Our contributions are summarized as follows:

- We introduce a novel residual encoder for the sparse-to-dense image-driven problem, which exploits lightweight gated convolutions [141] to process dense visual data and sparse geometric data together in a single branch at very little cost (Sec. 4.3.1). This design results in a much faster and more versatile network, with respect to the current approaches that process the data using multi-branch architectures and interconnections at various levels of the network [98, 100, 95, 93, 142]. Our encoder combines the advantages of a gating system, to handle different types of input in a single encoder, and of a residual architecture [62], allowing us to use deeper networks with respect to common inpainting solutions [143, 95], thanks to the efficient fusion and propagation of features at various resolutions and depth, without using skip connections that would increase the computational burden of the network [95]. As a result, the method meets real-time constraints even for the highest image and depth resolutions (Sec. 4.4.2).

- We introduce asymmetric feature compression and flattening for depth completion of gravity-aligned indoor panoramic imaging (Sec. 4.3.2), exploiting the intrinsic characteristics of equirectangular projections of indoor scenes [10, 107]. While gravity-aligned features have been employed earlier for depth estimation [10], they have not been used for designing depth completion networks. In this setting, this type of encoding remarkably maximizes the visual and geometric information gathered from a panoramic input, allowing, at the same time, the gathering of multi-resolution features and the use of a lightweight self-attention module (i.e., 1 layer, 4 heads) as bottleneck. Such an attention module allows the network to find the short- and long-range relations between parts of the scene, typical of man-made environment and panoramic images [54], relating features both spatially and at various levels of network depth (Sec. 4.3.1). Other state-of-the-art approaches, instead, employ dilated convolutions [95] as bottleneck, which are common in visual inpainting [143], renouncing to exploit deep-level features and, thus, losing part of the long-term information.
- We show how our approach is capable to handle a large variety of sparsity patterns and delivers excellent results when trained on synthetic data and applied to various real-world configurations with or without fine tuning (Sec. 4.4). In order to increase the robustness to various sampling patterns, we also complement approaches based on theoretical noise models for moderately dense and uniform RGB-D capture [144, 30] with a data augmentation module designed to model LiDAR behavior (Sec. 4.3.3). Such an augmentation is fundamental to increase the performance of our model in the LiDAR case, and increases also the performance of other methods, whose advertised accuracy was instead related to a specific capture pattern (Sec. 4.4.3).

We evaluated our approach on a variety of panoramic indoor scenes, ranging from commonly available panoramic indoor benchmarks [6, 93, 145] to novel real-world captures with mobile devices. Our results demonstrate that our approach outperforms current state-of-the-art solutions in terms of speed and accuracy (Sec. 4.4).

4.2 Datasets

In order to cover a large variety of use cases, we created a novel dataset leveraging on synthetic data generated by sampling the large-scale Structured3D [108] photo-realistic synthetic dataset, containing $3.5K$ house designs created by professional designers with a variety of ground truth 3D structure annotations, including $21,000$

photo-realistic full-panoramic (i.e., 1024×512 equirectangular format) indoor scenes. The main advantage of such a synthetic dataset is that it provides a fully accurate dense ground-truth for color and depth, which is not available with other common large-scale datasets, such as Matterport3D [6] or Stanford2D-3D-S [109], whose completeness, even if based on multi-view, is still limited by visibility and sensor limitations. For training purposes, we associate to each panoramic image and ground truth dense depth a sparse depth created through a sampling process that simulates a variety of setups. 50% of the depths simulate LiDAR setups, 25% RGB-D setups, and 25% data coming from SfM/stereo pipelines. The LiDAR setups emulate multi-beam mobile devices, selecting with equal probability 0, 16, 32, 48, 64, 80, and 96 beams on a rotating platforms. LiDAR simulation is performed by a parametric sampling process [146, 147, 148], using configurations mimicking *Velodyne* devices with 30° to 40° vertical FOV. The 0-beam case is included to simulate pure visual capture, while for the other multi-beam setups the depth coverage ranges from 16 beams (6% of pixels having depth values) to 96 beams (38%). As an extreme case, we also include a case where we have no depth input (i.e., data is purely visual, and depth maps have 0% valid pixels. Representative examples are included in Fig. 4.4. Moreover, to evaluate the method on different kinds of sparsity patterns, we simulate data coming from low-cost depth cameras using Bernoulli sampling [144] and input from SfM/stereo pipelines using a SIFT detector to place samples at feature locations. Training data is, thus, augmented with two parameterizations of Bernoulli samplings (24.68% and 6.17% of visible pixels having a depth), as well as with two different SIFT settings (with 0.91% and 2.99% valid depth pixels). Each of these 4 configurations comprise 12.5% of the training data. Representative examples are included in Fig. 4.5.

In order to validate the generalization capabilities of the model and the suitability of training on synthetic data, models trained on this dataset are tested both on S3D data and on completely novel data coming from other capture setups, including real-world ones.

Furthermore, as another important point of our work, we tested our model with a real-world sparse and challenging capture campaign, not included in any of the training datasets, but supporting a dense capture as dense ground truth. Thus, we produce a novel dataset from a real LiDAR RGB-D acquisition (i.e., mobile device with 2 Velodyne VLP-16 and a registered Garmin spherical camera - Fig. 4.7) and a ground truth dense depth acquisition through a $FaroFocus3DX330TLS$. Each sparse scan takes about 300 $milliseconds$ and produces about 16% of pixels with valid depth. We have acquired, in a multi-floor and multi-room environment, about 150 scenes in equirectangular format aligned with dense ground truth and sparse depth maps. Note that the gravity alignment of the poses is directly the one

provided by the tracking tools in the mobile device and has not been corrected through dense depth registration. This choice results in tilted sparse-dense pairs, which also provide us with a real-world benchmark to evaluate the robustness of our system to misalignment with respect to gravity direction (see Sec. 4.1). We use such a real-world benchmark for testing without any fine tuning, after training on *S3D-SD*, also demonstrating transfer-learning capabilities.

4.3 Network architecture and training

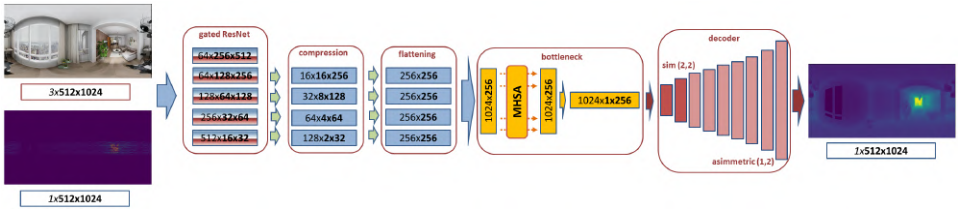


Figure 4.2: ***Network architecture.*** *Our network is constituted by a single-branch encoder-decoder, which processes together the dense visual and sparse geometric data. A residual-gated encoder takes as input 4 channels (RGB + sparse depth) returning fused features at different resolution. Multi-resolution features are compressed, flattened and passed to a MHSA- single layer module (i.e., bottleneck). Decoding proceeds symmetrically to the encoder, but without using gating, to reach the final output resolution.*

Our network is designed to directly infer a panoramic depth map from a single equirectangular image registered with a sparse depth map. Fig. 4.2 illustrates its structure for a 512×1024 input map. The architecture, is, however, fully scalable with respect to input resolution (Sec. 4.4).

The network input is given by the concatenation of the $3 \times 512 \times 1024$ RGB image with the $1 \times 512 \times 1024$ sparse depth map. On input, the RGB image is dense and contains a color value for each pixel. Valid pixels in the sparse depth map contain the distance from the camera in metric scale, while invalid pixels contain a zero.

The feature extraction is performed by 5 layers, each one having a residual architecture inside [62]. In order to process dense visual data and sparse geometric data together, each block is built around specific gated convolutions. The indoor panoramic format is also specifically handled through spherical padding and ELU activations. Encoding layers are described in Sec. 4.3.1. Similarly to other state-of-the-art solutions for 3D from RGB data [107, 54, 10, 12], we start from the assumption that, in architectural indoor spaces, vertical and horizontal features have different characteristics along and across the gravity direction. We apply such concepts in our

context by compressing the extracted features (i.e., 4 deeper feature maps) through an anisotropic contractive encoding that preserves the horizontal dimension and compresses the vertical one (Sec. 4.3.2). The resulting 4 feature maps, containing information at different spatial and depth levels, are flattened and concatenated in a single, sequential latent feature of feature dimension$\times$sequence length. The encoding of the latent feature as a sequence allows the network to use a multi-head self-attention module (MHSA) [140] as bottleneck, leveraging complementary features in distant portions of the image and depth measurements rather than only local regions to support reconstruction. This makes it possible to cope with large changes due to occlusions and to take into account the short- and long-range relations between parts of the scene typical of man-made environment. As a result of these design choices, decoding proceeds very fast and without the need for skip connections, as it can just consist of a series of convolutions, upsampling and activations until the output resolution is reached.

Our model is trained end-to-end supervised by sparse-dense depth map couples (Sec. 4.4.1), without specific assumptions on sparsity patterns, which are learned from training data. In addition to use variable depth density for RGB-D situation, we introduce a LiDAR-specific augmentation module that generates parametric LiDAR capture patterns at run-time during training (Sec. 4.3.3).

4.3.1 Feature extraction

The joined feature encoding of the mixed RGB+depth input is performed by a cascade of 5 blocks - i.e., 1 convolution-pooling block followed by 4 residual blocks. Given the spherical nature on the image, we adopt circular padding along the horizon for convolutions, to overcome the longitudinal boundary discontinuity, and reflection padding to alleviate the singularities at the poles [149].

Each residual block follows the $ResNet$ scheme, involving two convolutions and one upsampling layer [62]. Here, for each convolution layer, we introduce a dynamic gating approach to efficiently process dense visual data and sparse geometric data together.

In a generic (vanilla) convolutional layer, for each pixel located at (y,x) in an input feature map F_n having n channels, the same filters are applied to produce the output for a generic convolutional filter.

However, the sparse depth channel does not contain all valid pixels, but for single channel tasks, like pure inpainting without RGB guidance, partial [150] convolutions can be adopted to make the convolution dependent only on valid pixels. Indeed, such solution is not very efficient for our problem, since, essentially, partial convolu-

tions act as single-channel hard-gating, heuristically classifying each spatial location to be either valid or invalid, and setting to zeros or ones the mask in next layer no matter how many pixels are covered by the filter range in previous layer [143].

In our case, instead, we introduce a multi-channel *gated convolution* approach, where a multi-channel soft mask is automatically learned from data, taking decisions that jointly consider the sparse depth and the dense color channel. While gated convolutions are often adopted for pure image synthesis combined with dilated convolutions [151, 152, 143], here we use such a soft masking to model a kind of implicit confidence for multi-source features.

For each gated convolutional layer, gated features F'_m are:

$$\begin{aligned} G_m &= conv(W_{g1}, conv(W_{gk}, F_n)) \\ F_m &= conv(W_f, F_n) \\ F'_m &= \sigma(G_m) \odot \psi(F_m) \end{aligned} \tag{4.1}$$

where σ is the Sigmoid function, whose output values are within $[0, 1]$, ψ is an activation function (in this paper we use ELU [153] to remove the need for batch normalization), W_{g1}, W_{gk} and W_f are different sets of convolutional filters, used, respectively, to compute the gates (W_{g1}, W_{gk}) and features (W_f), and F_n is the input feature map.

In terms of computational complexity, the use of gated convolution should almost doubles the number of parameters in comparison to a standard, vanilla convolution [143]. To cope with this problem, we adopt here a lightweight solution, also called depth-separable convolution [141], which reduces the number of parameters and processing time while maintaining the effectiveness. Thus, we decompose a gated convolution soft mask G_m with $k_h \times k_w \times n \times m$ into a depth-wise convolution [141] (i.e., $k_h \times k_w$ kernel) followed by a 1×1 kernel convolution. Such solution has only $k_h \times k_w \times n + n \times m$ parameters, resulting in a less overall computational cost for all the encoder without measurable loss in efficiency for our problem (Sec. 4.4.4).

Our encoder returns 4 feature maps having different depth and spatial size (Fig. 4.2), gathering fused information from both visual and geometric input. Beside data fusion, propagating these levels avoids using skip connections between encoders and decoders, such as those used by several other methods [95, 154, 98] to retrieve fine-grained details, drastically reducing the computational complexity (see Tab. 4.1). At the same time, propagating this information together in a deep architecture is not simple and requires an efficient compression system. To this end, we introduced a specific compression process described in Sec. 4.3.2.

4.3.2 Feature compression and decoding

In order to support an efficient gathering of information from the extracted features, taking into account the peculiar characteristics of indoor environments, we perform a specifically designed feature compression exploiting our knowledge of preferential directions. We start from the assumption that gravity plays an important role in the design and construction of interior environments, so world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. Moreover, the amount of information contained in the spherical equirectangular projection degrades significantly going towards the poles, and even disappears completely in the input depth due to the hardware limitations of the instrument.

According to these assumption, we perform an anisotropic contractive encoding that reduces the vertical direction while keeping the horizontal direction unchanged, so that separated vertical features can be better preserved. Specifically, we reduce the vertical dimension by a factor of 8 through an asymmetric convolution module with stride $(2,1)$, applied 3 times, that contains a 2D convolution and an ELU module. We apply such a compression for each encoded feature map (i.e., 4 maps, Sec. 4.3.1). Finally, compressed features are reshaped to the same size and joined in a flattened latent feature, $L_s = (l_0 \dots l_s)$, as a sequence of s feature vectors of dimension l (i.e., s horizontal size of the less deep feature map - $s = 256$ and $l = 1024$ for a 512×1024 input).

Such a compressed representation contains a variety of information about the geometry of the scene, both local and non-local, which can be exploited to recover missing depth samples. In our case, we aim to leverage complementary features in distant portions of the image rather than only local regions, to support both depth completion and recovery. To do that, we adopt a single-layer multi-head self-attention (MHSA) scheme [140]. Our self-attention module takes the latent features $L \in \mathbb{R}^{s \times l}$ as input, and outputs a self-attention weight matrix $A \in \mathbb{R}^{s \times s}$:

$$A = softmax\left(\frac{(LW_q)(LW_k)^T}{\sqrt{l}}\right) \tag{4.2}$$

where $W_q, W_k \in \mathbb{R}^{l \times l}$ are learnable weights. The MHSA module has a particularly lightweight design with 4 heads and only 1 inner layer. We have verified experimentally that increasing the number of layers and heads does not affect performance.

Once passed to the MHSA module, the decoding of the latent feature ($1 \times 1 \times s$ in Fig. 4.2) is very fast, through convolutions, upsampling modules, and ELU activations, until we reach the target output resolution ($1 \times h \times w$ in Fig. 4.2).

4.3.3 Training strategy

During the training phase, we compute the weights of the network using a supervised training approach that exploits databases matching indoor equirectangular images with their correspondent sparse and dense depth maps (Sec. 4.4.2 for datasets details).

Coping with variable distributions of sparse depth samples

The distribution of the samples of the sparse maps can vary considerably depending on the acquisition methods. While sparse-dense datasets from structured-light sensors are available [93], it is not so for LiDAR data, even if these sensors are increasingly used also in indoor environments (Sec. 4.1). Generating those sampling patterns cannot be simply done by generic noise models (e.g., [144, 30]), but must take into account striping.

To this end, we adopt a sparsity simulation module to produce, under parametric control, different types of LiDAR patterns starting from a dense ground truth. Such a module can be used to generate specific, defined capture setup (e.g., 1 scan with fixed parameters), or to randomize sparsity at training time, thus augmenting the data to make the model more robust to different inputs. Such a module extends existing generators [155, 146, 147] to provide run-time sparse samples extracted from ground truth dense depth maps.

Our sparsity simulator is driven by a limited number of parameters, that can be eventually randomized to augment data: the number of heads (sensors) and their position and orientation, and for each sensor, the horizontal aperture (i.e., 360 degrees), the vertical aperture, and the number of laser beams (e.g., 16 for a Velodyne16-like device, etc.). Furthermore, a small 3D random noise is applied to simulate real-device noise. Head aperture and beams parameters are bounded to match to realistic setups (e.g., beams are multiple of 16, etc.).

It should be noted that even a *0 beams* case is contemplated during augmentation. This case allows the network to work even if there is no geometric input. In this case the prediction performance is aligned with that of recent state-of-the-art image-based methods [10, 107] (Sec. 4.4).

Using this augmentation module as a complement to those based on noise models, in addition to increase robustness, allows us to avoid locking the training to a specific device sampling pattern, since sparse data is generated from ground truth dense maps. In particular, as we will see in Sec. 4.4, differently from most previous work, we can train the model on purely synthetic datasets, and apply it to real-world

data captured with a specific device even without any fine-tuning.

Loss function

Independently from the type of sparse depth distribution, learning is driven by a loss function combining two data terms:

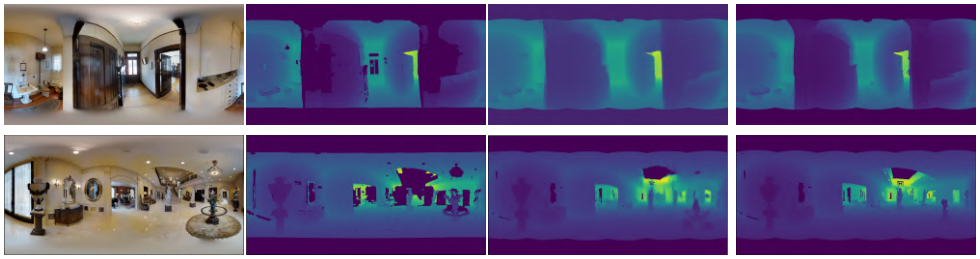
$$\mathscr{L}_{data} = \mathscr{L}_d + \mathscr{L}_{ss} \tag{4.3}$$

where L_d is the robust *Adaptive Reverse Huber Loss (BerHu)* [63], which has proven to be effective in many recent works for panoramic depth estimation [81, 107, 10]. To further take into account structural information, we add the structural loss L_s, based on the Structural Similarity Index Measure (SSIM) [156], which measures the preservation of highly structured signals with strong neighborhood dependencies. Since SSIM is higher if the two compared images are more structurally similar, we define $\mathscr{L}_{ss} = 1 - SSIM(D_{gt}, D_p)$, where D_gt is the ground truth dense depth and D_p is the final inferred depth.

4.4 Results

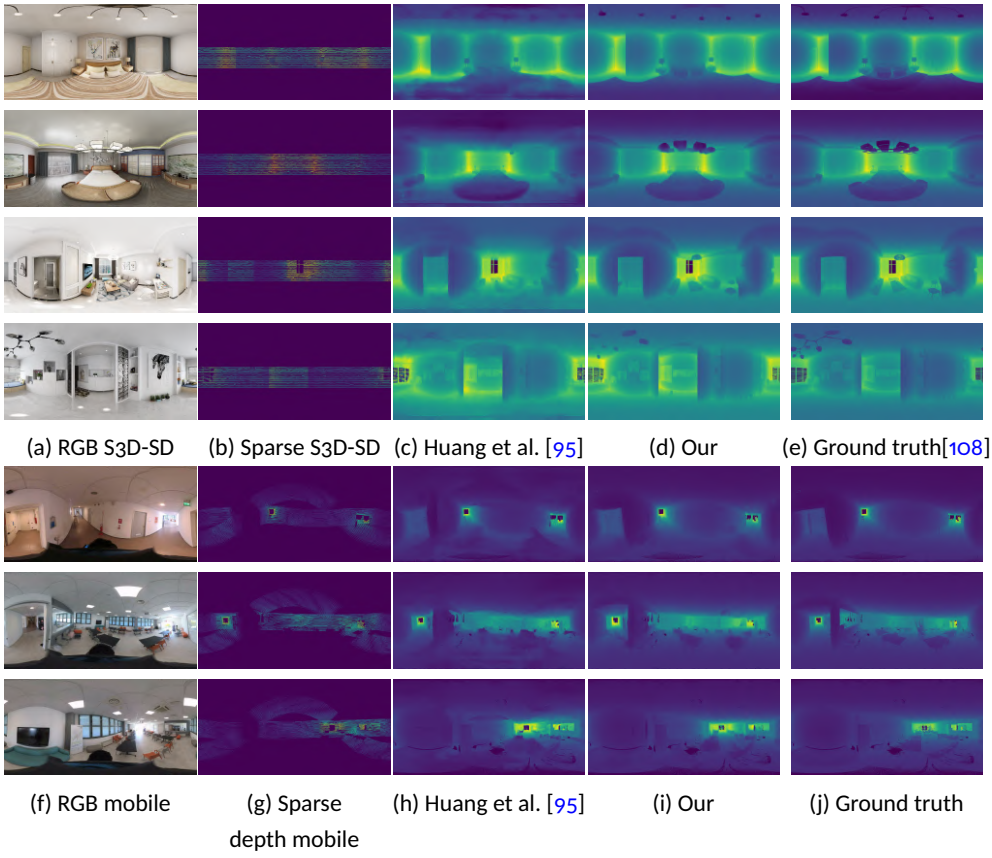
Our approach is implemented with PyTorch 1.5.1 and has been tested on a large number of indoor scenes.

Source code and models will be available to the public at *https://github.com/crs4/PanoDPC*.



(a) RGB IN [93] (b) Sparse IN [93] (c) Our (d) Ground Truth [93]

Figure 4.3: ***Qualitative results on* Matterport3D-SD *dataset [93].*** *Masked samples in the results are missing samples in the ground truth.*



(a) RGB S3D-SD (b) Sparse S3D-SD (c) Huang et al. [95] (d) Our (e) Ground truth[108]

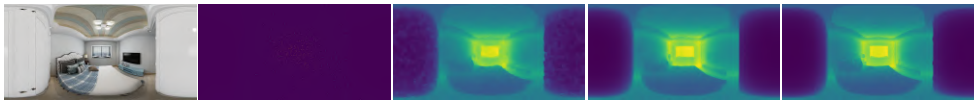
(f) RGB mobile (g) Sparse depth mobile (h) Huang et al. [95] (i) Our (j) Ground truth

Figure 4.4: ***Qualitative performance on S3D-SD with a LiDAR configuration with 32 beams and on real mobile LiDAR indoor capture.*** *Qualitative results with the same setup of Tab. 4.2. Our results are compared to the Huang et al. [95] approach trained with the same equirectangular augmented S3D-SD dataset with varying sparsity patterns.*

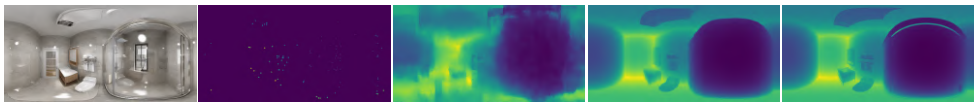
4.4.1 Benchmark datasets

Real-world capture of indoor environments is usually performed using a variety of settings, including panoramic cameras aligned with LiDAR-based setups (e.g., Velodyne) or stitching of structure-light-based sensors (e.g., Matterport). The limitations of these devices for indoor use [30] makes it difficult to find data corresponding to all the various use cases coupled with reliable full-frame ground truth data.

For training purposes, we employ in this paper the standard *Matterport3D-SD* (i.e., Matterport 3D sparse depth) [93] as well as a new dataset created on purpose that builds on Structured3D [108], dubbed *S3D-SD* (i.e., Structured 3D sparse depth).



(a) RGB S3D [108] (b) Sparse S3D Bernoulli (c) Huang et al. [95] (d) Our (e) Ground truth [108]



(f) RGB S3D [108] (g) Sparse S3D SIFT (h) Huang et al. [95] (i) Our (j) Ground truth [108]

Figure 4.5: ***Qualitative performance on S3D-SD with different input depth sparsity patterns.*** *Qualitative results using simulated input from low-cost depth cameras using Bernoulli sampling and simulated input from SfM/stereo pipelines, using a SIFT detector to place samples. Our results are compared to the Huang et al. [95] approach trained with the same equirectangular S3D-SD dataset.*

(a) RGB S3D [108] (b) Sparse S3D-SD PC (c) Huang et al. PC [95] (d) Our PC (e) Ground truth PC [108]

Figure 4.6: ***Qualitative performance on S3D-SD by point cloud (PC).*** *In these examples, 3D point clouds are obtained by unprojecting depth maps, using the same setting of Tab. 4.2, and visualizing them from a standard point of view. Note how the proposed approach improves reconstruction especially in regions where clear geometric structures from the architectural layout are present.*

Training and testing with Matterport3D-SD

Matterport3D was the first one to provide full-view indoor poses with paired sparse and dense depth maps, and for this reason, it has become a popular benchmark in recent papers and surveys [95, 30, 98]. For the sake of comparison with other results, and to show the behavior of our method on high-quality structured-light data, we thus include an analysis of our performance by training and testing our method on *Matterport3D-SD* compared to state-of-the-art works that use it. This dataset, however, is limited to a single kind of device operating in reasonably cooperative environments that ensure rather dense capture, so that even classical

infilling or hybrid data-driven solutions may be adopted with some success [30]. Fig. 4.3 shows representative examples. For this reason, we complement the dataset with much more challenging examples that cover other setups and less cooperative interiors.

Training and testing with S3D-SD

In order to cover a large variety of use cases, we created a novel dataset leveraging on synthetic data generated by sampling the large-scale Structured3D [108] photo-realistic synthetic dataset, containing $3.5K$ house designs created by professional designers with a variety of ground truth 3D structure annotations, including $21,000$ photo-realistic full-panoramic (i.e., 1024×512 equirectangular format) indoor scenes. The main advantage of such a synthetic dataset is that it provides a fully accurate dense ground-truth for color and depth, which is not available with other common large-scale datasets, such as Matterport3D [6] or Stanford2D-3D-S [109], whose completeness, even if based on multi-view, is still limited by visibility and sensor limitations. For training purposes, we associate to each panoramic image and ground truth dense depth a sparse depth created through a sampling process that simulates a variety of setups. 50% of the depths simulate LiDAR setups, 25% RGB-D setups, and 25% data coming from SfM/stereo pipelines. The LiDAR setups emulate multi-beam mobile devices, selecting with equal probability 0, 16, 32, 48, 64, 80, and 96 beams on a rotating platforms. LiDAR simulation is performed by a parametric sampling process [146, 147, 148], using configurations mimicking *Velodyne* devices with 30° to 40° vertical FOV. The 0-beam case is included to simulate pure visual capture, while for the other multi-beam setups the depth coverage ranges from 16 beams (6% of pixels having depth values) to 96 beams (38%). As an extreme case, we also include a case where we have no depth input (i.e., data is purely visual, and depth maps have 0% valid pixels. Representative examples are included in Fig. 4.4. Moreover, to evaluate the method on different kinds of sparsity patterns, we simulate data coming from low-cost depth cameras using Bernoulli sampling [144] and input from SfM/stereo pipelines using a SIFT detector to place samples at feature locations. Training data is, thus, augmented with two parameterizations of Bernoulli samplings (24.68% and 6.17% of visible pixels having a depth), as well as with two different SIFT settings (with 0.91% and 2.99% valid depth pixels). Each of these 4 configurations comprise 12.5% of the training data. Representative examples are included in Fig. 4.5.

In order to validate the generalization capabilities of the model and the suitability of training on synthetic data, models trained on this dataset are tested both on S3D data and on completely novel data coming from other capture setups, including

real-world ones.

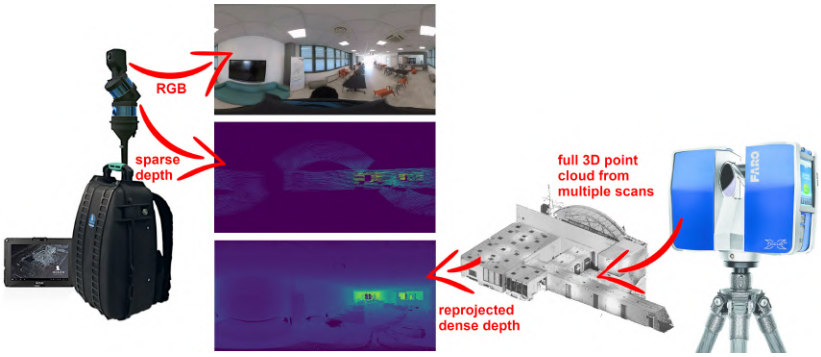


Figure 4.7: ***Mobile RGB+LiDAR setup.*** *To test our approach on a real-world panoramic RGB+LiDAR acquisition, we exploit a backpacked mobile scanner equipped with a full-view panoramic camera for the RGB capture and two LiDAR heads for sparse depth capture. Ground-truth dense depth for each pose is provided by reprojecting data coming from multiple poses of a static scanner.*

Validating on novel real-world captured data

Furthermore, as another important point of our work, we tested our model with a real-world sparse and challenging capture campaign, not included in any of the training datasets, but supporting a dense capture as dense ground truth. Thus, we produce a novel dataset from a real LiDAR RGB-D acquisition (i.e., mobile device with 2 Velodyne VLP-16 and a registered Garmin spherical camera - Fig. 4.7) and a ground truth dense depth acquisition through a $FaroFocus3DX330TLS$. Each sparse scan takes about $300\ milliseconds$ and produces about 16% of pixels with valid depth. We have acquired, in a multi-floor and multi-room environment, about 150 scenes in equirectangular format aligned with dense ground truth and sparse depth maps. Note that the gravity alignment of the poses is directly the one provided by the tracking tools in the mobile device and has not been corrected through dense depth registration. This choice results in tilted sparse-dense pairs, which also provide us with a real-world benchmark to evaluate the robustness of our system to misalignment with respect to gravity direction (see Sec. 4.1). We use such a real-world benchmark for testing without any fine tuning, after training on *S3D-SD*, also demonstrating transfer-learning capabilities.

4.4.2 Experimental setup and computational performance

We trained the network using the Adam optimizer [129] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, on four NVIDIA RTX 2080Ti GPUs (11GB VRAM) with a batch size of 8 and a learning

rate of 0.0001. For all benchmarks we adopt their original splits. Our new real-world dataset is not used for training, but for testing after training on synthetic data. With the given setup the best valid epoch was around 170 epochs for *S3D-SD* and *Matterport3D-SD*. The average training speed on 4 GPUs is about $105ms$ for each 512×1024 input image and depth pair.

Table 4.1: ***Computational cost and performance.*** *Our method is compared to the best performing state-of-the-art competitors.*

Method	Size	Params	FLOPs↓	ms/frame↓
Ma et al. [157]	512×1024	26.10 M	765.1 G	137
GAENet [158]	512×1024	4.06 M	60.12 G	39
PENet [159]	512×1024	131.67 M	487.4 G	167
packNet+SAN [100]	512×1024	76.99 M	304.7 G	149
NLSPN [98]	512×1024	26.23 M	829.86 G	167
Huang et al. [95]	512×1024	13.10 M	1624.9 G	105
Our	512×1024	22.11 M	**38.2 G**	**16**
Our	1024×2048	44.14 M	**211.7 G**	**67**
Our	2048×4096	132.22 M	**1319.3 G**	**384**

Tab. 4.1 shows our computational complexity stats, compared with several state-of-the-art methods for the inference of a 512×1024 image and depth map. Our computational cost, in terms of GFLOPs, is significantly lower than for competing solution. Note that this increased performance is also with respect to networks with a lower number of parameters but with a far more complex structure. Moreover, our method produces depth maps directly from equirectangular inputs without pre- or post-processing steps and can thus be directly integrated in production systems without additional overhead.

As a result, the inference performance of our network guarantees a low-latency generation of dense depth, and we can therefore support full instantaneous frame-by-frame depth map generation directly at acquisition. In our case, starting from a 512×1024 image and depth map, we infer depth in under 16ms on a single NVIDIA RTX 2080Ti, which is much faster than a single rotation of typical LiDARs covering a 360° view (e.g., 50ms to 200ms per rotation for a Velodyne VLP-16). The lean network structure also leads to a good scalability, as demonstrated by results with larger images included at the bottom of Tab. 4.1. We can, in particular, generate 2Kx4K depth images from equally-sized inputs in less than 0.4s.

4.4.3 Quantitative and qualitative evaluation

We evaluated our method with the same error metrics which are common to prior depth prediction and completion works and surveys [30, 100, 93, 95, 160]: mean absolute error (MAE), mean squared error (MSE), root mean square error of linear measures (RMSE) and three relative accuracy measures δ_n ($n = 1, 2, 3$), defined as the fraction of pixels where the relative error is within a threshold of 1.25^n. For MAE, MSE, and RMSE, smaller is better (i.e., unit is $meter$), while for δ_n larger is better.

Table 4.2: ***Quantitative comparison on S3D-SD/LiDAR and real LiDAR capture***. *We show our performance evaluated on standard metrics and compared to the recent state-of-the-art approaches which are comparable with us. Here we present results simulating a 360° capture with 40° vertical FOV (-30 to 10 degrees) and 32 active beams in the synthetic dataset, and results using a real mobile device with 2 Velodyne VLP-16 and a registered Garmin spherical camera with ground truth obtained using a Faro Focus3D X 330 TLS (see Sec. 4.4.1).*

Method	S3D-SD / LiDAR 32 beams						
	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1↑	δ_2↑	δ_3↑
GAENet [158]	0.086	0.394	0.160	0.149	0.466	0.753	0.889
packNeSAN [100]	0.052	0.286	0.125	0.614	0.596	0.867	0.954
Ma [157]	0.044	0.286	0.104	0.591	0.587	0.895	0.964
PENet [159]	0.028	0.210	0.090	0.595	0.671	0.930	0.976
NLSPN [98]	0.023	0.185	0.084	0.840	0.723	0.943	0.982
Huang [95]	0.017	0.138	0.068	0.830	0.824	0.960	0.987
Our	**0.003**	**0.038**	**0.022**	**0.944**	**0.982**	**0.993**	**0.997**

Method	Mobile LiDAR 16+16 beams						
	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1↑	δ_2↑	δ_3↑
GAENet [158]	0.041	0.472	0.105	0.202	0.230	0.555	0.748
packNeSAN [100]	0.027	0.404	0.078	0.539	0.278	0.603	0.842
Ma [157]	0.018	0.366	0.051	0.434	0.424	0.723	0.895
PENet [159]	0.010	0.252	0.035	0.512	0.578	0.835	0.969
NLSPN [98]	0.011	0.260	0.035	0.746	0.610	0.841	0.937
Huang [95]	0.009	0.197	0.030	0.745	0.763	0.886	0.947
Our	**0.003**	**0.088**	**0.024**	**0.822**	**0.922**	**0.986**	**0.997**

We compare our results with state of the art solutions for both indoor or generic scenes, for which the full code was available [158, 100, 157, 159, 98, 95] and an end-to-end training with equirectangular format was possible. The methods were adapted with minimal modifications to equirectangular images. We use 1024×512 for all tests.

Tab. 4.2 summarizes our performance and comparisons with related works using the augmented S3D-SD dataset to train every baseline compared (see Sec. 4.4.1), and LiDAR-specific examples for the inference. To select the training and the testing set, we adopt the official Structured3D split [108].

For synthetic tests, we considered all the simulated LiDAR configurations (i.e., 16 to 96 beams and various FOVs) discussed in Sec. 4.4.1. In Tab. 4.2, for clarity, we summarize only the results and comparisons for a 40° vertical FOV and 32 active beams case, since other S3D-SD/LiDAR tests follow the same performance trend, see Fig. 4.8.

We also include results on the real-world scenes acquired with the mobile LiDAR system (i.e., here named $mobileLiDAR\ 16+16$), compared to ground truth dense depth acquisition through a $FaroFocus3DX330TLS$ (i.e., all models trained with $s3D\ LiDAR$).

Both the real-world benchmark and the synthetic data limited to LiDAR are used only as a testing set, without any fine-tuning, thus providing evidence of transfer learning capability.

Despite our lower computational complexity, already discussed in Sec. 4.4.1, our method outperforms competitors for every condition, showing that simply adapting general purpose pipelines to the specific panoramic indoor problem leads to unsatisfactory results.

Fig. 4.4 presents qualitative results using the S3D LiDAR and mobile LiDAR test-sets adopted in Tab. 4.2. Here, we compare our method with the method of Huang et al. [95], which is the best performing among competitors in terms of quantitative results. In this case, with only a few stripes available from the scanner, our method benefits from its specific compression and information gathering features (Sec. 4.3.1) to recover more details in the final depth map.

Fig. 4.6 shows additional experiments, where geometric visualization is obtained by unprojecting the depth map into 3D point clouds. Following the same setup of Tab. 4.2 and Fig. 4.4, we show, respectively: the RGB input (a); the sparse input depth as a point cloud Fig. 4.6(b); the point cloud Fig. 4.6 predicted by the best competitor [95] (c); our prediction Fig. 4.6(d); and the ground truth point cloud Fig. 4.6(e). The illustrations complements the other qualitative and quantitative results with an easy-to-read illustration of the 3D reconstruction of the scene from a reference point of view. The performance improvement offered by the proposed approach is especially visible in regions where clear geometric structures (walls, ceilings or floor) are present.

Fig. 4.7 shows instead examples of scenes acquired with the mobile backpacked device. Numerical data is presented in Tab. 4.2). As for Fig. 4.4 experiments, our method successfully complete the map, with better accuracy than competitors. Furthermore, is also visually evident that the data acquired with the mobile backpacked device presents a significant misalignment with respect to the direction of gravity, also variable along the user's trajectory, which results in a distortion of the equirectangular projection. The consistent results also in this case show that our method is robust with respect to such an inclination, tested in a real and mobile user-case. Note that, in practice, such inclinations can be reduced before entering the depth estimation pipeline, by using on-board IMUs as well as by aligning successive poses. We show here uncorrected results, to also demonstrate the possibility of using the pipeline we present for frame-by-frame inference, without any latency connected to the integration of multiple frames or the need for assistance from external sensors.

Table 4.3: ***Quantitative comparison on S3D-SD with Bernoulli and SIFT sparsity***. *We show our performance, compared to ground truth and other approaches, testing two different sparsity patterns: Bernoulli pattern, with 1.97% of visible pixels and SIFT detector pattern, with 0.1 contrast, 5 edge threshold and no more than 8k extracted features, thus resulting in 0.91% of visible pixels (see Sec. 4.4.1).*

Method	S3D-SD / Bernoulli sparsity						
	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1↑	δ_2↑	δ_3↑
GAENet [158]	0.093	0.410	0.161	0.149	0.465	0.748	0.885
packNeSAN [100]	0.021	0.183	0.091	0.622	0.723	0.953	0.986
Ma [157]	0.049	0.280	0.102	0.441	0.679	0.895	0.954
PENet [159]	0.036	0.248	0.109	0.416	0.629	0.894	0.969
NLSPN [98]	0.018	0.162	0.054	0.834	0.813	0.961	0.985
Huang [95]	0.003	0.043	0.021	0.911	0.979	0.994	0.997
Our	**0.002**	**0.025**	**0.018**	**0.946**	**0.991**	**0.997**	**0.998**

Method	S3D-SD / SIFT sparsity						
	MSE↓	MAE↓	RMSE↓	SSIM↑	δ_1↑	δ_2↑	δ_3↑
GAENet [158]	0.093	0.410	0.161	0.149	0.465	0.748	0.885
packNeSAN [100]	0.070	0.352	0.149	0.673	0.471	0.787	0.915
Ma [157]	0.005	0.044	0.024	0.938	0.981	0.993	0.996
PENet [159]	0.040	0.259	0.118	0.499	0.557	0.859	0.960
NLSPN [98]	0.037	0.235	0.096	0.814	0.697	0.903	0.963
Huang [95]	0.025	0.177	0.084	0.774	0.766	0.931	0.974
Our	**0.003**	**0.035**	**0.020**	**0.943**	**0.987**	**0.995**	**0.998**

For completeness, we performed a further comparison of performance for different sparsity patterns. Tab. 4.3 summarizes the results obtained by emulating the pattern of low-cost structured light sensors (by a Bernoulli distribution [144]) and the pattern of a SIFT detector, emulating the typical sparse input that can be received from a SfM pipeline. Some qualitative examples with these patterns are illustrated in Fig. 4.5. Even in this situation our method demonstrates consistent performance, proving to be a versatile approach even when heterogeneous inputs vary.

Fig. 4.8 summarizes the results of our experiments on the ability to cope with different levels of sparsity, tackling both purely visual input and several multi-beam LiDAR configurations. We illustrate our performance in comparison with the competitor method [95] that best performed in our experiments. We show the results on four different sparsity cases, ranging from no depth information to a full vertical FOV scan with 96 beams (38% pixel coverage, see Sec. 4.4.1 for details). For clarity, only the δ_1 metric is included in the graph, since the other metrics have, as shown in Tab. 4.2, a similar behavior.

The continuous lines illustrate the performance of the models when trained on the augmented S3D-SD dataset (i.e., same setup of Tab. 4.2 experiments). The results indicate that our model, together with the proposed augmentation strategy, guarantees good performance for every type of sparsity. For the extreme case of a pure visual input, results are in-line with dedicated state-of-the-art [10, 107] approaches for panoramic depth estimation. On the other hand, the performance of the other approach [95] strongly depends on the number of available geometric samples. When training the model without data augmentation (dotted lines in the figure), but simply including in the training set the configuration used for testing, the performance of both models rapidly decays when moving away from the sampling used for training, even though our method remains superior at all sparsity levels. This experiment highlights how other methods can also benefit from our augmentation strategy, as it increases generalization without effects on use-case-specific performance.

For completeness, Tab. 4.4 summarizes our performance on *Matterport3D-SD* [93], compared to the results of other state-of-the-art approaches on the same benchmark [161, 162, 93, 30].

As discussed in Sec. 4.4.1, such a benchmark presents a low-challenging sparsity distribution. The majority of the state-of-the-art solutions which adopted this benchmark are not end-to-end deep learning networks, but hybrid pipelines [93], mainly focused on small-view perspective depth infilling [163]. Due to their hybrid nature, a direct computational complexity comparison is not feasible. It is also difficult, to create omnidirectional pipelines without major modifications to the

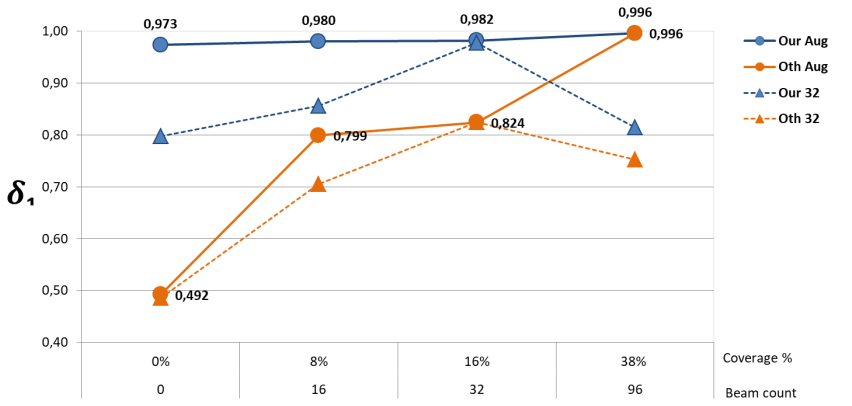


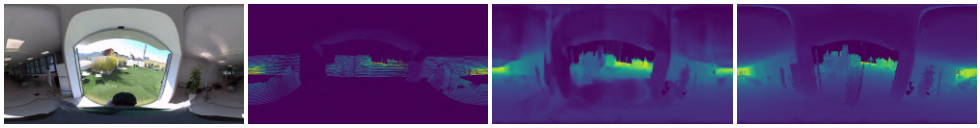
Figure 4.8: ***Performance with variable sparsity level.*** *The graph depicts the value of δ_1 as a function of input depth sparsity for our method and for the best competing method [95]. Continuous lines represent models trained with our augmentation strategy. Dotted lines show the same models but trained without augmentation (i.e., 40 degrees sparse coverage with 32 active beams)*

code. In order to provide a uniform and fair evaluation in terms of prediction accuracy, we adopt here their official baselines and pre-trained models for perspective views, testing them with the original perspective viewports provided by Zhang et al. [93], and comparing the results for our code by extracting from the single equirectangular image we produce the perspective views required for testing. It should be noted that the exposed results for compared methods, thus, do not include the additional error due to the subsequent process of stitching the results necessary to obtain the final omnidirectional view, or other effects due to pipeline modifications in case of adaptation to equirectangular projections.

We show our performance in the last two rows of Tab. 4.4. The bold row provides results obtained by training with *Matterport3D-SD* [93] training set, as for the compared methods, while, to also demonstrate our transfer learning capabilities, the other row summarizes the results obtained by inferring depth using the model trained with S3D-SD, with no fine-tuning. In both cases, our method provides consistent performance, well in line or outperforming other baselines that have been designed for this use-case. Although not directly comparable with the perspective results of the other pipelines (see Tab. 4.4), we show in Fig. 4.3 some qualitative results on the *Matterport3D-SD* dataset [93].

Table 4.4: *Quantitative comparison on* Matterport3D-SD. *We show our performance evaluated on standard metrics and compared to the recent state-of-the-art approaches on the indoor dataset provided by Zhang et al. [93]. We compare against the competitors best performance using their original perspective baselines, without considering additional error due to post-processing and stitching.*

Dataset	Method	MAE↓	RMSE↓	δ_1↑	δ_2↑	δ_3↑
M3D SD [93]	MRF [161]	0.618	1.675	0.651	0.780	0.856
	AD [162]	0.610	1.653	0.688	0.754	0.868
	Zhang et al. [93]	0.461	1.316	0.781	0.851	0.888
	Huang et al. [95]	0.342	1.092	0.850	0.911	0.936
	Xiong et al. [30]	0.462	0.866	0.863	0.930	0.942
	Our trained S3D-SD	0.464	0.803	0,834	0,908	0,942
	Our trained M3D SD [93]	**0.332**	**0.555**	**0.936**	**0.961**	**0.973**



(a) RGB IN (b) Sparse IN (c) Prediction (d) Ground truth

Figure 4.9: *Bad case.* *Results on almost-outdoor environment. Sparse samples from outdoor part, not properly masked, negatively affect the whole reconstruction.*

4.4.4 Ablation study

Our ablation experiments are presented in Tab. 4.5, with our configuration highlighted in bold. To test the key components of the approach, we use results obtained with S3D-SD, using for testing the LiDAR configuration with 3D beams (i.e., same configuration of Tab. 4.2, 32 beams). The variations discussed in the ablation study are within the design space of our approach. For example, the use of gating in the encoder is essential for the model to work. Not using it leads to inconsistent results.

The first row of Tab. 4.5 presents a case without using some key-solutions of our model: multi-resolution features (MRF), asymmetric feature compression (AFC), multi-head self-attention feature refinement (MHSA), structural-similarity loss (SSIM) and data augmentation (AUG). Here we use the deeper layer of the residual feature encoder (see Sec. 4.3.1), and we perform a standard symmetric compression along the horizontal and vertical direction. This first case, which represents a common gated encoder-decoder scheme, demonstrates how this design is not sufficient to guarantee adequate performance without the subsequent contributions we have introduced. In the second row, we show the performance obtained by

Table 4.5: ***Ablation study performed on S3D-SD, using the LiDAR 32 beams confuguration for testing.*** *MRF: multi-resolution features; AFC: asymmetric feature compression; MHSA: MHSA encoder; SSIM: SSIM loss; AUG: sparse data augmentation; LWGC: light-weight instead of standard gated convolution.*

MRF	AFC	MHSA	SSIM	AUG	LWGC	Param	Gflops	MAE	RMSE	δ_1
					✓	13.10	112.92	0.954	2.233	0.748
✓					✓	20.01	188.21	0.765	1.877	0.821
✓	✓				✓	20.01	43.15	0.312	1.384	0.877
✓	✓	✓			✓	22.11	38.16	0.121	0.084	0.951
✓	✓	✓	✓		✓	22.11	38.16	0.075	0.066	0.978
✓	✓	✓	✓	✓	✓	**22.11**	**38.16**	**0.038**	**0.022**	**0.982**
✓	✓	✓	✓	✓		31.86	61.62	0.035	0.021	0.985

introducing multi-resolution features (MRF), which hallows gathering of information without using skip connections [95, 100]. Such a solution, without an efficient feature compression results in a significant increase of computational complexity. The third row shows the benefits of asymmetric vertical compression (AFC), both in terms of lower computational complexity and in terms of accuracy. The fourth row shows instead the effects or using or not the MHSA module, without using specific losses or augmentation. It should be noted that MHSA feature refinement has a very low computational cost, but with a tangible increment of performance. The fifth and sixth rows show the increment in performance using augmentation, that limits overfitting.

At last, the seventh row shows that, in a setup using standard gated convolution instead of our light-weight choice (Sec. 4.3.1), performance is not improved despite the noticeable increment of computational cost.

4.4.5 Limitations and future works

In our experiments, we experienced that the worst results are for datasets that do not closely match the assumptions of a closed indoor space, which are used in our design to construct an efficient network architecture (see Sec. 4.1). Fig. 4.9 illustrates an example from a real-world capture. In this case, the sparse samples from the outdoor part, not properly masked, also negatively affect the reconstruction of the surrounding indoor parts.

It should be noted that the method has been specifically designed to exploit features in indoor structures. This behavior is mainly due to asymmetric feature compression and flattening of gravity-aligned indoor panoramic imaging (Sec. 4.3.2), which, in

addition to providing efficient information gathering, allows the use of a transformer (MHSA) to retrieve the wide panoramic context. Without such indoor assumptions, compression, flattening and self-attention are poorly effective. This design provides advantages in the prediction of depth for interior structures, as demonstrated by our results, while limiting the applicability of the method to scenes matching the assumptions.

Since such a domain-specific network design has shown to provide significant performance improvements with respect to more generic solutions, it is interesting to further extend this work by exploiting domain-specific constraints. One direction for future work would be to further exploit the indoor-specific design, e.g., by incorporating indoor-specific loss functions designed for architectural structures composed of large smooth surfaces, not necessarily planar, joining at possibly sharp edges [12]. Another direction would be, instead, to use the same concepts to design networks for other specific application contexts (e.g., outdoors, industrial plants), incorporating knowledge on plausible structures (e.g., presence of pipes) into the network representation and loss functions.

4.5 Conclusions

We have presented a novel end-to-end deep learning solution for rapidly estimating a dense spherical depth map of an indoor environment starting from a single image and a sparse depth map. To realize a lightweight and efficient single-branch network, we combine and extend several technical solutions to offer a novel way to solve this specific problem. We adopted a residual encoder with a dynamic gating system to extract multi-resolution features from hybrid visual-geometric input. In order to efficiently gather such amount of information and to avoid onerous interconnections between encoder and decoder, we introduced a specific compression and feature flattening which exploits the characteristics of typical man-made environments and panoramic view. End-to-end training was instead carried out by introducing a data augmentation scheme capable of making it robust and versatile as the sparsity changes. As a result, our compact network outperforms in terms of speed and accuracy current solutions for color-guided sparse depth prediction and completion.

4.6 Bibliographic notes

The content of this chapter has been adapted from an article accepted for publication in Computational Visual Media [11]. I am the joint first author of this work, and

I have significantly contributed to the conceptualization, methodology, implementation, testing, and validation of the method, as detailed in Chapter 1.

Chapter 5

Reconstructing a 3D architectural room layout from a single omnidirectional image

While the previous chapter focused on the estimation of per-pixel geometric information, here we tackle the problem of recovering the 3D shape of the bounding permanent surfaces of a room from a single panoramic image. We introduce, in particular, a novel deep learning technique capable to produce, at interactive rates, a tessellated bounding 3D surface from a single 360-degree image. Differently from prior solutions, we fully address the problem in 3D, significantly expanding the reconstruction space of prior solutions. A graph convolutional network directly infers the room structure as a 3D mesh by progressively deforming a graph-encoded tessellated sphere mapped to the spherical panorama, leveraging perceptual features extracted from the input image. Important 3D properties of indoor environments are exploited in our design. In particular, gravity-aligned features are actively incorporated in the graph in a projection layer that exploits the recent concept of multi head self-attention, and specialized losses guide towards plausible solutions even in presence of massive clutter and occlusions. Extensive experiments demonstrate that our approach outperforms current state of the art methods in terms of accuracy and capability to reconstruct more complex environments.

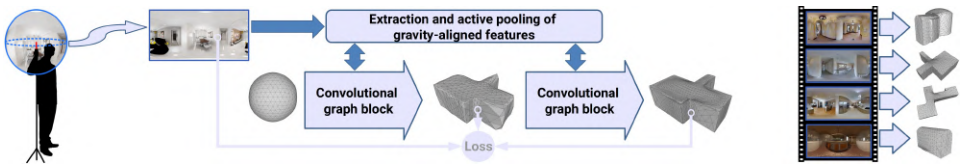


Figure 5.1: ***Method overview.*** *From a single cluttered panoramic image, our end-to-end deep network recovers, at interactive rates, a watertight 3D mesh of the underlying architectural structure. The graph convolutional network, trained using indoor-specific losses, exploits multi-scale gravity-aligned features and active pooling to deform a tessellated sphere to the correct geometry. Reconstructed models may include curved walls, sloped or stepped ceilings, domes, and concave shapes.*

5.1 Introduction

The rapid estimation of the overall 3D shape of a room from monocular visual input is a key component of indoor reconstruction pipelines [102]. The goal is to transform a single image of a furnished room into the 3D layout surface determined by joining the walls, ceilings, and floor that bound the room's interior. In this context, much of the effort is concentrated on 360° images, since they provide the widest single-shot coverage and their capture is widely supported [164, 165]. The problem is very challenging, due to the intrinsic characteristics of indoor environments, where furniture and other indoor elements mask large areas of the structures of interest, and concave room shapes generate vast amounts of self-occlusions (Fig. 5.2). Thus, indoor reconstruction requires very wide context information and must exploit very specific geometric priors for structural recovery [1].

In recent years, deep-learning solutions have emerged as a very promising way to cope with these problems for depth estimation in indoor spaces [81, 107, 10]. Thanks to the capability of these techniques to discover hidden relations from large data collections, many priors imposed by pure geometric reasoning approaches can be relaxed. However, 3D layout reconstruction is more complex than depth estimation, since it does not simply assign a depth to each visible pixel, but must extrapolate large portions of the invisible structure, which can be occluded not only by objects but by the structure itself, leading to multiple intersections per view ray. Current approaches cope with that complexity by operating in very restrictive solution spaces. In particular, most methods target variants of the Manhattan World model (MWM: horizontal floors and ceilings, vertical walls meeting at right angles) [102], such as the Indoor World model (IWM: MWM with single horizontal ceiling and floor) [103] or the Atlanta World model (AWM: vertical walls with single horizontal ceiling and floor) [45]. Moreover, the most effective approaches recover the layout by exploiting projections to lower-dimensional spaces before expanding them to

3D. However, the combination of 1D/2D projections with restrictive priors limits the reconstruction capability to very few regular shapes and makes reconstruction less robust to occlusion (see Sec. 2.7).

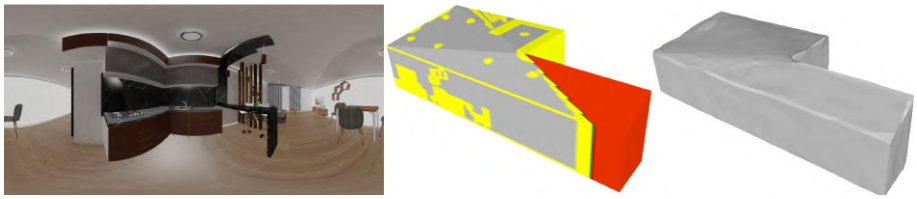


Figure 5.2: *Layout occlusion.* *Left: panoramic image. Middle: room shape, with occlusions from walls (red) or from furniture (yellow). Only 31% of the surface of interest is visible. Right: plausible 3D reconstruction generated by our method.*

In this work, we introduce a novel technique, dubbed *Deep3DLayout*, that exploits a graph convolutional network (GCN) to directly infer a watertight 3D mesh representation of the room shape from a gravity-aligned panoramic image. Such an approach significantly expands the solution space, covering a much wider class of interior environments than prior solutions, including concave rooms with curved or stepped walls or ceilings (Fig. 5.1). Indoor priors, less restrictive that previous ones, are taken into account in the network structure, as well as in the carefully crafted loss functions that drive training, without resorting to 1D/2D projections. In particular, the mesh, represented as a 3D graph-encoded object, is initialized as a tessellated sphere mapped to spherical coordinates and deformed towards the correct geometry by leveraging indoor-specific perceptual features extracted from the input panoramic image. To cope with large occlusion and take into account the typical characteristics of interior environments, we encode image information as gravity aligned features (GAF), which are representative of the architectural indoor model mentioned above, and we exploit a multi head self-attention (MHSA) approach to efficiently associates GAFs to 3D vertices during deformation, taking into account short- and long-range relations, thereby coping with occlusions. To train the network, our indoor-specific loss functions drive the mesh towards architecturally plausible watertight 3D structures favoring models defined by the intersection of smooth surfaces, not necessarily planar, possibly intersecting at sharp edges. Our main contributions are summarized as follows:

- We define the indoor layout as a 3D graph-encoded object, exploiting GCNs to infer the room structure as a 3D mesh (Sec. 5.3.1 and Sec. 5.3.2). Previous state-of-the-art methods for indoor panoramic scenes (e.g., [45, 102]) used, instead, simplified connected structures for the layout (Sec. 2.7), and required a post-processing step to obtain the 3D geometry [102, 103].

- We introduce a novel way to associate panoramic image features to 3D vertices in an indoor environment. We exploit GAFs to efficiently preserve receptive fields according to an indoor shape hypothesis (Sec. 5.3.3), refining and incorporating them in the graph with a MHSA approach (Sec. 5.3.4). Unlike static projections used for 3D object reconstruction [112, 113], our active element is very robust to severe occlusion.
- We introduce a domain-specific loss function that combines specialized data and regularization terms to guide reconstruction towards a plausible architectural model (Sec. 5.4). Since these priors are integrated in the training process, no further post-processing is necessary to regularize the output, and inference occurs at interactive rates.

Our extensive benchmarks demonstrate how we improve the state-of-the-art both in terms of accuracy and in terms of capability to reconstruct more heterogeneous environments (Sec.5.5). To grant reproducibility, code and data are made available.

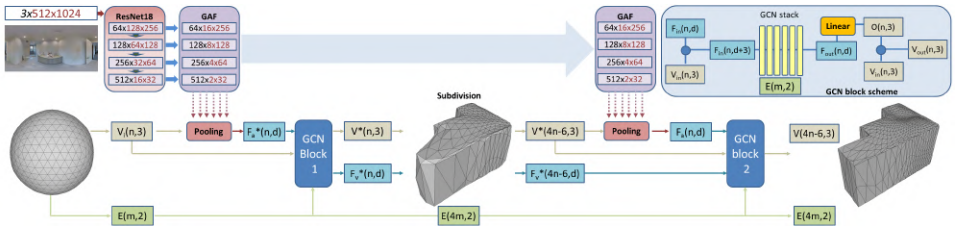


Figure 5.3: ***Deep3DLayout pipeline.*** *Our end-to-end deep learning technique maps an equirectangular image to a 3D mesh representing the bounding surface or the room. Two GCN blocks deform an input icosphere (Sec. 5.3.1) by offsetting its vertices (see Sec. 5.3.2). The first block starts from a first pooling of the GAF features $F^*(n,d)$ to return a low-res estimation of the mesh $M^*(V^*,E_i)$. This low-res representation M^* is then refined to poll refined GAF features $F^*(4n-6,d)$, which drive the second GCN block. The output of the second block is the final refined mesh model $M(V(4n-6,3),E(4m,2))$.*

5.2 Method overview

Our goal is to recover, from a single panoramic image, a representative 3D model of the boundary surfaces of the architectural layout of the environment in which the photograph has been taken. We assume that the environment around the viewer is a closed volume fully bounded by walls, ceiling and floor. These surfaces are assumed to be only partially visible, not only due to the presence of furniture and wall-hangings, but because of the commonality of self-occluding concave environments (e.g., L-shaped rooms). Since we have to cope with significant amounts of missing or ambiguous information, we need to use prior knowledge on the nature

of interior environments to guide reconstruction. Contrary to previous works, however, we avoid doing so by topologically and geometrically constraining the output model (e.g., forcing vertical walls and/or planar walls and ceilings), or by explicitly performing operations valid only in restricted cases (e.g., projections and reasoning in a 2D floor plan). Our solution, instead, is to drive the reconstruction of a general geometric shape in the most plausible direction by exploiting domain knowledge for network design and problem regularization.

5.2.1 Geometric model

The most general topological model of the recovered boundary surface is a closed 3D surface homeomorphic to a sphere, that we can represent as a triangulated mesh. We, therefore, use such a 3D mesh as the output representation of our network. Geometrically, we assume that vertices have unconstrained spatial positions, but that the shape is most likely obtained from the intersection of *smooth* surfaces, not necessarily planar, possibly intersecting at *sharp* edges. These characteristics, which drive the learning process through crafted loss functions (Sec. 5.4), are typical of the most common indoor structures [1].

5.2.2 Network design

Our network recovers the room structure by progressively deforming a 3D mesh so that its shape matches the environment seen by the viewer (Sec. 5.3.2). Since we have a spherical panorama as input, we can initialize the mesh to a 3D sphere, and use spherical coordinates to establish correspondences with the input image (Sec. 5.3.3). Moreover, as we do not know, for a given panorama, where geometric features may be positioned, we initialize the sphere to a geodesic polyhedron obtained by regular subdivision of an icosahedron (also known as *icosphere*). Mesh deformation is then driven by associating image features to mesh vertices. Since we expect, as consequence of architectural design, that a certain part of the structural elements will develop along the gravity direction, we extract gravity-aligned features (GAF) (Sec. 5.3.3) and refine the association with vertices by exploiting long- and short-range relations, which allows us to cope with large occlusions (see Sec. 5.3.4). To increase robustness, we also employ a coarse-to-fine approach, in which we first target the reconstruction of a coarse mesh starting from the initial sphere, and then refine the coarse mesh to a finer one. This approach results in the end-to-end pipeline illustrated in Fig. 5.3, consisting of a dual-stage mesh deformation network (see Sec. 5.3.1 and Sec. 5.3.2), driven by an image feature network (see Sec. 5.3.3 and Sec. 5.3.4). The mesh deformation network includes two GCN blocks (see Sec. 5.3.1) deforming the input icosphere by offsetting its vertices (see Sec. 5.3.2).

The image feature network, instead, performs feature pooling based on the current vertex positions. It includes a CNN encoder to encode GAFs from the input image (see Sec. 5.3.3), and a *multi-layer spherical pooling* system to refine the association of GAFs to vertices. In order to support our coarse-to-fine-approach, the first GCN block starts from a first pooling of the GAF features $F^*(n,d)$ to return a low-res estimation of the mesh M^*. This low-res representation M^* is then refined (in this paper 4 times the number of initial faces) to perform a further GAF pooling $F^*(4n-6,d)$, which drives the second GCN block. The output of the second block is the final mesh model $M(V(4n-6,3),E(4m,2))$ (see Sec. 5.3.4 for details). The network thus performs reconstruction using a fully 3D approach, looking for a solution in 3D space without resorting to any projection to a 2D layout or a 1D corner list.

5.2.3 Training and loss function design

Learning is performed using a supervised training approach that exploits databases matching spherical panoramas to the geometric representation. We assume, as in all recent works, that the examples are gravity-aligned, i.e., with the Y axis of the image pointing in the real-world vertical direction. All commonly available annotations of indoor panoramic layouts are already gravity-aligned and provided as closed shapes, and thus can be easily represented as closed meshes with the correct orientation (Sec. 5.5). The loss function used for training must embed our knowledge of the problem without overly constraining the solution space. We thus combine a data term, measuring the quality of fit with respect to training data, with regularization terms that drive the solution towards plausible reconstruction hypotheses based on our expected 3D models, favoring reconstructions in which shapes are likely to be composed of large smooth surfaces, not necessarily planar, joining at sharper edges. As the shape is represented in a graph, we can define these terms as differentiable higher order functions across neighboring nodes. It is important to note that these terms are computed with operators on the boundary surface, without resort to 2D or 1D projections (Sec. 5.4).

5.3 Network structure

Our end-to-end network maps panoramic images to a mesh representation. In the following, we first detail the encoding of the mesh model (Sec. 5.3.1) and the mesh deformation network based (Sec. 5.3.2). Finally, we discuss the gravity aligned features encoding (Sec. 5.3.3) and the multi-res spherical pooling (Sec. 5.3.4).

5.3.1 Room model as a 3D graph-encoded object

In our 3D graph-encoded layout the mesh is represented as a graph (V, E, F), where $V(n,3)$ is the set of n vertices in the mesh, $E(m,2)$ is the set of m edges, each one connecting two vertices, and $F(n,d)$ are the feature vectors of dimension d coming our of the pooling layer and associated to vertices (Sec. 5.3.4). Vertices are defined in the camera reference frame, setting the origin at center of the spherical image, and the Z axis pointing upwards.

5.3.2 Mesh Deformation Network

The mesh deformation network is a sequence of two GCN blocks (Fig. 5.3). It starts from an initial sphere $S(V_i, E_i)$, having $V_i(n,3)$ vertices and connectivity $E_i(m,2)$, and returns a final output model $M(V,E)$ having $V(4n-6,3)$ vertices and connectivity $E(4m,2)$. Each block, internally, consists of a cascade of GCN layers (i.e. 6 layers) followed by a final linear transform which returns the vertex offsets $O(n,3)$, used to compute the vertex displacements $V(n,3)$ (Fig. 5.3 upper right detail). Each GCN layer l is defined as:

$$f_v^{\,out} = W_0 f_v^{\,in} + \sum_{q \in \mathscr{E}} W_1 f_v^{\,in} \tag{5.1}$$

where $f_v^{\,l} \in F_l(n,d_l)$ are the feature vectors attached to vertices, d_l are the feature channels at level l, $f_v^{\,l+1} \in \mathbb{R}^{d_{l+1}}$ are the feature vectors on vertex $v \in V(n,3)$ before and after the convolution, and $\mathscr{E}(v)$ are the neighboring vertices of v specified in $E(m,2)$; W_0 and W_1 are the learnable parameter matrices of $d_l \times d_{l+1}$ that are applied to all vertices. Note that $W1$ is shared for all edges, and thus Eq. 5.1 works on nodes with different vertex degrees [112].

The first convolutional block takes as input a set of aligned image features $F_a(n,d)$ (i.e., F_a self-attention features, see Eq. 5.3), obtained by pooling the GAF features with the vertices $V_i(n,3)$, and the initial connectivity $E_i(m,2)$. The output of this first block is a set of deformed vertices $V^*(n,3)$ and a set of vertex features $F_v{}^*(n,d_l)$.

Before the second step, both the intermediate mesh $M^*(V^*,E_i)$ (i.e., $V^*(n,3)$ vertices with $E_i(m,2)$ connectivity) and the associated vertex features $F_v{}^*(n,d_l)$ are refined by following the subdivision scheme proposed by Gkioxari et al. [113]. Specifically, we subdivide each triangle mesh by adding a new vertex at the center of each edge and dividing each face into four new faces. Vectors of vertex features are also subdivided by averaging the values of the features at the two vertices which form each edge. After the subdivision, we obtain a refined mesh with $V^*(4n-6,3)$ vertices and $E(4m,2)$ edges, and the refined vertex features $F_v{}^*(4n-6,d_l)$.

We exploit the new vertex set $V^*(4n-6,3)$ to pool refined GAF features $F_a(4n-6,d)$, so we pass refined GAF as input to the second convolutional block, together with vertices $V^*(4n-6,3)$ and $F_v{}^*(4n-6,d_l)$ (i.e., the residual interpolated features from the first block). As a result the second block returns the final vertex displacement $V(4n-6,3)$ (Fig. 5.3). The final model $M(V,E)$ is then given by vertices $V(4n-6,3)$ and by the subdivided connectivity $E(4m,2)$.

While the design of our network is scalable, all the results in this paper have been produced by a network that has been sized in accordance with available datasets (Sec. 5.5). In particular, we use as input/output for the first block a mesh with 642 vertices and 1280 faces (1920 edges), while for the second block we have 2562 vertices and 5120 faces (7680 edges). We found that, using available benchmarks, these triangulation are enough both to cover the whole spherical scene with an uniformly distributed number of vertices (i.e., block 1), as well as to provide a reliable representation of the targeted indoor structures (i.e. block 2).

We also studied different multi-stage architectures with variable number of faces, similar to architectures for general-purpose object reconstruction [112]. However, we experienced that the illustrated dual stage scheme performs better (Sec. 5.5.4) in our context. This is due to the combination of two factors differentiating our problem from generic object reconstruction methods targeted to recover details of the entire visible surface of the object, starting from images with a small field-of-view [112, 113]. First of all, our panorama covers a full 360° FOV. This requires a reasonably dense coverage in the initial mesh to ensure a good starting angular resolution, especially when coping with occlusions. Second, our targeted indoor structure is characterized by a low number of clustered geometric details, as the target shape is composed of large portions of piecewise uniform surfaces. We are therefore not targeting a final uniformly dense mesh.

5.3.3 Gravity-aligned Features Encoding

A central component of our network architecture is the combination of the features extracted from the images with the vertices encoded in the graph. As these features are present at many scales, the common architectural choice is to use convolutional residual networks for extracting relevant low/mid/high-level features from the input tensor. Such networks contain a contractive encoding part that progressively decreases the input image resolution through a series of convolutions and pooling operations, giving higher-level neurons with large receptive fields. As we work on panoramic images, these features can be effectively distributed over the whole geometric context and cover wide areas.

In order to support an efficient pooling of the image features, taking into account the peculiar characteristics of indoor environments (Sec. 5.3.4), we perform a specifically designed anisotropic contractive encoding exploiting our knowledge of preferential directions.

We start from the assumption that gravity plays an important role in the design and construction of interior environments, so world-space vertical and horizontal features have different characteristics in most, if not all, man-made environments. Such concept is exploited in several recent works for depth estimation from indoor panoramic images [10]. According to this assumption, we perform an anisotropic contractive encoding that reduces the vertical direction while keeping the horizontal direction unchanged, so that separated vertical features can be better preserved. Specifically, in our approach, we start by encoding features from *ResNet-18* layers (Fig. 5.3). We chose this light-weight architecture to maintain interactive inference rates (Sec. 5.5), and, in order to compensate for the low depth of the network, we simultaneously exploit the last four layers, instead of only the deepest one. In this regard, we have also tested other deeper encoders, such as *ResNet-50* [62] and *HarDNet* [166], finding only a marginal increase in performance against an increased time cost.

Anisotropic contractive encoding is then applied to the features coming out of *ResNet-18* by performing an asymmetric convolution with stride $(2,1)$ applied 3 times, achieving a reduction along the vertical direction by a factor of 8. Each convolution is followed by ELU activation function, thus removing the need for batch normalization [71]. We apply this encoding for each one of the last 4 ResNet-18 layers, obtaining the 4 GAF layers $G0, G1, G2, G3$ (Fig. 5.3), which are the latent features ready for vertex pooling. As discussed in Sec. 5.3.4, this compressed multi-scale representation contains useful information to recover the underlying structure, including locally-visible features and non-local structure information.

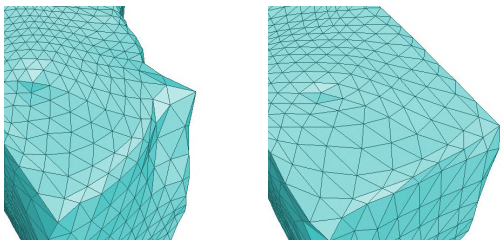


Figure 5.4: *Effect of MHSA.* *Qualitative difference in not using (left) or using (right) the MHSA transformer when pooling image features.*

5.3.4 Multi-layer spherical pooling with self-attention

In pipelines for generic 3D object reconstruction, the objects is observed from an external viewpoint and within a restricted field of view, and the shape of the object is reconstructed from local features visible. Thus, it is possible to simply pool image features from the 2D projection of the associated vertex on the image, which can be readily obtained by assuming known camera intrinsic matrix [112, 113]. In that case, the main problem for the pooling layer is the interpolation of nearby features, which in our case, would mean combining nearby GAF features.

In our case, by contrast, in addition to feature interpolation, we have to cope with major occlusion problems, caused by a vast amount of clutter and by the structure itself, as discussed in Sec. 5.2. We cannot restrict us to simply statically combine nearby features in image space, but need to take into account short and long range relationships in the image to perform an effective pooling. This has to be done using an active process, that learns the importance of local and non-local features for a given neighborhood. To this end, we introduce a specific pooling system for combining our GAFs.

Given the 3D vertex positions $V(n,3)$, we poll the four gravity feature layers $G0$, $G1$, $G2$, $G3$, encoded as described at Sec. 5.3.3, through the following spherical projection:

$$u = \frac{\arctan(x/y)}{\pi} \quad v = \frac{\arctan(z/\sqrt{x^2+y^2})}{2\pi} \tag{5.2}$$

where x, y, z are the world coordinates of a vertex $v \in V(n,3)$ and $u, v \in G$ normalized coordinates in image feature space.

For each vertex v_i, we concatenate the features extracted from the four layers into a single feature $g_i{}^*$ associated to the vertex (in this paper the feature dimension is $64+128+256+512=960$). This solution has the advantage of associating information at the vertex at different resolutions, keeping at the same time a low number of parameters for each layer. After this pooling, we obtain a *latent feature* representation $F_g = (g_0 \dots g_n)$, as a sequence of n feature vectors of dimension d. However, due to occlusions, this compressed representation contains a variety of information that may or may not be useful to recover the underlying structure. In fact, it contains both local-visible features and non-local structure information, as well as features from clutter or occluders. In order to efficiently retrieve useful information from this representation, we adopt a self-attention strategy. Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [140], that has had important successes in tasks where one must capture global dependencies, such as image synthesis [167].

In our case, we aim to leverage complementary features in distant portions of the image rather than only local regions to support reconstruction. We do that by learning a set of attention weight vectors used for refining important spatial features.

Our self-attention module takes the latent features $F_g \in \mathbb{R}^{n \times d}$ as input and outputs a self-attention weight matrix $A \in \mathbb{R}^{n \times n}$:

$$A = softmax\left(\frac{(F_g W_q)(F_g W_k)^T}{\sqrt{d}}\right) \tag{5.3}$$

where $W_q, W_k \in \mathbb{R}^{d \times d}$ are learnable weights.

We exploit the attention matrix in Eq. 5.3 to obtain the refined latent feature $F_a = A(F_g W_v) \in \mathbb{R}^{n \times d}$, where $W_v \in \mathbb{R}^{d \times d}$ are learnable weights. Such a self attention approach is applied in a multi-head fashion (MHSA) [140], to let the model jointly attend to information from different representation sub-spaces at different positions. This amounts to running r attention modules in parallel. In our case we use $r = 4$, denoting 4 attention weights for each image spatial feature. These refined features, combined through a learning process, are then associated to the vertices of our model. Fig. 5.4 shows a qualitative example of the effect of using MHSA to pool feature with respect to statically combined local features.

5.4 Training and loss functions

During the training phase, we compute the parameters of the network using a supervised training approach that exploits databases matching gravity-aligned spherical panoramas of cluttered scenes to the their boundary layout representation.

Our loss functions are designed to combine data terms that measure the quality of fit with respect to training data, with regularization terms that drive the solution towards a plausible reconstruction of an indoor environments. As the shape is represented in a graph, we can define all these terms as differentiable functions that compute geometric properties by accessing neighboring nodes. As we perform a coarse-to-fine reconstruction in a single end-to-end network, see Sec. 5.3.2, these losses are applied with the same weights for both the intermediate and final mesh.

Due to the nature of typical human-built structures, we expect that our models will be composed of large smooth surfaces, not necessarily planar, joining at possibly sharp edges. Such a characterization is less restrictive then typical indoor priors based on planar surfaces and vertical/horizontal alignments (e.g., variations of MWM, IWM, AWM), and includes common structures such as curved walls, vaults,

and domes, that we seek to represent with a limited number of vertices. We incorporate this knowledge in our data terms by measuring the dissimilarity in surface positions and orientations between predicted and ground truth meshes, as well as the fitting of sharp features present in the ground truth model. Data terms have thus the following form:

$$\mathscr{L}_{data} = \lambda_c \mathscr{L}_{pos} + \lambda_n \mathscr{L}_{norm} + \lambda_{sh} \mathscr{L}_{sharp} \tag{5.4}$$

where $\mathscr{L}_{pos}$ is the positional loss, $\mathscr{L}_{norm}$ is the orientation loss, and $\mathscr{L}_{sharp}$ is the sharpness loss. λ_c, λ_n, and λ_{sh} are weights that tune the relative importance of the terms (see Sec. 5.5 for details).

Positional and orientation terms, as usual in 3D reconstruction, are computed by uniformly sampling the ground truth and predicted surface meshes and summing the contributions at each point. We adopt the differentiable mesh sampling operation proposed by Gkioxari et al. [113], sampling a point cloud Q from the ground-truth mesh, and a point cloud P from the mesh prediction, retrieving at each sample point the position p and its unit normal n_p. Given a point p in a point cloud A, let $N(A,p) = argmin_{a \in A} \|p-a\|$ be the nearest neighbor of p in A, and $n_{N(A,p)}$ its normal. We then define the positional term from the bidirectional *chamfer distance* between point clouds P and Q

$$\mathscr{L}_{pos} = |P|^{-1} \sum_{p \in P} \|p - N(Q,p)\|^2 + |Q|^{-1} \sum_{q \in Q} \|q - N(P,q)\|^2 \tag{5.5}$$

and the orientation term from the bidirectional *normal distance*

$$\mathscr{L}_{norm} = -|P|^{-1} \sum_{p \in P} \left| n_p \cdot n_{N(Q,p)} \right|^2 - |Q|^{-1} \sum_{q \in Q} \left| n_q \cdot n_{N(P,q)} \right|^2 \tag{5.6}$$

These two terms are averaged over the surface, and large areas would dominate the few sharp edges, which are important in indoor environments as they appear, e.g., at the connection of walls among themselves or of walls with ceiling or floor. As we target low-poly reconstruction, in order to preserve such features, we want do drive vertices in the prediction to snap to ground truth feature edges. Given a ground truth mesh, we start by calculating, at mesh loading time, a sharpness value based on cosine similarity for each of its edges, i.e. $e_{sharp} = 1 - n_0 \cdot n_1$, where n_0 and n_1 are the normal vectors of two triangles sharing the edge e, and mark as feature all edges with $e_{sharp} > \tau$ (with $\tau = 0.5$ for this paper). This measure favors considering as features angles around 90 degrees, which are common in architecture. We then uniformly sample all the extracted feature edges to obtain a point cloud S_e. We then compute

$$\mathscr{L}_{sharp} = |S_e|^{-1} \sum_{q \in S_e} \|q - N(P,q)\|^2 \tag{5.7}$$

Note that, differently from positional and orientation terms, sharpness is unidirectional, as we want to have ground-truth feature edges ground truth only attract close-by vertices in the prediction, leaving the others unchanged.

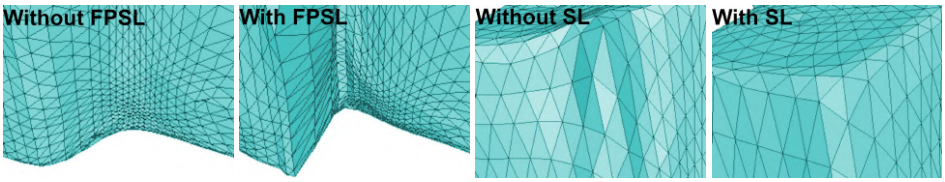


Figure 5.5: ***Effect of FPSL.*** *The first two images shows the difference in using or not the feature-preserving smoothness loss (FPSL - Eq. 5.11); the second two images show the difference in using or not the sharpness loss (SL - Eq. 5.7).*

Using data terms alone, the network may generate very large deformations to closely fit the ground truth, which is harmful especially in the first training iterations, when the estimation is far from ground truth and large vertex movements would compute inconsistent solutions, letting the optimizer stuck in local minima. We therefore introduce regularization losses to counteract this effect, while at the same time driving the solution towards plausible reconstructions in areas where data terms provide little information:

$$\mathscr{L}_{reg} = \lambda_e \mathscr{L}_{edge} + \lambda_s \mathscr{L}_{smooth} \tag{5.8}$$

where $\mathscr{L}_{edge}$ is an edge regularization term, $\mathscr{L}_{smooth}$ is a curvature regularization term, and λ_e and λ_s are weights that tune the relative importance of these terms. Regularization weights are smaller than the data weights since these terms must support data fitting and not counteract it (see Sec. 5.5 for numerical details).

Edge regularization tends to favor uniform distribution of vertices in the predicted mesh, and is computed by:

$$\mathscr{L}_{edge} = |E|^{-1} \sum_{(i,j) \in E} \left\| v_i - v_j \right\|^2 \tag{5.9}$$

where v_i and v_j are the vertices of a common edge $e_{ij} \in E$. The combination of this weight with $\mathscr{L}_{sharp}$ has the effect of nicely distributing vertices around sharp features.

In addition to regularize positions, we also aim to regularize curvature, to avoid small curvature variations while preserving sharp features. We do that by first computing the discrete mean curvature normal [168] of each predicted vertex v_i,

i.e., the unit length surface normal n_i at the vertex v_i scaled by the discrete mean curvature $\bar{k}_i$:

$$\bar{k}_i n_i = \frac{1}{4A(v_i)} \sum_{(i,j)\in E} (\cot \alpha_{ij} + \cot \beta_{ij})(v_j - v_i) \tag{5.10}$$

where $A(v_i)$ is the sum of the areas of all triangles containing vertex v_i, α_{ij} and β_{ij} are the two angles opposite to the common edge $e_{ij} \in E$, $v_j \in S[i]$, assuming $S[i]$ the set of neighboring vertices to v_i. We use Eq. 5.10 to discretize the Laplacian matrix $L \in \mathbb{R}^{n\times n}$, so that the tensor $K_H = \|LV\| \in \mathbb{R}^{n\times 1}$ contains the discrete mean curvature for all vertices [169]. Directly minimizing this term as done in 3D object reconstruction [113] would lead to uniform smoothing, causing a degradation of sharp structural features of an indoor environment. Thus, we introduce an exponential curvature-aware weight term:

$$\mathscr{L}_{smooth} = |V|^{-1} \sum_{i\in V} e^{-|K_{Hi}|} |K_{Hi}| \tag{5.11}$$

The introduced exponential weight reflects what we expect from our indoor model, as it penalizes low-curvature vertices, forcing them to lie on a plane or on a constant-uniform curvature surface, while avoiding to penalize feature vertices with a more marked curvature.

The contribution of each individual term is analyzed in Sec. 5.5. Some qualitative effects are also illustrated in Fig. 5.5.

5.5 Results

Our approach was implemented using *PyTorch* [170] and *PyTorch3D* [171] and has been tested on a large variety of indoor scenes. Code and data will be made available at *https://github.com/crs4/Deep3DLayout*

5.5.1 Benchmark datasets

In order to provide a comparison with state-of-the-art work, we analyze results standard publicly available datasets [72, 109, 49, 108], containing thousands of indoor scenes and commonly adopted for benchmarking 3D layout recovery [54, 45, 107, 103, 85]. However, due to the focus of prior works, these benchmarks mostly consist of MWM structures [102]. Since our method is more general, we extend the testing set with the publicly available *AtlantaLayout* [45] dataset, which also contains rooms with curved walls or meeting at non-right angles. In addition, we prepared a specific dataset, called *Pano3DLayout*, containing 106 more complex

Method	MatterportLayout					
	IoU3D↑	IoU2D↑	CD↓	$F\tau 0.1\uparrow$	$F\tau 0.3\uparrow$	$F\tau 0.5\uparrow$
LayoutNet [73]	75.78	78.02	1.96	49.16	78.45	84.20
DuLaNet [74]	75.62	78.86	0.82	51.55	80.20	86.88
HorizonNet [54]	78.45	81.28	0.79	56.14	85.35	91.67
AtlantaNet [45]	80.67	82.55	0.56	59.73	88.13	93.62
HoHoNet [107]	80.25	83.06	0.65	59.00	87.67	92.54
Led2Net [103]	81.70	84.12	0.37	64.24	93.12	97.80
Zeng [85]	-	-	-	-	-	-
Deep3DLayout (ours)	**85.38**	**86.45**	**0.18**	**77.92**	**98.91**	**99.78**

Method	Stanford					
	IoU3D↑	IoU2D↑	CD↓	$F\tau 0.1\uparrow$	$F\tau 0.3\uparrow$	$F\tau 0.5\uparrow$
LayoutNet [73]	76.78	80.34	0.96	34.89	78.20	82.53
DuLaNet [74]	80.02	83.44	0.65	39.35	82.89	87.15
HorizonNet [54]	82.77	86.12	0.23	45.88	88.03	94.83
AtlantaNet [45]	82.36	85.70	0.18	46.45	88.92	95.27
HoHoNet [107]	82.44	85.75	0.22	45.92	88.15	94.65
Led2Net [103]	83.60	87.12	0.18	49.23	91.77	98.10
Zeng [85]	86.21	-	-	-	-	-
Deep3DLayout (ours)	**89.39**	**90.11**	**0.01**	**84.66**	**99.94**	**99.99**

Table 5.1: ***Comparison on MWM datasets.*** *We compare our method, according to indoor layout and 3D reconstruction metrics, to recent state-of-the-art approaches on the MatterportLayout [6] and Stanford [109] MWM datasets.*

environments, not included in previous benchmarks, such as, for example, scenes with sloped or stepped ceilings, domes, and interconnections of different rooms.

Ground-truth layout meshes were created without resorting to manual annotation. For new synthetic scenes, we simply used the watertight mesh generated with *PyMeshlab* [172] from the same model used for rendering with interior objects removed. For real-world scenes, *PyMeshlab* was used to transform to a watertight mesh the global dense point clouds available with *Matterport3D* [6].

5.5.2 Experimental setup and timing performance

We trained the network using the Adam optimizer [129] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, on four NVIDIA RTX 2080Ti GPUs (11GB VRAM) with a batch size of 8 and a learning rate of 0.0001. The adopted weights for loss function are 1.0 for the position and

normal distances and 0.1 for all the other losses (see Sec. 5.4). We found that these figures work well on models in the metric scale, and we convert other units to meters prior to training. As a result, our models are already in metric scale. We experienced that the scale estimation, compared to using normalized meshes, adds an important information to the final result at a negligible cost.

Our method uses triangulated meshes as ground truth models (Sec. 5.3.1). Newly modeled scenes in Pano3DLayout are modeled directly as watertight meshes stored as collections of vertices and faces, while existing 2.5D datasets [72, 109, 6, 108] are triangulated at run-time using *trimesh* [173] from the original representations in terms of 1D collection of corners on the image horizon plus the height of the layout.

The computational complexity of our method is relatively low with respect to comparable works, since the model has $23.8M$ of learnable parameters. As an example, HorizonNet [54], which is the baseline for several other methods [45, 103], includes about $57M$ of parameters.

As a result, the inference performance of our network is compatible with interactive rates, and we can therefore support model generation directly at acquisition time, to support, e.g., augmented reality applications and/or interactive editing. Even though we generate full 3D models without resorting to 1D or 2D reductions, we can predict the results, starting from a 512×1024 image at a rate of $27 fps$ on a single NVIDIA RTX 2080Ti.

It should be noted that our results are obtained through an end-to-end network that takes directly as input the gravity-aligned image and produces directly as output the 3D mesh. In this work, the 360 data are mostly well-aligned, so we do not apply any pre-processing. This condition is fulfilled at virtually no cost by all capture setups that include a IMU sensor and could incorporate our network without any modification. For more general cases, we might consider including a 360 gravity alignment block to align the input. Several deep learning solutions exist that perform this task at interactive rates [110]. For several competitors, pre- and post-processing operations may be more costly. For instance, the image pre-processing adopted by many of the compared methods [73, 74, 54, 85, 107, 103], that has to be applied both on the training and testing sets, takes about $3 seconds$ per image.

5.5.3 Quantitative and qualitative evaluation

We compared our reconstruction performance to the one achieved by latest state of the art methods [73, 74, 54, 45, 85, 107, 103]. Tab. 5.1 summarizes the results the Indoor World scenes comprising commonly available benchmark datasets [72, 109,

Method	AtlantaLayout					
	IoU3D↑	IoU2D↑	CD↓	$F\tau 0.1\uparrow$	$F\tau 0.3\uparrow$	$F\tau 0.5\uparrow$
Led2Net [103]	75.68	77.45	0.92	33.69	65.67	75.09
AtlantaNet [45]	80.25	84.30	0.48	34.28	67.56	80.55
Deep3DLayout (ours)	**89.88**	**90.51**	**0.10**	**87.01**	**99.90**	**99.98**

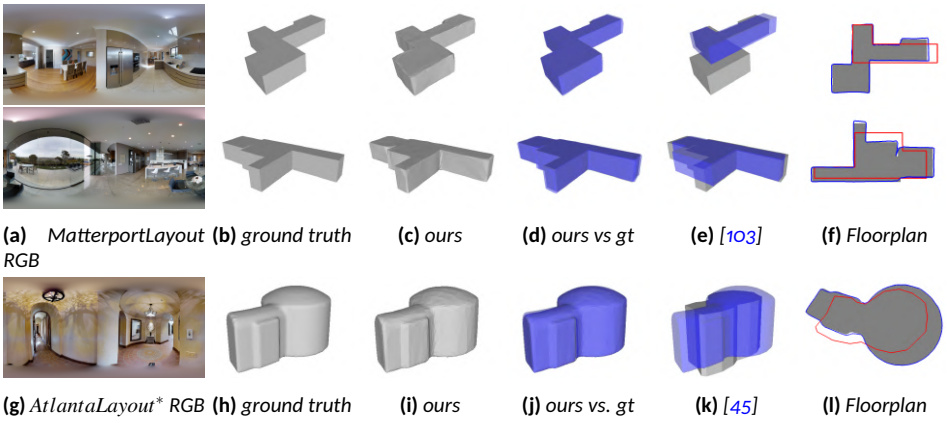
Method	Pano3DLayout					
	IoU3D↑	IoU2D↑	CD↓	$F\tau 0.1\uparrow$	$F\tau 0.3\uparrow$	$F\tau 0.5\uparrow$
Led2Net [103]	39.61	57.20	485.49	29.36	64.91	67.23
AtlantaNet [45]	69.21	78.54	2.24	35.45	65.46	68.35
Deep3DLayout (ours)	**83.28**	**89.15**	**0.02**	**69.82**	**98.76**	**99.08**

Table 5.2: ***Comparison on non-MWM dataset.*** *We compare our method, according to indoor layout and 3D reconstruction metrics, to recent state-of-the-art approaches on the publicly available non-MWM AtlantaLayout dataset [45] and on our new Pano3DLayout release. For comparison, we choose best-performance methods for which source code and pre-trained models are available.*

6, 108], while Tab. 5.2) presents the results on the more challenging non MWM scenes from AtlantaLayout [45] and Pano3DLayout.

We evaluated all methods using error metrics relevant to our task. Since the target is not to reconstruct the full visible scene, but to infer the underlying severely occluded layout, we resort to spatial measures rather than pixel error metrics. In particular, we complemented standard metrics for indoor layout reconstruction, such as intersection-over-union [102] (i.e., *IoU2D*, *IoU3D*), which were adopted as benchmark by all the competing methods [74, 54, 45, 85, 107, 103] with proper 3D reconstruction metrics [174], such as *Chamfer distance (CD)* and *F-score*, commonly adopted for 3D object reconstruction, which provide additional information, especially for complex scenes.

We refer to Zou et al. [102] for details on the indoor layout metrics. It should be noted, however, that we use *IoU2D* solely with the purpose of facilitating the comparison with prior works on models with vertical walls and flat floors. We computed this measure by extracting the 2D plan through planar sectioning according to the Y axis. As our work solves the problem in 3D, the other included 3D measures are more appropriate. Moreover, the *IoU3D* estimation adopted by all mentioned competing methods is usually obtained by the product of a 2D error (i.e., room footprint) and the height error, assuming a constant layout height. Since our method works directly in 3D space and is not limited to single-height layouts, we implemented full-3D routines to calculate both *IoU3D* and *IoU2D* using *PyMeshlab* [172]. We experimentally verified, with the available codes of the compared methods, that



(a) *MatterportLayout RGB* **(b)** *ground truth* **(c)** *ours* **(d)** *ours vs gt* **(e)** *[103]* **(f)** *Floorplan*

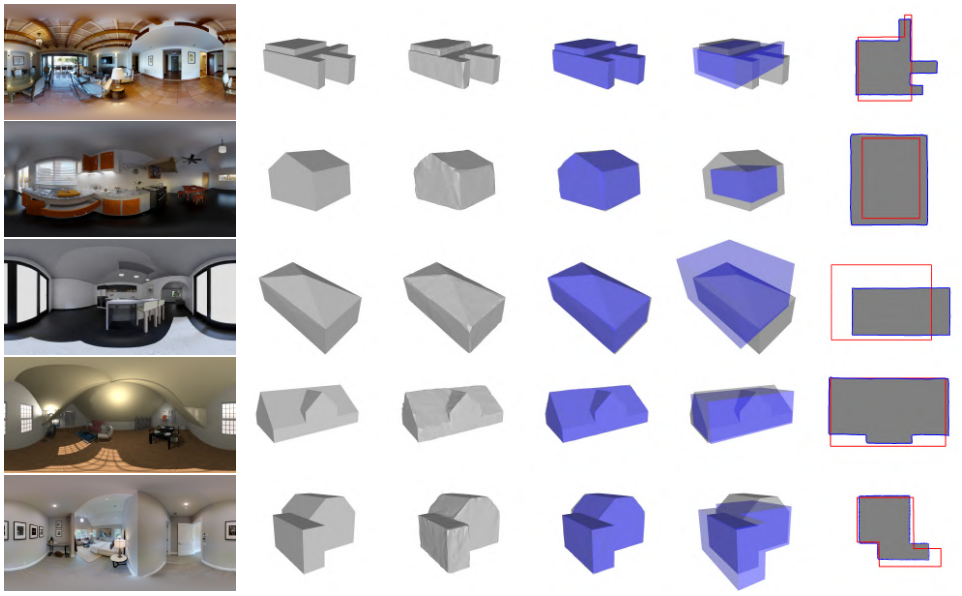
(g) *AtlantaLayout* RGB* **(h)** *ground truth* **(i)** *ours* **(j)** *ours vs. gt* **(k)** *[45]* **(l)** *Floorplan*

Figure 5.6: ***Qualitative comparison.*** *Qualitative comparison on publicly available datasets. We show the input image, the ground truth model, our prediction, our prediction in overlay with ground truth, competitor prediction in overlay with ground truth and the 2D floorplan comparison (grey ground truth,* blue *ours, red competitor). The presented scenes contains multiple connected rooms partially visible from a single point-of-view, as well as non-MWM corners, curved walls and ceiling. Fig.5.6h full ground truth, including the dome, was recovered from the Matterport3D [6] meshes.*

when dealing with a single ceiling and single floor scenes, our implementation is consistent with the restricted one adopted by Zou et al. [102]. Therefore, all the statistics provided in Tab. 5.1 and Tab. 5.2 are calculated using our full-3D measures, except for the method of Zeng et al. [85], whose source code is not available, where we expose the performances declared in their paper, based on the assumption of a single elevation per model.

The Chamfer distance (CD) and the F-score are presented for all the methods for which source code and data are available. To obtain such measures, we uniformly sample 10000 points from the result and the ground truth mesh [112, 113] and compute measures by comparing those samples. Specifically, CD measures the distance of each point to the other set, while F-score represents the harmonic mean of precision and recall, obtained by computing the percentage of points in prediction or ground truth that can find a nearest neighbor from the other within a distance threshold τ [174]. In Tab. 5.1 and Tab. 5.2 we present, respectively, F-score for $\tau = 0.1, \tau = 0.3, \tau = 0.5$, which are typical metric distances used in indoor benchmarks [113]. For CD, smaller is better, while for the F-Score larger is better.

Results in Tab. 5.1 summarize the results obtained on the *MatterportLayout* [49] and the *Stanford2D-3D-S* [109] datasets. For training and testing, we follow the same official split described by Zou et al. [102], and adopted by the compared works. Both *MatterportLayout* and *Stanford2D-3D-S* mainly contain Indoor World



(a) *Pano3DLayout RGB* **(b)** *ground truth* **(c)** *ours* **(d)** *ours vs. gt* **(e)** *[45]* **(f)** *Floorplan*

Figure 5.7: ***Qualitative comparison on non-MWM scenes.*** *Qualitative comparison on non-MWM scenes (*Pano3DLayout*). We show the input image, the ground truth model, our prediction, our prediction in overlay with ground truth, competitor prediction in overlay with ground trutht and the 2D floorplan comparison (grey ground truth,* blue *ours, red competitor). Our approach has consistent performance for a variety of model kinds, in particular for complex structures, such as domes and sloping roofs.*

scenes, that is scenes with walls meeting at right angles and rooms have a single horizontal floor and a ceiling. As discussed in previous sections, all compared methods, except ours, adopt some form of post-process regularization on the output that exploits the Indoor World assumptions. Our method, on the other hand, without any postprocessing, outperforms other methods with all metrics. Such difference in performance is more pronounced, in particular, with the F-score and Chamfer metrics.

While the size of our network can be parameterized in terms of mesh sizes, all the results are presented for a final mesh size of 2562 vertices and a coarse mesh size of 642 vertices, which produced the best results for our 512×1024 image data. These numbers are not surprising, since using coarser meshes would reduce our capability to represent smooth curves (e.g., domes), while denser meshes would overly reduce the image feature size associated to each vertex. As an example, our setting of (642,2562) vertices achieves $F\tau 0.1 = 64.24$ for MatterportLayout,

while reducing the mesh to (162,642) vertices reduces the score to $F\tau 0.1 = 37.43$, and increasing the mesh to (2562,10242) vertices achieves only $F\tau 0.1 = 64.78$ at a much higher storage and computational cost.

Fig. 5.6 illustrates some examples from publicly available benchmarks [49, 45]. We show, respectively, the input equirectangular image, the ground truth 3D model, our predicted results, our prediction with the ground truth overlay and the prediction with a competitor method with ground truth overlay. We choose for comparison the methods of Wang et al. [103] and Pintore et al. [45], which have, respectively, the best performance for Indoor World and Atlanta World environments at the time of this writing. The presented scenes contain multiple connected rooms partially visible from a single point-of-view, as well as non-MWM corners, curved walls and ceiling. In all cases our method outperform the reconstruction obtained with the other methods, which is not surprising since we are more flexible in terms of expected output geometry.

On the other hand, MWM cases (Fig. 5.6b) are particularly challenging for our method, since we do not impose any constrain of this kind, while the expected results is a regularized, planar layout. All the methods compared in tab. 5.1 share the same MWM regularization post-processing of HorizonNet [54], but, in many cases, the layouts obtained with post-processing are visually plausible, but not correct in many cases (e.g., Fig. 5.6b). In particular, the differences are more marked in case of strong occlusions, where our method returns a reconstruction generally returns a much more reliable reconstruction (e.g., Fig. 5.6b, top). This seems to be related to the fact that our network, which works in full 3D and is fully data-driven, is more robust towards occlusions with respect to methods relying on 2D/1D projects and post-process regularization.

Fig. 5.6g presents a case from *AtlantaLayout* that violates the Atlanta World assumption since there is a dome rather than an horizontal ceiling). In this case our method provides a faithful reconstruction (Fig. 5.6i), while methods that approximate the Atlanta World model provide partially correct reconstructions since the curved ceiling causes an error in scale estimation, which propagates to an error on the footprint (e.g, Fig. 5.6k)/ In Tab. 5.2 we present results for more complex scenes not limited by the Indoor World assumption. We show the results with our novel *Pano3DLayout* dataset, which includes more challenging cases, such as domes, sloped or stepped ceilings and more. We compare our results with competing methods which have best performance on the same data and for which training code has been made available by the authors [103, 45]. All the methods presented, included ours, are trained on the *MatterportLayout* dataset and fine-tuned with a specific training set, respectively from the *AtlantaLayout* and *Pano3DLayout* dataset, following the same

data splitting for fine tuning adopted by other compared baselines [54, 45]. The 106 Pano3DLayout scenes were split into 66 for fine tuning and 40 for testing. Training speed is $\approx 0.04s/img$ on 4 GPUs. Training time on the full MatterportLayout is 1 minute/epoch. Reported results are for 3200 epochs.

The results show that our approach guarantees consistent performance with different kinds of models, in particular in the case of more complex structures such as domes and sloping roofs. On the contrary, the performance of the methods based on the Indoor World and Atlanta World hypotheses are not able to maintain adequate performance on these more complex cases. This tendency is evident also in the qualitative comparisons of Fig. 5.6 and Fig. 5.7. For the competing methods, besides the predictable error on the roofs, there is a remarkable scale error. This is due to the fact that the proportions of the structure in all these approaches are obtained under the hypothesis that the surfaces can only be vertical or horizontal, and that, therefore there is always a homography between the boundaries of the ceiling and the floor [175]. This constraint is clearly violated on these complex scenes.

5.5.4 Ablation Study

Baseline					Structured3D			
MLP	GAF	MHSA	FPSL	SL	IoU3D↑	$F\tau 0.1\uparrow$	$F\tau 0.3\uparrow$	$F\tau 0.5\uparrow$
-	-	-	-	-	49.13	53.58	70.02	76.98
✓	-	-	-	-	63.93	55.24	71.45	80.20
✓	✓	-	-	-	75.61	67.24	79.11	85.78
✓	✓	✓	-	-	83.34	70.16	93.55	98.82
✓	✓	✓	✓	-	84.98	78.66	97.12	99.02
✓	✓	✓	✓	✓	91.45	80.65	98.74	99.18

Table 5.3: ***Ablation study.*** *The ablation study, performed on the* Structured3D *dataset [108], demonstrates how our proposed design choices improve the accuracy of prediction. Results show only comparable-stable cases that actually increase it. We show in the last row the full architecture setup.Legend: MLP: multi-layer pooling; GAF: gravity aligned features; MHSA: multi-head self-attention; FPSL: feature preserving smoothness loss; SL: sharpness loss.*

Tab. 5.3 summarizes the results of our ablation experiments. To test the key components of our approach, we exploit the Structured3D dataset [108], a synthetic dataset containing over 21,000 rendered rooms with ground truth 3D structure annotations. This recent dataset has not yet been adopted by the comparable works surveyed in Sec. 5.5.3, but provides an additional valuable benchmark for our method. Fig. 5.4 and Fig. 5.5 visually illustrates examples of behavior related to these ablation experiments.

Since we designed an end-to-end network, we show design variations that lead to comparable-stable cases. To this end, we highlight five representative key-choices: the MLP (multi-layer pooling), compared to using only the last ResNet layer, the GAF (gravity aligned features), compared to standard image features encoding (see Sec. 5.3.3), the MHSA (multi-head self-attention) module (see Sec. 5.3.4), the FPSL (feature-preserving smoothness loss), compared with standard Laplacian smoothness, and the use of SL (sharpness loss) (see Sec. 5.4). The variations discussed in the ablation study are those that consistently match the encoder and decoder components of our specific architecture and that better characterize our approach.

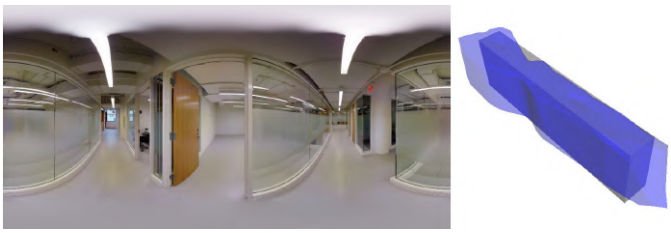


Figure 5.8: ***Failure case****. Example of bad reconstruction.*

The first row of Tab. 5.3 shows a configuration starting from the last layer of a ResNet encoder, without using any anisotropic contractive encoding (i.e., GAF) and MHSA feature pooling, and without a specific indoor loss function, such as FPSL and SL. The second row of Tab. 5.3, instead, shows the same setup of the first row but exploiting the last 4 layers of the ResNet encoder. It should be noted that this configuration provides results of a variation of our technique that bears similarity with mesh-growing methods, such as Mesh-RCNN [113] and Pixel2Mesh [112], adapted to interior panoramic views, but without the indoor-specific features. The numerical performance clearly show that just adapting mesh growing approaches to the task is not sufficient.

Exploiting GAFs, at row 3 of Tab. 5.3, considerably improves performance, by efficiently preserving the receptive field according to the hypothesis that indoor environments are constructed taking into account the gravity direction. Row 4 shows instead the performances of the whole network without using specifically designed loss functions. Even though results are somewhat consistent, reconstruction lacks many details and misses large feature edges connecting the main architectural surfaces, as also highlighted by Fig. 5.5. Row 5 and 6 show the increase in performance by applying FPSL and SL. Although the metrics $F\tau 0.3$ and $F\tau 0.5$ are almost the same using the sharpness loss SL, a significant difference is present in

the $IoU3D$, where this objective function greatly improves the detection of sharp details (see Fig. 5.5).

Pano3DLayout (synthetic scenes)				
Misalignment	IoU3D↑	$F\tau 0.1\uparrow$	$F\tau 0.3\uparrow$	$F\tau 0.5\uparrow$
$\pm 0^\circ$	89.01	70.90	97.95	98.99
$\pm 1^\circ$	88.15	69.72	96.92	98.13
$\pm 2^\circ$	85.52	56.14	85.35	91.67
$\pm 5^\circ$	76.67	34.50	78.35	89.20

Table 5.4: ***Robustness to gravity-alignment errors.*** *Comparison of reconstruction performance on synthetic scenes of Pano3DLayout by introducing gravity alignment errors.*

Our approach assumes that input images are already gravity-aligned, a constraint met by all common datasets and that can be achieved in most common setups using IMUs or automatic image upright adjustment solutions [10, 107]. In order to test the robustness to our method to moderate variations in gravity alignment, we report in Tab 5.4 the results obtained by introducing various degrees of alignment error (0°, $\pm 2^\circ$, $\pm 2^\circ$, $\pm 5^\circ$) on the synthetic scenes included in Pano3DLayout. The method appears fairly robust to small alignment errors ($\leq \pm 2^\circ$), and degrades as soon as input images are severely misaligned. As these tests were performed without any retraining, we expect that further robustness can be achieved through data-augmentation with misaligned examples, as done in previous work on depth estimation [10, 107].

5.6 Conclusions

We presented an end-to-end deep learning approach to directly recover, at interactive rates, the 3D layout of an indoor structure from a single panoramic image. Differently from prior solutions, all the components of our method address the problem in 3D, without resorting to 1D or 2D projections, and we produce as output a closed 3D mesh rather than a 2.5D model with strong planarity or surface orientation priors. By taking into account the properties of indoor environments in the network design and in the loss specification, we were able to produce an indoor-specific solution which is efficient to train and use. In particular, inference times are well within interactivity constraints, and quantitative and qualitative results show significant improvements with respect to state-of-the-art methods in terms of accuracy and capability to reconstruct non-MWM environments.

The method has also limitations. First of all, the problem is inherently ambiguous and, as all purely-image-based solutions, reconstructions may be far from reality in

several situations. Fig. 5.8 shows an example of failure of our reconstruction due in this case to the abundant presence in the scene of transparent and specular walls, combined with repetitive structures inside and outside the targeted scene. Limitations more specific to our approach stem from the tessellated mesh representation. In particular, reconstruction by deformation from a single origin generates denser and more detailed meshes near the origin, and less detailed ones as one moves away from the origin and occlusions increase, and thus the precision depends on mesh tessellation size. Moreover, while our 3D mesh model is significantly more flexible than current solutions exploiting MWM priors, our spherical mesh topology is far from being sufficient to represent all sorts of architectural environments, since several elements of the architectural structure, such as pillars, stairs, septal walls or openings cannot be represented with a single closed surface. Including holes (doors, windows) seems feasible as a direct extension of our end-to-end single pass method deforming a spherical mesh, while extending the approach to other topologies is not trivial. We plan to tackle this problem by exploiting semantic information to handle internal architectural elements and details, separating the reconstruction into several layers. Moreover, we also plan to extend this methodology to multiple images and/or additional geometric information (e.g., RGB-D), in order to support larger and more articulated indoor environments, such as multi-room structures.

5.7 Bibliographic notes

The content of this chapter has been adapted from an article published in ACM Transactions on Graphics and presented at SIGGRAPH Asia 2021 [12], in which I was one of the primary authors of the paper. I have significantly contributed to the conceptualization, methodology, testing, implementation, and validation of the method, as detailed in Chapter 1. An interesting follow-up of our approach has been recently proposed by Dong et al. [176]. Their work extends our solution based on mesh representation to total-scene understanding using a transformer architecture.

Chapter 6

Conclusion

This thesis has introduced novel techniques that advance the state-of-the-art in 3D reconstruction of indoor environments, with a focus on methods the infer depth and layout information from a single panoramic image, eventually enriched with sparse depth. This final chapter provides a concise summary of the achieved results and briefly discusses the potential directions for future work.

6.1 Overview of achievements

The research comprising this thesis has been focused on deep learning solutions for inferring from a single 360° image of an indoor environment, eventually enriched with very sparse depth information, a dense depth map that provides the distance to the viewer of every visible point and the structure of the architectural layout of the imaged environment, i.e., the closed surface formed by the walls, ceiling, and floor of the room in which the photo was taken. In my discussion of background material and analysis of related work (Chapter 2), I have highlighted how solutions to these problems form fundamental building blocks of reconstruction pipelines, and summarized the significant research efforts that have been made in the past towards their solution.

The results presented in this dissertation highlight how the introduced techniques represent a progress of the state-of-the-art. All the presented methods share the fact that they take directly as input data in equirectangular format, as produced by devices and without any kind of prior processes, and produce their output through an end-to-end deep learning solution. All the techniques exploit the fact that

input is gravity-aligned, and that gravity-aligned processing of images throughout specially designed networks can directly exploit long- and short-range relations among gravity-aligned world-space features.

In particular, my main achievements have been the following:

- **An innovative end-to-end technique for deep dense depth estimation from a single indoor panorama** (Chapter 3). The main technical contributions of this work are the compact representation of the scene into vertical slices of the sphere, the exploitation long- and short-term relationships among slices to recover the equirectangular depth map, and the maintenance of high-resolution information in the extracted features even with a deep network.
- **A novel end-to-end deep learning solution for rapidly estimating a dense spherical depth map of an indoor environment from both dense visual data and sparse geometric data as input** (Chapter 4). This work significantly extends the above method by incorporating the processing of sparse (and even optional) depth information inside a lightweight single-branch network, employing a dynamic gating system to process together dense visual data and sparse geometric data.
- **An innovative method for layout reconstruction** (Chapter 5), that, differently from prior layout estimation solutions addresses the problem fully in 3D, using a graph-convolutional network for mapping a single 360-degree image into a tessellated bounding 3D surface representing the union of walls, floor, and ceiling. Gravity-aligned features are actively incorporated in the graph in a projection layer based on multi head self-attention, and specialized loss terms guide towards plausible solutions even in presence of massive clutter and occlusions.

6.2 Discussion and future directions

As illustrated in the previous chapters, my work has resulted in methods and implementations that have introduced important conceptual contributions and have shown to achieve beyond-state-of-the-art performance on a number of benchmark datasets.

While I refer to the individual chapters to an in-depth analysis of the results obtained on the individual tasks, there are some common considerations that can be made. First of all, all three techniques exploit specific characteristics of the capture setup (in particular, gravity-alignment) and of the imaged environment (in particular, a world-space alignment with gravity that makes it possible to exploit regularities

of vertical features along the horizontal direction). These characteristics have consistently led to network designs that exploit asymmetric contractions and various ways to combine long- and short-range features. As the various ablation tests have shown, the specific networks designed provide sizeable advantages over more generic alternatives, which demonstrates the benefit of creating custom solutions for interior capture, rather than using generic networks for outdoor or generic-shape 3D reconstruction. Creating specific networks, however, has also the disadvantage of relying on specific characteristics on the environments, leading to major failures as soon as the imaged environment does not match with the expected ones. While more robust than geometry-reasoning methods, the solutions devised still present limitations in terms of applicability, as shown in the failure case analyses presented in the previous chapter.

Another important limitation of the current solutions, which is, however, currently shared with all the competing methods (see discussions in Chapter 3, Chapter 4, and Chapter 5), is in terms of size of processed input. While the presented solutions are generally lightweight and the network design is scalable, all the tests have generally been performed at image sizes that are smaller than what is currently achievable with panoramic cameras. All the available benchmarks are typically performed at the 1024x512 resolution, and seldom at larger size, while industrial cameras are more detailed. One important avenue for future work is, thus, to evaluate the scaling of these techniques to larger datasets. This will require not only the scaling of the networks, but, also the generation of large annotated datasets to serve as ground truth.

The problems that I have tackled, moreover, have also a different nature. Deep dense estimation or completion is, in itself, a problem that requires a well-defined per-pixel output, while layout reconstruction is a more abstract task. While the solution presented here is significantly more flexible than prior ones, since we can generate a reasonably complex layout homeomorphic to a sphere that can include a variety of features, including large free-form surfaces joining at sharp angles, while competing solution are typically limited to Manhattan or Atlanta-world environments. Such a representation can be useful for a variety of needs (see Chapter 5), but is far from being an accurate representation of all possible environments. In particular, reconstruction by deformation from a single origin generates denser and more detailed meshes near the origin, and less detailed ones as one moves away from the origin and occlusions increase, and thus the precision depends on mesh tessellation size. Moreover, the spherical mesh topology is far from being sufficient to represent all sorts of architectural environments, since several elements of the architectural structure, such as pillars, stairs, septal walls or openings cannot be represented with a single closed surface. Including holes

(doors, windows) seems feasible as a direct extension of our end-to-end single pass method deforming a spherical mesh, while extending the approach to other topologies is not trivial. Moreover, the method could be also improved by taking into account, as for depth completion, the optional availability of sparse depth information.

Our monocular reconstruction methods have been applied to obtain geometry and layout, but, in principle they can be extended to new problems such as semantic extraction and reconstruction of the visual channel. A particular case of view synthesis that I have experimented with, in a follow-up work with respect to what presented in this this is diminished reality [13]. By exploiting concepts coming from depth estimation, where we synthesize per-pixel information for all the visible pixels, and layout estimation, where we have concentrated only on the permanent structures (walls, floor, and ceiling), we have designed a network that, given suitable examples, estimates the depth and the color of the imaged room emptied of all clutter [13]. As for the networks presented in Chapter 3 and Chapter 4, the input and outputs are both in equirectangular format, and provided as per-pixel information. We have shown how this representation can serve as a basis for many image editing operations.

The application to visual synthesis/room emptying shows how the designed networks can serve as building blocks for more complex applications, including additional channels. Another area where we see important future works is in the area of (sparse) multi-view reconstruction. In particular, a straightforward extension of our monocular analysis methods would be to exploit them for cases in which we capture a minimum amount of data in a multi-room environment (e.g., one or two photos per room), without going towards full multi-view. This setting is very common, and research solutions, instead of starting from (a large set of) common features among views, try to first extract the maximum amount of information from single views, to then exploit in a later fusion phase [177]. Since the methods discussed in this thesis have shown remarkable performance in single-image analysis, it can be expected that they can also benefit such extreme multi-view pipelines.

Our work on SliceNet [10] (Chapter 3) has been the subject of a number of follow-ups that have built upon it, analyzed our behavior, and/or used it as baseline for further enhancements. In particular, Yu et al. [178] have shown that reflective objects, that are not handled directly by our method, are likely to produce artifacts. As an example, in Fig. 6.1), column 2 (originally included in Yu et al. [178]), artifacts are present in the case of mirrors or windows. Since reflecting materials are abundant in interior environments, one future direction is to improve SliceNet (as well as our other solutions) to better handle these situations. The problem is challenging,

as the detection of mirrors in single-view situations is a malformed problem that requires imposing priors. Since we work in restricted environments (indoors), we can expect these mirrors/reflecting images to share some common characteristics (e.g., stemming the typical shape and location of windows in common apartments), and we can expect that data-driven solutions could learn those hidden relations from data. The creation of challenging data sets with realistic mirrors and windows would be an important contribution for creating solid more robust indoor reconstruction methods.

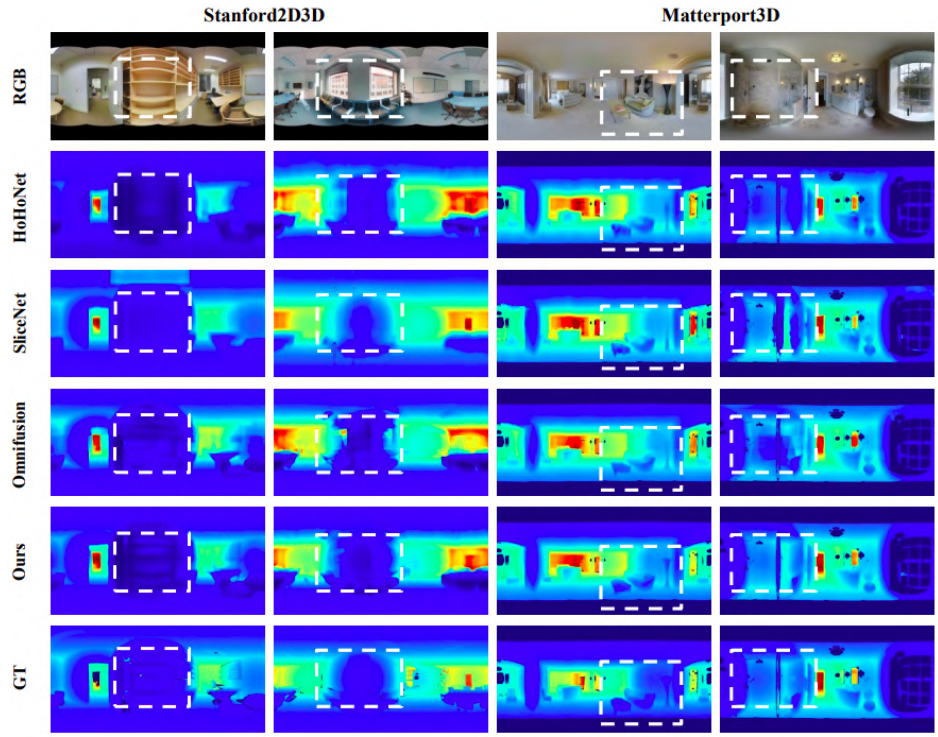


Figure 6.1: ***Examples of failure with reflective materials.*** *Original image published by Yu et al. [178]. Our method (SliceNet [10]) is in the second row.*

One possible direction of improvement of the methods for reconstruction from panoramic imaging is to take into account also the characteristics of the different setups used to capture panoramic images. As hardware solutions are variable, the captured images have different distortion characteristics [133], that could be taken into account to improve the quality of reconstructions. This would entail, however, not only the creation of distortion-specific methods (e.g., in terms of specific losses),

but also the creation of datasets that include those distortions, much as we have done for the simulation of laser scanning.

Another very interesting future direction is to study/analyze how our methods, based on supervised learning, would work in a self-supervised scenario, that would replace comparisons with ground truth with consistency measures. A recent example is the work of Wang et al. [131], that analyzes a self-supervised problem for monocular 360 depth estimation. To do that, their training process takes three adjacent panoramas extracted from video sequences and estimates the depth map and camera motions, thus replacing the need for ground truth data with the need for a multi-view training dataset. In this work, moreover, Wang et al. [131] also show how the violation of gravity alignment constraints negatively affects solutions that exploit them [10]. This effect was already studied in our work, and did not pose problems in a single-view setting, where the training dataset was gravity-aligned and at inference time it was possible to perform alignment prior to entering the network.

While our work targeted single-view estimation, a future extension would be to expand them in a multi-view context. One future direction for 3D reconstruction concerns volumetric reconstruction [179, 180, 181] using the truncated signed distance function (TSDF) representation inside approaches to generate consistent scene geometry from the fusion of multiple depth maps. As a representative example, Jang et al. [179] propose an approach designed for short trajectories of an omnidirectional video camera to get 3D reconstruction, facing not just depth estimation but also posed camera estimation, spherical rectification (aligning epipolar lines with horizontal image scanlines) and texture atlas reconstruction. The integration of our indoor-specific solutions for layout estimation and depth estimation within this class of approaches is an interesting avenue for future work. On one hand, our methods could provide more refined and regularized depth maps for fusion in specific classes of indoor environments, thanks to the incorporation of specific constraints (e.g., Atlanta-world and/or gravity alignment). On the other hand, our methods, in a multi-view setting, could also be revised to take into account multi-view consistency, eventually also in a self-supervised framework [131].

6.3 Publications

The scientific results obtained during this PhD work also appeared in related publications, for which I significantly contributed to the conceptualization, methodology, and validation of the developed method. These main publication, sorted by their introduction in this thesis, are the following:

- **SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation.**
  Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti, In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Pages 11531-11540, 2021. Selected as oral presentation.
  DOI: CVPR46437.2021.01137.
  — This is the original work that introduced the concept of slicing and gravity-aligned features for solving depth inference from a single omnidirectional image (Chapter 3). I have significantly contributed to the methodology, implementation, testing, and validation of the method.

- **Deep Panoramic Depth Prediction and Completion for Indoor Scenes.**
  Giovanni Pintore, Eva Almansa, Armando Sanchez, Giorgio Vassena, and Enrico Gobbetti, in Computational Visual Media, 2023.
  DOI: 10.1007/s41095-023-0358-0 — This is the original work that introduced a lightweight single-branch network, which employs a dynamic gating system to process together dense visual data and sparse geometric data, exploiting the concept of slicing and gravity-aligned features from a single omnidirectional image. Also, it is introduced a new augmentation strategy to make the model robust to different types of sparsity, including those generated by various structured light sensors and LiDAR setups (Chapter 4) expands over the previous approach by also exploiting optional sparse depth information, without any assumption on the sparsity pattern. I am joint first author of this work, to which I have contributed very significantly in all phases, including conceptualization, methodology, implementation, testing, and validation of the method, and can be considered my main achievement.

- **Deep3DLayout: 3D reconstruction of an indoor layout from a spherical panoramic image.**
  Giovanni Pintore, Eva Almansa, Marco Agus, and Enrico Gobbetti. 2021. ACM Trans. Graph. 40, 6, Article 250, 12 pages. 2021
  DOI: 10.1145/3478513.3480480
  — This is the original work that are exploited important 3D properties of indoor environments in the design. In particular, gravity-aligned features are actively incorporated in the graph in a projection layer that exploits the recent concept of multi head self-attention, and specialized losses guide towards plausible solutions even in presence of massive clutter and occlusions. (Chapter 5). I have significantly contributed to the conceptualization, methodology, implementation, testing, and validation of the method.

In addition, during the course of my thesis, I have also contributed to the following related publication, which have not been included in this work:

- **Instant Automatic Emptying of Panoramic Indoor Scenes.**
  Giovanni Pintore, Marco Agus, Eva Almansa, and Enrico Gobbetti, Proc. ISMAR. and published in IEEE Transactions on Visualization and Computer Graphics, vol. 28, no. 11, pp. 3629-3639, 2022. [Journal Article]
  DOI: 10.1109/TVCG.2022.3202999
  — In this work is introduced a novel light-weight end-to-end deep network that, from an input 360° image of a furnished indoor space automatically returns, with very low latency, an omnidirectional photorealistic view and architecturally plausible depth of the same scene emptied of all clutter. In this case, I have contributed to the validation of the approach by performing tests on all the included benchmarks, both coming from publicly available sources and custom user-captured data.

6.4 Demonstration videos

In the context of the EVOCATION project, I have also illustrated the outcomes of my research in the following demonstration videos that is available on the project web site at the URL evocation.eu/videos/:

- **Pilot 2 - indoor mapping for AEC: Automatic 3D reconstruction of structured indoor environments** — Demo video.
  This video presents the results of applying the techniques presented in this thesis to both publicly available benchmark data and data captured within the EVOCATION project.

Bibliography

[1] Giovanni Pintore, Claudio Mura, Fabio Ganovelli, Lizeth Fuentes-Perez, Renato Pajarola, and Enrico Gobbetti. "State-of-the-art in Automatic 3D Reconstruction of Structured Indoor Environments". In: *Comput. Graph. Forum* 39.2 (2020), pp. 667–699.

[2] Satoshi Ikehata, Hang Yang, and Yasutaka Furukawa. "Structured Indoor Modeling". In: *Proc. ICCV*. 2015, pp. 1323–1331.

[3] 3DVista. *3DVista:Professional Virtual Tour software*. https://www.3dvista.com. 1999.

[4] StructionSite. *VideoWalk*. https://www.structionsite.com/products/videowalk/. 2016.

[5] Reconstruct Inc. *Reconstruct: A Visual Command Center*. https://www.reconstructinc.com/. 2016.

[6] Matterport. *Matterport3D*. https://github.com/niessner/Matterport. [Accessed: 2023-03-03]. 2017.

[7] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. "A Survey of Surface Reconstruction from Point Clouds". In: *Comput. Graph. Forum* 36.1 (2017), pp. 301–329.

[8] Giovanni Pintore, Claudio Mura, Fabio Ganovelli, Lizeth Fuentes-Perez, Renato Pajarola, and Enrico Gobbetti. "Automatic 3D Reconstruction of Structured Indoor Environments". In: *SIGGRAPH 2020 Courses*. 2020, 10:1–10:218.

[9] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. *CVPR2023 Tutorial on Automatic 3D modeling of indoor structures from panoramic imagery*. http://vic.crs4.it/vic/cvpr2023-tutorial-pano. 2023.

[10] Giovanni Pintore, Marco Agus, Eva Almansa, Jens Schneider, and Enrico Gobbetti. “SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation”. In: *Proc. CVPR*. 2021, pp. 11536–11545.

[11] Giovanni Pintore, Eva Almansa, Armando Sanchez, Giorgio Vassena, and Enrico Gobbetti. “Deep Panoramic Depth Prediction and Completion for Indoor Scenes”. In: *Computational Visual Media* (2023). DOI: 10.1007/s41095-023-0358-0.

[12] Giovanni Pintore, Eva Almansa, Marco Agus, and Enrico Gobbetti. “Deep3DLayout: 3D Reconstruction of an Indoor Layout from a Spherical Panoramic Image”. In: *ACM TOG* 40.6 (2021), 250:1–250:12.

[13] Giovanni Pintore, Marco Agus, Eva Almansa, and Enrico Gobbetti. “Instant Automatic Emptying of Panoramic Indoor Scenes”. In: *IEEE Transactions on Visualization and Computer Graphics* 28.11 (2022), pp. 3629–3639. DOI: 10.1109/TVCG.2022.3202999.

[14] Tarek Abu Haila, Moonisa Ahsan, Eva María Almansa Aranega, Joao Cardoso, Paolo Cignoni, Lizeth Joseline Fuentes Perez, Enrico Gobbetti, Fabio Marton, Renato Pajarola, Ruggero Pintus, Giovanni Pintore, Martin Ritz, Arslan Siddique, Armando Arturo Sánchez Alcázar, Matteo Sgrenzaroli, Pedro Santos, and Mana Takhsha. *Interim report on the development of new geometry and material acquisition techniques and processing methods*. Deliverable D2.2. EVOCATION MCSA-ITN-2018 813170, Sept. 2022. URL: https://cordis.europa.eu/project/id/813170/results.

[15] Moonisa Ahsan, Eva María Almansa Aranega, Lizeth Joseline Fuentes Perez, Enrico Gobbetti, Fabio Marton, Renato Pajarola, Ruggero Pintus, Giovanni Pintore, Armando Arturo Sánchez Alcázar, and Matteo Sgrenzaroli. *New technologies for high-level structured reconstruction with visual and depth sensing*. Deliverable D2.5. EVOCATION MCSA-ITN-2018 813170, Feb. 2023. URL: https://cordis.europa.eu/project/id/813170/results.

[16] Armando Arturo Sánchez Alcázar, Matteo Sgrenzaroli, Giorgio Paolo Maria Vassena, Eva María Almansa Aranega, Enrico Gobbetti, Giovanni Pintore, Lizeth Joseline Fuentes Perez, Luciano Arnaldo Romero Calla, and Renato Pajarola. *Indoor Mapping for AEC*. Deliverable 6.2. EVOCATION MCSA-ITN-2018 813170, May 2023. URL: https://cordis.europa.eu/project/id/813170/results.

[17] Giovanni Pintore, Ruggero Pintus, Fabio Ganovelli, Roberto Scopigno, and Enrico Gobbetti. “Recovering 3D existing-conditions of indoor structures from spherical images”. In: *Computers & Graphics* 77 (Dec. 2018), pp. 16–29. DOI: 10.1016/j.cag.2018.09.013.

[18] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. “Unsupervised Monocular Depth Estimation with Left-Right Consistency”. In: *Proc. CVPR*. 2017, pp. 270–279.

[19] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. “A comparison and evaluation of multi-view stereo reconstruction algorithms”. In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. Vol. 1. IEEE. 2006, pp. 519–528.

[20] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. “A survey of structure from motion*.” In: *Acta Numerica* 26 (2017), pp. 305–364.

[21] Thiago L. T. da Silveira, Paulo G. L. Pinto, Jeffri Murrugarra-Llerena, and Cláudio R. Jung. “3D Scene Geometry Estimation from 360° Imagery: A Survey”. In: *ACM Comput. Surv.* 55.4 (Nov. 2022). ISSN: 0360-0300. DOI: 10.1145/3519021. URL: https://doi.org/10.1145/3519021.

[22] Michael Firman. “RGBD datasets: Past, present and future”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2016, pp. 19–31.

[23] R. Cabral and Y. Furukawa. “Piecewise Planar and Compact Floorplan Reconstruction from Images”. In: *Proc. CVPR*. 2014, pp. 628–635.

[24] Tero Jokela, Jarno Ojala, and Kaisa Väänänen. “How people use 360-degree cameras”. In: *Proc. International Conference on Mobile and Ubiquitous Multimedia*. 2019, pp. 1–10.

[25] Sunghoon Im, Hyowon Ha, François Rameau, Hae-Gon Jeon, Gyeongmin Choe, and In So Kweon. “All-Around Depth from Small Motion with a Spherical Panoramic Camera”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 156–172.

[26] Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. “6-DOF VR videos with a single 360-camera”. In: *2017 IEEE Virtual Reality (VR)*. 2017, pp. 37–44. DOI: 10.1109/VR.2017.7892229.

[27] Marc Eder, Mykhailo Shvets, John Lim, and Jan-Michael Frahm. “Tangent Images for Mitigating Spherical Distortion”. In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[28] Manuel Rey-Area, Mingze Yuan, and Christian Richardt. "360monodepth: High-resolution 360deg monocular depth estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3762–3772.

[29] Kari Pulli, Habib Abi-Rached, Tom Duchamp, Linda G Shapiro, and Werner Stuetzle. "Acquisition and visualization of colored 3D objects". In: *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*. Vol. 1. IEEE. 1998, pp. 11–15.

[30] Xin Xiong, Haipeng Xiong, Ke Xian, Chen Zhao, Zhiguo Cao, and Xin Li. "Sparse-to-Dense Depth Completion Revisited: Sampling Strategy and Graph Construction". In: *Proc. ECCV*. 2020, pp. 682–699.

[31] Ville V Lehtola, Harri Kaartinen, Andreas Nüchter, Risto Kaijaluoto, Antero Kukko, Paula Litkey, Eija Honkavaara, Tomi Rosnell, Matti T Vaaja, Juho-Pekka Virtanen, et al. "Comparison of the selected state-of-the-art 3D indoor scanning and point cloud generation methods". In: *Remote sensing* 9.8 (2017), p. 796.

[32] Kang Chen, Yu-Kun Lai, and Shi-Min Hu. "3D indoor scene modeling from RGB-D data: a survey". In: *Computational Visual Media* 1 (2015), pp. 267–278.

[33] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. "Floornet: A unified framework for floorplan reconstruction from 3d scans". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 201–217.

[34] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. "Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2661–2670.

[35] Ruiqi Guo and Derek Hoiem. "Support surface prediction in indoor scenes". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 2144–2151.

[36] Zhaoyin Jia, Andrew Gallagher, Ashutosh Saxena, and Tsuhan Chen. "3d-based reasoning with blocks, support, and stability". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1–8.

[37] New York University. *NYU-Depth V2*. https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html. [Accessed: 2023-03-03]. 2012.

[38] Yiyi Liao, Jun Xie, and Andreas Geiger. "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D". In: *arXiv.org* 2109.13410 (2021).

[39] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. “DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene From Sparse LiDAR Data and Single Color Image”. In: *Proc. CVPR*. 2019, pp. 3308–3317.

[40] Alican Mertan, Damien Jade Duff, and Gozde Unal. “Single image depth estimation: An overview”. In: *Digital Signal Processing* (2022), p. 103441.

[41] Erick Delage, Honglak Lee, and Andrew Y Ng. “A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image”. In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 2418–2428.

[42] V. Hedau, D. Hoiem, and D. Forsyth. “Recovering the spatial layout of cluttered rooms”. In: *Proc. ICCV*. 2009, pp. 1849–1856.

[43] David C Lee, Martial Hebert, and Takeo Kanade. “Geometric reasoning for single image structure recovery”. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 2136–2143.

[44] James M Coughlan and Alan L Yuille. “Manhattan world: Compass direction from a single image by bayesian inference”. In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. IEEE. 1999, pp. 941–947.

[45] Giovanni Pintore, Marco Agus, and Enrico Gobbetti. “AtlantaNet: Inferring the 3D Indoor Layout from a Single 360 Image Beyond the Manhattan World Assumption”. In: *Proc. ECCV*. 2020, pp. 432–448.

[46] Claudio Mura, Oliver Mattausch, Alberto Jaspe Villanueva, Enrico Gobbetti, and Renato Pajarola. “Automatic Room Detection and Reconstruction in Cluttered Indoor Environments with Complex Room Layouts”. In: *Computers & Graphics* 44 (Nov. 2014), pp. 20–32. DOI: 10.1016/j.cag.2014.07.005.

[47] David C Lee, Martial Hebert, and Takeo Kanade. “Geometric reasoning for single image structure recovery”. In: *Proc. CVPR*. 2009, pp. 2136–2143.

[48] Clara Fernandez-Labrador, José M Fácil, Alejandro Perez-Yus, Cédric Demonceaux, Javier Civera, and José J Guerrero. “Corners for Layout: End-to-End Layout Recovery from 360 Images”. In: *ArXiv e-print arXiv:1903.08094* (2019).

[49] Chuhang Zou, Jheng-Wei Su, Chi-Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung-Kuo Chu, and Derek Hoiem. “3D Manhattan Room Layout Reconstruction from a Single 360 Image”. In: *ArXiv e-print arXiv:1910.04099* (2019).

[50] Wenqi Xian, Zhengqi Li, Matthew Fisher, Jonathan Eisenmann, Eli Shechtman, and Noah Snavely. “UprightNet: geometry-aware camera orientation estimation from single images”. In: *Proc. ICCV*. 2019, pp. 9974–9983.

[51] R. Jung, A. S. J. Lee, A. Ashtari, and J. Bazin. “Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment”. In: *Proc. IEEE VR*. 2019, pp. 1–8.

[52] Benjamin Davidson, Mohsan S. Alvi, and Joao F. Henriques. “360 Camera Alignment via Segmentation”. In: *Proc. ECCV*. 2020, pp. 579–595.

[53] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel Recurrent Neural Networks”. In: *Proc. ICML*. 2016, pp. 1747–1756.

[54] Cheng Sun, Chi-Wei Hsiao, Min Sun, and Hwann-Tzong Chen. “HorizonNet: Learning room layout with 1D representation and pano stretch data augmentation”. In: *Proc. CVPR*. 2019, pp. 1047–1056.

[55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[56] Ethem Alpaydin. *Machine learning*. Mit Press, 2021.

[57] A. Saxena, M. Sun, and A. Y. Ng. “Make3D: Learning 3D Scene Structure from a Single Still Image”. In: *IEEE TPAMI* 31.5 (2009), pp. 824–840.

[58] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. “Deep learning for monocular depth estimation: A review”. In: *Neurocomputing* 438 (2021), pp. 14–33.

[59] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network”. In: *Advances in Neural Information Processing Systems 27*. 2014, pp. 2366–2374.

[60] D. Eigen and R. Fergus. “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture”. In: *Proc. ICCV*. 2015, pp. 2650–2658.

[61] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. “Deeper Depth Prediction with Fully Convolutional Residual Networks”. In: *Proc. 3DV*. 2016, pp. 239–248.

[62] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *Proc. CVPR*. 2016, pp. 770–778.

[63] Sophie Lambert-Lacroix and Laurent Zwald. “The adaptive BerHu penalty in robust regression”. In: *Journal of Nonparametric Statistics* 28 (2016), pp. 1–28.

[64] J. Lee, M. Heo, K. Kim, and C. Kim. “Single-Image Depth Estimation Based on Fourier Domain Analysis”. In: *Proc. CVPR*. 2018, pp. 330–339.

[65] F. Liu, Chunhua Shen, and Guosheng Lin. “Deep convolutional neural fields for depth estimation from a single image”. In: *Proc. CVPR*. 2015, pp. 5162–5170.

[66] Peng Wang, Xiaohui Shen, Zhe Lin, S. Cohen, B. Price, and A. Yuille. “Towards unified depth and semantic prediction from a single image”. In: *Proc. CVPR*. 2015, pp. 2800–2809.

[67] Y. Cao, Z. Wu, and C. Shen. “Estimating Depth From Monocular Images as Classification Using Deep Fully Convolutional Residual Networks”. In: *IEEE Trans. on Circuits and Systems for Video Technology* 28.11 (2018), pp. 3174–3182.

[68] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci. “Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation”. In: *Proc. CVPR*. 2018, pp. 3917–3925.

[69] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. “Deep Ordinal Regression Network for Monocular Depth Estimation”. In: *Proc. CVPR*. 2018, pp. 2002–2011.

[70] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. “Unsupervised Learning of Monocular Depth Estimation and Visual Odometry With Deep Feature Reconstruction”. In: *Proc. CVPR*. 2018, pp. 340–349.

[71] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. “OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas”. In: *Proc. ECCV*. 2018, pp. 453–471.

[72] Yinda Zhang, Shuran Song, Ping Tan, and Jianxiong Xiao. “PanoContext: A Whole-Room 3D Context Model for Panoramic Scene Understanding”. In: *Proc. ECCV*. 2014, pp. 668–686.

[73] Chuhang Zou, Alex Colburn, Qi Shan, and Derek Hoiem. “LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image”. In: *Proc. CVPR*. 2018, pp. 2051–2059.

[74] Shang-Ta Yang, Fu-En Wang, Chi-Han Peng, Peter Wonka, Min Sun, and Hung-Kuo Chu. “DuLa-Net: A Dual-Projection Network for Estimating Room Layouts from a Single RGB Panorama”. In: *Proc. CVPR*. 2019, pp. 3363–3372.

[75] H. Cheng, C. Chao, J. Dong, H. Wen, T. Liu, and M. Sun. “Cube Padding for Weakly-Supervised Saliency Prediction in 360 Videos”. In: *Proc. CVPR*. 2018, pp. 1420–1429.

[76] Yu-Chuan Su and Kristen Grauman. “Learning Spherical Convolution for Fast Features from 360 Imagery”. In: *Advances in Neural Information Processing Systems 30*. 2017, pp. 529–539.

[77] Keisuke Tateno, Nassir Navab, and Federico Tombari. “Distortion-Aware Convolutional Filters for Dense Prediction in Panoramic Images”. In: *Proc. ECCV*. 2018, pp. 732–750.

[78] Gregoire Payen de La Garanderie, Amir Atapour Abarghouei, and Toby P. Breckon. “Eliminating the Blind Spot: Adapting 3D Object Detection and Monocular Depth Estimation to 360 Panoramic Imagery”. In: *Proc. ECCV*. 2018, pp. 812–830.

[79] Y. Su and K. Grauman. “Kernel Transformer Networks for Compact Spherical Convolution”. In: *Proc. CVPR*. 2019, pp. 9434–9443.

[80] Nikolaos Zioulis, Antonis Karakottas, Dimitris Zarpalas, Federic Alvarez, and Petros Daras. “Spherical View Synthesis for Self-Supervised 360° Depth Estimation”. In: *Proc. 3DV*. 2019, pp. 690–699.

[81] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. “BiFuse: Monocular 360 Depth Estimation via Bi-Projection Fusion”. In: *Proc. CVPR*. 2020, pp. 462–471.

[82] M. Eder, P. Moulon, and L. Guan. “Pano Popups: Indoor 3D Reconstruction with a Plane-Aware Network”. In: *Proc. 3DV*. 2019, pp. 76–84.

[83] W. Yin, Y. Liu, C. Shen, and Y. Yan. “Enforcing Geometric Constraints of Virtual Normal for Depth Prediction”. In: *Proc. ICCV*. 2019, pp. 5683–5692.

[84] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. “Geometric Structure Based and Regularized Depth Estimation From 360 Indoor Imagery”. In: *Proc. CVPR*. 2020, pp. 889–898.

[85] Wei Zeng, Sezer Karaoglu, and Theo Gevers. “Joint 3D Layout and Depth Prediction from a Single Indoor Panorama Image”. In: *Proc. ECCV*. 2020, pp. 666–682.

[86] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. “Confidence propagation through CNNs for guided sparse depth regression”. In: *IEEE TPAMI* 42.10 (2019), pp. 2423–2436.

[87] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. “Learning guided convolutional network for depth completion”. In: *IEEE TIP* 30 (2020), pp. 1116–1129.

[88] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. “Sparse and noisy LiDAR completion with RGB guidance and uncertainty”. In: *Proc. MVA*. 2019, pp. 1–6.

[89] Sihaeng Lee, Janghyeon Lee, Doyeon Kim, and Junmo Kim. “Deep architecture with cross guidance between single image and sparse LiDAR data for depth completion”. In: *IEEE Access* 8 (2020), pp. 79801–79810.

[90] Saif Imran, Yunfei Long, Xiaoming Liu, and Daniel Morris. “Depth Coefficients for Depth Completion”. In: *Proc. CVPR*. 2019, pp. 12438–12447.

[91] Changgyoon Oh, Wonjune Cho, Yujeong Chae, Daehee Park, Lin Wang, and Kuk-Jin Yoon. “BIPS: Bi-modal Indoor Panorama Synthesis via Residual Depth-Aided Adversarial Learning”. In: *Proc. ECCV*. 2022, pp. 352–371.

[92] Pushmeet Kohli Nathan Silberman Derek Hoiem and Rob Fergus. “Indoor Segmentation and Support Inference from RGBD Images”. In: *Proc. ECCV*. 2012, pp. 746–760.

[93] Yinda Zhang and Thomas Funkhouser. “Deep depth completion of a single RGB-D image”. In: *Proc. CVPR*. 2018, pp. 175–185.

[94] Xinjing Cheng, Peng Wang, Yanqi Zhou, Chenye Guan, and Ruigang Yang. “Omnidirectional Depth Extension Networks”. In: *Proc. ICRA*. 2020, pp. 589–595.

[95] Yu-Kai Huang, Tsung-Han Wu, Yueh-Cheng Liu, and Winston H Hsu. “Indoor depth completion with boundary consistency and self-attention”. In: *Proc. CVPR Workshops*. 2019, pp. 1070–1078.

[96] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. “Generative image inpainting with contextual attention”. In: *Proc. CVPR*. 2018, pp. 5505–5514.

[97] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Proc. MICCAI*. 2015, pp. 234–241.

[98] Jinsun Park, Kyungdon Joo, Zhe Hu, Chi-Kuei Liu, and In So Kweon. “Non-local Spatial Propagation Network for Depth Completion”. In: *Proc. ECCV*. 2020, pp. 120–136.

[99] Yu-Kai Huang, Yueh-Cheng Liu, Tsung-Han Wu, Hung-Ting Su, Yu-Cheng Chang, Tsung-Lin Tsou, Yu-An Wang, and Winston H. Hsu. “S3: Learnable Sparse Signal Superdensity for Guided Depth Estimation”. In: *Proc. CVPR*. 2021, pp. 16706–16716.

[100] Vitor Guizilini, Rares Ambrus, Wolfram Burgard, and Adrien Gaidon. “Sparse Auxiliary Networks for Unified Monocular Depth Prediction and Completion”. In: *Proc. CVPR*. 2021, pp. 11078–11088.

[101] Ruyu Liu, Guodao Zhang, Jiangming Wang, and Shuwen Zhao. “Cross-modal 360° depth completion and reconstruction for large-scale indoor environment”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.12 (2022), pp. 25180–25190.

[102] Chuhang Zou, Jheng Wei Su, Chi Han Peng, Alex Colburn, Qi Shan, Peter Wonka, Hung Kuo Chu, and Derek Hoiem. “Manhattan Room Layout Reconstruction from a Single 360 Image: A Comparative Study of State-of-the-Art Methods”. In: *International Journal of Computer Vision* 129 (2021), pp. 1410–1431.

[103] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. “LED2-Net: Monocular 360 Layout Estimation via Differentiable Depth Rendering”. In: *Proc. CVPR*. 2021, pp. 12956–12965.

[104] Derek Hoiem, Alexei A. Efros, and Martial Hebert. “Recovering Surface Layout from an Image”. In: *International Journal of Computer Vision* 75.1 (2007), pp. 151–172.

[105] J. Xu, B. Stenger, T. Kerola, and T. Tung. “Pano2CAD: Room Layout from a Single Panorama Image”. In: *Proc. WACV*. 2017, pp. 354–362.

[106] H. Yang and H. Zhang. “Efficient 3D Room Shape Recovery from a Single Panorama”. In: *Proc. CVPR*. 2016, pp. 5422–5430.

[107] Cheng Sun, Min Sun, and Hwann-Tzong Chen. “HoHoNet: 360 Indoor Holistic Understanding with Latent Horizontal Features”. In: *Proc. CVPR*. 2021, pp. 2573–2582.

[108] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. “Structured3D: A Large Photo-realistic Dataset for Structured 3D Modeling”. In: *Proc. ECCV*. 2020, pp. 519–535.

[109] Stanford University. *BuildingParser Dataset*. http://buildingparser.stanford.edu/dataset.html. [Accessed: 2019-09-25]. 2017.

[110] Raehyuk Jung, Aiden Seung Joon Lee, Amirsaman Ashtari, and Jean-Charles Bazin. “Deep360Up: A Deep Learning-Based Approach for Automatic VR Image Upright Adjustment”. In: *Proc. IEEE VR*. 2019, pp. 1–8.

[111] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. “Atlas: End-to-End 3D Scene Reconstruction from Posed Images”. In: *Proc. ECCV*. 2020, pp. 1–18.

[112] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. “Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images”. In: *Proc. ECCV*. 2018, pp. 55–71.

[113] G. Gkioxari, J. Johnson, and J. Malik. “Mesh R-CNN”. In: *Proc. ICCV*. 2019, pp. 9784–9794.

[114] Edward Smith, Scott Fujimoto, Adriana Romero, and David Meger. “GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects”. In: *Proc. ICML*. 2019, pp. 5866–5876.

[115] CRS4 VISUAL COMPUTING. *CRS4 ViC Research Datasets*. http://vic.crs4.it/download/datasets/. [Accessed: 2019-09-25]. 2018.

[116] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. “Recognizing scene viewpoint using panoramic place representation”. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 2695–2702.

[117] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. “The Replica Dataset: A Digital Replica of Indoor Spaces”. In: *ArXiv e-print arXiv:1906.05797* (2019).

[118] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. “Habitat: A Platform for Embodied AI Research”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[119] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. “Habitat 2.0: Training Home Assistants to Rearrange their Habitat”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2021.

[120] Jiale Xu, Jia Zheng, Yanyu Xu, Rui Tang, and Shenghua Gao. “Layout-Guided Novel View Synthesis From a Single Indoor Panorama”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[121] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. “OpenMVG: Open multiple view geometry”. In: *International Workshop on Reproducible Research in Pattern Recognition*. Springer. 2016, pp. 60–74.

[122] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. “MatryODShka: Real-time 6DoF Video View Synthesis using Multi-Sphere Images”. In: *European Conference on Computer Vision (ECCV)*. Aug. 2020. URL: https://visual.cs.brown.edu/matryodshka.

[123] Armando Arturo Sánchez Alcázar; Giovanni Pintore; Matteo Sgrenzaroli. *Indoor3Dmapping dataset*. https://doi.org/10.5281/zenodo.6367381. [Accessed: 2022-04-25]. 2022.

[124] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. “State of the Art on 3D Reconstruction with RGB-D Cameras”. In: *Comput. Graph. Forum* 37.2 (2018), pp. 625–652.

[125] Yang Yang, Shi Jin, Ruiyang Liu, Sing Bing Kang, and Jingyi Yu. “Automatic 3d indoor scene modeling from single panorama”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3926–3934.

[126] Yeonkun Lee, Jaeseok Jeong, Jongseob Yun, Wonjune Cho, and Kuk-Jin Yoon. “SpherePHD: Applying CNNs on a spherical polyhedron representation of 360° images”. In: *Proc. CVPR*. 2019, pp. 9181–9189.

[127] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”. In: *Proc. NIPS*. 2015, pp. 802–810.

[128] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proc. ICCV*. USA, 2015, pp. 1026–1034.

[129] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *ArXiv e-print arXiv:1412.6980* (2014).

[130] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. “Neural window fully-connected crfs for monocular depth estimation”. In: *Proc. CVPR*. 2022, pp. 3916–3925.

[131] Fu-En Wang, Yu-Hsuan Yeh, Yi-Hsuan Tsai, Wei-Chen Chiu, and Min Sun. “Bifuse++: Self-supervised and efficient bi-projection fusion for 360 depth estimation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.5 (2022), pp. 5448–5460.

[132] Zhijie Shen, Chunyu Lin, Kang Liao, Lang Nie, Zishuo Zheng, and Yao Zhao. “PanoFormer: Panorama Transformer for Indoor 360-degree Depth Estimation”. In: *Proc. ECCV*. 2022, pp. 195–211.

[133] Shaohua Gao, Kailun Yang, Hao Shi, Kaiwei Wang, and Jian Bai. “Review on panoramic imaging and its applications in scene understanding”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–34.

[134] Zishuo Zheng, Chunyu Lin, Lang Nie, Kang Liao, Zhijie Shen, and Yao Zhao. “Complementary bi-directional feature compression for indoor 360-degree semantic segmentation with self-distillation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023, pp. 4501–4510.

[135] Adrian Lopez-Rodriguez, Benjamin Busam, and Krystian Mikolajczyk. “Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data”. In: *International Journal of Computer Vision* (2022). DOI: 10.1007/s11263-022-01726-1.

[136] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. “Monocular Depth Estimation with Affinity, Vertical Pooling, and Label Enhancement”. In: *Proc. ECCV*. 2018, pp. 232–247.

[137] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. “Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End”. In: *Proc. CVPR*. 2020, pp. 12014–12023.

[138] Jason Ku, Ali Harakeh, and Steven L Waslander. “In defense of classical image processing: Fast depth completion on the cpu”. In: *Proc. CRV*. 2018, pp. 16–22.

[139] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* 12.37 (2013), pp. 1231–1237.

[140] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 6000–6010.

[141] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. “Contextual residual aggregation for ultra high-resolution image inpainting”. In: *Proc. CVPR*. 2020, pp. 7508–7517.

[142] Yanchao Yang, Alex Wong, and Stefano Soatto. “Dense Depth Posterior (DDP) From Single Image and Sparse Range”. In: *Proc. CVPR*. 2019, pp. 3348–3357.

[143] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. “Free-form image inpainting with gated convolution”. In: *Proc. ICCV*. 2019, pp. 4471–4480.

[144] Fangchang Ma and Sertac Karaman. “Sparse-to-dense: Depth prediction from sparse depth samples and a single image”. In: *Proc. ICRA*. 2018, pp. 4796–4803.

[145] Kujiale.com. *Structured3D Data*. [Accessed: 2019-09-25]. 2019. URL: https://structured3d-dataset.org/.

[146] Tao Wu, Hao Fu, Bokai Liu, Hanzhang Xue, Ruike Ren, and Zhiming Tu. “Detailed Analysis on Generating the Range Image for LiDAR Point Cloud Processing”. In: *Electronics* 10.11 (2021), p. 1224.

[147] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. “Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving”. In: *arXiv preprint arXiv:1906.06310* (2019).

[148] Yunwang Li, Sumei Dai, Yong Shi, Lala Zhao, and Minghua Ding. “Navigation simulation of a Mecanum wheel mobile robot based on an improved A* algorithm in Unity3D”. In: *Sensors* 19.13 (2019), p. 2976.

[149] Vasileios Gkitsas, Vladimiros Sterzentsenko, Nikolaos Zioulis, Georgios Albanis, and Dimitrios Zarpalas. “PanoDR: Spherical Panorama Diminished Reality for Indoor Scenes”. In: *Proc. CVPR*. 2021, pp. 3716–3726.

[150] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. “Image Inpainting for Irregular Holes Using Partial Convolutions”. In: *Proc. ECCV*. 2018, pp. 89–105.

[151] Fisher Yu and Vladlen Koltun. “Multi-Scale Context Aggregation by Dilated Convolutions”. In: *Proc. ICLR (Poster)*. 2016, 1:1–1:13.

[152] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. “Pluralistic image completion”. In: *Proc. CVPR*. 2019, pp. 1438–1447.

[153] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. “Fast and accurate deep network learning by exponential linear units (ELUs)”. In: *Proc. ICLR Posters* (2015).

[154] Vitor Guizilini, Jie Li, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. “Robust semi-supervised monocular depth estimation with reprojected distances”. In: *Proc. CoRL*. 2020, pp. 503–512.

[155] Jesús Morales, Victoria Plaza-Leiva, Anthony Mandow, Jose Antonio Gomez-Ruiz, Javier Serón, and Alfonso García-Cerezo. “Analysis of 3D scan measurement distribution with application to a multi-beam lidar on a rotating platform”. In: *Sensors* 18.2 (2018), p. 395.

[156] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE TIP* 13.4 (2004), pp. 600–612.

[157] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. “Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera”. In: *Proc. IEEE ICRA*. 2019, pp. 3288–3295.

[158] Wenchao du, Hu Chen, Hongyu Yang, and Yi Zhang. “Depth Completion Using Geometry-Aware Embedding”. In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pp. 8680–8686. DOI: 10.1109/ICRA46639.2022.9811556.

[159] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. “Penet: Towards precise and efficient image guided depth completion”. In: *Proc. IEEE ICRA*. 2021, pp. 13656–13662.

[160] A. Eldesokey, M. Felsberg, and F. Khan. “Confidence Propagation through CNNs for Guided Sparse Depth Regression”. In: *IEEE TPAMI* 42.10 (2020), pp. 2423–2436.

[161] Alastair Harrison and Paul Newman. “Image and sparse laser fusion for dense scene reconstruction”. In: *Field and Service Robotics*. 2010, pp. 219–228.

[162] Junyi Liu and Xiaojin Gong. “Guided depth enhancement via anisotropic diffusion”. In: *Pacific-Rim conference on multimedia*. 2013, pp. 408–417.

[163] Xuehan Xiong and Daniel Huber. “Using Context to Create Semantic 3D Models of Indoor Environments”. In: *Proc. BMVC*. 2010, pp. 45.1–45.11.

[164] Kevin Matzen, Michael F. Cohen, Bryce Evans, Johannes Kopf, and Richard Szeliski. “Low-cost 360 Stereo Photography and Video Capture”. In: *ACM TOG* 36.4 (2017), 148:1–148:12.

[165] Sheng Yang, Beichen Li, Yan-Pei Cao, Hongbo Fu, Yu-Kun Lai, Leif Kobbelt, and Shi-Min Hu. “Noise-Resilient Reconstruction of Panoramas and 3D Scenes Using Robot-Mounted Unsynchronized Commodity RGB-D Cameras”. In: *ACM TOG* 39.5 (2020), 152:1–152:15.

[166] Ping Chao, Chao-Yang Kao, Yushan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. “HarDNet: A Low Memory Traffic Network”. In: *Proc. ICCV*. 2019, pp. 3551–3560.

[167] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. “Self-attention generative adversarial networks”. In: *Proc. ICML*. 2019, pp. 7354–7363.

[168] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. “Discrete Differential-Geometry Operators for Triangulated 2-Manifolds”. In: *Visualization and Mathematics III*. 2003, pp. 35–57.

[169] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. “A Sketch-Based Interface for Detail-Preserving Mesh Editing”. In: *Proc. SIGGRAPH*. 2005, pp. 1142–1147.

[170] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. “Automatic differentiation in PyTorch”. In: *Proc. NIPS Workshop on Autodiff*. 2017.

[171] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. “Accelerating 3D Deep Learning with PyTorch3D”. In: *arXiv preprint arXiv:2007.08501* (2020).

[172] Alessandro Muntoni and Paolo Cignoni. *PyMeshLab*. 2021. DOI: 10.5281/zenodo.4438750.

[173] Dawson-Haggerty et al. *trimesh*. Version 3.2.0. 2019. URL: https://trimesh.org/.

[174] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. “Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction”. In: *ACM TOG* 36.4 (2017), 78:1–78:13.

[175] Alex Flint, Christopher Mei, David Murray, and Ian Reid. “A Dynamic Programming Approach to Reconstructing Building Interiors”. In: *Proc. ECCV*. 2010, pp. 394–407.

[176] Yuan Dong, Chuan Fang, Zilong Dong, Liefeng Bo, and Ping Tan. “PanoContextFormer: Panoramic Total Scene Understanding with a Transformer”. In: *arXiv preprint arXiv:2305.12497* (2023).

[177] Mohammad Amin Shabani, Weilian Song, Makoto Odamaki, Hirochika Fujiki, and Yasutaka Furukawa. “Extreme structure from motion for indoor panoramas without visual overlaps”. In: *Proc. CVPR*. 2021, pp. 5703–5711.

[178] Haozheng Yu, Lu He, Bing Jian, Weiwei Feng, and Shan Liu. “PanelNet: Understanding 360 Indoor Environment via Panel Representation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 878–887.

[179] Hyeonjoong Jang, Andreas Meuleman, Dahyun Kang, Donggun Kim, Christian Richardt, and Min H Kim. “Egocentric scene reconstruction from an omnidirectional video”. In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), pp. 1–12.

[180] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. “SimpleRecon: 3D reconstruction without 3D convolutions”. In: *European Conference on Computer Vision*. Springer. 2022, pp. 1–19.

[181] Huiyu Gao, Wei Mao, and Miaomiao Liu. “VisFusion: Visibility-aware Online 3D Scene Reconstruction from Videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17317–17326.

Appendix A

Curriculum Vitae

I am Eva Almansa and will briefly summarized my achievements. I worked as an Early-stage Researcher (ESR) in the Visual and Data-intensive Computing (ViDiC) Group of CRS4, Italy from 2020-2023, supported by a **Marie Sklodowska-Curie Fellowship** in the **EVOCATION MSCA-ITN** (H2020 Research and Innovation Funding Programme grant 81370). My scientific and research interest are Computer Vision and Computer Graphics. Within the context of EVOCATION, I was pursuing my PhD at University of Cagliari (UniCa) from 2020-2023 (exam 2024) on the topic of Automatic 3D reconstruction in indoor environments from a single omnidirectional image. I hold a **Bachelor of Science in Computer Science and Artificial Intelligence** in 2015, and a **Master Degree in Data Science** in 2017, both awarded by the University of Granada (Spain).

Career focus

Highly motivated to continue in the field in **Computer Science**, especially finding clever models from **Machine Learning and Deep Learning techniques**. As for programming, I am most familiar with Python, Pytorch, R, Spark and other 3D libraries such as Open3D, Trimesh, Pytorch3D, VTK (and more) and developing algorithms in Linux, also using Docker, Blender (as a beginner designer) and SQL/No-SQL tools. Regarding my expertise in Computer Vision, most of my learning has been during my PhD as a **Marie Sklodowska-Curie Fellow** (2020-2023) in the

EVOCATION project. Besides, I have been developing software professionally and, also, in research projects using Python and C++, as well as developing algorithms in both Linux and Microsoft Windows. Moreover, I have co-authored one conference paper and three journal articles related to Computer Vision and Computer Graphics, during my PhD and three more conference papers on optimization algorithms during my first years as a researcher.

Contact Information

Name	**Eva Almansa**
Portfolio	github.io/EvaAlmansa_Portfolio
LinkedIn	Linkedin.com/eva-almansa-2a582bb4
Semantic Scholar	semanticscholar.org/eva-almansa/16308166
ORCID	0000-0002-7288-0989

Student Volunteer

Student Volunteer at SIGGRAPH, 2022 (remote) and 2023 (in-person, Los Angeles USA).

Education

2020 - 2024	**Ph.D in Computer Science, specialty Computer Vision and Graphics / Marie Sklodowska-Curie Fellow** Doctoral program in Dept. of Mathematics and Computer Science, University of Cagliari (UniCa), Italy. Main topics: **Computer Vision, Computer Graphics and Research.** Techniques: The results achieved so far have been four approaches that I will briefly mention. The first one transforms a **panoramic image into a dense depth map**, the second approach transforms the **panoramic image into a mesh representation of the architectural layout**, the third one is focused on **diminished reality which gets an uncluttered environment with both geometric and image synthesis**, while the last one is **transform both RGB and sparse depth map into dense depth map**. These works have been published in major conferences/journals - **CVPR 2021, TOG/SIGGRAPH Asia 2021, TVCG/ISMAR 2022, and CVM 2023**.
2015 - 2017	**Master Degree in Data Science** University of Granada (Spain). Main topics: **Soft Computing, Machine Learning, Data Mining, Big Data and Research.** Master's Final Project published: Fernandez A., Almansa E., and Herrera F., **Chi-Spark-RS: An Spark-built evolutionary fuzzy rule selection algorithm in imbalanced classification for big data problems**, 2017 IEEE Int. Conf. on Fuzzy Systems, 2017: 1-6. **DOI**: 10.1109/fuzz-ieee.2017.8015520 **Source Code**: github.com/EvaAlmansa/Spark-Chi-RS
2011 - 2015	**Bachelor of Science in Computer Science and Artificial Intelligence** University of Granada (Spain). Main topics: **Computer Vision, Soft Computing, Machine Learning, Techniques of intelligent systems, Advanced computer models**, and more.

Work Experience

2020 - 2023	**Marie Sklodowska-Curie Early Stage Researcher** Visual and Data-Intensive Computing (ViDiC) Group, Center for Advanced Studies, Research and Development in Sardinia (CRS4), Italy. EVOCATION project (www.evocation.eu) **My work' summary at CRS4** crs4.it.people/eva.almansa Topic: Computer Vision and Computer Graphics focused on 3D reconstruction in indoor environments.
2020 - 2020	**Machine Learning Scientist** Funditec SME, Madrid (Spain). Topic: Data science focused on industrial renewable energy and writing project proposals.
2018 - 2020	**Data Scientist and Software Developer** TurningTables SME, Granada (Spain). Topic: Simulation of electrical systems in an Energy Community System. Data analysis and Software development. SCRUM methodology.
2018 - 2018	**Researcher** University of Oviedo (Spain). Topic: Intelligent soft sensors for automotive rechargeable batteries. Data science and application of soft computing techniques. Almansa et al. **Health Assessment of Automotive Batteries Through Computational Intelligence-Based Soft Sensors: An Empirical Study**, International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 Leon, Spain. **DOI**: 10.1007/978.3.319.67180.2.5

2015 - 2018	**Data Scientist and Optimization Algorithm Developer** Joint collaboration between University of Granada and SHS Consultores SME, Sevilla (Spain). Topic: High performance computing and data analysis for the planning and optimization of resources in highly complex scenarios, like passenger transport and aviation. Data analysis in Big Data context.

Selected Scientific Publications

Journal Articles

1. **Deep Panoramic Depth Prediction and Completion for Indoor Scenes.** Giovanni Pintore*, Eva Almansa*, Armando Sanchez, Giorgio Vassena, and Enrico Gobbetti, Computational Visual Media, 2023.
   DOI: 10.1007/s41095-023-0358-0 — View PDF.

2. **Instant Automatic Emptying of Panoramic Indoor Scenes.** Giovanni Pintore, Marco Agus, Eva Almansa, Enrico Gobbetti, IEEE Transactions on Visualization and Computer Graphics, 28(11): 3629-3639, November 2022.
   DOI: 10.1109/TVCG.2022.3202999 — View PDF.

3. **Deep3DLayout: 3D reconstruction of an indoor layout from a spherical panoramic image.** Giovanni Pintore, Eva Almansa, Marco Agus, Enrico Gobbetti, ACM Transactions on Graphics, 40(6): 250:1-250:12, December 2021.
   DOI: 10.1145/3478513.3480480. — View PDF.

Conference Papers

1. **SliceNet: deep dense depth estimation from a single indoor panorama using a slice-based representation.**
   Giovanni Pintore, Eva Almansa, Marco Agus, Jens Schneider, Enrico Gobbetti, In Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Pages 11531-11540, 2021.
   DOI: 10.1109/CVPR46437.2021.01137 — View PDF.