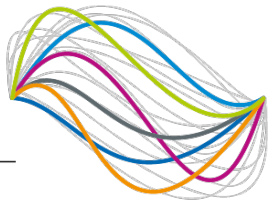
# Indoor Scene Understanding using Non-Conventional Cameras

Clara Fernandez-Labrador

Universidad Zaragoza

1542



UBFC

UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

**THESE DE DOCTORAT DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE**

**PREPAREE A L'UNIVERSITE DE BOURGOGNE ET A L'UNIVERSITE DE ZARAGOZA**

Ecole doctorale n° 37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat en instrumentation et informatique de l'image

Par

Mme FERNANDEZ LABRADOR CLARA

# Indoor Scene Understanding using Non-Conventional Cameras

Thèse soutenue le 03/12/2020

Composition du Jury :

| | | |
|---|---|---|
| M Chateau Thierry | Professeur à l'Université Clermont Auvergne | Président |
| M Aubry Mathieu | Chercheur HDR, Ecole des Ponts ParisTech | Rapporteur |
| M Wolf Christian | Maître de Conférences HDR INSA de Lyon | Rapporteur |
| M Funkhouser Thomas | Professeur à l'Université de Princeton | Examinateur |
| Mme Murillo Ana C. | Professeur à l'Université de Zaragoza | Examinatrice |
| M Demonceaux Cédric | Professeur à l'Université de Bourgogne-Franche-Comté | Codirecteur de thèse |
| M Guerrero José J. | Professeur à l'Université de Zaragoza | Codirecteur de thèse |

To my family.

# Acknowledgements

I will never forget these three years of my life. I am not going to say that the road was easy, but I do say that I enjoyed and learned from each and every moment and that I was incredibly fortunate to meet a big number of great people on my way.

I want to start with a big thanks to my advisors. To Josechu, because I would probably never have embarked on this adventure if not for you, and to Cédric, for opening so many doors to me. To both of you, thank you for all the advice, for believing in me, the discussions, and also for all the freedom you gave me during these three years. Thanks for all the encouragement and support.

I spent the first year and a half of my PhD in Zaragoza, my hometown and where I received most of my education (except for my two Erasmus stays in Italy). I met Alejandro the first day I set foot in the laboratory as he was going to supervise my master thesis. Huge thanks for being my mentor and showing me what the research world was like. Thanks for your patience and your endless humor, which was especially reflected in all your reviews and feedback, and also in all our after-work plans! It has always been a pleasure working with you. Also all my respect to Jesús, for properly "vampirizing" the younger generations. I really like your energy, your pulp fiction dance, and your wise advice. Chema, we have the record in hours drinking coffee and discussing research and life. You have always inspired me and remembered me that I was capable of everything when I had my doubts. Companion of all (non-virtual) conferences, trips and hikes. This is an enormous thanks! Jose, for teaching me there is no bad moment to drink a beer and philosophize. I need to highlight here our wine tasting in the surroundings of Le Creusot, a favourite day! Berta, because we shared the bachelor's, master's and PhD and sooo many trips, from weekend trips to the Pyrenees to new year's eve in Lapland. Lorenzo, adopted son of the lab, thank you for making me cry with laughter on many occasions and for being the music of our group. I also want to thank Julia, because it does not happen all the time that you can learn from your students, and it was the case with you. I'm very glad that I had the opportunity to supervise your bachelor thesis. I want to thank as well Javier Civera for his valuable feedback and suggestions in the shared publications and to Ana Cris, for

organizing the reading group and encouraging all of us to share these weekly meetings. To Iñigo, for your always inappropriate comments. Besides, there were uncountable juepinchos, barbeques, fiestas de veterinaria and enriching coffee conversations, and for those I thank as well Leo, Edgar, Rafa, Carlos, Richard, Melani and Emma.

In the summer of 2018, I was very lucky to join the International Computer Vision Summer School. It was my very first international event of the PhD and there, I not only learned a lot of new things, but I also met awesome people that I happily welcomed for the festivities of my city and with whom I continue meeting at various conferences and events. Especial mention goes to Patri, Raul, Andrés, Daniel and Andrea. Thanks for all the great moments!

In April 2019 I moved to Le Creusot to start the second half of my PhD. There I was incredibly fortunate to meet Yanis, a master student of Cédric (now PhD student in Paris) who was my absolute friend during my stay there. Thanks for your endless happiness and for contributing to squeeze all the possibilities of Le Creusot by running up and down the mountains, doing French-Spanish tandems, aqua cycling and more! Despite being a very small city, Le Creusot gave me many experiences, where I can include climbing to the highest places, day trips to Chalon-sur-Saône or Beaune, and participating in a Ramadan diner. I thank for those to David, Marc, Daniel, Tibault, Devesh, and Luigia.

Zurich, my hair stands on end. It is incredible how in such a short time you can feel 100% at home, mixed at the same time with all the excitement of the unknown. I cannot thank more Luc Van Gool for giving me the opportunity to do a research visiting in the Computer Vision Laboratory at ETH and with it, to get to know the city where I want to live at least for the next few years of my life.

The only way to start this is with an infinite thanks to Ajad and Danda. The best good cop/bad cop couple. Working with you was a really exciting and inspiring experience. At the research level, you are a real whirlwind, always energetic, wanting to discuss new ideas, positive, passionate about research, talented and really enjoying what you do. At the level of everything else, you are super attentive, always down for a karaoke, a dinner, a match of kicker, a boating journey,... I cannot enumerate all the plans we did and do together, nor express what I have learned from you two. Thank you for taking care of me since my first day. Ajad, thanks for the thousands of tips, your visits to the B floor to discuss and talk about everything and for the online sport sessions during the lock down. Danda, my academic sibling, thanks for your always constructive criticisms and for discovering me the happier time of the day, 10:10. Also, in the same pack, I cannot forget about Thomas, always around and the best Spanish

speaker of the lab, this team wouldn't be the same without you, although you put ketchup on the sweet potatoes...

To my everyday team in CVL, thanks for all the lunch times, sushi dinners, fruitful discussions and infinite laughs in the lab. Menelaos, my companion of gossiping at any time in the lab with coffee on hand and the only one calling me by my two names and four surnames. Dario, the bad guy of Dietikon and best listener, thanks for sharing with me the love for the Asia mensa and for the Tannenbar bar coffee. Evan, all imagination and creativity, with more ideas for start-ups than any one else. Thanks for all the board game nights and for the alternative dances. David, always happy to finish my food (and all the food in general) and great salsa dance teacher. I hope your sofa is ok. To the rest of BIWIs, you are all incredible and a great group of people, thank you for integrating me so well from the beginning and for the thousands of activities: apéros, boating, paintball, board game nights, beers in bqm and a great etc!

In March 2020, due the world-wide lockdown caused by the Covid-19, I had to run back to Zaragoza, where I spent a couple of months at home with my family again, of course totally unexpected. They were weird times, but with them it was easier, and I got the time I needed to write a big part of the pages present in this thesis.

July started with a new trip to Zurich to start my summer internship at Disney Research Studios. Hayko, we already met during my first period in Zurich, but I never imagined such a good connection! Sharing brain was never so easy and working was never so fun. Thanks for welcoming me on the topogang board and letting me work with you in such a cool project! Besides that, thanks for the good vibes, the overfitted kicker games, the commuting by kayak and the starlinks from the rooftop. Continuing with the Disney team, thanks Sik for making me a better programmer, for the gin-tonics and all the homemade food that reminds me of home! And of course to Leo, discrete and always with a smile (happy to eat burrata/ borracha) and ready to destroy us in kicker. I want to thank as well Simone, for all the Thursdays and for thinking about me for the postdoc position at MTC.

Let's move to the Oculus team (saying "my Facebook friends" always created confusion). I love the fact that I met all of you, randomly and separately, at different conferences, and never imagined that we would all end up being friends and living in the same city. To Mariano, for being the forerunner of the karaoke nights (that ended-up being known as Clara-Oke nights), for organizing awesome hikes and for your eternal patience and humor when I sabotage you in all the games. I still owe you some churros. To Alberto, for all the padel matches, the board game nights, your black

humor and all the wines. To Alejo and Ruben, for the best korsikas in the summer. It is always good to have a small "Spanish Mafia" when one is abroad.

I cannot forget about my flatmates in Zurich. I can't imagine a better combination of people to live with or do home office with. Thanks for being also very good friends, for calming me down in every moment of stress and for all the improvised celebrations and parties, apéros and walks to discover the city.

And of course, infinite thanks to all those who were already there before starting this adventure and who lived with me, almost in first person, each of these experiences.

First of all, to Francesco. We met weeks before I started my PhD and, since then, you have been my life partner and my particular Michelin Star Chef. Thank you for all the love, support and inspiration. You always made this path easier and exciting at the same time, making me feel a perfect balance regardless of the distance. You also blindly believed in me. I couldn't be more fortunate to have you. I love you.

To my parents and sister, whose favorite sport has been to accompany me by car, regardless of the kilometers, to each move. You have been a fundamental pillar of this trip and of life.

And to the rest of my lifelong friends, because they won the gold medal for organizing the best farewells and welcomes to me, because they already assumed that I will leave and return constantly. For coming to visit me wherever I go. For your infinite love.

Clara Fernández Labrador
December 2020.

# Abstract

Humans understand environments effortlessly, under a wide variety of conditions, by the virtue of visual perception. Computer vision for similar visual understanding is highly desirable, so that machines can perform complex tasks by interacting with the real world, to assist or entertain humans. In this regard, we are interested in indoor environments, where humans spend nearly all their lifetime. This thesis specifically addresses the problems that arise during the quest of the hierarchical visual understanding of indoor scenes, and the required sensing mechanism for the same. On the side of sensing the wide 3D world, we propose to use non-conventional cameras, namely 360° imaging and 3D sensors. On the side of understanding, we aim at three key aspects: room layout estimation; object detection, localization and segmentation; and object category shape modelling, for which novel and efficient solutions are provided.

The focus of this thesis is on the following underlying challenges. First, the estimation of the 3D room layout from a single 360° image is investigated, which is used for the highest level of scene modelling and understanding. We exploit the assumption of Manhattan World and deep learning techniques to propose models that handle invisible parts of the room on the image, generalizing to more complex layouts. At the same time, new methods to work with 360° images are proposed, highlighting a special convolution that compensates the equirectangular image distortions. Second, considering the importance of context for scene understanding, we study the problem of object localization and segmentation, adapting the problem to leverage 360° images. We also exploit layout-objects interaction to lift detected 2D objects into the 3D room model. The final line of work of this thesis focuses on 3D object shape analysis. We use an explicit modelling of non-rigidity and a high-level notion of object symmetry to learn, in an unsupervised manner, 3D keypoints that are order-wise correspondent as well as geometrically and semantically consistent across objects in a category. Our models advance state-of-the-art on the aforementioned tasks, when each evaluated on respective reference benchmarks.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

"*Sometimes science is more art than science. Lot of people don't get that.*"
— Rick Sanchez

## 1.1 Motivation

*Can we create autonomous algorithms that understand the scenes as humans do?* The world is made of objects with a wild variety of shapes, appearances and structures. We humans see the world through the images formed by the light reflected from the objects in our environment. These images allow our brain to understand the shape and texture of the objects, crucial for higher level understandings. Moreover, we understand the visual scene effortlessly, under a wide variety of conditions, and this is because human perception emerges from the genetic code fueled by millions of years of evolution and, at the same time, from a lifetime of experience. This understanding is achieved from a specific viewpoint, from which the scene is observed. In order to automatically reproduce this understanding with algorithms, we need a camera to capture the light of the scene at some location and additionally, intelligent models that reason about the visual input.

The human visual system perceives an horizontal angle of view of about 140°, without considering the eyes movement. For comparison, the horizontal field of view (FoV) of conventional cameras ranges from 40° to 60°. Moreover, human stereopsis allows a 3D perception that may not be directly achieved from single 2D conventional images. Such reduced field of view or the lack of depth perception, crucially limits the goal of developing intelligent systems to match the performance of human vision.

An increasing demand to extend the cameras FoV, led to the appearance of fisheye and catadioptric lenses, achieving up to 360° of horizontal FoV. Today, 360° images can be easily obtained with special lenses, but also with camera arrays or automatic image stitching algorithms [18]. Ultra wide-angle lenses have demonstrated to be beneficial, particularly in indoor scenarios, for many different tasks including indoor scene understanding [170] or visual odometry [172]. This is not surprising, since a FoV of 360° allows to see the whole scene at once, giving a strong context about the space, allows to track features longer, greatly increasing the robustness of visual localization, and allows features to be more evenly distributed in space, which stabilizes pose estimation. Additionally, with the rapid development of 3D acquisition technologies, 3D sensors are becoming increasingly available and affordable, including 3D scanners, LIDAR sensors, and RGB-D cameras. Depth perception significantly contributes to solve several challenging tasks related to 3D object shape analysis [2, 94]. As an example, when we look at an image of a 3D object, we see only its projection from a specific viewpoint. Therefore, different viewpoints may create entirely different renders, limiting the use of 2D images in applications where shape information is critical, or when shape abstraction itself is the scope of the study. Therefore, the choice of the sensor has a tremendous impact on the robustness and accuracy of the developed models. We are particularly motivated to see if scenes can be understood better beyond the traditional sensors, betting for 360 images and 3D point clouds. And, to be more specific, we are interested on exploring this understanding on indoor scenarios.

*Why are indoor scenes "special"?* Around 10.000 years ago, there was a time of transition from a hunter-gatherer mode of subsistence to an agricultural way of life. This enabled humans to live in more permanent settlements, to the point that nowadays, humans spend approximately 90 percent of their time in interior spaces [73], which means more than six days per week. This shocking fact translates into an urgent need to understand well indoor scenarios to improve life quality indoors. We therefore are concerned about giving machines the required visual sensing mechanism to understand the indoor environments where they often operate, to assist or entertain. This understanding is not trivial, as there are hundreds of different man-made scenarios. To put an example, the Places Database[174] is a repository that contains 10 million scene pictures, comprising a large and diverse list of the types of environments encountered in the world, consistent with real-world frequencies of occurrence. They divide the dataset into 365 scene categories and classify them into indoors, outdoors and outdoors man-made classes. As a result, 159 categories out of the 365 belong to indoor scenes, 80 to outdoors and 159 to outdoors man-made (some of the categories

are classified as outdoors and outdoors man-made at the same time). According to these numbers, around the 80% of the images taken are from man-made scenarios. Man-made scenarios can differ for many reasons, mainly due to the use for which they are designed or due to the time or cultural environment in which they are built. These differences are usually in terms of the scenes layout or the type of objects we can find inside them e.g. supermarkets and theaters. Even between different scenes that belong to the same category we can see these differences, as one does not act the same way in a home bedroom, a hotel bedroom or a nursery. Achieving a complete understanding of indoor scenes, would equip the discipline of computer vision with many exciting tools, thus making it more powerful and ubiquitous. But the challenges that need to be solved are numerous. To start, the layout can range from very simple i.e. 4 walls, to highly complex layouts as in museums. To continue, the diversity of objects of interest is very high, many of which appear infrequently. Indoor spaces usually contain many instances, generating in the visual scenes clutter and high degree of occlusions, which makes very hard to know the separability of objects and surfaces. Not to say that objects come in various shapes, sizes and in different poses. Additionally, while some spaces can be recognized by global spatial properties e.g. corridors, others are better characterized by the objects inside e.g. bookstores [108]. In fact, proposed solutions to problems such as scene recognition or semantic segmentation [108, 7] have demonstrated a high performance on outdoor scenarios, while performing poorly in the indoor domain. This suggests that indoor scenes require special and dedicated algorithms for their understanding.

Solving the aforementioned challenges is not only stimulating, but also necessary for many exciting applications. One clear example, that is revolutionizing the real estate industry, are the virtual tours of homes[1], which are helping sellers sell faster while feeling more confident, and buyers understand the home layout and imagine what it would be like to call it home. Such virtual tours are also becoming very popular to visit museums or art galleries, in part as a consequence of the COVID-19 lockdown across the world. In fact, the Arts & Culture initiative by Google currently offers virtual visits of about 500 museums throughout the world, including the MoMA, Amsterdam's Rijksmuseum, the National Gallery and the Palace of Versailles, to name a few. Indoor scene understanding is also vital for autonomous mobile robots such us vacuum cleaners[2], surveillance drones, or assistive robots, that need to move freely in the same space as humans do, or even get to move in complex spaces where less

[1] https://www.zillow.com/marketing/3d-home/
[2] https://www.irobot.es/roomba/

exposure of humans is desired, like buildings under construction, mines or hospital areas with contagious people. Autonomous mobile robots can be also useful for visual data collection or 3D modeling and domestic robots with cognitive abilities can be specially helpful to take care of visually impaired people, which represent the 17% of the world's population [98], and elderly people, whose number is expected to reach 1.5 billion by the year 2050 [147]. Another example, which is getting more and more demanded, is virtual and augmented reality for education, games or interior design[3]. This technology requires a detailed level of understanding of the scene for various reasons, such as delimiting a safe area for the user or reasoning about the real-virtual objects interaction.

We strongly believe that bringing together indoor scene understanding and non-conventional cameras, such as 360 and 3D sensors, has many possibilities. Some of them are scene recognition, structure analysis such as room layout reconstruction or floor plan estimation, object detection, object-layout interaction, shape analysis, object pose estimation, saliency prediction, etc. While all of these problems are exciting, we focus in this thesis on some of them, selected for their relevance, not only for the task itself, but also for their potential use or benefit for other vision tasks. More specifically, the contributions of this thesis are focused on the hierarchical understanding of indoor environments, that we summarize in three different levels of details:

- Layout level: understanding of the main structure of an indoor scenario.
- Scene level: localization of objects and their distribution in the 3D scene.
- Object level: geometry and shape modelling of large collection of objects.

## 1.2 Context and Challenges

Computer Vision is the science that seeks for giving computers a full three-dimensional scene understanding from images, by emulating the brain's ability to make sense of what the eyes see. *Is this idea that challenging?* In the early 1960s, Seymour Papert, one of the pioneers of artificial intelligence, did not think so, and proposed to a couple of his students to solve 'Computer Vision' as a summer project [100]. However, today we are still far away from seeing methods that achieve human-level robustness and generalization. So, *what makes Computer Vision so demanding and why do we care about it?*

[3] https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050

Visual perception is a very complex piece of our organic technology. It not only involves our eyes and visual cortex, but also takes into account our uncountable personal experiences and interactions with the world, as well as our abstract understanding of concepts and mental models of objects. In order to understand how our learning process works, early studies analyze the principles of object perception with human infants [133]. Understanding how we learn in our earliest stage of life, gives the hints as to how we have to design learning algorithms. Modern Computer Vision models aim at reproducing how our brain shapes all the inputs we receive, from the simplest features to the most detailed understanding, largely by observation of the geometry of the world. It is equally important to understand how positive experiences and failures help to the learning process, as human infants learn through a mixture of semi-supervised and unsupervised learning.

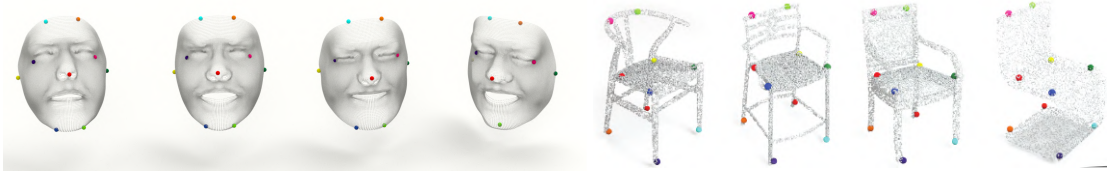
The early optimism of Seymour Papert led to huge improvements in computer vision models, and also in the capabilities of the computers that run them. The 1980's saw the backpropagation algorithm for neural networks being laid out by Geoffrey Hinton [116] and ten years later, Yann LeCun and Yoshua Bengio among others, proposed the first convolutional neural network (CNN) architecture [78]. The rapid advancements of Machine Learning and Deep Learning techniques [76] brought further life to the field 2012 onwards, where backpropagation and CNNs became ubiquitous in AI. And now, we live in an exciting time for computer vision, since we are producing more visual data than ever before, and we count on powerful algorithms to process it. Even if the field has been able to take great leaps in recent years, and even to surpass humans in some tasks [127, 58], significantly more efforts are required to achieve truly autonomous systems that enable complex tasks, like home robotics or autonomous driving. These complex tasks require a deep understanding of 3D scenes, across multiple levels, connecting vision, graphics and robotics research.

This thesis shows how to combine geometry and deep learning techniques for a hierarchical understanding of indoor environments. The proposed methods advance state-of-the-art at three different levels that are detailed below. An overview of our results is given in Figure 1.1.

**Layout level.** *What is the configuration of this room? how much space do I have?* These are the first questions that need to be answered when we arrive to a new space. In order to answer them, we need to know the scene structure. The scene structure can be simply defined by a set of geometric primitives. They can be a set of planes, corresponding to the walls, ceiling and floor. They can be defined by lines, representing

(a) Layout level: understanding of the main structure of an indoor scene.


(b) Scene level: localization and segmentation of objects and their distribution in the 3D scene.



(c) Object level: geometry and shape understanding of collection of objects per categories.

Fig. 1.1 Examples of the variety of algorithms developed in this thesis. Hierarchical understanding of indoor environments, at different levels of details: (a) layout level, (b) scene level, (c) object level.

the edges or intersections between planes. Or they can be points, as the corners of the room, being the intersection between three planes or two lines. Finding these geometric primitives is a complex task, as indoor scenes usually contain many object instances that partly occlude the walls. Moreover, depending on the camera viewpoint and the configuration of the room, some walls may occlude part of the room.

Indoor scenes have some advantages that can be leveraged to predict the room layout. Interior spaces have well-defined structures that can be described based on the main directions of the space. In 1999, Coughlan and Yuille [24] observed that, for the majority of the cases, we can identify three mutually orthogonal directions in the scene, notably in the indoor domain. They named this assumption "Manhattan world", whereby indoor scenes are three-dimensional grids, where all walls are at right angles to each other and to the floor and ceiling planes [25]. Shortly after, Schindler and Dellaert [123] extended this idea to include multiple groups of orthogonal directions, which they refer to as "Atlanta world". These main directions are given by the vanishing points, obtained as the intersection of parallel lines in the 3D world. Identifying the main directions, we are allowed to recover both extrinsic and intrinsic camera parameters, as well as to extract the 3D structure of the room from a single image. The Manhattan world assumption has led the vast majority of work on monocular layout estimation since 2006 [31], with impressive results. However, most of the works since then, use conventional images with reduced field of view, which limits the accessible spatial information [80, 60, 61, 124, 79]. As a consequence, there is a dominant simplification whereby previous works predict cuboid layouts, i.e., rectangular 3D boxes. In Chapter 2, we propose novel techniques to solve the 3D layout recovery problem, from 360 images. We use the aforementioned scene geometry assumptions, combined with deep learning techniques, to recover cuboid and non-cuboid layouts. An example of some qualitative results achieved by our proposed methods is shown in Figure 1.1a.

**Scene level.** *Which kind of objects are in this room? How are they organized inside the 3D layout?* We need to address the aforementioned questions in order to recognize the scenario, understand the free space available or interact with the objects in the 3D scene. The progress of object detectors has gone through two historical periods, the second starting in 2014 with the deep learning advancements. Early object detectors were built mainly based on handcrafted features. In 2001, P. Viola and M. Jones re-purposed classifiers to perform detection. To detect an object, an object classifier was evaluated using sliding windows at various locations and scales in a test image [149, 150]. Histogram of Oriented Gradients (HOG) feature descriptor was proposed in

2005 [29] to balance the feature invariance (including translation, scale, illumination, etc) and to discriminate different objects categories. Deformable Parts Models (DPM) [38], came at the peak of the traditional object detection methods. These methods detect objects by learning the relationships between HOG features of object parts, and introduced very valuable insights, like bounding box regression, which are still popular today. In 2013, Selective Search [146] proposed an alternative to sliding windows, by detecting regions with high "objectness". From 2014, the advancements in deep learning techniques revolutionised object detectors, achieving real-time and accurate predictions. Mainly, they can be divided into two-stage detectors, standing out the Region-based Convolutional Neural Netowrks (R-CNN) family [51, 50, 112, 57], and one-stage detectors like YOLO, among others, [109, 110, 85]. We discuss these modern methods and more in Chapter 3. In Chapter 3, considering the importance of context for scene understanding, we show how to adapt a deep learning based method to detect objects in indoor 360 images, and combine the detection with geometric reasoning about the 3D scene structure. By leveraging the contextual cues given by the 3D layout, we can benefit from a joint reasoning about the scene, e.g. objects are located in the free space defined by the room walls, usually aligned with the main directions, etc. An example of some qualitative results achieved by our proposed methods is shown in Figure 1.1b.

We offer solutions to the two fundamental problems presented in the layout level and the scene level using single large field of view images. Images are rich in color, texture, reside in a regular domain and, moreover, a large field of view provides a great context to achieve a high-level understanding of the scene, i.e., its structure and how and where the objects are found in this space. We bet for 360 images to exploit the full view of the scene offered, and propose novel techniques to deal with their inherent distortions. On the contrary, if we want to understand the details, geometry, shape and 3D configuration of the objects, we need to go deeper into its own structure. In such case, it is preferable to work directly on 3D. By nature, 3D point clouds are irregular (with regard to their density), unordered (and therefore invariant to permutations of their members), lack color information and often suffer from sensor noise. For example, on average, 30% of chairs have more than 50% missing depth pixels [54]. Nevertheless, geometric and shape reasoning in 3D space is preferred, where the objects of interest also reside.

**Object level.** *What does a table look like? what is the relative pose between all the chairs around it?* Automatic understanding of shapes in 3D data is an active field of research. Landmarks or keypoint locations are usually used for shape abstraction, being a key building block for downstream tasks such as 3D reconstruction [96], geometric registration [164], shape generation [167] or human body pose [14]. Explore collections of object shapes allows a deep understanding of the geometry and the semantics of objects. If we further focus on understanding shapes per object category, we can also aim at getting correspondences between them, favouring the aforementioned applications. However, it is not trivial to get correspondences between 3D objects in a category. 3D objects go through shape variations, either because of being deformable (like the human body) or when two different objects of one category are compared. Moreover, not all objects in a category necessarily have the same semantic parts, e.g., chair arms are not present in all the category instances. The challenges are exacerbated in the practical cases of misaligned data, which deserves special attention since real data is never aligned. Figure 1.1c contains some qualitative results of our work presented in Chapter 4, where we propose to model shapes in a category, with non-rigidity, in the keypoints space. We also include the notion of symmetry to deal with input misalignments and improve keypoints predictions. We satisfactorily find 3D keypoints, automatically and in an unsupervised manner, that meaningfully represent objects' shape and their correspondences can be simply established order-wise across all objects in a category.

All the methods presented in this thesis combine traditional geometric reasoning with novel deep learning techniques. In recent years, we have observed a host of works addressing traditional computer vision problems with deep learning, often ignoring the problem theory and relying on ground truth labels for supervision. But we must not forget the underlying reasoning that is behind the tackled problems. Understanding the scene geometry, e.g., depth, shape, pose, motion, etc., is essential for many vision tasks. Building learning methods that directly leverage the geometric properties of the scene or allow a combination, can improve and simplify the learning process, compared to the case of just relying on semantic representations. We will see as well, in the Chapter 4 of the document, that geometry becomes especially useful for unsupervised learning, where no ground-truth labels are available. Unsupervised learning is particularly exciting because getting large amounts of labeled data is not always possible and it offers a far more scalable framework.

## 1.3 Contributions

The next section details each specific line of work followed, pointing out the contributions and associated publications that came out as a result of our work [43, 42, 41, 53, 40].

### 1.3.1 Peer-Reviewed Publications

**Room Layout Estimation.** Our first line of work is presented in Chapter 2. The chapter shows an evolution of our research on the layout level, divided in two different sections (Section 2.4 and Section 2.5). We not only advance state of the art on the layout estimation problem, but we also demonstrate how to effectively exploit the advantages of the 360 images, without the limitations of their inherent distortions.

Our contributions in this line are mainly motivated to provide fast methods that enable a faithful prediction of the indoor room layout. At the time of writing, previous works needed expensive pre and post-processing stages and were limited to cuboid layouts (rooms with only four walls). In our first work, we propose a data abstraction method to get potential structural room corners. Based on those corners, we only need a reduced number of layout hypotheses to achieve closed room solutions that satisfy the actual shape of the scenes. During the hypotheses generation, our model is able to place non-visible corners in the image, so that rooms meaningfully satisfy the Manhattan World assumption. Our second work demonstrates how to create CNNs to work directly with 360 images, exploiting all their context and minimizing extra pre- and post- processing time. It also presents a special type of convolution that adapts the size and shape of the kernel to the equirectangular image distortions. We demonstrate that equirectangular convolutions have several advantages to work with 360 images, allowing to recover more accurate room layouts, much faster than previous methods.

Associated publications:

- "Corners for Layout: End-to-End Layout Recovery from 360 Images"
  **Clara Fernández Labrador***, J. María Fácil*, Alejandro Pérez Yus, Cédric Demonceaux, Javier Civera, José J. Guerrero.
  IEEE Robotics and Automation Letters.
  ICRA, 2020. Paris, France.
  WiCV with CVPR 2019. Long Beach, California.

- "PanoRoom: From the Sphere to the 3D Layout"
  **Clara Fernández Labrador**, J. María Fácil, Alejandro Pérez Yus, Cédric

Demonceaux, José J. Guerrero
3DRMS with ECCV 2018. Munich, Germany.

- "Layouts from Panoramic Images with Geometry and Deep Learning"
  **Clara Fernández Labrador**, Alejandro Pérez Yus, Gonzalo López Nicolás, José J. Guerrero
  IEEE Robotics and Automation Letters.
  IROS, 2018. Madrid, Spain.
  WiCV with ECCV 2018. Munich, Germany.

**Object Recognition.** The second line of work, related to the scene level, is presented in Chapter 3. In this chapter we present, up to our knowledge, the first object detection system working directly on 360 images, focused on indoor scene understanding. We extend an existing dataset [67], with object semantic segmentation and object localization labels on 360 images. We study how to adapt existing CNNs, in this case designed for the task of object detection, to match the nature of the equirectangular image input. Additionally, we show the potential of exploiting the 2D room layout to improve the instance segmentation masks, and how to leverage the 3D layout to generate 3D object bounding boxes directly from the instance masks.

Associated publication:

- "What's in my Room? Object Recognition on Indoor Panoramic Images"
  Julia Guerrero-Viu*, **Clara Fernández Labrador***, Cédric Demonceaux, José J. Guerrero.
  ICRA, 2020. Paris, France.

**Object Category Shape Modelling.** In the third line of work, we work on a more detailed understanding of the objects, with a focus on shape abstraction. We propose a method to learn 3D keypoints from a collection of objects of some category, so that they meaningfully represent objects' shape and their correspondences can be simply established order-wise across all objects. Up to our knowledge, we are the first proposing to solve category-specific 3D keypoints detection directly from 3D point clouds, with misaligned data and in an unsupervised manner. This work is related to the object level understanding and is presented in Chapter 4.

Associated publication:

- "Unsupervised Learning of Category-Specific Symmetric 3D Keypoints from Point Sets"
  **Clara Fernández Labrador**, Ajad Chhatkuli, Danda Pani Paudel, José J. Guerrero, Cédric Demonceaux, Luc Van Gool.
  ECCV, 2020. Glasgow, Scotland.

### 1.3.2 Open-Source Software/ Datasets

We have released the following open-source software and datasets:

- Source code and dataset for the work "Corners for Layout: End-to-End Layout Recovery from 360 Images" can be downloaded from the project website: https://cfernandezlab.github.io/CFL/
- 360 Scene Understanding. I created the GitHub repository *360 Scene Understanding*, where different tools to work with 360 images, developed during my PhD, are available: https://github.com/cfernandezlab/360-Scene-Understanding.
- The extended dataset for the work "What's in my Room? Object Recognition on Indoor Panoramic Images" can be downloaded from the project website: https://webdiis.unizar.es/~jguerrer/room_OR/.
- Source code for "Unsupervised Learning of Category-Specific Symmetric 3D Keypoints from Point Sets" can be downloaded from github: https://github.com/cfernandezlab/Category-Specific-Keypoints.

### 1.3.3 Research Stays

During my PhD I did the following research stays abroad:

**Computer Vision Laboratory, ETH Zurich.** From August 2019, I spent 7 wonderful months as a visiting researcher in the Computer Vision Laboratory at ETH Zurich lead by Professor Luc Van Gool, where I enjoyed working together with Dr. Ajad Chhatkuli and Dr. Danda Pani Paudel.

Additionally to our ECCV submission [40], we worked on the understanding of the Industry Foundation Classes (IFC) data model and on the development of translator tools between IFC and point clouds. The IFC schema is a standardized (ISO 16739-1:2018) data model intended to describe building industry data. The schema incorporates not only 3D geometry but also all the relevant data relating to the building,

its components and the project schedules. We also evaluated object detection pipelines on 3D point clouds generated from IFC models. This part of the work is not presented in the thesis.

**Disney Research Studios.** During the summer of 2020 I did a 3 months internship at Disney Research Studios in Zurich, Switzerland. During this stay, I had the opportunity to work under the supervision of Dr. Hayko Riemenschneider on 3D shape correspondences, at the intersection between computer vision and computer graphics.

### 1.3.4 Supervision of Students

During my PhD I also had several enriching and rewarding advising experiences.

- Juan Carlos Medina (Bachelor Thesis in Industrial Engineering at the University of Zaragoza – 2018): "Single View Layout Reconstruction".
- Julia Guerrero Campo (Bachelor Thesis in Computer Science at the University of Zaragoza – 2019): "Object Recognition in 360 Images".

## 1.4 Outline

This dissertation is divided into the following chapters:

**Chapter2** answers many questions regarding the 3D layout estimation from single view problem and how to leverage 360 images. This chapter guides through two approaches showing an evolution of our research in this field.

**Chapter3** introduces and proposes solutions to the problem of object detection using 360 images. Additionally, we explore how to leverage object-layout interaction to place the detected objects inside the 3D room model.

**Chapter4** explores how to automatically discover 3D keypoints from a collection of objects of the same category, so that they are correspondent. For the first time, we propose to do so using misaligned 3D point clouds, including the notion of symmetry and in an unsupervised manner.

**Chapter5** gives the final discussion of this thesis and ideas for future work on the presented problems.

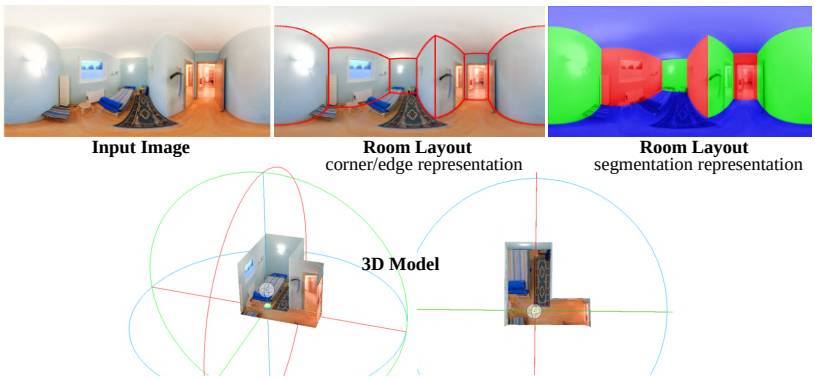
# Chapter 2

# Room Layout Estimation

"*There is geometry in the humming of the strings, there is music in the spacing of the spheres.*"

— Pythagoras

*This chapter guides through two approaches showing an evolution of our research in the task of room layout estimation. The problem of 3D layout recovery in indoor scenes has been a core research topic for over a decade. However, there are still several major challenges that remain unsolved. Among the most relevant ones, a major part of the state-of-the-art methods make implicit or explicit assumptions on the scenes –e.g. box-shaped or Manhattan layouts. Also, current methods are computationally expensive and not suitable for real-time applications like robot navigation and AR/VR. At the end of this chapter, we will end up with a fast geometric deep learning model, flexible and robust to camera pose, that generalizes to cuboid and non-cuboid layouts from single 360 images.*



## 2.1 Introduction

Room layout estimation aims at finding the 3D box that best fits an indoor scene regardless of truncations or occlusions, where the boundaries of the 3D box are the intersections between walls, ceiling and floor.

We are particularly excited to solve the layout estimation problem due to its utility for many real applications and for other computer vision tasks. 3D scene modeling is a key technology in several emerging application markets, such as augmented and virtual reality, intelligent robot navigation or navigational aid for visually impaired people [120]. And also for more traditional ones, like real estate [84]. For the former applications, it is highly useful to have a precise reasoning about the available space, e.g. where the subject can move. For the latter, every time more and more companies opt for democratizing this technology to deliver value to their users. Knowing the room layout, also provides a strong prior for other visual tasks like depth recovery [36, 23], realistic insertions of virtual objects into indoor images [71], indoor object recognition [8, 131], indoor place recognition [65], human pose estimation [46] or scene reconstruction/ rendering [66].

A large variety of methods have been developed to estimate room models using multiple input images [145, 44] or depth sensors [168], which deliver high-quality reconstruction results. For the common case when a single RGB image is available, the problem becomes considerably more challenging. In fact, inferring 3D information from a single monocular image is one of the holy grails of computer vision. The problem itself is ill-posed: depth is irrecoverably lost. However, prior knowledge about the scene geometry and semantics can help resolve some of the ambiguities. The Manhattan world assumption, proposed by Coughlan and Yuille [24], has led the vast majority of work on monocular layout estimation, whereby indoor scenes are three-dimensional grids. Consequently, all walls are at right angles to each other and perpendicular to the floor and ceiling planes. This assumption allows to recover both extrinsic and intrinsic camera parameters, as well as to extract the 3D structure of the room from a single image.

A crucial limitation of previous works [31, 80, 60, 61, 124, 79, 91, 173] lies in the use of conventional images with limited field of view (FoV). On the one hand, this prevents the reconstruction of the real closed geometry of the whole room. This limitation leads to the over-simplification of the room types, e.g. 4-wall layouts, often underfitting the richness of real indoor spaces. On the other hand, the ceiling does not usually appear in conventional images, being nevertheless an important part to detect the main structure of the room, as it usually has much less occlusions. In this regard,

the approximate horizontal FoV of the human vision system is almost 180°, without considering eyes movement. However, the FoV of a standard camera is much smaller, only around 50 °. Even the new smartphones only have an horizontal FoV of around 70°. This crucially limits the use of context cues to understand the surrounding scene. For example, we expect a bedroom to have at least one bed or a kitchen to have a fridge. However, if we use a camera with a small FoV, depending on the direction the camera looks at, there might not be a bed or a fridge, while there might be a table that does not give us much information about the scene. Since the goal of computer vision is to mimic the human visual perception, it is not fair to ask computer vision algorithms to match the performance of human vision with such reduced FoV.

Therefore, a more recent research direction looks to extend the FoV. Lopez-Nicolas *et al.* in [87] perform the layout recovery using a catadioptric system. In [104], layout hypotheses are made combining fisheye images with depth information that provides scale. But the real impact comes with the 360° images, which nowadays can be easily obtained with camera arrays, special lenses or automatic image stitching algorithms [18]. Panoramic images have broken the barriers of performance on this task. These images allows to acquire the whole scene at once and hence, it is possible to exploit their wide FoV to generate closed room solutions based on the best consensus distributed around the scene. [69] shows the advantages of having a complete scene view over partial views of the same scene [80]. However, the methods for conventional cameras are not suitable for wide FoV images, due to the image distortions. This limitation becomes a major bottleneck in some recent works that use 360 images [170, 158, 160], as extra work is needed to leverage conventional algorithms.

To summarize, the problem of estimating the 3D room layout from a single view is not trivial, as it is an ill-posed problem. Additionally, there are several major challenges that still remain unsolved. Most existing methods use conventional images with reduced FoV, that prevents generating closed room solutions. Such reduced FoV also leads to room simplifications, e.g. simple 4-wall cuboids. Moreover, state of the art methods are still far from being solved in real-time, as expensive pre- and/or post-processing steps are needed. Additionally, if we want to leverage wide FoV images, traditional methods are not suitable and new ones have to be developed.

In this chapter we provide a 3D understanding of the room layout beyond the field of view, proposing several approaches that are presented in Sections 2.4 and 2.5. Our goal is threefold: *i)* provide faithful scene geometry predictions, with the motivation to leave behind the 4-wall room simplification, *ii)* create faster methods, avoiding

expensive pre- and post- processing stages and *iii)* propose effective ways to leverage the advantages offered by 360 images.

In Section 2.4, we present our first work in this direction. We propose a model for recovering room models, generalizing to cuboid and non-cuboid layouts. The method combines traditional geometric reasoning and deep learning techniques to get already potential structural lines and corners, from which the layout hypotheses are generated. Working directly with potential structural primitives, leads to a meaningful reduction of the number of hypotheses needed and consequently, to a reduction in the post-procesing computation time. An important contribution for generalizing to non-cuboid layouts is presented in the hypotheses generation process. We observed that both conventional and deep learning algorithms, struggle to detect structural corners that are non-visible in the image due to occlusions. This is in fact a very common scenario where specially floor corners are occluded by the objects in the scene. Additionally, depending on the camera viewpoint and the complexity of the scene, some walls may occlude entire parts of the room. To solve this problem, we allow the model to add extra corners during the hypotheses generation, so that rooms meaningfully satisfy the Manhattan World assumption.

The second method is presented in Section 2.5. In this section, we present a deep learning model to estimate structural lines and corners directly on panoramic images. The advantage is twofold. First, it allows avoiding expensive pre-processing algorithms to leverage methods designed for conventional images. Second, working directly on the panorama allows to take advantage of all the context. In this section, we also make the observation that the use of standard convolutions in equirectangular images can lead to a loss of performance. We propose a novel convolution that adapts the kernel size and shape to the equirectangular image distortions, bringing numerous advantages. We demonstrate how the obtained predictions are not only more accurate, but also more robust to camera pose variations. The performance improvement allows us to predict layouts in an end-to-end manner, and relax the scene assumptions. Predicting the corners in an end-to-end fashion, makes our method up to 100 times faster than previous approaches.

## 2.2 Related Work

The seminal monocular approach to automatically recover 3D reconstructions was [31], which shows how prior knowledge about indoor scenes, i.e. floor-wall boundaries, can be learned using a dynamic Bayesian network. In parallel, Lee et al. [80] generate layout

hypotheses from detected line segments, and select the best-fitting one evaluating with an Orientation map. While effective, Orientation maps get limited with the presence of clutter, since no reasoning about the lines is made. Motivated by the problem of the presence of clutter, [60] models the layout of the room with an aligned 3D box while localizing visible objects. This inspiring idea was followed by [61, 124]. However, the 3D box simplification does not match reality in many cases, being hence constrained to this particular room geometry and unable to generalize to other room configurations. Typically, these methods follow a proposing-ranking scheme and rely on Geometric Context [63] to evaluate. Geometric Context improves clutter detection compared with the Orientation maps, but provides worse results at the higher parts of the scenes. More recently, [125] introduces the concept of integral geometry and pairwise potentials decomposition which results in an efficient structured prediction framework.

Since 2012, CNNs achieved breakthrough performance in a wide range of applications such as image classification [76], segmentation [7], detection [109] optical flow [137] and keypoint detection [121]. This unprecedented level of data abstraction inspired by neuronal processes, became popular in all areas of Computer Vision, including that of estimating the layout of rooms. Mallya *et al.* [91] train a Fully convolutional Network (FCN) to jointly predict informative edges and geometric context from conventional images. The pixel-wise edge labeling distinguishes between background, wall-floor edge, wall-wall edge and wall-ceiling edge. More recently, other works focus on pixel-wise edge labeling. [113, 169] address the problem in a coarse-to-fine manner. Instead, the proposal of [173] is inspired by mechanics concepts. Alternatively, Dasgupta *et al.* [30] propose a FCN to predict semantic surface labels of the rooms, providing separate belief maps of the walls, ceiling and floor of the scene. All these methods require extra computation added to the forward propagation of the network to retrieve the actual layout. In [79], an end-to-end network predicts the layout corners in a perspective image, as well as a label that indicates which corners are visible. Afterwards, the room type is inferred within a limited set of manually chosen configurations. Other deep learning works extract an estimation of the depth or/and surface normals from simple RGB images, which also produces an interesting outcome for the problem of room layout estimation [36, 77]. The main drawback of these CNNs is that they are designed to work on conventional images with limited FoV, with the aforementioned consequent limitations.

While layout recovery from conventional images has progressed rapidly with both conventional methods and deep learning, the works that address these challenges using omnidirectional images are still very few in comparison. Omnidirectional cameras have

the potential to improve the performance of the task: their 360° field of view captures the entire viewing sphere surrounding its optical center, allowing to acquire the whole room at once and hence to predict layouts with more visual information. PanoContext [170] was the first work that extended the frameworks designed for perspective images to panoramas. It recovers the room layout, which is also assumed as a simple 3D box, and bounding boxes for the most salient objects inside the room. Pano2CAD [158] extends the method to non-cuboid rooms, but it is limited by its dependence on the output of object detectors. [160] treats the problem as a graph with lines and superpixels as nodes, solving it with complex geometric constraints instead. The most recent works along this line are contemporary to the last approach of this chapter. LayoutNet [175] trains a FCN from panoramas and vanishing lines, generating the layout models from edge and corner maps, and DuLa-Net [161] predicts Manhattan-world layouts leveraging a perspective ceiling-view of the room. All of these approaches require pre- or post-processing steps like line and vanishing point extraction or room model fitting, that increase their cost.

In the last two years, the main improvements in layout recovery from panoramas have come from the application of deep learning. The high-level features learned by deep networks have proven to be as useful for this problem as for many others. Nevertheless, these techniques entail other problems such as the lack of data or overfitting. In this regard, state-of-the-art methods require additional pre- and/or post-processing. As a consequence they are slow, and this is a major drawback considering the aforementioned applications for real-time layout recovery.

In addition to all the challenges mentioned above, we also notice that there is an incongruence between panoramic images and conventional CNNs. The space-varying distortions caused by the equirectangular representation makes the translational weight sharing ineffective. Very recently, Cohen et.al. [22] did a relevant theoretical contribution by studying convolutions on the sphere using spectral analysis. However, it is not clearly demonstrated whether Spherical CNNs can reach the same accuracy and efficiency on equirectangular images. A related work [140] proposes distortion-aware convolutional filters to solve dense prediction tasks such as depth prediction and semantic segmentation by leveraging commonly used datasets with annotations for perspective images during training.

In the following sections we present the two different approaches proposed for the room layout estimation problem.

## 2.3 Background and Theory

**Panorama Geometry.** A big part of the solutions proposed in this thesis to the problem of 3D indoor scene understanding use panoramic images. Therefore, we start explaining the basics of the spherical camera geometry, which will help us progress smoothly to the actual solutions. For convenience, we will use the terms equirectangular image, 360 image, spherical image and panoramic image interchangeably.

We can define a central camera as a collection of rays passing through a single point in a space, which is the camera center. For the particular case of a spherical camera model, it consists of a camera centered inside a surface of a unit sphere.

*How do we fit the surface of the unit sphere onto a single image?* According to the Gauss's Theorema Egregium, the Gaussian curvature of an embedded smooth surface in $R^3$ is invariant under the local isometries. Since the sphere of radius $r$ has constant positive curvature $1/r^2$ and a flat plane has zero constant curvature, these two surfaces are not isometric. This means that a piece of paper cannot be bent onto a sphere without crumpling and conversely, the sphere surface cannot be unfolded onto a plane without distorting. Thus, all planar projections of a sphere have distortions. Among all the possible planar projections of the sphere, the Equirectangular projection is usually preferred in computer vision as it preserves distances between points i.e. it is equidistant, meaning that the image grid can be indexed directly with spherical coordinates. This is because Equirectangular projection maps meridians and parallels of the sphere to vertical and horizontal straight lines of constant spacing respectively. The projection however, is neither equiareal nor conformal. This inevitably generates some distortions that are more pronounced near the poles, where the areas get stretched horizontally to the entire width of the image, i.e. the entire top edge corresponds to a single point, as does the lower edge. Further, the left and the right edges of the image, are the same spot in reality, losing the continuity of the scene in the image.

Let's denote the resolution of the equirectangular image to be $W \times H$ pixels. Because the spherical images covers 360° field of view horizontally and 180° field of view vertically, we know that $W = 2H$, and the focal length is $\frac{W}{2\pi}$ pixels. To take these images, the camera is typically positioned so that the top of the projection sphere is pointing to the sky. Therefore, we can safely assume that the horizontal vanishing line of the ground plane is at 0 height of the image coordinate $[-\frac{W}{2}, \frac{W}{2}] \times [-\frac{H}{2}, \frac{H}{2}]$. Otherwise, we can perform an upright-alignment.

We formulate the relation between a point in a space, a point on the unit sphere surface and a point in the equivalent image plane.

The first step consists of projecting the scene point $\mathsf{X}$ onto the unit sphere; therefore:

$$\mathsf{x} = \frac{1}{\|\mathsf{X}\|}\mathsf{X} \tag{2.1}$$

In the spherical coordinate system, a point on the unit sphere is represented by the ordered triple $(\rho, \theta, \phi)$, where $\rho = 1$ is the radial distance, $0 \leq \theta \leq \pi$ is the polar angle and $0 \leq \phi \leq 2\pi$ is the azimuth angle. Since $\mathsf{x}$ is a point on a unit sphere, it is possible to express it in spherical coordinates, such that,

$$\mathsf{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\cos(\theta)\sin(\phi) \\ \sin(\theta) \\ \cos(\theta)\cos(\phi) \end{bmatrix}, \tag{2.2}$$

Converting this into image coordinates $(u, v)$:

$$u = (\frac{\phi}{2\pi} + \frac{1}{2})W \quad ; \quad v = (-\frac{\theta}{\pi} + \frac{1}{2})H. \tag{2.3}$$

Therefore, angles $\phi$ and $\theta$ are defined as:

$$\phi = (u - \frac{W}{2})\frac{2\pi}{W} \quad ; \quad \theta = -(v - \frac{H}{2})\frac{\pi}{H}. \tag{2.4}$$

**Lines and Vanishing Points in Panoramas.** For a straight line in the world, there exists a plane that includes the line itself and the spherical camera center. The intersection of the plane and the spherical surface yields an arc segment on a great circle onto the unit sphere and thus it appears as a curved line segment in the equirectangular image. We express a particular $i$-th line in the panorama by the normal vector $\mathsf{n}_i$ of the plane where its great circle lies in.

In order to infer 3D information of the scene from a single 2D image, further assumptions are needed. The Manhattan world assumption [24] has led the vast majority of work on monocular layout estimation, whereby there exist three dominant orthogonal directions in the scene. These dominant directions are given by the vanishing points. This is a safe assumption, particularly for indoor scenes, as has been widely demonstrated [31, 80, 60, 61, 124, 79]. Vanishing points in the image plane can be defined as the points where parallel lines in the 3D space meet in the image. Therefore, whereas in conventional images they do so in one single vanishing point, in spherical images, parallel lines intersect in two antipodal vanishing points.

The three orthogonal vanishing points also construct a complete rotation, that can be used to resolve the misorientation of the camera. For example, once the vertical vanishing point of an image is detected, the image can be rotated by placing the

vanishing point on the y-axis. Therefore, the vanishing points allow to rotate the panorama so that it is pointing perpendicularly to one of the room walls and upright aligned.

## 2.4 Layouts with Geometry and Deep Learning

In this section, we describe our first proposal for the room layout estimation problem. From a single 360 image, our model combines strategically conventional and deep learning techniques to predict potential room corners on the image. This step leads to a meaningful reduction of the number of room hypotheses needed and consequently, to an improvement in terms of efficiency. Moreover, we observed that both traditional and deep learning algorithms struggle to detect room corners that are non-visible in the image due to occlusions. The proposed approach generalizes to cuboid and non-cuboid room configurations by inferring additional corners during the layout hypotheses generation so that the final layout satisfies the Manhattan World assumption. This idea demonstrates to be very powerful since allows us to localize a priori invisible corners and correct errors from initial predictions.

### 2.4.1 Structural Lines

This section begins describing how we extract lines and vanishing directions directly on panoramic images, using only geometric reasoning. We then show how to abstract the data to obtain potential structural lines by a combination with a deep learning approach.

**Lines and Vanishing Points in Panoramas.** We first describe how we detect lines and vanishing directions in the panoramic images, taking into account the spherical image geometry described in Section 2.3.

PanoContext [170] was the pioneer work proposing to solve the room layout problem from single panoramic images. In order to get the lines on the image, they split the panorama into several overlapping perspective views, run the Line Segment Detection (LSD) algorithm [151], and warp then all detected line segments back to the panorama. Then they use Hough Transform to find all vanishing directions. A more recent approach followed the same procedure [175]. This method not only suffers from very high computational cost but also from loss of accuracy in the detected line segments due to the partitions.

However, there are some works whose main purpose is precisely obtaining lines and vanishing points directly on panoramas. This is the case in [11]. They solve the problem by a branch-and-bound framework associated with a rotation space search. This method guarantees a global optimum but such strategies may be computationally expensive and less robust.

We detect lines and vanishing directions by a RANSAC-based algorithm whose input is directly the rgb panorama, resulting in entire and unique line segments thus avoiding duplicate lines coming from different splits and improving the overall efficiency of the method. The proposed method is as follows. First, we run a Canny edge detector on the panorama and cluster contiguous edge points in edge groups considering them as candidate lines of the image. We represent edge points as their spatial ray projection into the 3D space $\mathsf{r}_{ij} \in \mathbb{R}^3$, $j \in \{1, 2, \ldots, N\}$, following Equations (2.4) and (2.2). We represent a particular $i$-th edge group as a tuple of edge points $\mathsf{R}_i = (\mathsf{r}_{i1}, \mathsf{r}_{i2}, \ldots, \mathsf{r}_{iN})$. Iteratively, two edge points of each group are randomly selected to compute a possible normal direction for the edge group $\mathsf{n}_i = (\mathsf{r}_{i1} \times \mathsf{r}_{i2})$. The number of inliers is evaluated, i.e. how many rays fulfill the condition of perpendicularity with the computed normal under an angular threshold $\tau_{th} = \pm 0.5°$, $|arccos(\mathsf{n}_i \cdot \mathsf{r}_{ij}) - \frac{\pi}{2}| \leq \tau_{th}$. After a certain number of iterations the process outputs, for each edge group, the model leading to the highest number of inliers giving the $\mathsf{n}_i$ that fits the line best. Edge groups that do not share a common normal direction are discarded whereas the others are kept as actual lines of the panorama. The algorithm returns all the lines in the panorama, i.e. those belonging to the structure of the room but also those belonging to the objects, clutter, etc. These lines accurately catch the edges shown in the image and satisfy some desired properties: they are accurate straight line segments and follow three main orthogonal directions in the scene. For each line, we get the pixels it occupies in the planar projection of the image, the corresponding coordinates on the surface of the sphere, and the normal direction $\mathsf{n}_i$ of the plane that includes the line itself and the spherical camera center.

We obtain the three orthogonal vanishing directions with another RANSAC-based algorithm, considering $\mathsf{vp}_k = \mathsf{n}_a \times \mathsf{n}_b$ where $n_a$ and $n_b$ are the normal vectors of two world parallel lines and $k = x, y, z$. Eventually we select the three vanishing points $(\mathsf{vp}_x, \mathsf{vp}_y, \mathsf{vp}_z)$ that have the most number of inlier lines, exploiting that normal vectors of lines must be orthogonal to the three main directions $|arccos(\mathsf{n}_i \cdot \mathsf{vp}_k) - \frac{\pi}{2}| \leq \tau_{th}$.

Lines are classified according to the Manhattan directions (vanishing points), discarding those lines whose normal vector is not perpendicular to any of the main

directions. The lines with the same Manhattan direction are shown in identical color in Figure 2.3 (top-left).

**Structural lines introducing deep learning.** The main piece of information we use to create layout hypotheses are lines. However, in cluttered scenes it is difficult to know whether they come from actual wall intersections or from other elements of the scene. Proceeding with all the lines leads to an intractable number of hypotheses. In order to tackle this problem, we propose to evaluate the extracted lines on the panoramic image introducing a deep learning approach.

CNNs have been successfully applied to extract complex features such as corners [79] or structural edges [91]. However, they have not been trained to deal with omnidirectional images and then, they are very inaccurate when used directly on panoramas. Besides that, there does not exist any dataset collecting panoramic images with enough amount and variety of labeled data required to train a deep neural network. Thus, we do not directly train an end-to-end CNN and decide instead, to adapt our image geometry to an existing CNN. Here, we adapt the Fully Convolutional Network (FCN) proposed by Mallya and Lazebnik [91]. This network was trained to estimate probability maps representing the room edges, even in the presence of clutter and occlusions. This idea seems very powerful to us, since it is a very simple representation that intuitively encodes the structure of a room. Our proposal is to combine such rough yet meaningful information with more accurate geometric cues such as lines.

There are two learning strategies that were used thus far in order to take advantage from conventional methods with panoramic images, see Figure 2.1. Strategy I, fast but inaccurate, applies the CNN directly on the equirectangular image. Strategy II, more accurate but slow, samples multiple tangent planar projections to obtain overlapping perspective images, to which the CNN is applied independently to obtain local results that are then projected back to the original panorama. The latter strategy was used by PanoContext [170] to detect the lines on the panorama among other steps in their pipeline.

In Fig 2.2 we show that the accuracy of the predicted edge maps substantially improves when we use Strategy II, specially in those cases where the result of applying directly the FCN on the panorama is completely uninformative (first and third columns). This is not only due to the image projection, as suggested in [135], but also from the difference in the field of view, i.e. if a pretrained CNN learns to predict edge maps seeing two walls, a piece of floor and luckily a piece of ceiling, it is very hard that it generalizes to the whole room structure that is seen in panoramic images.

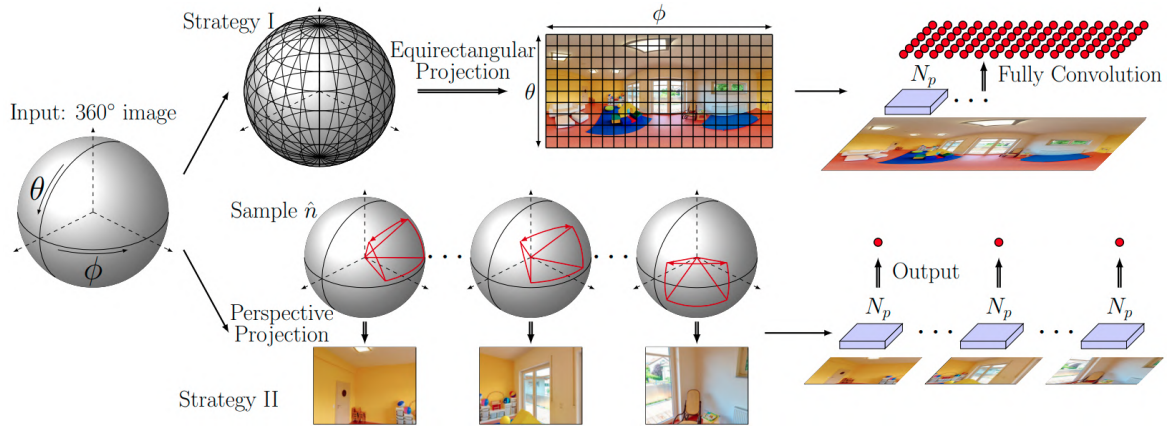



Fig. 2.1 Learning strategies with equirectangular images. Figure from [135]

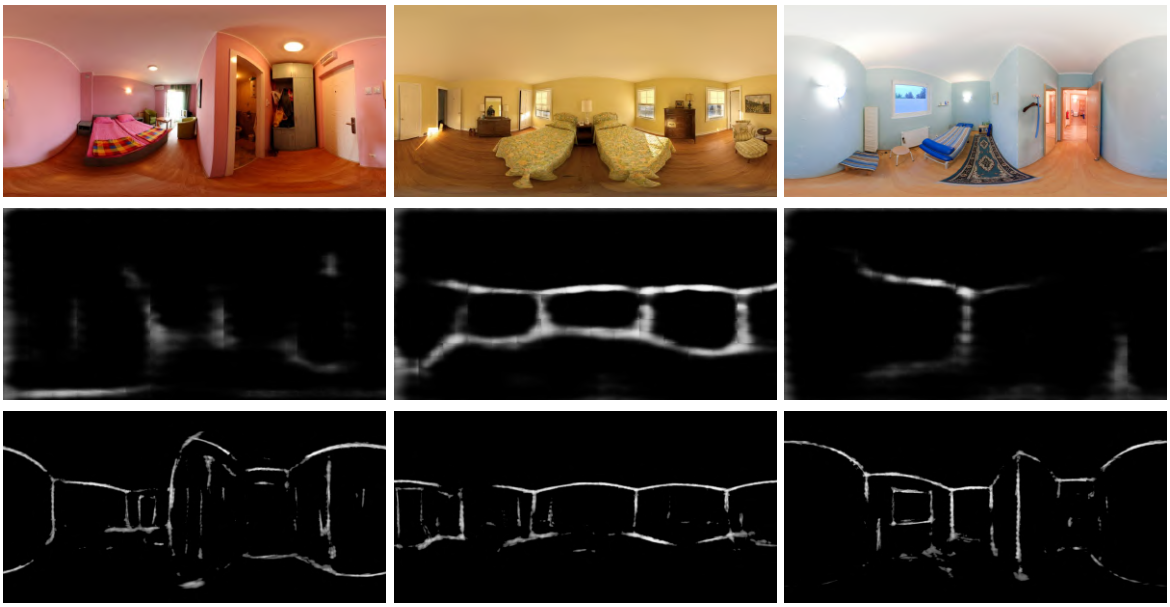


Fig. 2.2 Edge maps comparison. **Top**: Input RGB images. **Middle**: Edge maps obtained using [91] with strategy I. **Bottom**: Edge maps obtained using [91] with strategy II and the proposed discretization of the sphere.

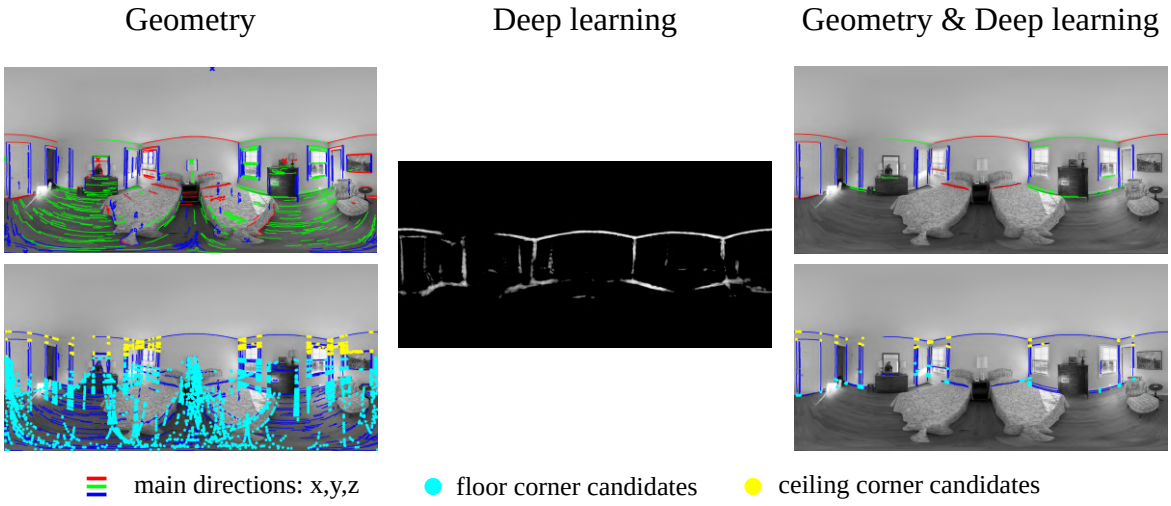



Fig. 2.3 **Left**: Full set of oriented lines and corners extracted with only geometric reasoning (canny edge detector and RANSAC). **Middle**: Edge Map obtained using [91]. **Right**: Resulting structural lines and corners after combining geometry an deep learning. The proposed method keeps only potential structural lines that generate already good corner candidates.

Therefore, to apply the FCN, we split the panoramas into a set of overlapping perspective images with a FoV similar to conventional images ($\sim$70°) and planar projection. We run the algorithms in each of them separately to obtain local results and finally stitch them all back to the panorama as in [157, 170, 158]. For the discretization of the sphere, previous methods use the spherical coordinates from uniform distributions $\theta \in (-\pi/2, \pi/2)$ and $\phi \in (-\pi, \pi)$, which is not adequate since the density increases as we get closer to the poles. Instead, we use an algorithm based on the golden section spiral [52, 55]. For any given number of points, it results in an evenly distribution with bins covering areas of similar size equally distant from their closest neighbor. We empirically choose 60 points, *i.e.* 60 perspective images.

To improve the predicted edge maps, we avoid noise by removing low probability pixel values below a certain threshold, 0.2 out of 1, empirically chosen. When the virtual perspective images are stitched back to the panorama, there are some overlapping regions that we solve by choosing the maximum value of probability to not lose information.

Once we independently get the edge map of the panorama given by the FCN [91] and the full set of lines estimated by our geometric approach i.e. canny edge detector and RANSAC, we proceed to filter the non-structural lines. We associate a score to each extracted line. The score is calculated as the sum of the probability values of

the pixels the line occupies on the edge map. We remove those lines whose score, normalized with the line length, is below a certain threshold, while the others are classified as structural lines. An example of this process can be observed in Figure 2.3, which shows the advantage of merging traditional geometric reasoning with deep learning techniques. Those lines that belong to clutter such as those from the parquet, the tables and even many windows, pictures and doors are removed, while most relevant lines to recover the structure of the room remain for further stages. With this operation the number of lines may be reduced to one-third or even a quarter of the original detections depending on the scene.

### 2.4.2 Room Layout Estimation

Our goal is to extract the main structure of an indoor environment *i.e.* the distribution of floor, ceiling and walls, abstracting all objects within rooms. For this purpose we develop a method to generate layout hypotheses from corners, using the predicted subset of lines, that are already potential structural lines. The way we represent mathematically the room layout and consequently the output of the proposed algorithm, is the set of 3D corner coordinates of the room, ordered clockwise and up to a scale and the corresponding 2D corner coordinates in the image.

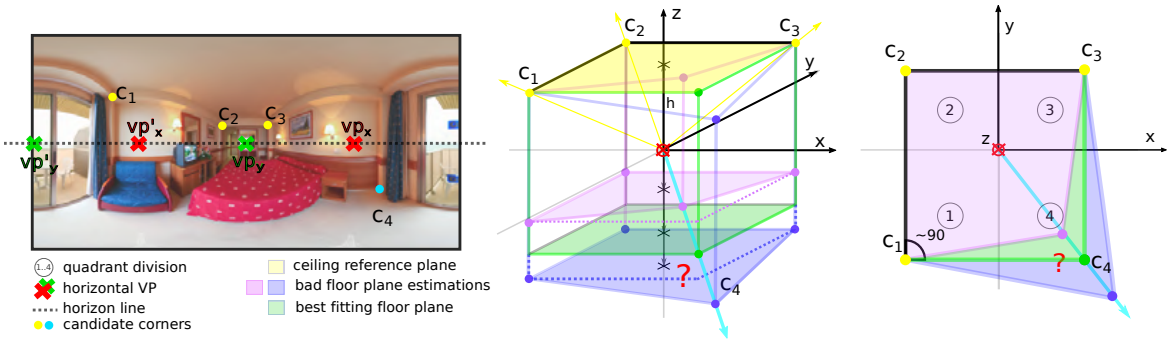



Fig. 2.4 **Room solution**. Our algorithm returns a solution such that the walls connected by the corners are as perpendicular as possible (green) following the Manhattan world assumption. See how non-Manhattan solutions (pink and purple) result in floor planes that are not parallel to the ceiling plane (yellow). The solution also provide us an estimated room height for the layout hypothesis.

#### Candidate corners extraction

Our layout generation process is based on corners, i.e. structural intersections between two walls and ceiling or floor. In a Manhattan World, two line segments are enough

to define a corner. We intersect the predicted lines among themselves in pairs, as long as they do not cross each other and they have different directions $(x, y, z)$. The direction vector of the corner is computed using the lines intersecting at that corner, $c_{ab} = (n_a \times n_b)$. Thanks to the previous line filtering step, the extracted corners are already good candidates to generate room layout hypotheses. Figure 2.3 shows the large difference between obtaining corners with the original set of lines (left) and with the subset of structural lines (right). By removing non-structural lines, the number of corners gets vastly reduced, yet the important ones remain detected. This reduction makes further stages of the method faster and more efficient while improving the reliability of the results, since most corner candidates coming from clutter and irrelevant structures are not considered.

Panoramic images have the advantage of providing a full view of the room, allowing us to look around, up and down in the scene. This is unlike the conventional images, where the ceiling and some walls use to be out of the FoV. Taking this into account, we carry out a classification of the detected corners following two criteria:

a) **Vertical direction.** Corners detected below the horizon line ($l_H$), which in central panoramas is at the middle row, are considered as floor corner candidates and those detected above, are considered as ceiling corner candidates.

b) $XY$**-plane.** We divide the scene into four quadrants around the camera center using the horizontal VPs as quadrant dividers, $\mathcal{Q} = \{q_1, q_2, q_3, q_4\}$. Hence, e.g. to determine when a corner belongs to the fourth quadrant: $c \in q_4 \iff c_x \in \mathbb{R}^+ \wedge c_y \in \mathbb{R}^-$.

See Figures 2.4 and 2.5 for more details about the representation.

### Layout hypotheses generation

Many works simplify the room layout to have only four walls. Usually, this simplification comes from the lack of contextual information when conventional images are used [60, 61, 124]. However, more recent works using 360 images also adopt this simplification [170]. Here, we handle more complex designs which will be faithful to the actual shapes of the rooms, introducing the possibility of estimating in-between hidden corners when required, i.e. when they are occluded by clutter or due to scene non convexity. We generate layout hypotheses by means of an iterative method that attempts to join consecutive corners with alternatively oriented walls. We assume the following:

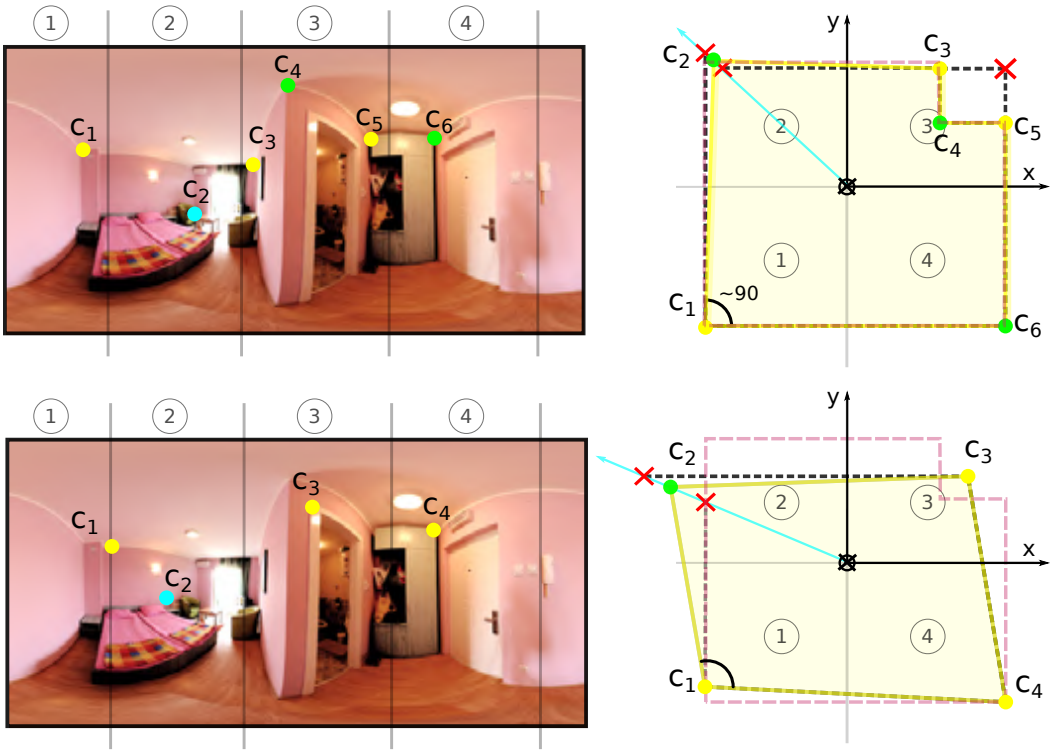



Fig. 2.5 **Layout hypothesis generation**: We show two possible layout hypotheses for the same scene. Yellow (ceiling) and cyan (floor) corners are randomly sampled by the algorithm to start the corresponding hypothesis. Green corners are estimated by the algorithm to satisfy the Manhattan World assumption and to potentially localize non-visible corners in the image. Top: valid hypothesis. Bottom: non-valid hypothesis.

a) **Manhattan world.** There are three main orthogonal directions to each other that define the indoor scene.

b) **Ceiling-floor parallelism.** Floor corners are on the same floor plane and ceiling corners are directly above the floor ones. The normal direction of both planes is the vertical vanishing direction $\mathsf{vp}_z$.

c) **Camera height.** Since no depth information is available, we need to assume the distance from the camera to the floor or ceiling planes. This is trivial as results are up to scale, but needed to predict the total height of the room. Generally, the distance to the floor is assumed. We observed that the predicted ceiling corners are more reliable, being in a less cluttered area, and we assume the distance to the ceiling plane instead.

The proposed algorithm randomly samples a group of corners among the predicted ones, $\mathcal{G}_c$, which are ordered clockwise in the $XY$-plane. The number of sampled

corners $N_{\mathcal{G}_c}$ may vary at each iteration and can be directly related to the maximum number of walls that our algorithm can handle, $N_W^{max} = 2(N_{\mathcal{G}_c} - 1)$. For example, we can draw room layouts with six walls from a minimum number of four corners, allowing the algorithm to introduce two new corners that may be occluded in the image. Additionally, we observe that Manhattan World rooms always have an even number of walls and an odd number of corners at each quadrant. As an example, a simple layout has only one corner per quadrant, while more complex layouts may have three or even five corners at some of their quadrants. Therefore, the proposed quadrant division provides a convenient way to sample corners. The corner sampling must include corners from at least three quadrants, so that the corner in the remaining quadrant can be estimated assuming closed Manhattan layouts, and there must be at least one corner of each hemisphere, so that the total room height can be predicted.

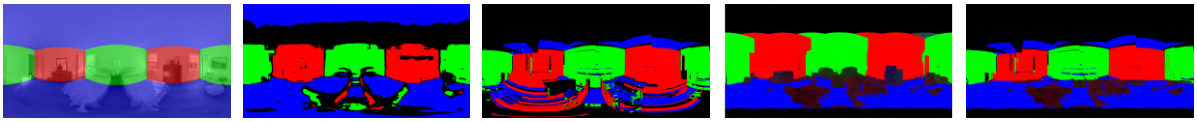
We proceed with the geometric reasoning in 2D with a top view of the 3D scene, see right side of Figure 2.4 or 2.5. Note that we do not have the real 3D coordinates of the corners, but only the 3D ray that goes from the center of the spherical image through the corner position on the surface of the sphere.

We use Figure 2.4 to describe how our hypotheses generation algorithm works. First, the 3D rays of the sampled ceiling corners are intersected into a reference ceiling plane at an assumed distance, obtaining the potential 3D ceiling corners $c_1, c_2$ and $c_3$ (yellow). We keep the 3D ray of the sampled floor corner (cyan), as the distance to the floor is yet unknown. Then, we use the Manhattan world requirement to estimate the correct floor corner $c_4$ position along its 3D ray, so that the walls connected by the corners are as perpendicular as possible ($90° \pm 5°$). This process returns a Manhattan World solution for the room that also allows us to compute the complementary distance to the floor plane that verifies the ray equation.

In Figure 2.5 we show a complex layout and two possible layout hypotheses depending on the initial sampled corners. We first show a valid hypothesis (top), with sampled corners $\mathcal{G}_c = \{c_1, c_2, c_3, c_5\}$. This means that the algorithm will be able to solve a layout hypothesis with $N_W^{max} = 6$. After projecting the ceiling corners into a reference plane, a joining corner process starts from $c_1$. As before, we find the optimal floor corner position along its ray using its nearest corners and draw an intermediate solution, $c_2$. In the third quadrant, taking into account the direction $(x - y)$ from previous unions, our algorithm selects the best solution for $c_4$ by choosing the one which produces alternatively oriented consecutive walls. In the empty quadrant, Manhattan walls from nearest corners give $c_6$. We also show a non-valid hypothesis (bottom),

with sampled corners $\mathcal{G}_c = \{c_1, c_2, c_3, c_4\}$. Following the same process, the corners are orderly joined resulting in this case a non-Manhattan layout.

A further explanation of the algorithm can be found in Appendix A.



(a) Layout hypothesis ($\mathcal{I}^{H_i}$) (b) Normals Map ($\mathcal{I}^{NM}$) (c) Orientation Map ($\mathcal{I}^{OM}$) (d) Geometric Context ($\mathcal{I}^{GC}$) (e) Merge Map ($\mathcal{I}^{MM}$)

Fig. 2.6 (a) Example of labeled image generated from layout hypotheses. (b)-(e) Visual representation of how each of the *reference maps* $\mathcal{I}^{\mathcal{R}}$, looks like.

### Layout hypotheses evaluation

We evaluate a number of layout hypotheses $N_h$ to get the best and final room layout solution. For each hypothesis $H_i$, we generate a segmented image $\mathcal{I}^{H_i}$, encoding the orientation of the predicted surfaces, i.e. walls in $x$, walls in $y$ and floor/ceiling in $z$. In Figure 2.6 (a) there is an example of a segmented image, where each orientation is encoded with a different color.

We evaluate the segmented image $\mathcal{I}^{H_i}$ by comparing it to a reference map $\mathcal{I}^{R}$ that roughly encodes the orientation of the pixels, and can be obtained from several methods. We compute the ratio of pixels that are equally oriented in $\mathcal{I}^{H_i}$ and $\mathcal{I}^{R}$ over the total size of the image $(H, W)$, that we name Equally Oriented Pixel ratio $(EOP)$:

$$EOP\left(\mathcal{I}^{H_i}, \mathcal{I}^{R}\right) = \frac{1}{H \cdot W} \sum_{x,y,z}^{C} \sum_{i,j}^{H,W} \mathcal{I}^{H_i} \cap \mathcal{I}^{\mathcal{R}},$$

being $C$ the number of channels corresponding to the labels i.e. orientations $x,y,z$.

In this work, we test four methods to compute the reference map $\mathcal{I}^{\mathcal{R}}$. The four methods are designed for conventional images so we repeat the same process as in Section 2.4.1 to compute them. The Orientation Map [80], $\mathcal{I}^{OM}$ (Figure 2.6 (c)), and Geometric Context [60], $\mathcal{I}^{GC}$ (Figure 2.6 (d)) are two methods widely used in the literature to evaluate room models [63, 80]. Recently, [170, 68] combine the strengths of both of them in one single map, that we name Merge Map, $\mathcal{I}^{MM}$ (Figure 2.6 (e)).

We additionally propose to use a Normal Map ($\mathcal{I}^{NM}$). We choose the work from Eigen and Fergus [36], which proposes a multiscale CNN that returns depth prediction,

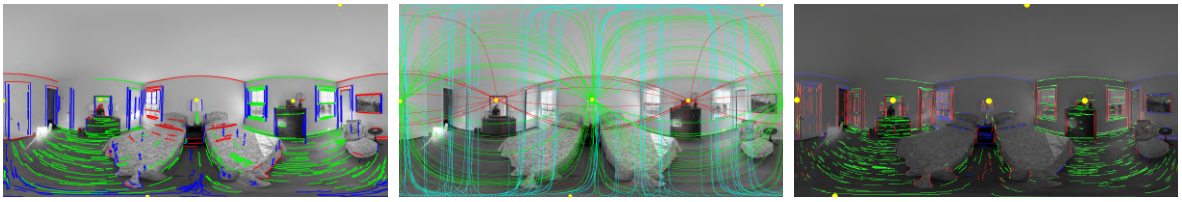
surface normal estimation and semantic labeling of indoor images. Here, we take advantage of the surface normal estimation to create our reference map. We first split the panorama into local perspective images through a discretization of the sphere (See Section 2.4.1), keeping the spherical coordinates (polar and azimuthal angles) of their position on the surface of the sphere. We evaluate [36] on the local images, receiving as output the corresponding images encoding the x, y and z components of the normal at each pixel. In order to stitch the local results back to the panorama, we need to rotate the normals to set them in a common reference frame. More specifically, we rotate the predicted normals of each local image by the azimuthal angle of its position on the surface of the sphere. Additionally, we rotate the normals by the vanishing directions of the scene to resolve any misorientation of the camera. Overlapping areas are tackled by doing the per-pixel average to achieve a continuity of the overall image. The resulting Normal Map is shown in Figure 2.6 (b). We also determine whether or not the normals from each pixel belong to a main direction (VPs) and label them accordingly, i.e. $x$ direction in red, $y$ direction in green and $z$ direction in blue. We set to black the pixels that do not belong to any main direction. It can be noticed in Figure 2.6 that the ceiling is the worst estimated part by all the methods. This happens because the ceiling does not usually appear in conventional images.

### 2.4.3 Experimental Results

We evaluate our proposal using 360 images of indoor scenarios from two public datasets. In particular, most of our quantitative results have been obtained from a subset of 85 panoramas of bedrooms and living rooms of the SUN360 dataset [157]. Additionally, we also show results on the Stanford (2D-3D-S) dataset [6].

For each panorama we create the ground truth as a segmented image $\mathcal{I}^{GT}$, like those in Figure 2.6, where each pixel encodes the direction of the surface it belongs to. A previous ground truth was provided by [170], but images were labeled assuming rooms have only 4 walls.

The accuracy of our results is evaluated by computing $EOP\left(\mathcal{I}^{H_b}, \mathcal{I}^{GT}\right)$, measuring the ratio of equally-oriented pixels between the best layout hypothesis and the ground truth. Each EOP value shown is a median of 10 times performing the experiment. The number of hypotheses drawn ($N_h$) is specified in each experiment. For the experiments we allow the algorithm to initially select from three to five corners, i.e. to solve layouts with four to eight walls.



(a) Ours (b) Bazin et.al. [11] (c) PanoContext [170]

Fig. 2.7 Lines and vanishing points detection using three different methods.

**Lines and vanishing points.** The proposed algorithm in Section 2.4.1 works directly on the equirectangular image, allowing us to obtain unique line segments, avoiding thus duplicate lines coming from different splits.

In [170] the panorama is split in order to run a specific algorithm that only works with perspective images, warping then all detected line segments back to the panorama, whereas in [11], the problem is solved by a branch-and-bound framework associated with a rotation space search, working directly on panoramas. For [170] we run directly the code provided by the authors. [11] does not provide any code and there are no public experimental results with omnidirectional images. However, same authors provide code of previous work [12, 10] that is used for this evaluation.

Our RANSAC-based algorithm achieves really similar results to [170, 11] being also much faster, ∼3.8s per image in our proposal, compared to ∼67s per image with [11] and ∼42s per image using [170]. Visual results from each work are shown in Figure 2.7.

**Advantages of combining geometric reasoning with deep learning.** A comparative study showing the effects of selecting structural lines (Section 2.4.1) can be found in Figure 2.8. For this experiment we choose $N_h = 100$ and the $\mathcal{I}^{NM}$ as reference map. Each point represents an image. We show in red the EOP when the complete set of lines predicted by our geometric approach (G) is used to obtain the candidate corners. We show in green the results when only the subset of structural lines obtained combining geometry and deep learning (G+DL) is used. Mean and especially median values highlight the improvement when combining traditional approaches with deep learning: 0.889 vs. 0.925. The detection of structural lines allows to remove clutter effectively, which translates into better accuracy.

**Reference maps.** We compare the performance of our model using the four alternative reference maps in the evaluation step. For this experiment we also consider

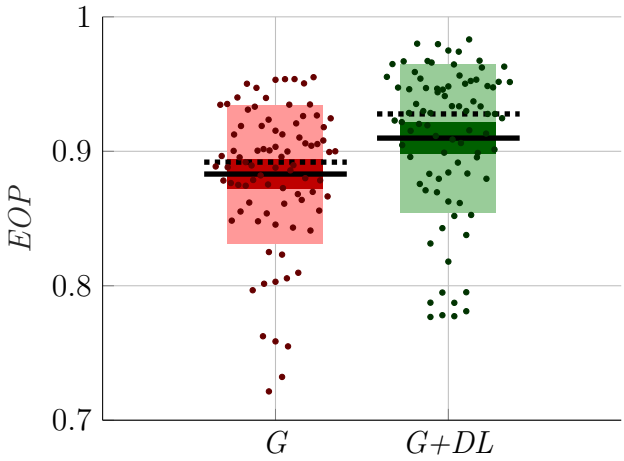



Fig. 2.8 **Advantages of combining.** Here we highlight the advantages of using structural lines from Geometry and Deep Learning combination [91] over lines obtained only with Geometry. The mean is represented in solid black and the median in dotted black. Also the standard deviation is shown in light color and jittered raw data are plotted for each group.

| | *EOP* | *Computing Time* |
|---|---|---|
| Normal Map ($\mathcal{I}^{NM}$) | **0.925±0.061** | 243.36±1.42 |
| Orientation Map ($\mathcal{I}^{OM}$) | 0.906±0.133 | **23.54±4.16** |
| Geometric Context ($\mathcal{I}^{GC}$) | 0.883±0.114 | 174.07±13.28 |
| Merge Map ($\mathcal{I}^{MM}$) | 0.923±0.147 | 197.61±17.44 |

Table 2.1 Ratio of equally-oriented pixels when comparing the best final hypotheses, $\mathcal{I}^{H_b}$, with the ground truth $\mathcal{I}^{GT}$, evaluating in each case with a *reference map*. Also the computing time in seconds of generating each map is shown.

$N_h = 100$. Table 2.1 shows the median EOP value and the computing time of creating each map. In terms of accuracy, $\mathcal{I}^{NM}$ and $\mathcal{I}^{MM}$ perform similarly in median, although the smaller standard deviation of the $\mathcal{I}^{NM}$ indicates more consistent results. Both are considerably better than $\mathcal{I}^{OM}$ and $\mathcal{I}^{GC}$. However, the $\mathcal{I}^{OM}$ is about ten times faster to compute than the $\mathcal{I}^{NM}$ and, therefore, its usage would be recommendable if the priority lies in getting fast results in spite of losing some accuracy. The smaller standard deviation on the computing time of the $\mathcal{I}^{NM}$ shows that it does not vary through images, unlike the others whose time depends on scene-specific features such as the number of lines.

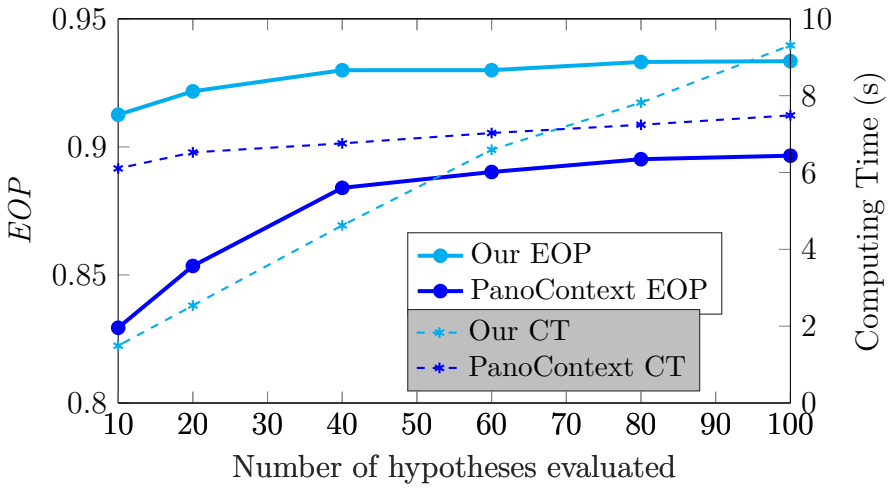



Fig. 2.9 **Comparison with PanoContext** [170] (with only four-wall rooms). We show the ratio of equally-oriented pixels and computing time against the number of hypotheses. Our method outperforms PanoContext and is able to provide much better results and much faster with fewer hypotheses.

**Comparison with the state of the art.** We perform a comparison with PanoContext [170] since it is, to our knowledge, the only directly related method with available code. In Figure 2.9 we show the EOP ratio and the computing time necessary to generate the hypotheses for each method, varying the number of hypotheses $N_h$. Our method clearly outperforms [170], being the difference larger when only a few hypotheses are considered. Although the difference decreases as the amount of hypotheses rises, when both methods reach a stable EOP value, our proposal achieves better results. Moreover, our method with only 10 hypotheses (91,26%) beats [170] with 100 hypotheses (89.66%). This shows the good performance of our structural lines selection which increases the likelihood of getting good hypotheses with only a few attempts. Computing times show again bigger difference when fewer hypothesis are evaluated. Only rooms up to 4 walls are considered in this evaluation to get a fair numerical comparison, but our method is also able to deal with more complex rooms, see Figure 2.10 for a qualitative comparison.

**Evaluation in SUN360 and Stanford 2D-3D-S datasets.** Besides the 85 images from the SUN360 dataset, we additionally tested our method with 25 panoramas from the Stanford (2D-3D-S) dataset. In Table 2.2 we show the EOP ratio reached in both datasets. Several reasons can be associated with the fact that our proposal works better with SUN360 dataset. On the one hand, panoramas from the Stanford dataset

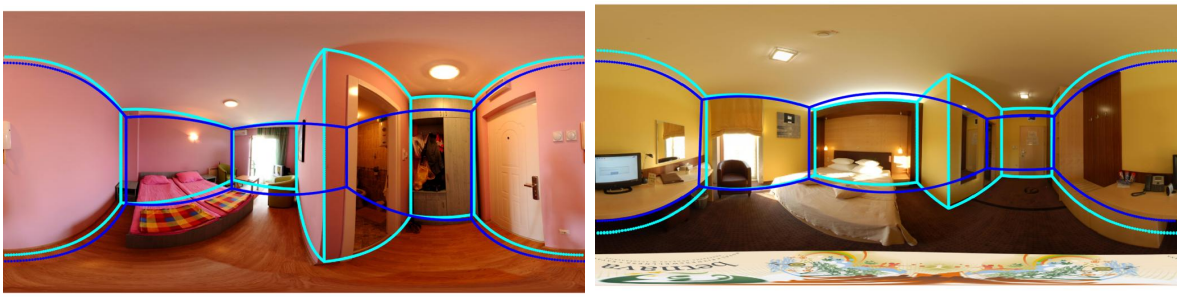


Fig. 2.10 **Comparison with PanoContext** [170] in complex geometries. Our method (cyan) is able to find 6 walls whereas [170] (dark blue) always finds just 4 walls.

| Dataset | Category | EOP ($N_h$ =100) |
|---|---|---|
| LSUN360 | bedroom | 0.921 |
| | livingroom | 0.933 |
| Stanford (2D-3D-S) | area1 | 0.873 |
| | area3 | 0.885 |

Table 2.2 Ratio of equally-oriented pixels evaluated in different scenarios from two public datasets.

Fig. 2.11 **Top**: challenging corridor well estimated by our approach on Stanford (2D-3D-S) dataset. **Bottom**: a clear case of failure.

Fig. 2.12 Layout predictions (cian) and ground truth (red) on the SUN360 dataset. **Left**: cuboid layouts. **Right**: non-cuboid layouts.

do not cover full view vertically, leaving a black mask that can lead to confusions in the limits when extracting structural lines. On the other hand, indoor scenes represented in the Stanford dataset show more challenging scenarios like cluttered laboratories or corridors instead of bedrooms and living-rooms (see Figure 2.11). Still, our method achieves more than 87% of Equally Oriented Pixels in this dataset.

Additional qualitative results on the SUN360 dataset are shown in Figure 2.12.

## 2.5 CFL: Corners for Layout

We note that using conventional methods with panoramic images represents a major bottleneck in most recent methods and the final result after combining the local predictions is not as good as we would desire. This is not surprising as panoramic images contain scene parts, such as the ceiling or visually less conspicuous regions, that generally do not appear in conventional images. Additionally, we make the observation that using standard convolutions with equirectangular images may lead to a loss of performance for several reasons. First, pre-training on conventional images is key due to the lack of training 360 data. However, the pre-trained feature space is non-distorted, which makes the pre-training less effective on the distorted space of the equirectangular

images. Moreover, the right and the left side of the equirectangular images are the same spot in reality. If we apply standard convolutions on these images, the network simply do not understand the continuity of the scene, as the kernel moves limited by the image borders.

In this section, we demonstrate how to create deep learning algorithms to work directly with 360 images, exploiting all their context and reducing the processing time. Additionally, we propose a special type of convolution, named EquiConv, that adapts the size and shape of the kernel to the equirectangular image distortions. EquiConvs can directly substitute the standard convolutions, and demonstrated to have several advantages to work with 360 images. We specially design a Fully Convolutional Network (FCN) for panoramic images to solve the problem of room layout estimation. Compared to previous methods, the proposed network not only outputs edge maps, but also corner maps. We get more structural information and the joint learning helps to reinforce the quality of both map types. We name our method Corners for Layout (CFL).

### 2.5.1 Equirectangular Convolutions

Spherical images are receiving an increasing attention due to the growing number of omnidirectional sensors in drones, robots and autonomous cars. We observe that a naïve application of convolutional networks to an equirectangular projection, is not, in principle, a good choice due to the space-varying distortions introduced by such projection.

In this section we present a convolution that we name EquiConv, which is defined in the spherical domain instead of the image domain and it is implicitly invariant to equirectangular representation distortions. The kernel in EquiConvs is defined as a spherical surface patch –see Figure 2.13. We parametrize its receptive field by the angles $\alpha_w$ and $\alpha_h$. Thus, we directly define a convolution over the field of view. The kernel is rotated and applied along the sphere and its position is defined by the spherical coordinates ($\phi$ and $\theta$ in the figure) of its center. Unlike standard kernels, that are parameterized by their size $k_w \times k_h$, with EquiConvs we define the angular size ($\alpha_w \times \alpha_h$) and resolution ($r_w \times r_h$). In practice, we keep the aspect ratio, $\frac{\alpha_w}{r_w} = \frac{\alpha_h}{r_h}$, and we use square kernels, so we will refer the field of view as $\alpha$ ($\alpha_w = \alpha_h$) and the resolution as $r$ ($r_w = r_h$) respectively from now on. In this work, we choose values of resolution and field of view to be the same as the image.

Although we use by default the same resolution and field of view from the image in our model, it can be different. As we increase the resolution of the kernel, the angular distance between the elements decreases, with the intuitive upper limit of not

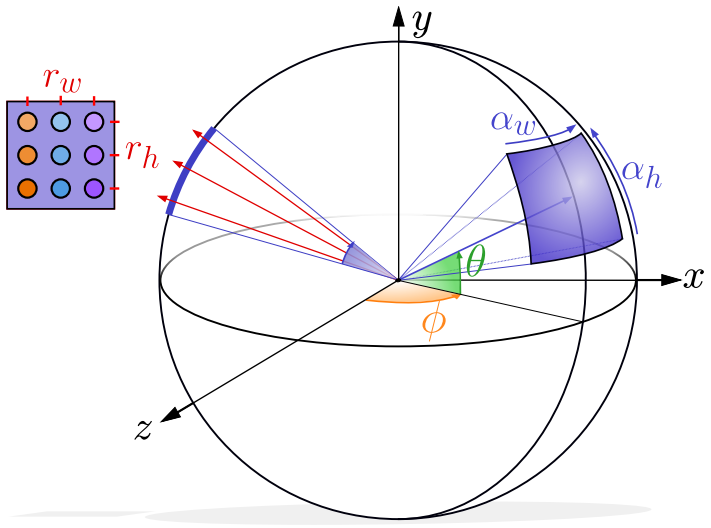



Fig. 2.13 **Spherical parametrization of EquiConvs**. The spherical kernel, defined by its angular size ($\alpha_w \times \alpha_h$) and resolution ($r_w \times r_h$), is convolved around the sphere with angles $\phi$ and $\theta$.

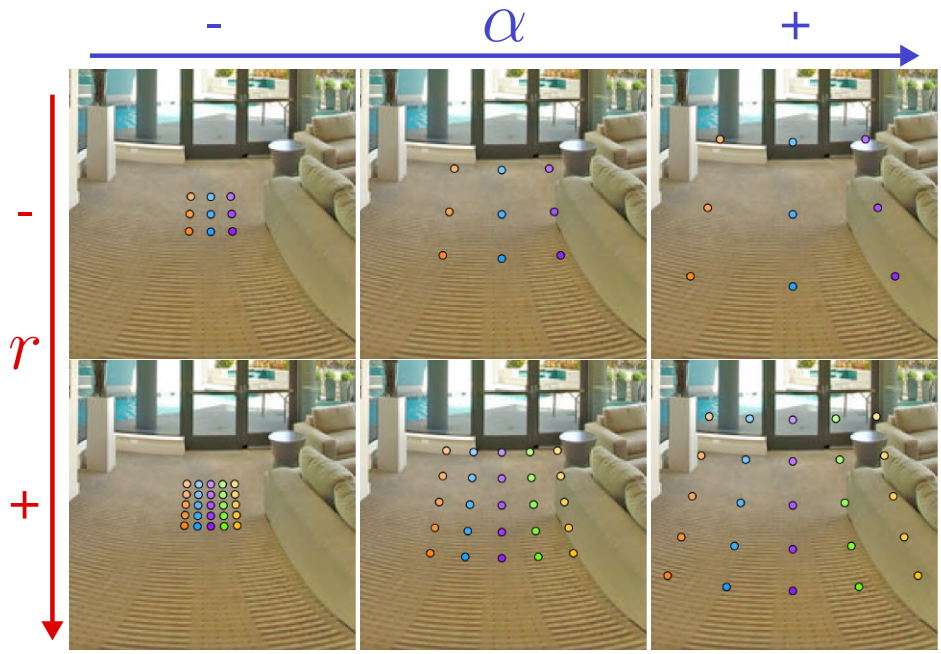



Fig. 2.14 **Effect of changing field of view $\alpha$ (rad) and resolution $r$ in EquiConvs**. 1 column shows a narrow field of view $\alpha = 0.2$. 2 column shows a wider kernel keeping its resolution (atrous-like), $\alpha = 0.5$. 3 column shows an even larger field of view for the kernel, $\alpha = 0.8$. Notice how the kernel adapts to the equirectangular distortion. Rows are resolutions $r = 3$ and $r = 5$.

giving more resolution to the kernel than the image itself. In other words, the kernel is defined in a sphere, being its radius less or equal to the image sphere radius. Therefore, EquiConvs can also be seen as a general model for spherical Atrous Convolutions [20, 21] where the kernel size is what we call resolution, and the rate is the field of view of the kernel divided by the resolution. An example of the differences of EquiConvs by modifying $\alpha$ and $r$ can be seen in Figure 2.14.

### EquiConvs Details

Deformable convolutions are introduced in [27], with the idea of learning offsets from target tasks which are added to the regular kernel locations in order to augment the spatial sampling of Standard Convolutions.

Inspired by this work, we deform the shape of the kernels according to the geometrical priors of the equirectangular image projection. To do that, we generate offsets that are not learned but fixed given the spherical distortion model and constant over the same horizontal locations. Here, we describe how to obtain the distorted pixel locations from the original ones.

Let us define $(u_{0,0}, v_{0,0})$ as the pixel location on the equirectangular image where we apply the convolution operation (*i.e.* the image coordinate where the center of the kernel is located). First, we define the coordinates for every element in the kernel and afterwards we rotate them to the point of the sphere where the kernel is being applied. We define each point of the kernel as follows,

$$\hat{p}_{ij} = \begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \\ \hat{z}_{ij} \end{bmatrix} = \begin{bmatrix} i \\ j \\ d \end{bmatrix}, \tag{2.5}$$

where $i$ and $j$ are integers in the range $[-\frac{r-1}{2}, \frac{r-1}{2}]$ and $d$ is the distance from the center of the sphere to the kernel grid. In order to cover the field of view $\alpha$,

$$d = \frac{r}{2\tan(\frac{\alpha}{2})}. \tag{2.6}$$

We project each point into the sphere surface by normalizing the vectors, and rotate them to align the kernel center to the point where the kernel is applied.

$$p_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix} = R_y(\phi_{0,0}) R_x(\theta_{0,0}) \frac{\hat{p}_{ij}}{|\hat{p}_{ij}|}, \tag{2.7}$$

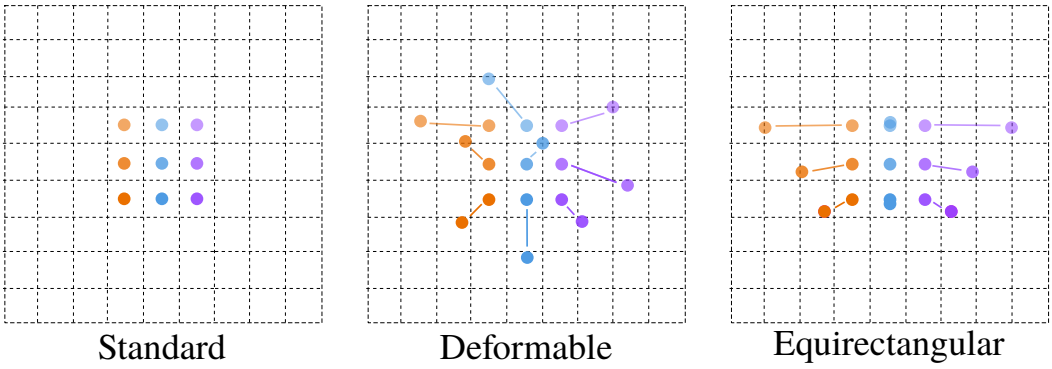



Fig. 2.15 **Effect of offsets on a $3 \times 3$ kernel.** Left: Regular kernel in Standard Convolution. Center: Deformable kernel in [27]. Right: Spherical surface patch in EquiConvs.


Fig. 2.16 **EquiConvs on spherical images.** We show three kernel positions to highlight the differences between the offsets. As we approach to the poles (larger $\theta$ angles) the deformation of the kernel on the equirectangular image is bigger, in order to reproduce a regular kernel on the sphere surface. Additionally, with EquiConvs, we do not use padding when the kernel is on the border of the image since offsets take the points to their correct position on the other side of the 360° image.

where $R_a(\beta)$ stands for a rotation matrix of an angle $\beta$ around the $a$ axis. $\phi_{0,0}$ and $\theta_{0,0}$ are the spherical angles of the center of the kernel –see Figure 2.13, and are defined as

$$\phi_{0,0} = (u_{0,0} - \frac{W}{2})\frac{2\pi}{W} \quad ; \quad \theta_{0,0} = -(v_{0,0} - \frac{H}{2})\frac{\pi}{H}, \tag{2.8}$$

where $W$ and $H$ are, respectively, the width and height of the equirectangular image in pixels. Finally, the rest of elements are back-projected to the equirectangular image domain. First, we convert the unit sphere coordinates to latitude and longitude angles:

$$\phi_{ij} = \arctan\,(\frac{x_{ij}}{z_{ij}}) \quad ; \quad \theta_{ij} = \arcsin\,(y_{ij}). \tag{2.9}$$

And then, to the original 2D equirectangular image domain:

$$u_{ij} = (\frac{\phi_{ij}}{2\pi} + \frac{1}{2})W \quad ; \quad v_{ij} = (-\frac{\theta_{ij}}{\pi} + \frac{1}{2})H. \tag{2.10}$$

In Figure 2.15 we show how these offsets are applied to a regular kernel; and in Figure 2.16 three kernel samples on the spherical and on the equirectangular images.

### 2.5.2 Learning Corners for Layout

In this section, we describe our end-to-end approach that receives as input a single 360° image and estimates the 2D room corner coordinates from which the 3D room layout is directly generated. The proposed network architecture, see Figure 2.17, can be trained using Standard Convolutions (CFL StdConvs) and the proposed Equirectangular Convolutions implementation (CFL EquiConvs). Our FCN follows the encoder-decoder structure and builds upon ResNet-50 [59]. We replace the final fully-connected layer with a decoder that jointly predicts layout edges and corners locations already refined.

**Encoder.** Most of deep-learning approaches facing layout recovery problem have made use of the VGG16 [128] as encoder [91, 30, 79]. Instead, [173] builds their model over ResNet-101 [59] outperforming the state of the art. Here, we use ResNet-50 [59], pre-trained on the ImageNet dataset [118], which leads to a faster convergence due to the general low-level features learned from ImageNet. Residual networks allow us to increase the depth without increasing the number of parameters with respect to their plain counterparts. This leads, in ResNet-50, to capture a receptive field of $483 \times 483$ pixels, enough for our input resolution of $256 \times 128$ pixels.

**Decoder.** Most of the recent work [91, 175, 113] builds two output branches for multi-task learning, which increases the computation time and the network parameters. We instead propose a unique branch with two output channels, corners and edge maps, which helps to reinforce the quality of both map types. In the decoder, we combine two different ideas. First, skip-connections [114] from the encoder to the decoder. Specifically, we concatenate "up-convolved" features with their corresponding features from the contracting part. Second, we do preliminary predictions at lower resolutions which are also concatenated and fed back to the network following the spirit of [34], ensuring early stages of internal features aim for the task. We use ReLU as non-linear function except for the prediction layers, where we use Sigmoid.

**Objective output.** The ground truth (GT) for every panorama consists of a set of corner coordinates. With this coordinates we generate two probability maps, $m$, one represents the room edges ($m = e$), *i.e.* intersections between walls, ceiling and floor, and the other encodes the corner locations ($m = c$). Both maps are defined as $\mathcal{Y}^m = \{y_1^m, \ldots, y_i^m, \ldots\}$, with pixel values $y_i^m \in \{0, 1\}$. $y_i^m$ has a value of 1 if it belongs to an edge or a corner, and 0 otherwise. Dealing with the image at pixel level is very noise-sensitive so we do line thickening and Gaussian blur for easier convergence during training since it makes the loss progression continuous instead of binary. The loss will be gradually reduced as the prediction approaches the target.

Notice here that our target is considerably simpler than others that usually divide the ground truth into different classes. This contributes to the small computational footprint of our proposal. For example, [91, 173] use independent feature maps for background, wall-floor, wall-wall and wall-ceiling edges. Different segmentation images for left, front and right wall, ceiling and floor categories are used in [30]. In [79], they represent a total of 48 different corner types by a 2D Gaussian heatmap centered at the true keypoint location. Here, instead, we only use two probability maps, one for edges and another one for corners.

**Loss function.** Edge and corner maps are learned through a pixel-wise sigmoid cross-entropy loss function. Since we know a priori that the natural distribution of pixels in these maps is extremely unbalanced ($\sim 95\%$ have a value of 0), we introduce weighting factors to make the training stable. Defining as 1 and 0 the positive and negative labels, the weighting factors are defined as $w_t = \frac{N}{N_t}$, being $N$ the total number of pixels and $N_t$ the amount of pixels of class $t$ per sample. The per-pixel per-map loss $\mathcal{L}_i^m$ is as follows:

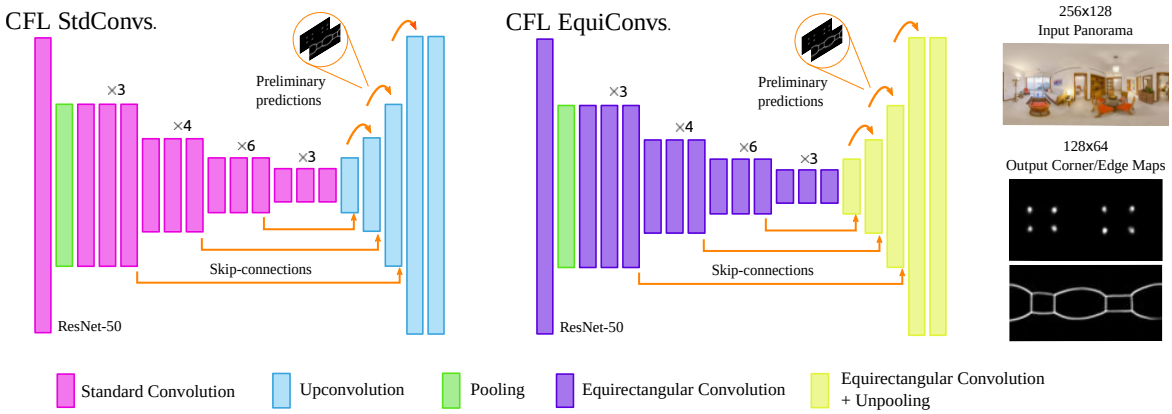



Fig. 2.17 **CFL architecture**. Our network is built upon ResNet-50, adding a single decoder that jointly predicts edge and corner maps. There are two network variations: CFL StdConvs, applies standard convolutions and upconvolutions on the equirectangular panorama, whereas CFL EquiConvs applies Equirectangular Convolutions and Equirectangular Convolutions + unpooling directly on the sphere.

$$\begin{aligned} \mathcal{L}_i^m &= w_1\Big(y_i^m\Big(-\log(\hat{y}_i^m)\Big)\Big) + \\ &+ w_0\Big((1-y_i^m)\Big(-\log(1-\hat{y}_i^m)\Big)\Big), \end{aligned} \tag{2.11}$$

where $y_i^m$ is the objective value for pixel $i$ in the map $m$ and $\hat{y}_i^m$ is the network output for pixel $i$ and map $m$. We minimize this loss at 4 different resolutions $k = \{1, \ldots, 4\}$, specifically in the network output ($k = 4$) and 3 intermediate layers ($k = \{1, \ldots, 3\}$). The total loss used to learn the probability maps is then the sum over all pixels, the 4 resolutions and both the edge and corner maps

$$\mathcal{L}_{maps} = \sum_{k=\{1,\ldots,4\}} \sum_{m=\{e,c\}} \sum_i \mathcal{L}_i^m[k]\,. \tag{2.12}$$

**From corner maps to 3D layout.** Current methods [175, 43, 170] use pre-computed vanishing points and posterior optimizations, being constrained to produce strict Manhattan 3D layouts. Aiming to a fast end-to-end simple model, CFL avoids extra computation and adopt a representation usually referred as Soft/Weak Manhattan [47] or Atlanta World [70]. Following this, horizontal directions are not necessarily orthogonal to each other, thus relaxing the model assumptions. To this end, we simply follow a natural transformation from corners coordinates to 2D and 3D layout. The 2D corners coordinates are the maximum activations in the probability map.


(a) The 2D corners coordinates are the maximum activations in the probability map. From the 2D corners, we can directly recover the 3D layout by doing a couple of assumptions.

(b) Assumptions. (i) ceiling and floor planes are parallel and oriented with the gravity direction, (ii) the camera is located at a certain height.

Fig. 2.18 **Layout from corner predictions**. From the corner probability map, the coordinates with maximum values are directly selected to generate the layout.

Assuming that the corner set is consistent, they are directly joined, from left to right, in the unit sphere space and re-projected to the equirectangular image plane. The 3D layout is inferred by only assuming ceiling-floor parallelism, leaving the wall structure unconstrained i.e. we assume that the floor corners are on the same plane and the top corners are directly above the floor ones, but we do not force the usual Manhattan perpendicularity between walls. Corners are projected to floor and ceiling planes given a unitary camera height (trivial as results are up to scale). See Figure 2.18.

We directly join corners from left to right, meaning that our model could not be able to infer the correct order of the corners if any wall is occluded because of the convexity of the scene. In those particular cases, the joining process should follow a different order. In Section 2.4.2 we propose a geometry-based post-processing that could alleviate this problem, but its cost is high and it requires the Manhattan World assumption.

In order to generate the 3D room layout, we do the following. We can define a plane as the set of all points $P = (x, y, z)$ such that $P \cdot N + d = 0$, where the normal $N = (n_x, n_y, n_z)$ is a normalized vector perpendicular to its surface and $d$ is the distance that separates it from the origin of coordinates in the direction of the normal. Since we assume ceiling-floor parallelism and a camera height, $N$ of both the floor and ceiling planes is equal and corresponds to the vertical direction $v_z$, and the distance $d$ from the floor to the camera is known. The distance to the ceiling is yet unknown. Additionally, thanks to the nature of spherical images, we can easily obtain the 3D ray $R(t) = O + \vec{V} \cdot t$ (parametric representation) going from the center of the sphere $O = (o_x, o_y, o_z)$ through the corner position, with normalized direction vector

$\vec{V} = (v_x, v_y, v_z)$. To obtain the normalized direction vector $\vec{V}$, we need the corner position in the sphere, thus we transform the image coordinates of the corners $(u, v)$ into spherical coordinates and then to the Euclidean 3D space. Equations for this are presented in Section 2.3.

In the first place, Eq (2.13) give us the angles that define the point $(u, v)$ in the sphere.

$$\phi = (u - \frac{W}{2})\frac{2\pi}{W} \quad ; \quad \theta = -(v - \frac{H}{2})\frac{\pi}{H} \tag{2.13}$$

Where W and H are the width and height of the equirectangular image. Second, once these rotations are known we can compute the direction of the ray. Therefore, using Eq (2.14) we can calculate $\vec{V}$.

$$\vec{V} = \begin{bmatrix} -\cos(\theta)\sin(\phi) \\ \sin(\theta) \\ \cos(\theta)\cos(\phi) \end{bmatrix} \tag{2.14}$$

The intersection between the corner ray and the corresponding floor or ceiling plane will give us the actual 3D corner point $P = (x, y, z)$ (up to scale), i.e. the intersection represents that point $P$ on the surface of the plane that verifies the ray equation: $(o_x + v_x \cdot t)n_x + (o_y + v_y \cdot t)n_y + (o_z + v_z \cdot t)n_z + d = 0$. The point $P$ of intersection would simply be the result of evaluating the calculated $t$, Eq (2.15), in the ray equation $R(t)$.

$$t = -\frac{o_x n_x + o_y n_y + o_z n_z + d}{v_x n_x + v_y n_y + v_z n_z} \tag{2.15}$$

Let's consider we have performed the operations to compute one corner point on the floor plane, $P^F = (x^F, y^F, z^F)$. The corresponding point on the ceiling plane ($P^C$) will be on top of it (*ie.* $x^F = x^C$ and $y^F = y^C$). Therefore, we can use this to compute $t^C$, Eq (2.16), and thus the ceiling point:

$$t^C = \frac{(x^F - o_x)}{v_x^C} \tag{2.16}$$

where $\vec{V}^C = (v_x^C, v_y^C, v_z^C)$ is computed as in (2.14) with the corresponding ceiling point in the image. Notice that with $P^C$ we have the information we were missing to recover the ceiling plane.

### 2.5.3 Experimental Results

We present a set of experiments to evaluate CFL using both Standard Convolutions (StdConvs) and the proposed Equirectangular Convolutions (EquiConvs). We do not only analyze the corner maps predicted by our model, but also the impact of each algorithmic component through ablation studies. We report the performance of our proposal in two different datasets, and show qualitative 2D and 3D models of different indoor scenes.

**Datasets.** We use two public datasets that comprise several indoor scenes, SUN360 [157] and Stanford (2D-3D-S) [6] in equirectangular projection (360°). The former is used for ablation studies, and both are used for comparison against several state-of-the-art baselines.

[1] SUN360 [157]: We use ∼500 bedroom and livingroom panoramas from this dataset labeled by Zhang *et al.* [170]. We use these labels but, since all panoramas were labeled as box-type rooms, we hand-label and substitute 35 panoramas representing more faithfully the actual shapes of the rooms. We split the raw dataset in 85% training scenes and 15% test scenes randomly by making sure that there were rooms of more than 4 walls in both partitions.

[2] Stanford 2D-3D-S [6]: This dataset contains more challenging scenarios like cluttered laboratories or corridors. In [175], they use areas 1, 2, 4, 6 for training, and area 5 for testing. For our experiments we use same partitions and the ground truth provided by them.

**Implementation details.** The input to the network is a single panoramic RGB image of resolution $256 \times 128$, unlike [175] that uses also vanishing lines as input. The outputs are the room layout edge map and corner map, both of them at resolution $128 \times 64$. A widely used strategy to improve generalization of neural networks is data augmentation. We apply random erasing, horizontal mirroring as well as horizontal rotation from 0° to 360° of input images during training. The weights are all initialized using ResNet-50 [59] trained on ImageNet [118]. For CFL EquiConvs we use the same kernel resolutions and field of views as in ResNet-50. This means that for a standard 3×3 kernel applied to a W×H feature map, $r = 3$ and $\alpha = r\frac{fov}{W}$, where $fov = 360°$ for panoramas. We minimize the cross-entropy loss using Adam [72], regularized by penalizing the loss with the sum of the L2 of all weights. The initial learning rate is $2.5e^{-4}$ and is exponentially decayed by a rate of 0.995 every epoch. We apply a dropout

| | | | Corners | | | | |
|---|---|---|---|---|---|---|---|
| **Conv.** | **IP** | **EM** | *IoU* | *Acc* | *P* | *R* | $F_1$ |
| | | | : 1 | : 1 | : 1 | : 1 | : 1 |
| StdConvs | - | - | 0.519 | 0.978 | 0.611 | 0.763 | 0.675 |
| StdConvs | - | ✓ | 0.531 | 0.979 | 0.639 | 0.749 | 0.685 |
| StdConvs | ✓ | ✓ | 0.569 | 0.982 | 0.684 | 0.761 | 0.718 |
| EquiConvs | - | - | 0.485 | 0.972 | 0.551 | 0.786 | 0.642 |
| EquiConvs | - | ✓ | 0.536 | 0.980 | 0.649 | 0.744 | 0.690 |
| EquiConvs | ✓ | ✓ | **0.580** | **0.983** | **0.697** | **0.762** | **0.726** |
| | | | *bigger is better* | | | | |

Table 2.3 **Ablation study on SUN360 dataset.** We show results for both Standard Convolutions (StdConvs) and our proposed Equirectangular Convolutions (EquiConvs) with some modifications: Using or not intermediate predictions (IP) in the decoder and edge map predictions (EM).

rate of 0.3.Although we label some panoramas more accurately to their actual shape, we still have a big unbalanced dataset. In order to overcome this problem, we choose a batch size of 16 and we force it always to include one example between those panoramas hand labeled by us (not box-type). This favors the learning of more complex rooms despite having few examples. The network is implemented using TensorFlow [1] and trained and tested in a NVIDIA Titan X. The training time for StdConvs is around 1 hour and the test time is 0.31 seconds per image. For EquiConvs, training takes 3 hours and test around 3.32 seconds per image.

**Network's output evaluation.** We measure the quality of our predicted probability corner maps using five standard metrics: intersection over union *IoU*, precision *P*, recall *R*, F1 Score $F_1$ and accuracy *Acc*. Table 2.3 summarizes our results and allows us to answer the following questions:

*What are the effects of different convolutions?* As one would expect, EquiConvs, aware of the distortion model, learn in a non-distorted generic feature space achieving accurate predictions, like StdConvs on conventional images [79]. Distortion understanding, additionally, gives the network other advantages. While StdConvs learn strong bias correlation between features and distortion patterns (*e.g.* ceiling line on the top of the image or clutter in the mid-bottom), EquiConvs are invariant to that. For this reason, the performance of EquiConvs does not degrade when varying the camera DOF pose – see Section 2.5.3. Additionally, EquiConvs allow to directly leverage


Fig. 2.19 **EquiConvs show more consistent qualitative results** whereas StdConvs simply do not understand that the image wraps around the sphere, losing the continuous context that these images provide.

networks pre-trained on conventional images. Specifically, this translates into a faster convergence, which is desirable as, to date, 360° datasets contain far less images than datasets with conventional images. In omnidirectional images, the right and the left edge are the same spot in reality so, another strength of EquiConvs lie in the fact that we can avoid padding when the kernel reaches the border of the image since offsets take the points to their correct position on the other side of the 360° image. This allows the model to understand the continuity of the scene. StdConvs, instead, simply do not understand that the image wraps around the sphere. As a consequence, in most cases when corners approach the borders, StdConvs predict these corners twice, i.e. at both ends, or the edges at one side would not coincide with the edges at the other side. This effect is highlighted in Figure 2.19 and further demonstrated in a supplementary video[1].

*How can we refine predictions?* There are some techniques that we can use in order to obtain more accurate and refined predictions. Here, we make pyramid preliminary predictions in the decoder and iteratively refine them, by feeding them back to the network, until the final prediction. Also, although we only use the corner map to recover the layout of the room, we train the network to additionally predict edge maps as an auxiliary task. This is another representation of the same task that ensures that the network learns to exploit the relationship between both outputs, *i.e.*, the network learns how edges intersect between them generating the corners. The improvement is shown in the Table 2.3.

[1]https://www.youtube.com/watch?v=dK_vsVYiPaAfeature=emb_logo

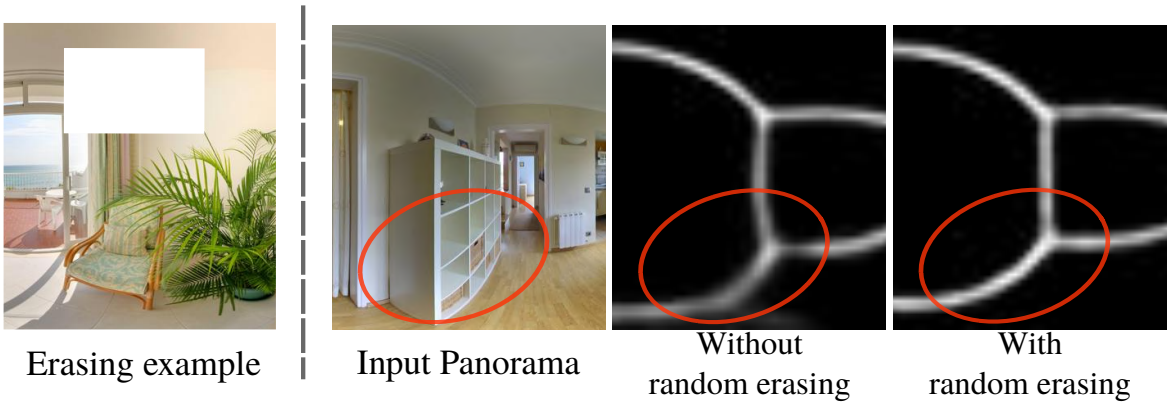



Fig. 2.20 **Augmenting the data with virtual occlusions.** Left: Image with erased pixels. Right: Input panorama and predictions without and with pixel erasing. Notice the improvement by random erasing.

*How can we deal with occlusions?* We do Random Erasing Data Augmentation. This operation randomly selects rectangles in the training images and removes its content, generating various levels of virtual occlusion. In this manner we simulate real situations where objects in the scene occlude the corners of the room layout, and force the network to learn context-aware features to overcome this challenging situation. Figure 2.20 illustrates this strategy with an example.

*Is it possible to relax the scene assumptions while keeping a good performance?* By avoiding constrained Manhattan 3D layout predictions we not only achieve better results compared with current arts, but also we save in computation. Additionally, our model overcomes the classic box-room simplification (four-walls room setups), even if we still have a largely unbalanced dataset after labeling some panoramas more accurately to their actual shape. We address this problem by choosing a batch size of 16 and forcing it to always include one non-box sample. This favors the learning of more complex rooms despite having few examples.

We also show in Figure 2.21 three examples of our predicted edge maps with different number of walls compared to the ground truth maps, to our previous approach [43] and to LayoutNet [175]. Our proposal is able to directly handle network outputs not limited to 4-wall rooms, which demonstrates that is possible to train strategically in a way that the network takes full advantage of the few different data that we have at our disposal.

| | | $F_1$ | $Acc$ | $IoU$ |
|---|---|---|---|---|
| **Trans** | StdConvs | $55.32 \pm 8.23$ | $95.46 \pm 1.3$ | $39.135 \pm 7.82$ |
| | EquiConvs | $\mathbf{59.55 \pm 8.95}$ | $\mathbf{96.21 \pm 1.14}$ | $\mathbf{43.47 \pm 8.83}$ |
| **Rot x** | StdConvs | $45.89 \pm 14.72$ | $93.44 \pm 3.18$ | $31.26 \pm 12.83$ |
| | EquiConvs | $\mathbf{46.2 \pm 15.1}$ | $\mathbf{94.43 \pm 2.18}$ | $\mathbf{31.625 \pm 13.41}$ |
| **Rot y** | StdConvs | $72.28 \pm 2.7$ | $98.21 \pm 0.21$ | $57.54 \pm 3.25$ |
| | EquiConvs | $\mathbf{72.96 \pm 2.02}$ | $\mathbf{98.29 \pm 0.14}$ | $\mathbf{58.44 \pm 2.44}$ |

Table 2.4 **Robustness analysis**. Values represent the mean value (*bigger is better*) $\pm$ standard deviation (*smaller is better*) in %. We apply three types of transformations to the panoramas: translations in $y$ dependant on the room height from $-0.3h$ to $0.3h$, rotations in $x$ from $-30°$ to $+30°$ and rotations in $y$ from $0°$ to $360°$. We do not use these images for training but just for testing in order to show the generalization capabilities of both models.

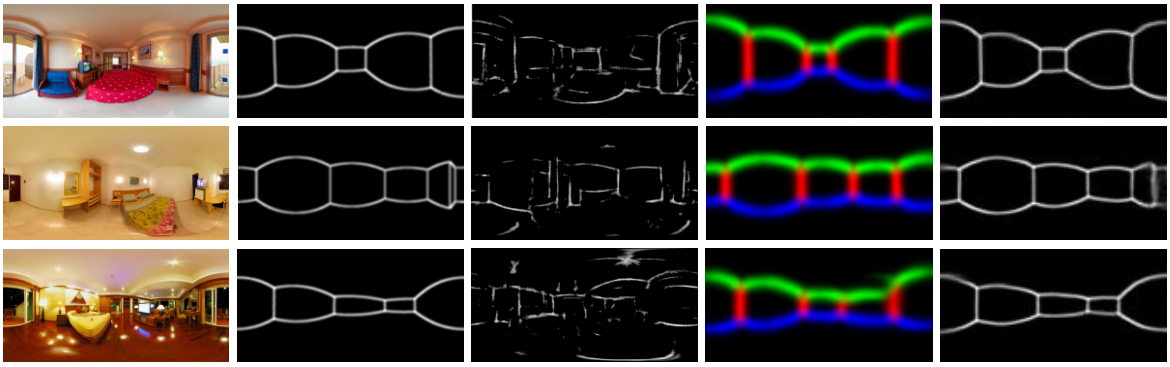


Fig. 2.21 **Predicted Edge Maps**. From left to right: Input RGB images, ground truth edge maps, results from [43], results from [175], our results. Observe that our FCN predicts cleaner edges around the boundaries and recovers faithful edge maps even when indoor scenes are not simple cuboids (middle row).

**Robustness analysis.** We test our model with previously unseen images where the camera viewpoint is different from that in the training set. The distortion in equirectangular projection is location dependent, specifically, it depends on the polar angle $\theta$. Since EquiConvs are invariant to this distortion, it is interesting to see how modifications in the camera extrinsic parameters (translation and rotation) affect the model performance using EquiConvs against StdConvs. When we generate translations (over vertical axis $y$) and rotations (over horizontal axis $x$), the shape of the layout is modified by the distortion, losing its characteristic pattern (which StdConvs use in its favor).

Since standard datasets have a strong bias when referring to camera pose and rotation, we synthetically render these transformations along our test set. The rotation

Fig. 2.22 **Synthetic images for robustness analysis.** Here we show two examples of panoramas generated with upward translation in $y$ and rotation in $x$ respectively.

is trivial as we work on the spherical domain. As the complete 3D dense model of the rooms is not available, the translation simulation is performed by using the existing information, ignoring occlusions produced by viewpoint changes. Nevertheless, as we do not work with wide translations the effect is minimal and images are realistic enough to prove the point we want to highlight (see Figure 2.22). For both experiments, we uniformly sample from a minimum to a maximum transformation and calculate the mean and standard deviation for all the metrics. What we see in Table 2.4 is that we obtain higher mean values by using EquiConvs. This means that this EquiConvs make the model more robust and generalizable to real life situations, not covered in the datasets, *e.g.* panoramas taken by hand, drones or small robots.

We also quantitatively analyzed the robustness of the model to rotation over the vertical axis $y$. Even though this rotation do not distort the shape of the layout like the previous extrinsic parameters, the incapability of StdConvs to wrap around the sphere and understand the continuity of the scene was a frequent source of failure as we showed in Figure 2.19 and the supplementary video. Table 2.4 compare both convolutions, where the numbers represent the mean of the results obtained from each panorama after doing all possible rotations (from 0° to 360° horizontally) and computing mean and standard deviation per panorama. Results show that EquiConvs not only have better overall performance, but the standard deviation is much smaller since there are no special cases that cause failure due to lack of continuity in the borders.

**Synthetic translations generation.** To generate the synthetic translations we use the 3D layout reconstruction from the ground truth. The idea is to obtain the color values of each pixel in the new synthetic image, with similar reasoning about ray to plane intersection that can be found in Section III. For each pixel of this image we can recover its spherical coordinates (see Equation (2.13)) and the direction of the ray

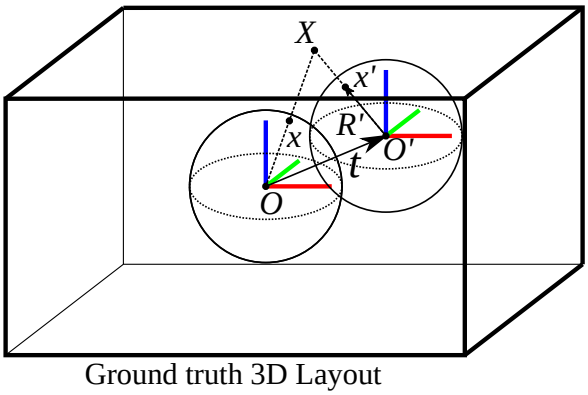



Fig. 2.23 Generation of synthetic translated image from the ground truth layout. Each point in the sphere in the new reference ($x'$) takes the color from the projection of the 3D intersection point $X$ to the sphere in the old reference ($x$).

emanating from the reference frame (see Equation. (2.14)). In Figure 2.23 we represent the point in the unitary sphere $x'$ and the corresponding ray $R'$ on the translated reference frame. We compute the intersection of the ray $R'$ and the ground truth layout as a ray to plane intersection as explained in Section III, specifically Equation. (2.15). To simulate the translation, the origin of the sphere $O'$ will be set accordingly as $O+t$. Since the layout has several planes and the ray could intersect more than one, the closest point is always selected. Once we have the 3D point $X$ we can change its reference frame (*ie.* subtract $t$) and project it back to the sphere before the translation to recover its point $x$ whose color value can be recovered by going back to the original equirectangular image. We go through all the pixels in the new synthetic image until it is completely filled with color values. Since we only have the ground truth 3D layout but not the complete 3D reconstruction, the objects in the scene could appear deformed because of the change in perspective if translations are too large. The effect of this change in perspective was not noticeable for the translations we applied, and it did not affect the results.

**3D Layout comparison.** We evaluate our layout predictions using three standard metrics, 3D intersection over union $3DIoU$, corner error $CE$ and pixel error $PE$, and compare ourselves against four approaches from the state of the art [170, 175, 43, 161]. Pano2CAD [158] has no source code available nor evaluation of layouts, making direct comparison difficult. The pixel error metric given by [175] only distinguishes between ceiling, floor and walls, $PE^{SS}$. Instead our proposed segmented mask distinguish between ceiling, floor and each wall separately, $PE^{CS}$, which is more informative since it also has into account errors in wall-wall boundaries. For all experiments, only

| Test | Method | $3DIoU$ | $CE$ | $PE^{SS}$ | $PE^{CS}$ |
|---|---|---|---|---|---|
| SUN360 | PanoContext [170] | 67.22 | 1.60 | 4.55 | 10.34 |
| | Fernandez [43] | - | - | - | 7.26 |
| | LayoutNet [175] | 74.48 | 1.06 | 3.34 | - |
| | DuLa-Net [161] | 77.42 | - | - | - |
| | CFL StdConvs | 78.79 | 0.79 | **2.49** | 3.33 |
| | CFL EquiConvs | **78.87** | **0.75** | 2.6 | **3.03** |
| Std.2D3D | Fernandez [43] | - | - | - | 12.1 |
| | LayoutNet [175] | 64.56 | 1.44 | 5.16 | - |
| | CFL StdConvs | 65.13 | **1.44** | **4.75** | **6.05** |
| | CFL EquiConvs | **65.23** | 1.64 | 5.52 | 7.11 |
| | | | *smaller is better* | | |

Table 2.5 **Layout results** on both datasets (in %), training on SUN360 data. *SS*: Simple Segmentation (3 categories): ceiling, floor and walls [175]. *CS*: Complete Segmentation: ceiling, floor, $\text{wall}_1$,..., $\text{wall}_n$ [43]. Observe how our method outperforms all the baselines in all the metrics.

SUN360 dataset is used for training. Table 2.5 shows the performance of our proposal testing on both datasets, SUN360 and Stanford 2D-3D. Results are averaged across all images. It can be seen that our approach outperforms the state of the art clearly, in all the metrics.

It is worth mentioning that our approach, not only obtains better accuracy but also it recovers shapes more faithful to the real ones, since it can handle non box-type room designs with few training examples. In Table 2.6 we show that, apart from achieving better localization of layout corners, our model is much faster. Our full method with EquiConvs takes 3.47 seconds (0.3 fps) to process one room and with StdConvs just 0.46 seconds (2.2 fps), which is a major advantage considering the aforementioned applications of layout recovery need to be real-time (robot navigation, AR/VR).

## 2.6 Qualitative Results

Here we show some qualitative results of our recovered layouts in SUN360 [157] and Stanford 2D-3D [6] datasets. Figure 2.24 shows some results on both datasets with the corresponding 3D reconstruction of the room. Figures 2.25 and 2.26 collect examples in SUN360 dataset and show indoor scenes with different geometries, not only cuboid shapes. Figure 2.27 shows examples in Stanford 2D-3D dataset. Panoramas in this

| Method | Computation Time (s) |
|---|---|
| PanoContext [170] | > 300 |
| LayoutNet [175] | 44.73 |
| DuLa-Net [161] | 13.43 |
| CFL EquiConvs | 3.47 |
| CFL StdConvs | **0.46** |

Table 2.6 **Average computing time per image.** Every approach is evaluated using NVIDIA Titan X and Intel Xeon 3.5 GHz (6 cores) except DuLa-Net, evaluated using NVIDIA 1080Ti GPU. Our end-to-end method is more than 100 times faster than other methods.

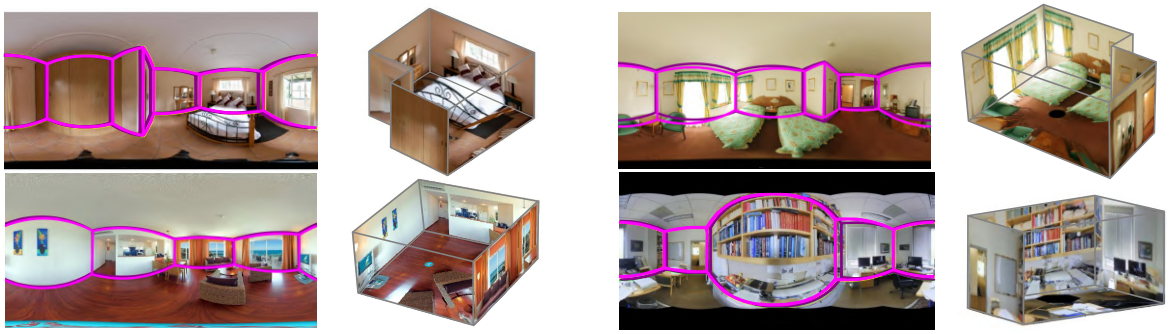


Fig. 2.24 Layout predictions (light magenta) and ground truth (dark magenta) on both datasets.

dataset do not cover full view vertically and the indoor scenes represent more challenging scenarios like cluttered laboratories or corridors.

## 2.7 Conclusion

In this chapter we present two different approaches that show an evolution of our research on the 3D room layout estimation problem from single 360 images.

We first propose a novel pipeline that combines geometry and deep learning to obtain structural lines and corners, from which the layout hypotheses are generated. We also demonstrate how to deal with non-visible structural corners by automatically predicting new corners during the hypotheses generation process, so that the generated room layouts satisfy the Manhattan world assumption. This idea allows us to generalize to cuboid and non-cuboid layouts, leaving behind the simplification of 4 wall rooms.

We additionally present a new deep learning model to predict structural lines and corners directly on panoramic images. The CNN allows us to avoid expensive

Fig. 2.25 Layout predictions (light magenta) and ground truth (dark magenta) on the SUN360 annotation dataset [157]. Best viewed in color.

Fig. 2.26 Layout predictions (light magenta) and ground truth (dark magenta) for **complex room geometries** on the SUN360 annotation dataset [157]. Best viewed in color.

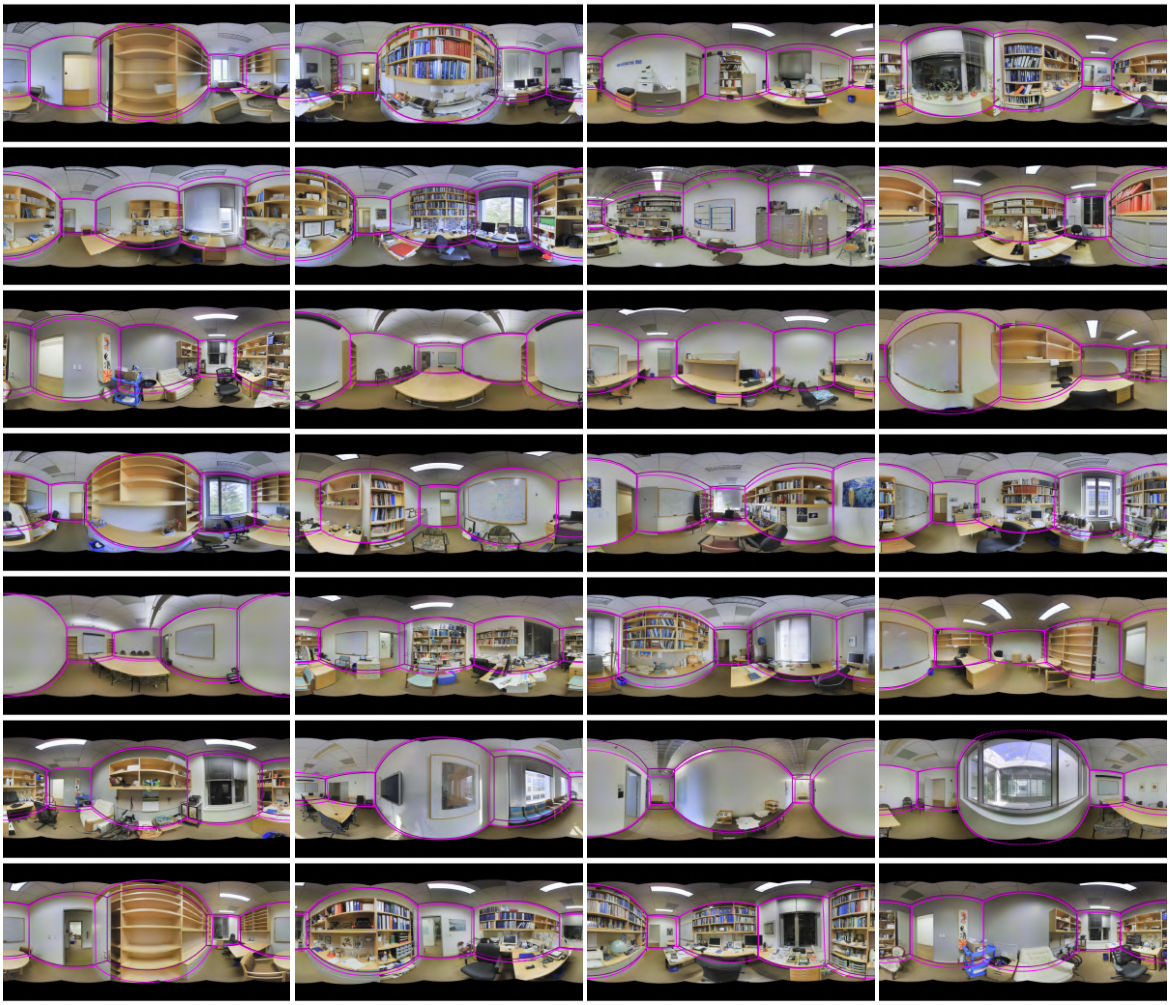


Fig. 2.27 Layout predictions (light magenta) and ground truth (dark magenta) on the Stanford 2D-3D annotation dataset [6]. Best viewed in color.

pre-processing stages improving the overall efficiency of the method. Additionally, working directly on panoramic images ensures a full leverage of the room context, giving better predictions.

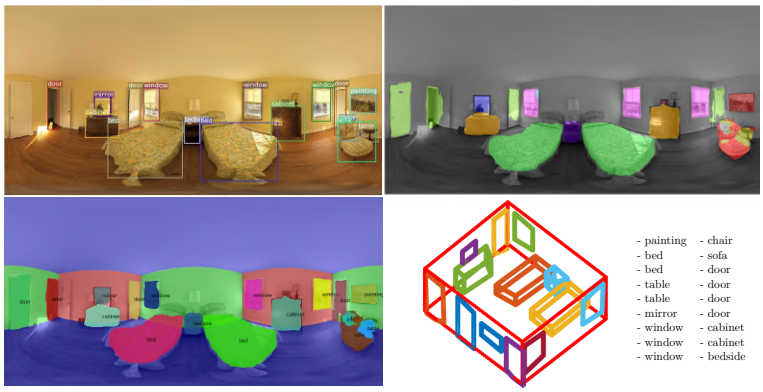
In the last approach, we present CFL, the first end-to-end algorithm for layout recovery in 360° images. Our experimental results demonstrate that our predicted layouts are more accurate than the state of the art. Additionally, the removal of extra pre- and post-processing stages makes our method much faster than other works. Finally, being entirely data-driven relaxes the geometric assumptions that are commonly used in the state of the art and limits their usability in complex geometries. We present two different variants of CFL. The first one, implemented using Standard Convolutions, reduces the computation in 100 times and it is very suitable for images taken with a tripod (recommended if the time is a critical issue). The second one uses our proposed implementation of Equirectangular Convolutions that adapt their shape to the equirectangular projection of the spherical image (recommended if looking for robustness and better generalization). This proves to be more robust to translations and rotations of the camera making it ideal for panoramas taken by a hand-held camera.

# Chapter 3

# Object Recognition

"*The Three R's of Computer Vision: Recognition, Reconstruction & Reorganization.*"
— Jitendra Malik

*In the last few years, there has been a growing interest in panoramic images. While several tasks have been improved thanks to the contextual information these images offer, object recognition in indoor scenes still remains a challenging problem that has not been deeply investigated. We provide an object recognition system that performs object detection and semantic segmentation tasks by using a deep learning model adapted to match the nature of equirectangular images. From these results, instance segmentation masks are recovered, refined and transformed into 3D bounding boxes that are placed into the 3D model of the room. The proposed method outperforms the state of the art by a large margin and shows a complete understanding of the main objects in indoor scenes.*



## 3.1 Introduction

The increasing interest in autonomous mobile systems, like drones, robotic vacuum cleaners or assistant robots, makes detection and recognition of objects in indoor environments a very important and demanded task.

Since recognizing a visual concept is relatively trivial for a human, it is worth considering the hard challenges inherently involved. Objects in images can be oriented in many different ways, vary their size, be occluded, blended into the environment because of their color or appearance, or affected by different illumination conditions, which changes drastically their aspect on the pixel level. Moreover, the concept behind an object's name is sometimes broad, including non-clear frontiers to other concepts. For example, where do you consider the limits between a sofa and an armchair?

Convolutional Neural Networks (CNNs) have already demonstrated to be the best known models to perform object recognition, as they are capable of dealing with those challenges by automatically learning objects' inherent features and correctly identify their intrinsic concepts.

However, images from conventional cameras have a small field of view, much smaller than human vision, which implies that contextual information cannot be as useful as it should. To overcome this limitation, a real impact came with the arrival of the 360° full-view panoramic images, which are recently arising more and more interest in the robotics and computer vision community, as they allow us to visualize, in a single image, the whole scene at the same time. Together with all of their potential we have to deal with challenges produced by their own spherical projection, such as distortion, or the lack of complete, labeled and massive datasets. This requires the development of specific techniques that take advantage of their strengths and allow working with panoramic images in an efficient and effective way.

In this Chapter, we propose an object recognition system that provides a complete understanding of the main objects in an indoor scene from a single 360° image in equirectangular projection. Our method extends the BlitzNet model [35] to perform both object detection and semantic segmentation tasks but adapted to match the nature of the equirectangular image input. We train the network to predict 14 different classes of main indoor scenes related objects. Results of the CNN are post-processed to obtain instance segmentation masks, which are successfully refined by taking advantage of the spatial contextual clues that the room layout provides. In this work, we not only show the potential of exploiting the 2D room layout to improve the instance segmentation mask, but also the possibility of leveraging the 3D layout to generate 3D object bounding boxes directly from the improved masks.

## 3.2 Related Work

Object detection field has been mainly dominated by two different approaches: one-stage and two-stage detectors. Two-stage detectors, as the first R-CNN[51] architecture followed by its variants Fast R-CNN[50], Faster R-CNN[112] and Mask R-CNN[57] achieve great accuracy but lower speed. They require firstly to refine proposals to obtain the features needed to classify the objects. On the other hand, one-stage detectors, following YOLO[110] and SSD[85] simultaneous bounding box refinement and classification, significantly reduce computational cost. They achieve real-time performing maintaining high accuracy, which is needed for most applications in autonomous mobile systems. SSD multi-scale pyramid idea proves to help in conducting more accurate detections and manage widely various object sizes, approach followed in most state-of-the-art object detectors.

While all those models optimize bounding box detection, not so many integrate in their pipeline the pixel-wise recognition needed for many applications. In this way, BlitzNet [35] is a one-stage multi-scale model that adds semantic segmentation and therefore recognizes objects at pixel level. It also proves the advantages of jointly learning two scene understanding tasks: object detection and semantic segmentation, which benefit from each other by sharing almost the complete network architecture.

However, state-of-the-art research mainly focuses on using conventional images. Their limited field of view prevents contextual information from being as crucial as it is in scene understanding for humans. Differently from outdoor object recognition, where thanks to the increasing research on autonomous driving, there are recent works using panoramic images [92] [162], there is no wide research on object recognition from indoor panoramas. A recent work that addresses this problem is [32], where Deng *et. al* use a R-CNN approach, and also evaluate their own implementation of DPM [45] on panoramas. The most relevant work on indoor panoramic object recognition is PanoContext [170]. It includes 2D object detection and semantic segmentation among other 3D scene understanding tasks, proving the potential of having a larger field of view for recognition problems. Their method, nevertheless, is based on geometrical reasoning and traditional computer vision feature extractors and can be still considered as state-of-the-art in indoor object recognition on panoramic images. Recent research on this kind of images includes 3D layout recovery [97] [69] [175] [41] and scene modeling [163], which provides global context and gives a 3D interpretation of the scene from a single view. In [89], they show that this tasks can also benefit and augment an omnidirectional SLAM. Combining object recognition and 3D layout recovery motivates our proposal to obtain the 3D recognition and location of main objects in our room.

## 3.3 Dataset extension

Panoramic images datasets with object recognition labels are not as standard or complete as conventional images ones [67] [132] [6]. Therefore, in this work we decide to extend the SUN360 database [67] with segmentation labels. For every panorama, we generate individual masks encoding each object's spatial layout. Additionally, we combine all the masks obtaining a semantic segmentation panoramic image with per-pixel classification. Bedroom and living room sets, formed by 418 and 248 images respectively, are used and 14 different object classes are considered. The dataset is divided into 85% for train and validation and 15% for test.

We generate segmentation masks based on 2D bounding points of the objects, taken from PanoContext [170] work. As shown in Figure 3.1 left, we project them on the spherical domain to follow distortion patterns in contours and to correctly manage objects that appear cropped on the horizontal image limits (see Section 2.3 for more details about the spherical geometry). To combine the binary masks and create the semantic segmentation panorama, with the lack of depth or other 3D information, an hypothesis of occlusion among objects is needed. We consider the assumption that objects are not in general completely occluded, and therefore for each pair of objects in conflict their area of overlap and size are computed. If area of overlap is bigger than a threshold, the smallest object is considered closer and completely visible and otherwise the biggest one is selected. With its evident limitations, this hypothesis experimentally proves to work well in most of the cases, allowing to correctly segment most of the visible and cuboid-shaped objects in images as shown in Figure 3.1. The complete **dataset** used in this work is released for public access and can be found in the project webpage[1].

## 3.4 Model

In this section we present our object recognition model, called *Panoramic BlitzNet*, that is based on the original CNN BlitzNet [35] but adapted to work specifically with complete equirectangular images. It addresses both object detection and semantic segmentation tasks, following BlitzNet architecture: a Fully Convolutional model that follows the encoder-decoder approach with skip connections. It performs multi-scale recognition and takes advantage of joint learning. Main changes to their base implementation include the use of the complete rectangular panorama, modifying the

[1] Available at https://webdiis.unizar.es/~jguerrer/room_OR/

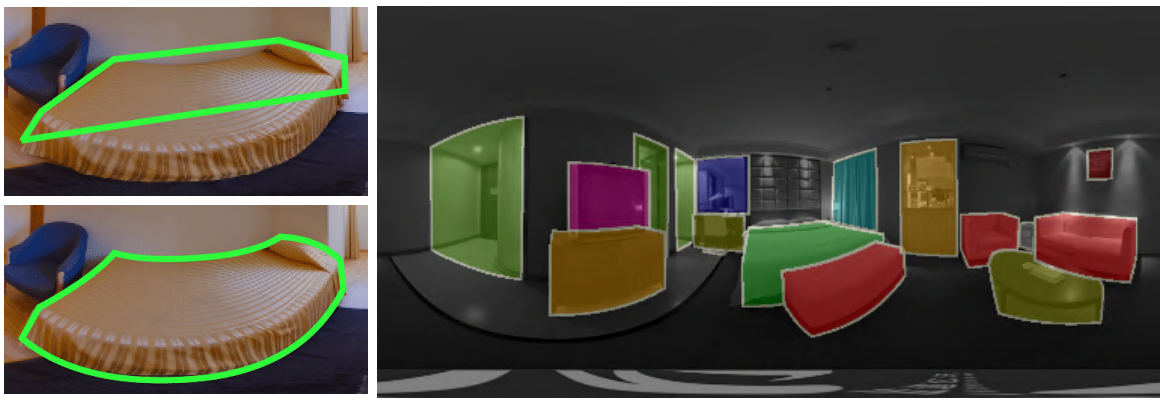


Fig. 3.1 Result of our method to **create semantic segmentation masks**, assuming hypothesis of occlusion. Notice on the left the differences between creating straight contours on image domain (top) vs. spherical domain (bottom).

input aspect ratio. We also change the anchor boxes proposals, as the new input shape needs to be considered because they are centered on pixels grid. Our bounding boxes proposals are done by firstly converting image to a regular grid, covering the whole rectangular-shaped image. Grid has different dimensions in each layer, from 128x256 to 1x2, because of the iteratively lower scale of the feature maps. In each grid cell 5 different proposals are created with 5 different aspect ratios: 1, 2, 1/2, 3 and 1/3, allowing the network to manage different object shapes.

Special mention deserves data augmentation as an important technique to avoid overfitting, particularly on non-massive datasets like in our case. Here, we modify the original data augmentation by removing random crops on images (contextual information is important) and adding horizontal rotation from 0° to 360° to cover all different positions on the sphere.

### 3.4.1 Dealing with 360° images distortion

We exploit the potential of omnidirectional images covering 360° horizontal and 180° vertical field of view represented in equirectangular projection. While these images allow us to analyse the whole scene at once taking advantage of all the context, they present great distortions due to their projection of the sphere. Here, we replace all standard convolutions of our *Panoramic BlitzNet* by equirectangular convolutions (EquiConvs [41]), to study their impact on the task of recognizing objects. With this kind of convolutions, the kernel adapts its shape and size accordingly to the distortions produced by the equirectangular projection. As mentioned in [41], the distortion

presented is location dependent, specifically, it depends on the polar angle. They demonstrate how EquiConvs can be really convenient to generalize to different camera positions since the layout shape can suffer from many variations. For the specific task of object recognition, the use of EquiConvs is definitely convenient even if the camera is always at the same place, since objects can be at many different locations inside the scene –*e.g.* objects closer to the camera will have greater distortions than objects around the horizon line. EquiConvs here play an important role since they can learn ignoring this distortion patterns and thus, being more able to learn real objects appearance. Additionally, one important challenge to accomplish our goal is represented by the need of extensive annotations for training object recognition. To this end, EquiConvs make the pre-training much more effective since this type of convolutions implicitly handle equirectangular distortion, being able to use previous weights from conventional images as if they were learnt on the same kind of images. We can therefore exploit the wealth of publicly available perspective datasets for training – SUN RGB-D [119] dataset in this case, which reduces the cost of annotations and allows training under a larger variety of scenarios. Moreover, standard convolutions do not understand that the image wraps around the sphere, loosing the continuity of the scene, while EquiConvs, working directly on the spherical space, avoid padding and exploit this idea. This makes them also very suitable for the objects that appear cut between the left and right side of the image.

### 3.4.2 From semantic to instance segmentation

Semantic segmentation masks allow us to pixel-wise classify scenes in object categories. One step further goes instance segmentation, which classifies each pixel not only to its category but also differentiating its concrete object instance, an essential stage to correctly locate them into the 3D reconstruction of the room. Without using any instance segmentation prior to be learnt, we add a simple post-processing to obtain pixel-wise instance classification of the scene, based on the outputs of the network. Considering each bounding box detection as a Gaussian distribution, it is assumed that 99% of the object is contained on it, so standard deviation in each dimension is taken as $\sigma_w = \frac{width}{6}, \sigma_h = \frac{height}{6}$, giving $3\sigma$ at each side of the mean, which is defined as the center of the bounding box. Then, each pixel from the semantic segmentation mask is assigned to the instance distribution with the minimum Mahalanobis distance. When none of the distances to distributions exceeds the chi-squared test threshold, the pixel is left as the classification in the initial semantic segmentation, a proposal that shown the best experimental results.

This assumption gives, consequently, more importance to bigger objects, which are usually better segmented by our model. In addition to providing an approach to create instance segmentation masks, we analyze the impact it can have in improving our initial segmentation, as shown in experiments.

### 3.4.3 From instance segmentation masks to 3D bounding boxes

If there is a task that has experimented a disruptive innovation with the emergence of 360° images, it is the room layout estimation problem [69, 175, 43, 41, 136]. In these works, from a single panorama, the goal is to recover the main structure of the room, i.e. disposition of the walls, ceiling and floor, not only in the image domain, but also a complete 3D reconstruction model up to scale.

The hypothesis here is that objects location and pose inside a 3D indoor space are not randomly distributed. Following the laws of physics, objects will be fairly constrained to lie on at least one supporting plane, in stable configurations and, in several cases, aligned with the room walls. This implies that the room layout provides strong spatial contextual cues as to where and how objects can be found. Thus, in order to provide a greater understanding of the scene, we analyze the potential of using the room layout as a *prior* for the object recognition task:

(i) We found that we can easily leverage the room layout in the image domain to improve the instance segmentation masks. Based on the contextual information given by the layout, there are a series of logical assumptions that we can immediately make –*e.g.* it is very unlikely to find doors not resting on the floor or paintings hanging in between two walls, i.e. we assume no floating objects and objects aligned to walls. During this process we also detect holes in masks and fill them.

(ii) The aforementioned methods provide 3D layout models of the rooms. This allows us to place the identified objects inside the 3D model of the room as long as they lie on the walls, or rest on the floor / ceiling aligned with the walls. Even if we only need the object to be in contact with a wall in order to recover its height, and not necessarily aligned to it, our observation is that when an object is in contact with a wall, it is mostly always aligned to it and vice versa, when an object is aligned to a wall, it is generally in contact with it. In this way, only by detecting the masks of the objects and with a good layout prior, we can obtain a very precise 2D representation of the objects and an initial estimate of the 3D understanding of the scene. To obtain the room layout, here we choose

our previous work [41] (Section 2.5). Additionally, we rely on the Manhattan World assumption [24], whereby there exist three dominant orthogonal directions defining the scene. To compute the vanishing directions of the scene we follow the our previous approach [43] (Section 2.4). The proposed RANSAC-based algorithm works directly on omnidirectional images running up to 5 times faster than other approaches.

For the 3D object recognition task, PanoContext [170] generates many cuboid hypotheses combining two approaches. First, it performs rectangle detection in six axis-aligned views projected from the original panorama. Then, it samples rays from the vanishing points to fit image segmentation boundaries obtained by selective search. Finally, the best cuboid whose projection has the largest intersection over union score with the segment is chosen. Here instead, we directly approximate every object mask with four lines by a RANSAC approach, classify the lines according to the vanishing directions to obtain the object orientation inside the scene, and lift the 2D predictions to 3D by interaction with the room layout planes (walls, ceiling and floor).

The proposed algorithm receives as input $K$ binary masks with same resolution as the original panorama $W \times H$, one for each of the $K$ instance objects in the scene. Each pixel has a value of 1 if it belongs to the object of interest and 0 otherwise. We also input the vanishing points, i.e. Manhattan directions of the scene $vp = (vp_x, vp_y, vp_z) \in \mathbb{R}^{3\times3}$, the 3D layout corners up to a scale $\hat{C}_L = (c_1, c_2, \ldots, c_N),\ c_k \in \mathbb{R}^3,\ k \in \{1, 2, \ldots, N\}$, and the 2D room layout segmentation map $\mathcal{L}_s$. The layout segmentation map, with resolution $W \times H$, is generated from the 2D layout corners. Each pixel encodes the direction $x, y$ or $z$ of the surface (walls, ceiling or floor) it belongs to. Mathematically, we define an object in 3D, in the same scale of the given room layout, by the tuple of corners of its approximate cuboid shape (or rectangular shape if it is a planar object) $\hat{C}_O = (c_1, c_2, \ldots, c_M),\ c_j \in \mathbb{R}^3,\ j \in \{1, 2, \ldots, M\}$, where $M = 8$ or 4. Consequently, the proposed algorithm outputs such set of corners for the $K$ objects detected in the scene if they satisfy our two assumptions, i.e. they are not floating and they are aligned to walls. We assume that objects not aligned with the main directions of the scene are not in contact with the room walls and therefore, their height is irrecoverable only with the layout prior.

For simplicity, we describe the method for one single object, which is applied in the same way for the $K - 1$ remaining objects. First, we find the polygon that encloses the mask of the instance object, $P_O = (p_1, p_2, \ldots, p_T),\ p_i \in \mathbb{R}^2,\ i \in \{1, 2, \ldots, T\}$, i.e. the object mask contour. A RANSAC approach receives as input the retrieved object polygon $P_O$ and returns the four lines that best fit the polygon (See Figure 3.2). The

procedure is similar to the RANSAC algorithm described in Section 2.4.1. We transform the polygon coordinates in the image into the corresponding ray directions in the 3D space using the spherical coordinates, $p_i \in \mathbb{R}^3$. Iteratively, two rays are randomly selected to compute a normal direction for a possible line as $\mathsf{n} = (p_{r1} \times p_{r2}), p_{r1}, p_{r2} \in P_O$, and the number of inliers is evaluated. We consider as inliers, all the rays fulfilling the condition of perpendicularity with respect to the computed normal under an angular threshold $\theta_{th} = \pm 0.5°$, $|arccos(\mathsf{n} \cdot p_i) - \frac{\pi}{2}| \leq \theta_{th}$. After some iterations, the algorithm returns the longest line found, i.e. the line with more inliers, which are saved and removed from the list of coordinates $P_O$ to continue looking for the remaining lines. The process is repeated several times. The best solution is returned as the four lines that, in total, contain the maximum number of the initial polygon coordinates, i.e. ideally $P_O$ empties. After getting the lines that best fit the object mask, we aim at getting their orientation according to the main directions of the scene. We consider that a line belongs to one concrete direction $k \in \{x, y, z\}$, when its normal direction in the 3D space fulfills the condition of perpendicularity with the direction $k$, $|arccos(\mathsf{n} \cdot vp_k) - \frac{\pi}{2}| \leq \theta_{th}$. If the object is oriented with the scene, we determine with which planes of the room it interacts by obtaining the intersection between the object binary mask and the layout segmentation map $\mathcal{L}_s$. Since the layout is known, also the mapping between its 2D and its 3D versions is known (See Section 2.5 for more details). Given the vanishing points and the 3D layout corners, we define each plane of the room in 3D (wall, ceiling or floor) $\Pi$ by the room corners that limit its extension, the normal of the plane $\mathsf{n}_\Pi \in \{x, y, z\}$ and its distance to the camera center $d_\Pi$. Planar objects, i.e. objects lying on a wall like doors, windows, mirrors or pictures, are simply projected to the corresponding wall in the 3D room layout. To do so, the image coordinates of the object corners are transformed into ray directions as usual, and then intersected with the corresponding room plane in 3D. For cuboid objects like beds, sofas or bedside tables, the object is placed in 3D similarly, using the object lines that intersect and align with the room planes. The same information is used to recover the dimensions (length, width and height) of the object. See Figure 3.2. With these ideas, we can place most of the objects inside the 3D scene and obtain a good understanding of the scene from the 2D segmentation masks. Some qualitative results are shown in Figure 3.3.

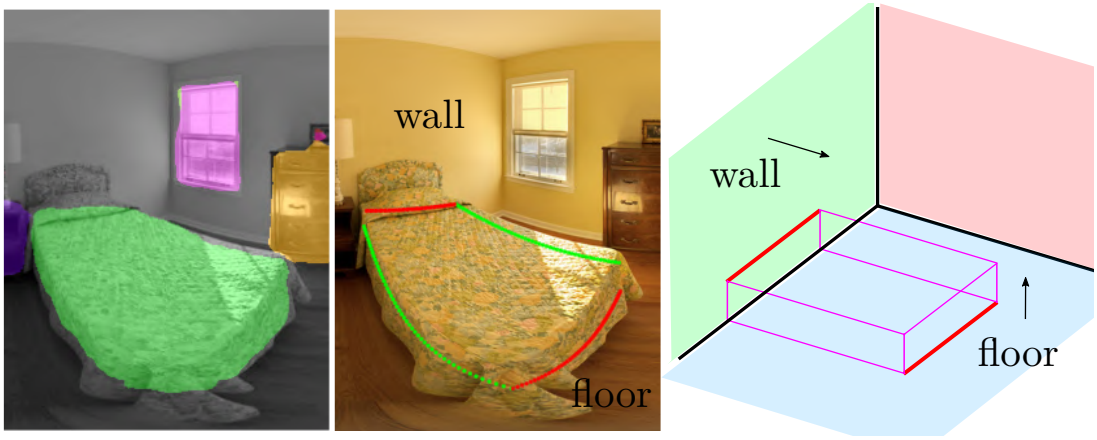



Fig. 3.2 **From mask to 3D**: We find the lines that best fit the object mask by a RANSAC algorithm and orient them accordingly to the main directions of the scene. The object dimensions are obtained trusting in the line resting on the wall and the line resting on the floor (red lines in the figure).

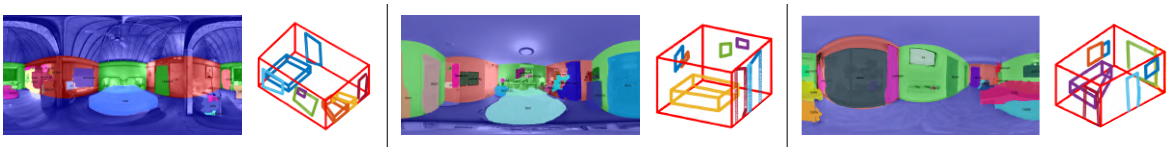


Fig. 3.3 Examples of 3D models obtained from instance objects masks and room layout knowledge [41].

## 3.5 Experimental Results

We evaluate our model by different experiments conducted on SUN360 [67] extended dataset, which are presented in this section. Experimental setup is explained in order to make our work reproducible, together with the detailed evaluation metrics.

**Experimental setup** The whole model is coded in Python 3.5 using the framework Tensorflow v1.13.1. All experiments were conducted on a single Nvidia GeForce GTX 1080 GPU. As in [35], we use ResNet-50 as feature extractor, Adam stochastic algorithm [72] for optimization and learning rate set to $10^{-4}$ and decreased twice during training. Experiments were conducted by changing that learning rate without noticeable influence. We use stride 4 in the last layer of the up-scaling stream and varying mini-batch sizes, which are stated in each experiment. All models are trained until convergence, measured with a random validation subset.

Based on our dataset characteristics, we decide to pre-train our network instead of initializing it randomly, that would conduct to a clear overfit to our data, as studied in the first experiment. Because of the lack of massive panoramic datasets, conventional images are used for pre-training: Firstly, we use the publicly available weights of ResNet50 backbone on ImageNet [117] dataset. With that initialization, we then train the whole network on SUN RGB-D [119], pre-processed to have the same common classes. This way we have an initialization for the complete model to be able to fine-tune with the panoramic images, possible thanks to a Fully Convolutional network where weights can be shared with variable input dimensions.

**Evaluation metrics** To evaluate detection performance we use typical mean average precision ($mAP$), considering that a predicted bounding box is correct if its intersection over union with the ground truth is higher than 0.3, as exact localization is better predicted in the segmentation branch. The average precision evaluates interpolated precision at all different recall levels, in its simplest definition, but can also be widely found in literature as weighted by the recall area that they represent. In this work we mostly use simple $AP$ and when using the second version it is referred as $AP^w$. Finally, when calculating the mean among all different classes we use a weighted mean as defined in equation 3.1, being $M$ the number of classes, $AP_i$ the average precision per class, $d_i$ the number of detections of class $i$ and $n$ the total number of detections. It is considered as a more representative metric because results calculated from objects with a minimum number of samples in the test set should be less significant when analyzing a global performance.

| | TRAIN $mAP^w$ | TEST $mAP^w$ | TRAIN $meanIoU$ | TEST $meanIoU$ |
|---|---|---|---|---|
| From scratch | 0.911 | 0.468 | 0.872 | 0.419 |
| ImageNet (ResNet) | 0.896 | 0.479 | 0.788 | 0.432 |
| SUN RGB-D | 0.728 | **0.516** | 0.742 | **0.461** |

Table 3.1 **Effect of initialization:** Results tested on original BlitzNet with different weights initializations. Notice how initializing with SUN RGB-D weights gives clearly better test results. Evaluation on train set is shown to observe the overfitting effect, specially clear in trained from scratch model.

Segmentation performance is measured with mean intersection over union ($mIoU$), as stated in equation 3.2, being $A_i$ the area formed by all pixels of class $i$ in the ground truth segmentation map, and $\hat{A}_i$ in the predicted segmentation map.

$$mAP = \sum_{i=1}^{M} \frac{d_i}{n} AP_i \tag{3.1}$$

$$mIoU = \frac{1}{M} \sum_{i=1}^{M} \frac{A_i \cap \hat{A}_i}{A_i \cup \hat{A}_i} \tag{3.2}$$

### 3.5.1 Initialization

First experiment was conducted before the development of our model, to verify the importance of pre-training. It uses original BlitzNet300 architecture, without modifying the base implementation to adapt for panoramas. Batch size is set to 16 and it is trained until convergence. When training, three different initializations are executed to compare: random initialization (trained all from scratch), ImageNet initialization of the feature extractor (ResNet) and pre-trained SUN RGB-D initialization of the complete model. Results are compared in Table 3.1, which shows that pre-training with SUN RGB-D followed by fine-tuning the complete network with panoramic dataset, gives the best performance results and generalizes better. ImageNet initialization converges faster, but it demonstrates higher overfitting than SUN RGB-D pre-training. This experiment shows that given a relatively small panoramic dataset, the use of a massive one of conventional images for pre-training allows the network to learn higher level characteristics of the objects, avoiding overfitting and being one of the keys for the success of the system.

| | $mAP^w$ | $mAP$ | $meanIoU$ |
|---|---|---|---|
| BlitzNet | 0.516 | 0.688 | 0.461 |
| *Panoramic BlitzNet* | **0.632** | **0.768** | **0.530** |

Table 3.2 **Effect of adapting the CNN for panoramas:** Comparison between results on panoramic images with BlitzNet vs. our proposed *Panoramic BlitzNet*.

| input | mIoU | backgr | bed | picture | table | mirror | window | curtain | chair | light | sofa | door | cabinet | bedside | tv | shelf |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ours | 53.0 | 90.7 | 61.7 | **32.1** | **75.2** | **42.3** | 55.8 | 54.0 | 55.1 | 31.4 | **34.6** | **63.6** | 48.5 | **40.7** | 52.2 | 57.4 |
| Ours+I | **53.1** | 90.7 | **63.3** | 30.1 | 75.0 | 41.5 | **56.7** | **55.3** | **55.5** | **34.0** | 31.8 | 62.9 | **48.8** | 40.2 | **53.5** | **57.5** |

Table 3.3 Semantic segmentation results before and after applying the instance segmentation post-processing. Initial semantic segmentation is taken from our CNN output.

### 3.5.2 Square versus Panoramic

This experiment compares the performance of our *Panoramic BlitzNet* network with the original model designed for conventional images. In this case, batch size is reduced to 4, for memory limitations in our GPU. As seen in Table 3.2, our adapted model demonstrates clearly better results, improving performance by a wide margin and supporting our assumption of the important benefits of a concrete model designed for panoramas.

Apart from avoiding distortions and crops to make it fit to a square shape, it also shows the benefits of using a wider field of view, which allows to consider the whole context of the room. This idea is a strong support for the potential of panoramic images, not only in object recognition but in many other visual tasks, at least in those related to indoor scene understanding problem.

### 3.5.3 StandardConvs versus EquiConvs

Evaluation of the influence that equirectangular convolutions have on object recognition task is a key point in this work. Comparison of both models can be seen in Tables 3.4 and 3.5 and Figure 3.6. There we show that our model with EquiConvs, adapting the kernel to manage the equirectangular distortion, perform better in both object detection and semantic segmentation tasks. Additionally, equirectangular convolutions have other advantages over standard convolutions. They make our pre-training on conventional images more meaningful, as managing distortions by the kernel allows to share the weights as if they were trained on the same kind of images. Therefore, we

| model | mAP | bed | picture | table | mirror | window | curtain | chair | light | sofa | door | cabinet | bedside | tv | shelf |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ∗ [39] | 29.4 | 35.2 | 56.0 | 21.6 | 19.2 | 21.8 | 29.5 | 26.0 | — | 22.2 | 31.9 | — | — | 31.0 | — |
| ∗ [32] | 68.7 | 76.3 | 68.0 | 73.6 | 58.7 | 62.6 | 69.5 | 68.0 | — | 72.5 | 67.3 | — | — | 70.0 | — |
| Ours$^{SC}$ | 76.8 | 94.9 | **85.0** | **83.3** | 71.9 | **72.2** | 72.2 | 71.9 | 35.0 | **89.3** | **75.5** | **57.9** | 87.9 | 91.1 | 30.5 |
| Ours$^{EC}$ | **77.8** | **95.3** | 83.9 | 82.1 | **76.2** | 70.9 | **75.9** | **80.9** | **41.0** | 85.4 | 72.5 | 55.6 | **91.4** | **93.3** | **40.2** |

Table 3.4 **Object detection** results on SUN360 test set with our method *Panoramic BlitzNet* using standard convolutions (SC) versus equirectangular convolutions (EC), compared with previous methods. ∗ Results trained and evaluated on a combination of datasets (including SUN360) by [32]

strongly believe that this type of convolutions help in avoiding overfitting to training data, which due to the particularities of the SUN360 dataset (camera pose does not vary and scenes are relatively similar) does not drastically damage test results, but will probably be crucial when working on different datasets. Finally, EquiConvs also prove to make detections with higher confidence as, when raising the confidence threshold to 0.95, their recall is maintained over 40% compared to 28% achieved with standard convolutions.

### 3.5.4 Instance segmentation

In this experiment we evaluate the influence of the instance segmentation post-processing in the segmentation performance. Our intuition was that the proposed instance segmentation method would imply an improvement to the initial segmentation maps because it gives higher confidence to bounding box detections, whose performance is clearly higher than segmentation's one in our model. Results, shown in Table 3.3, provide a comparison between the semantic segmentation output of the network and the semantic segmentation maps after the instance post-processing. Results are very similar and lead us to conclude that the post-processing does not prove to be influential in this way. However, qualitative results support our intuitive idea by showing some clearly improving cases that are remarked in Figure 3.4.

Our approach proves to work well on several different scenes by correctly separating same category objects, that initially overlapped in semantic maps, into different instances. Limitations of the method can be seen when the network fails detecting an object (bounding box), which is therefore not differentiated as an instance on the final map and when managing objects with complex shapes that can not be modelled with a gaussian distribution.

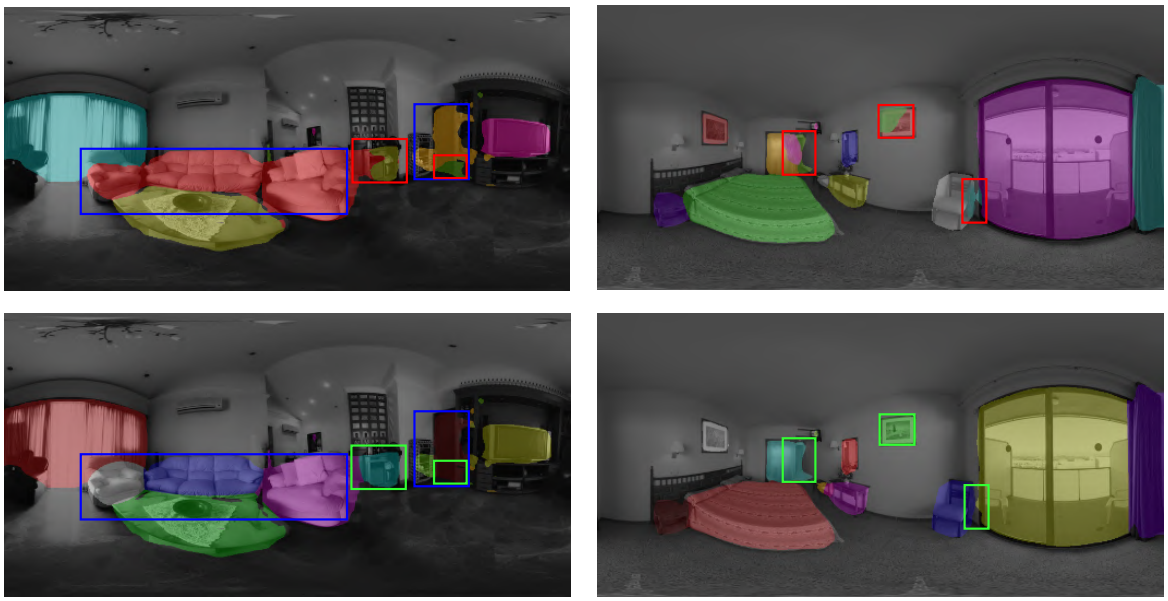


Fig. 3.4 **Instance segmentation post-processing** results. Top is initial semantic segmentation (output of CNN) and bottom is result of post-processing. Notice that apart from correctly differentiate among instances (highlighted in blue) it improves original segmentation (highlighted in red and green for failed and improved segmentation respectively).


Fig. 3.5 After combining the room layout with our segmentation masks, the model experiences a clear improvement as a whole. However, here we want to show a **failure case** where, when assuming that doors must reach the floor, we may have overlapping with other occluding objects in the image, damaging segmentation results but improving the door 3D localization.

| model | mIoU | backgr | bed | picture | table | mirror | window | curtain | chair | light | sofa | door | cabinet | bedside | tv | shelf |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [170] | 37.5 | 86.9 | **78.6** | 38.7 | 29.6 | 38.2 | 35.6 | — | 09.6 | — | 11.1 | 19.4 | 27.4 | 39.7 | 34.8 | — |
| Ours$^{SC}$ | 53.0 | 90.7 | 61.7 | 32.1 | 75.2 | 42.3 | 55.8 | 54.0 | 55.1 | 31.4 | 34.6 | 63.6 | 48.5 | **40.7** | 52.2 | 57.4 |
| Ours$^{EC}$ | 54.4 | **91.3** | 62.1 | 61.2 | 72.3 | 41.1 | 53.4 | 53.7 | 55.2 | 26.5 | 32.9 | **63.8** | 51.1 | 36.6 | **52.3** | **61.9** |
| Ours$^{LP}$ | **60.3** | 89.4 | 66.6 | **78.7** | **75.9** | **69** | **69.5** | **60.2** | **60.5** | **37.7** | **39.1** | 58.5 | **54.3** | **40.7** | 45.6 | 59.4 |

Table 3.5 **Semantic segmentation** results on SUN360 extended test set with our proposed model *Panoramic BlitzNet* using standard convolutions (SC) versus equirectangular convolutions (EC) and after including the layour prior (LP). Comparison with PanoContext [170].

We also analyze the improvement over our segmentation masks by leveraging the contextual information of the room layout. In our experiment, logical assumptions used for this refinement entail a significant improvement of up to 7.3% mIoU with respect to the baseline *Panoramic BlitzNet* with StdConvs, achieving a final **mIoU = 60.3%**. See Table 3.5. It should be noted that the classes that contribute most to this improvement are mirror, window and picture. However, while one would also expect a clear improvement in the door category, we have seen a drop in performance in some cases such as the one shown in Figure 3.5, although it definitely has a positive effect on its location in the 3D room space. As already supported by this preliminary experiment, we propose a promising method to noticeably benefit 2D and 3D object recognition tasks from room layout knowledge, and encourage the idea that it is worth continuing to work in this direction.

### 3.5.5 Comparison with the State of the Art

**Detection.** In Table 3.4 we show our detection results on the SUN360 extended dataset. Our Panoramic BliztNet with EquiConvs achieves very satisfactory results, with a global **mAP = 77.8%**. For completeness, we include here the results of [32], recent work on indoor panoramic object recognition with deep learning, together with their evaluation of the Deformable Parts Model (DPM) [39] on panoramas. Our method achieves the best results in detection for all 10 common classes compared to them. It is worth noting that our approach achieves these results just training with $\sim 400$ panoramas from the SUN360 dataset while they use additional panoramas to train their model. Since their dataset is not public and no code is available, we report directly the results collected in [32].

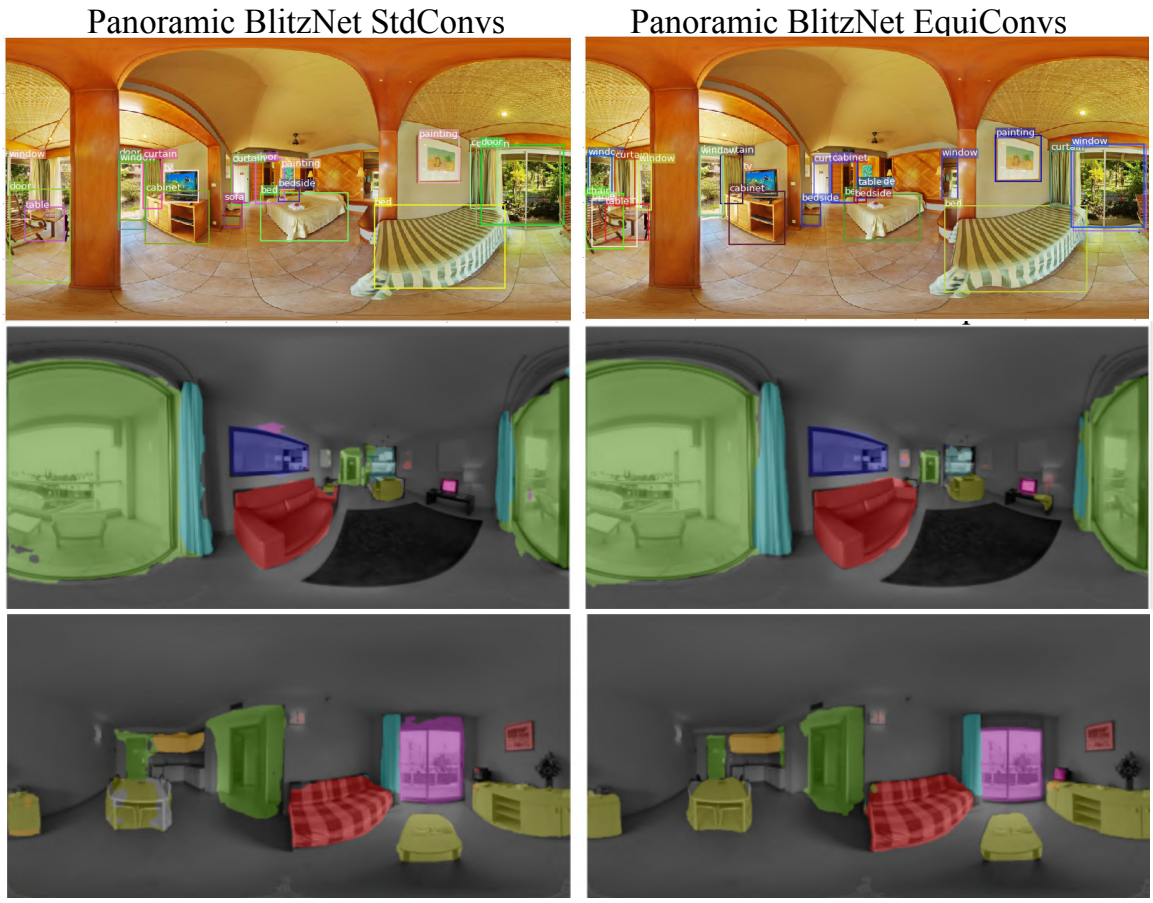



Fig. 3.6 **Qualitative evaluation of object detection and semantic segmentation:** Examples of results obtained with our *Panoramic BlitzNet* using both standard convolutions and EquiConvs [41].

**Segmentation.** Table 3.5 summarizes the semantic segmentation results on the SUN360 extended dataset. A direct comparison is possible with the work of PanoContext [170]. The results clearly show that our method significantly improves over the state of the art. In particular, we add three new object classes and boost **mIoU = 54.4%**, which represents an improvement of 16.9% over PanoContext's method. Additionally, our preliminary experiments on room layout leverage for 3D object detection, show that combining layout and objects under a common scene frame can really benefit the results.

## 3.6 Conclusion

From a single panoramic image, we propose a method that provides a complete understanding of the main objects in an indoor scene. By managing the inherent characteristics and challenges that equirectangular panoramas involve, we outperform state of the art in addition to creating a more complete system, which not only obtains 2D detection and pixel-wise segmentation of objects but also places them into a 3D reconstruction of the room. Exploiting the advantages of having a wider field of view in indoor environments, this visual system becomes a promising key element for future autonomous mobile robots. Future work includes the inclusion of instance segmentation predictions into the deep learning pipeline and a further study of the potential in combining layout recovery and object recognition tasks.

# Chapter 4

# Object Category Shape Modelling

"*- What are the three most important problems in computer vision? - Correspondence, correspondence, correspondence!*"

— Takeo Kanade

*Automatic discovery of category-specific 3D keypoints from a collection of objects of a category is a challenging problem. The difficulty is added when objects are represented by 3D point clouds, with variations in shape and semantic parts and unknown coordinate frames. We define keypoints to be category-specific, if they meaningfully represent objects' shape and their correspondences can be simply established order-wise across all objects in the category. We aim at learning such 3D keypoints, in an unsupervised manner, using a collection of misaligned 3D point clouds of objects from an unknown category. We model shapes defined by the keypoints using symmetric linear basis shapes without assuming the plane of symmetry to be known. The usage of symmetry prior leads us to learn stable keypoints suitable for higher misalignments. To the best of our knowledge, this is the first work on learning such keypoints directly from 3D point clouds for a general category. Using objects from four benchmark datasets, we demonstrate the quality of our learned keypoints by quantitative and qualitative evaluations.*

## 4.1 Introduction

A set of keypoints representing any object is historically of large interest for geometric reasoning, due to their simplicity and ease of handling. Keypoints-based methods [88, 142, 9] have been crucial to the success of many vision applications. A few examples include; 3D reconstruction [96, 28, 129], registration [164, 74, 90, 86], human body pose [126, 95, 19, 14], recognition [57, 122], and generation [139, 167]. That being said, many keypoints are defined manually, while considering their semantic locations such as facial landmarks and human body joints, to address the problem at hand. To further benefit from their widespread utility, several attempts have been made on learning to detect keypoints [64, 103, 171, 33, 166], as well as on automatically discovering them [4, 83, 82, 138]. In this regard, the task of learning to detect keypoints from several supervision examples, has achieved many successes [154, 103]. However, discovering them automatically from unlabeled 3D data –such that they meaningfully represent shapes and semantics– so as to have a similar utility as those of manually defined, has received only limited attention due to its difficulty.

As objects of interest reside in the 3D space, it is not surprising that 3D keypoints are preferred for geometric reasoning. For the given 3D keypoints, their counterparts in 2D images can be associated by merely using camera projection models [159, 62, 152]. However, being able to directly predict keypoints on provided 3D data (point clouds) has the advantage that the task can be achieved when multiple camera views or images are not available. In this work, we are interested on learning keypoints using only 3D structures. In fact, 3D structures with keypoints suffice for several applications including, registration [105], shape completion [93], and shape modeling [111]; without requiring their 2D counterparts.

When 3D objects go through shape variations, due to deformation or when two different objects of a category are compared, consistent keypoints are desired for meaningful geometric reasoning. Recall the examples of semantic keypoints such as facial landmarks and body joints. To serve a similar purpose, *can we automatically find keypoints that are consistent over inter-subject shape variations and intra-subject deformations in a category?* This is the primary question that we are interested to answer in this chapter. Furthermore, we wish to discover such keypoints directly from 3D point sets, in an unsupervised manner. We call these keypoints “category-specific", which are expected to meaningfully represent objects’ shape and offer their correspondence order-wise across all objects. More formally, we define the desired properties of category-specific keypoints as: i) generalizability over different shape instances and alignments in a category, ii) one-to-one ordered correspondences and semantic

consistency, iii) representative of the shape as well as the category while preserving shape symmetry. These properties not only make the representation meaningful, but also tend to enhance the usefulness of keypoints. Learning category-specific keypoints on point clouds, however, is a challenging problem because not all the object parts are always present in a category. The challenges are exacerbated when the practical cases of misaligned data and unsupervised learning are considered. Related works do not address all these problems, but instead opt for; dropping category-specificity and using aligned data [82], employing manual supervision on 2D images [103], or using aligned 3D and multiple 2D images with known pose [138]. The latter method achieves category-specificity without explicitly reasoning on the shapes. Yet another work leverages predefined local shape descriptors and a template model [26] specifically on faces.

In this chapter, we show that the category-specific keypoints with the listed properties can be learned unsupervised by modeling them with non-rigidity, based on unknown linear basis shapes. We further impose an unknown reflective symmetry on the deformation model, when considering categories with instance-wise symmetry. For categories where instance-wise symmetry is not applicable, we propose the use of symmetric linear basis shapes in order to better model, what we define as symmetric deformation spaces, e.g., human body deformations. This allows us to better constrain the pose and the shape coefficients prediction. Our proposed learning method does not assume aligned shapes [138], pre-computed basis shapes [103] or known planes of symmetry [134] and all quantities are learned in an end-to-end manner. Our symmetry modeling is powerful and more flexible compared to that of previous NRSfM methods [48, 134]. We achieve this by considering the shape basis for a category and the reflective plane of symmetry as the neural network weight variables, optimized during the training process. The training is done on a single input, circumventing the Siamese-like architecture used in [82, 164]. At inference time, the network predicts the basis coefficients and the pose in order to estimate the instance-specific keypoints. Using multiple categories from four benchmark datasets, we evaluate the quality of our learned keypoints both quantitatively and with qualitative visualization. Our experiments show that the keypoints discovered by our method are geometrically and semantically consistent, which are measured respectively by intra-category registration and semantic part-wise assignments. We further show that symmetric basis shapes can be used to model symmetric deformation space of categories such as the human body.

## 4.2 Related Work

Category-specific keypoints on objects have been extensively used in NRSfM methods, however, only few methods have tackled the problem of estimating them. In terms of the outcome, our work is closest to [138], which learns category-specific 3D keypoints by solving an auxiliary task of rigid registration between multiple renders of the same shape and by considering the category instances to be pre-aligned. Although the method shows promising results on 2D and 3D, it does so without explicitly modeling the shapes. Consequently, it requires renders of different instances to be pre-aligned to reason on keypoint correspondences between instances. A similar task is also solved in [103] for 6-degrees of freedom (DoF) estimation which uses low-rank shape prior to condition keypoints in 3D. Although, the low-rank shape modeling is a powerful tool, [103] requires supervision for heatmap prediction and relies on aligned shapes and pre-computed shape basis. [154] also predicts keypoints for categories with low-rank shape prior but the method is again trained on fully supervised manner. Moreover, all of the mentioned methods learn keypoints on images as heatmaps and thereafter lift them to 3D. Different from the other works, [26] exploits deformation model and symmetry to directly predict keypoints on 3D but requires a face template, aligned shapes and known basis. Shape modeling of category shape instances has been widely explored in NRSfM works. Linear low-rank shape basis[16, 144, 28], low-rank trajectory basis [3], isometry or piece-wise rigidity [141, 101] are some of the different methods used for NRSfM. Recently, a few number of works have used low-rank shape basis in order to devise learned methods [96, 75, 154, 134]. Another useful tool in modeling shape category is the reflective symmetry, which is also directly related to the object pose. Although [48] showed that the low-rank shape basis can be formulated with unknown reflective symmetry, its adaptation to learned NRSfM methods is not trivial. Recent methods, in fact, assume that the plane of symmetry is one among a few known planes [153]. Moreover, none of the methods formulate symmetry applicable for non-rigidly deforming objects such as the human body. A parallel work [155] on this regard models symmetry probabilistically in a warped canonical space to reconstruct 3D of different objects.

While shape modeling is a key aspect of our work, another challenge is to infer ordered keypoints by learning on unordered point sets. Despite several advances on deep neural networks for point sets [106, 107, 148], current achievements of learning on images dwarf those of learning on point sets. A related work learns to predict 3D keypoints unsupervised by again solving the auxiliary task of correctly estimating rotations in a Siamese architecture [17]. The keypoint prediction is done without

order by pooling features of certain point neighborhoods. Another previous work [164] proposes learning point features for matching, again using alignment as the auxiliary task. Matching such keypoints across shapes is not an easy task as the keypoints are not predicted in any order. In the following sections we show how one can model shape instances using the low-rank symmetric shape basis and use the shape modeling to predict ordered category-specific keypoints.

## 4.3 Background and Theory

### 4.3.1 Category-specific Shape and Keypoints

We represent shapes as point clouds, defined as an unordered set of points $\mathsf{S} = \{\mathsf{s}_1, \mathsf{s}_2, \ldots, \mathsf{s}_M\},\ \mathsf{s}_j \in \mathbb{R}^3,\ j \in \{1, 2, \ldots, M\}$. The set of all such shapes in a category defines the category shape space $\mathcal{C}$. We write a particular $i$-th category-specific shape instance in $\mathcal{C}$ as $\mathsf{S}_i$. For convenience, we will use the terms category-specific shape and shape interchangeably. The category shape space $\mathcal{C}$ can be anything from a set of discrete shapes to a smooth manifold of category-specific shapes spanned by a deformation function $\Psi_{\mathcal{C}}$. The focus of the work is on learning meaningful 3D keypoints from the point set representation of $\mathsf{S}_i$. To that end, this section defines category-specific keypoints and develops their modeling.

**Category-specific keypoints.** We represent category-specific keypoints of a shape $\mathsf{S}_i$ as a sparse tuple of points, $\mathsf{P}_i = (\mathsf{p}_{i1}, \mathsf{p}_{i2}, \ldots, \mathsf{p}_{iN}),\ \mathsf{p}_{ij} \in \mathbb{R}^3,\ j \in \{1, 2, \ldots, N\}$. Unlike the shape, its keypoints are represented as ordered points. Our objective is to learn a mapping $\Pi_{\mathcal{C}} : \mathsf{S}_i \rightarrow \mathsf{P}_i$ in order to obtain the category-specific keypoints from an input shape $\mathsf{S}_i$ in $\mathcal{C}$. Although not completely unambiguous, we can define the category-specific keypoints using the properties listed in Sec. 4.1. In mathematical notations they are:

(i) Generalization: $\Pi_{\mathcal{C}}(\mathsf{S}_i) = \mathsf{P}_i,\ \forall \mathsf{S}_i \in \mathcal{C}$.

(ii) Corresponding points and semantic consistency: Given $\mathsf{S}_a, \mathsf{S}_b \in \mathcal{C}$, we want $\mathsf{p}_{aj} \Leftrightarrow \mathsf{p}_{bj}$. Similarly, $\mathsf{p}_{aj}$ and $\mathsf{p}_{bj}$ should have the same semantics.

(iii) Representative-ness: $\text{vol}(\mathsf{S}_i) = \text{vol}(\mathsf{P}_i)$ and $\mathsf{p}_{ij} \in \mathsf{S}_i$, where vol(.) is the Volume operator for a shape. If $\mathsf{S}_i \in \mathcal{C}$ has a reflective symmetry, $\mathsf{P}_i$ should have the same symmetry.

### 4.3.2 Category-specific Shapes as Instances of Non-Rigidity

Several recent works have modeled shapes in a category as instances of non-rigid deformations [96, 75, 154, 134]. The motivation lies in the fact that such shapes often share geometric similarities. Consequently, there likely exists a deformation function $\Psi_{\mathcal{C}} : \mathsf{S}_T \rightarrow \mathsf{S}_i$, which can map a global shape property $\mathsf{S}_T$ (shape template or basis shapes) to a category shape instance $\mathsf{S}_i$. However, we argue that modeling $\Psi_{\mathcal{C}}$ is not trivial and in fact a convenient representation of $\Psi_{\mathcal{C}}$ may not exist in many cases. This observation, in fact, is what makes the dense Non-Rigid Structure-from-Motion (NRSfM) so challenging. On the other hand, one can imagine a deformation function $\Phi_{\mathcal{C}} : \mathsf{P}_T \rightarrow \mathsf{P}_i$, going from a global keypoints property $\mathsf{P}_T$ to the category-specific keypoints $\mathsf{P}_i$. The deformation function $\Phi_{\mathcal{C}}$ thus satisfies: $\mathsf{p}_{ij} \in \Phi_{\mathcal{C}}$ implies $\mathsf{p}_{ij} \in \Psi_{\mathcal{C}}$ and effectively, $\Phi_{\mathcal{C}} \subset \Psi_{\mathcal{C}}$, if the set order in $\mathsf{P}_i$ is ignored. Unlike $\Psi_{\mathcal{C}}$, the deformation function $\Phi_{\mathcal{C}}$ may be simple enough to model and use for estimating the category-specific keypoints $\mathsf{P}_i$. We therefore, choose to seek the non-rigidity modeling in the space of keypoints $\mathcal{P} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_L\}$, which functions as an abstraction of the space $\mathcal{C}$. Non-rigidity can be used to define the prediction function $\Pi_{\mathcal{C}}$ as below:

$$\Pi_{\mathcal{C}}(\mathsf{S}_i; \theta) = \Phi_{\mathcal{C}}(\mathsf{r}_i; \theta) = \mathsf{P}_i \tag{4.1}$$

where $\theta$ denotes the constant function parameters of $\Pi_{\mathcal{C}}$ and $\mathsf{r}_i$ is the predicted instance specific vector parameter. In our problem, we want to learn $\theta$ from the example shapes in $\mathcal{C}$ without using the ground-truth labels, supervised by $\Phi_{\mathcal{C}}$. In the NRSfM literature, two common approaches of modeling shape deformations are the low-rank shape prior [16, 144, 28, 3] and the isometric prior [141, 101]. In this chapter, we investigate the modeling using the low-rank shape prior, with instance-wise symmetry as well as symmetry of the deformation space.

### 4.3.3 Low-Rank Non-rigid Representation of Keypoints

The NRSfM approach of low-rank shape basis comes as a natural extension of the rigid orthographic factorization prior [143] and was introduced by Bregler et al. [16]. The key idea is that a large number of object deformations can be explained by linearly combining a smaller $K$ number of basis shapes at some pose. In the rigid case, this number is one, hence the rank is 3. In the non-rigid case, it can be higher, while the exact value depends on the complexity of the deformations. Consider $F$ shape instances in $\mathcal{C}$ and $N$ points in each keypoints instance $\mathsf{P}_i$. The following equation

describes the projection with shape basis.

$$\mathsf{P}_i = \Phi_{\mathcal{C}}(\mathsf{r}_i; \theta) = \mathsf{R}_i \, \mathrm{mat}(\mathcal{B}_{\mathcal{C}} \, \mathsf{c}_i) \tag{4.2}$$

where $\mathcal{B}_{\mathcal{C}} = (\mathsf{B}_1, \ldots, \mathsf{B}_K), \mathcal{B}_{\mathcal{C}} \in \mathbb{R}^{3N \times K}$ forms the low-rank shape basis. The rank is lower than the maximum possible rank of $3F$ or $N$ for $3K < 3F$ or $3K < N$. The vector $\mathsf{c}_i \in \mathbb{R}^K$ denotes the coefficients that linearly combines different basis for the keypoints instance $i$. Each keypoints instance is then completely parametrized by the basis $\mathcal{B}_{\mathcal{C}}$ and the coefficients $\mathsf{c}_i$. Next, the projection matrix $\mathsf{R}_i \in SO_3$ is simply the rotation matrix for the shape instance $i$.

Unlike in NRSfM, the problem of computing the category-specific keypoints, has $\mathsf{P}_i$ as unknown. Similar to NRSfM, the rest of the quantities in Eq. (4.2) – $\mathsf{c}_i$, $\mathcal{B}_{\mathcal{C}}$ and $\mathsf{R}_i$ are also unknown. This fact makes our problem doubly hard. First the problem becomes more than just lifting the 2D keypoints to 3D and second, the order of keypoints present in the NRSfM measurements matrix is not available. We intend to solve the aforementioned problems by learning based on Eq. (4.2), which is related to the deformation representation of $\Phi_{\mathcal{C}}$ in Eq. (4.1). Here, $\theta$ includes the global parameters or basis $\mathcal{B}_{\mathcal{C}}$ and $\mathsf{r}_i$ includes the instance-wise pose $\mathsf{R}_i$ and coefficients $\mathsf{c}_i$. To further reduce ambiguities on pose, we propose to also compute the reflective plane of symmetry for a category.

### 4.3.4 Modeling Symmetry with Non-Rigidity

Many object categories have shapes which exhibit a fixed reflective symmetry over the whole category. To discover and use symmetry, we consider two different priors: instance-wise symmetry and symmetric deformation space.

**Instance-wise symmetry.** Instance-wise reflective symmetry about a fixed plane is observed in a large number of rigid object categories (e.g. ShapeNet [165] and ModelNet [156]). Such a symmetry has been previously combined with the shape basis prior in NRSfM [48], however, a convenient representation for learning both the symmetry and the shapes have not been explored yet. A recent learning-based method [153, 134] uses the symmetry prior by performing an exhaustive search over a few planes in order to predict symmetric dense non-rigid shapes. However, such a strategy may not work when the shapes are not perfectly aligned. Instance-wise

symmetry can be included by re-writing Eq. (4.2) as follows:

$$\mathsf{P}_{i\frac{1}{2}} = \mathsf{R}_i \,\mathrm{mat}(\mathcal{B}_{\mathcal{C}\frac{1}{2}}\,\mathsf{c}_i), \quad \mathsf{P}_i = \begin{bmatrix}\mathsf{P}_{i\frac{1}{2}} & A_{\mathcal{C}}\mathsf{P}_{i\frac{1}{2}}\end{bmatrix} \tag{4.3}$$

where $\mathsf{P}_{i\frac{1}{2}} \in \mathbb{R}^{3\times N/2}$ represents one half of the category-specific keypoints. $\mathsf{P}_{i\frac{1}{2}}$ is reflected using $\mathsf{A}_{\mathcal{C}} \in \mathbb{R}^{3\times 3}$ and concatenated to obtain the final keypoints. Due to the exact instance-wise symmetry, we similarly can parametrize the basis as $\mathcal{B}_{\mathcal{C}\frac{1}{2}} \in \mathbb{R}^{3N/2\times K}$ to denote the shape basis for the first half of the keypoints. The reflection operator $\mathsf{A}_{\mathcal{C}}$ is parametrized by a unit normal vector $\mathsf{n}_{\mathcal{C}} \in \mathbb{R}^3$ of the plane of symmetry passing through the origin. The advantage of going from Eq. (4.2) to Eq. (4.3) should be apparent from the reduced dimensionality of the unknowns in $\mathcal{B}_{\mathcal{C}}$ as well as the additional second equality constraint of Eq. (4.3), which reduces the ambiguities in NRSfM [48].

**Symmetric deformation space.** In many non-rigid objects, shape instances are not symmetric. However, symmetry may still exist in the deformation space, e.g., in a human body. Suppose that a particular shape instance $\mathsf{S}_k \in \mathcal{C}$ has the reflective symmetry about $\mathsf{n}_{\mathcal{C}}$, which allows us to define its two halves: $\mathsf{S}_{k\frac{1}{2}}$ and $\mathsf{S}'_{k\frac{1}{2}}$ and thus correspondingly for all shape instances.

**Definition 1** (Symmetric deformation space)**.** $\mathcal{C}$ is a symmetric deformation space if for every half shape deformation instance $\mathsf{S}_{i\frac{1}{2}}$, there exists any shape instance $\mathsf{S}_j \in \mathcal{C}$ such that the $\mathsf{S}'_{j\frac{1}{2}}$ is symmetric to $\mathsf{S}_{i\frac{1}{2}}$.

The above definition also applies for the keypoints shape space $\mathcal{P}$. The instance-wise symmetric space is a particular case of the above. However, Eq. (4.3) cannot model the keypoints instances in the symmetric deformation space. We model such keypoints by introducing symmetric basis that can be weighted asymmetrically, thereby, obtaining the following:

$$\mathsf{P}_i = \mathsf{R}_i \begin{bmatrix}\mathrm{mat}(\mathcal{B}_{\mathcal{C}\frac{1}{2}}\,\mathsf{c}_i) & \mathrm{mat}(\mathcal{B}'_{\mathcal{C}\frac{1}{2}}\,\mathsf{c}'_i)\end{bmatrix} \tag{4.4}$$

where $\mathcal{B}'_{\mathcal{C}\frac{1}{2}}$ is obtained by reflecting $\mathcal{B}_{\mathcal{C}\frac{1}{2}}$ with $A_{\mathcal{C}}$ and $\mathsf{c}'_i \in \mathbb{R}^K$ forms the coefficients for the second half of the basis. Although Eq. (4.4) increases the dimension of the unknowns in the coefficients over Eq. (4.2), the added modeling of the symmetry of the deformation space and the reduced dimensionality of the basis can improve the final keypoints estimate. This brings us to the following proposition.

**Proposition 1.** *Provided that* $\mathcal{B}_{\mathcal{C}\frac{1}{2}}$ *and* $\mathcal{B}'_{\mathcal{C}\frac{1}{2}}$ *are symmetric about a plane, Eq.* (4.4) *approximates a symmetric deformation space if the estimates of* $\mathsf{c}_i$ *and* $\mathsf{c}'_i$ *come from the same probabilistic distribution.*

*Proof.* The two linear spaces due to the two basis $\mathcal{B}_{\mathcal{C}\frac{1}{2}}$ and $\mathcal{B}'_{\mathcal{C}\frac{1}{2}}$ are symmetric by Definition 1 as $\mathcal{B}_{\mathcal{C}\frac{1}{2}}$ is symmetric to $\mathcal{B}'_{\mathcal{C}\frac{1}{2}}$ for any $K \in \mathbb{Z}$. Let $\mathsf{c}_i \in \mathcal{L}$ and $\mathsf{c}'_j \in \mathcal{L}'$ represent the respective half coefficients for any two shape instances $i$ and $j$, where $\mathcal{L}$ and $\mathcal{L}'$ defines the spaces of the predicted half coefficient vectors. Consequently, the actual deformation spaces are symmetric to one another if $\mathcal{L}$ and $\mathcal{L}'$ are equal. We define $p : p(\mathsf{c}_i)$ as the probability distribution of $\mathsf{c}_i$ and $q : q(\mathsf{c}'_j)$ as the probability distribution of $\mathsf{c}'_j$. If $p$ and $q$ come from the same distribution, we approach $p = q$. Then we have:

$$\begin{aligned} &\text{if } \mathsf{c}_i = \mathsf{c}'_j, \\ &\text{either, } p(\mathsf{c}_i) = q(\mathsf{c}'_j) = 0, \\ &\text{or, } p(\mathsf{c}_i) > 0 \text{ and } q(\mathsf{c}'_j) > 0 \\ &\text{for all, } \mathsf{c}_i \in \mathcal{L}, \mathsf{c}'_j \in \mathcal{L}'. \end{aligned} \tag{4.5}$$

Condition (4.5) guarantees that $\mathcal{L} = \mathcal{L}'$ and thus we obtain a symmetric deformation space. □ □

Note that for condition (4.5) to be true, we do not require the two distributions to be equal, however, it is sufficient and desirable to have so. Therefore, Proposition 1 in the main text highlights such sufficient and desirable case. It is particularly meaningful when we are learning to predict the coefficients through stochastic methods such as a neural network training. In our network architecture indeed one can expect the distributions of these two vectors to be similar given the data exhibits such a symmetric deformation space, since the prediction branches of $\mathsf{c}_i$ and $\mathsf{c}'_i$ are very similar. Alternatively, one may also try to enforce the condition using a KL divergence loss.

As a consequence of Proposition 1, we can model keypoints in non-rigid symmetric objects with Eq. (4.4), while also tightly modeling the symmetry as long as we maintain the distribution of $\mathsf{c}$ and $\mathsf{c}'$ to be the same.

## 4.4 Learning Category-specific Keypoints

In this section, we use the modeling of $\Phi_{\mathcal{C}}$ to describe the unsupervised learning process of the category-specific keypoints. More precisely, we want to learn the function

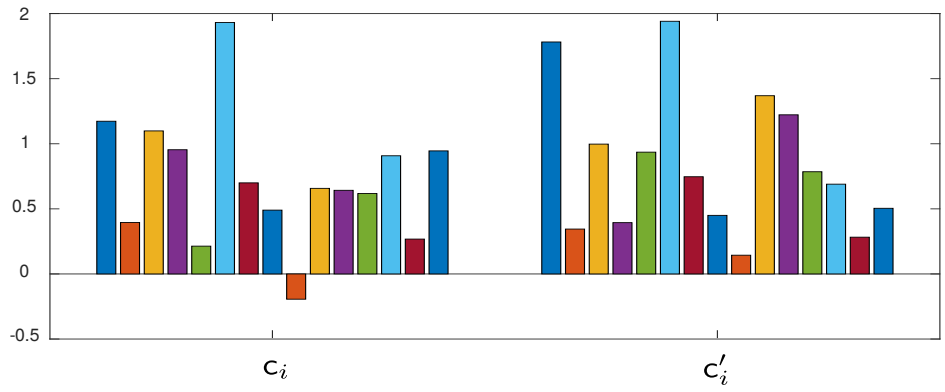



Fig. 4.1 **Coefficients distribution**. Mean values of $\mathsf{c}_i$ components (left) and $\mathsf{c}'_i$ components (right) for the Dynamic FAUST [15]. The mean of the variances for the different components are: $\mathsf{c}_i : 0.54$, $\mathsf{c}'_i : 0.50$. The figure shows that the network learns similar distribution for the coefficients $\mathsf{c}_i$ and $\mathsf{c}'_i$.

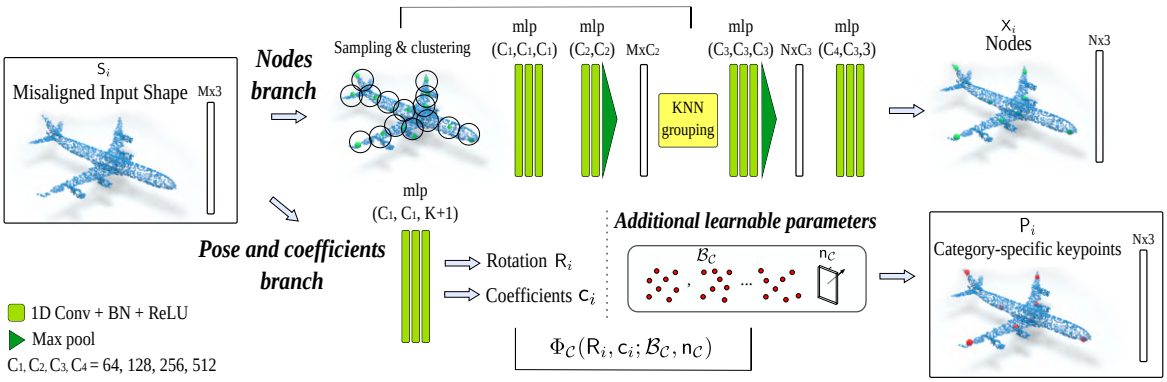



Fig. 4.2 **Network architecture:** The *pose and coefficients branch* and the *additional learnable parameters* generate the output category-specific keypoints. The *nodes branch* estimates the nodes that guide the learning process. "mlp" stands for multi-layer perceptron. Refer to Sec. 4.3 for the modeling, Sec. 4.4 for learning.

$\Pi_\mathcal{C} : \mathsf{S}_i \rightarrow \mathsf{P}_i$ as a neural network of parameters $\theta$, using the supervisory signal from $\Phi_\mathcal{C}$. In regard to learning keypoints on point sets, recent work [82] trains a Siamese network to predict order-agnostic keypoints stable to rotations for rigid objects [82]. Part of our network architecture is inspired from [82], which is based on PointNet [106]. However, we use a single input avoiding the expensive Siamese training. The network architecture is shown in Fig. 4.2, whose input consists of a single shape $\mathsf{S}_i$ misaligned in $SO_2$. This is reasonable since point clouds are usually aligned to the vertical direction. We describe the different components of the network architecture below.

**Node branch.** This branch estimates a sparse tuple of nodes that are potentially category-specific keypoints but are not ordered. We denote them as $\mathsf{X}_i = \{\mathsf{x}_{i1}, \mathsf{x}_{i2}, \ldots, \mathsf{x}_{iN}\}, \mathsf{x}_{ij} \in \mathbb{R}^3$ and $j \in \{1, 2, \ldots, N\}$. Initially, a predefined number of nodes $N$ are sampled from the input shape using the Farthest Point Sampling (FPS) and a local neighborhood of points is built for each node with point-to-node grouping [81, 82], creating $N$ clusters which are mean normalized inside the network. Every point in $\mathsf{S}_i$ is associated with one of these nodes. The branch consists of two PointNet-like [106] networks followed by a kNN grouping layer that uses the initial sampled nodes to achieve hierarchical information aggregation. Finally, the local feature vectors are fed into a Multi-Layer Perceptron (MLP) that outputs the nodes.

**Pose and coefficients branch.** We predict the quantities $\mathsf{R}_i$ and $\mathsf{c}_i$ with this branch. We use a single rotation angle to parametrize $\mathsf{R}_i$. The branch consists of an MLP that estimates the mentioned parameters. The output size varies depending on whether we are interested in symmetric shape instances as in Eq. (4.3) or symmetric basis as in Eq. (4.4), the size being double in the latter.

**Additional learnable parameters.** Several unknown quantities in Eq. (4.3) or (4.4) are constant for a category shape space $\mathcal{C}$. Such quantities need not be predicted instance-wise. We rather choose to optimize them as part of the network parameters $\theta$. They are the shape basis $\mathcal{B}_\mathcal{C} \in \mathbb{R}^{3N \times K}$ and the unit normal of the plane of symmetry $\mathsf{n}_\mathcal{C} \in \mathbb{R}^3$. We observed that a good choice for the number of shape basis is $5 \leq K \leq 10$. In fact, the generated keypoints are not very sensitive to the choice of $K$, as a large $K$ tends to generate sparser shape coefficients and similar keypoints. Depending upon the problem, alternate parametrization can be considered for $\mathsf{n}_\mathcal{C}$, e.g., Euler angles.

At inference time, we apply Non-Maximal Suppression obtaining the final $N'$ number of keypoints. Our method consistently provides $N'$ keypoints for all instances in the category, as they follow the same geometric model.

### 4.4.1 Training Losses

In order to adhere to the definitions of the category-specific keypoints introduced in Sec. 4.1 as well as our shape modeling, we design our loss functions as below.

**Chamfer loss with symmetry and non-Rigidity.** Eq. (4.1) suggests that the neural network $\Pi_\mathcal{C}$ can be trained with an $\ell_2$ loss between the node predictions $\mathsf{X}_i$ and

the deformation function $\mathsf{P}_i = \Phi_{\mathcal{C}}(\mathsf{R}_i, \mathsf{c}_i; \mathcal{B}_{\mathcal{C}}, \mathsf{n}_{\mathcal{C}})$, thus obtaining $\mathsf{P}_i = \mathsf{X}_i$. However, as confirmed by our evaluations as well as in [82], the $\ell_2$ loss does not converge as the network is unable to predict the point order. Alternatively, the Chamfer loss [37] does converge, minimizing the distance between each point $\mathsf{x}_{ik}$ in the first set $\mathsf{X}_i$ and its nearest neighbor $\mathsf{p}_{ij}$ in the second set $\mathsf{P}_i$ and vice versa.

$$\mathcal{L}_{chf} = \sum_{k=1}^{N} \min_{\mathsf{p}_{ij} \in \mathsf{P}_i} \|\mathsf{x}_{ik} - \mathsf{p}_{ij}\|_2^2 + \sum_{j=1}^{N} \min_{\mathsf{x}_{ik} \in \mathsf{X}_i} \|\mathsf{x}_{ik} - \mathsf{p}_{ij}\|_2^2, \tag{4.6}$$

The Chamfer loss in Eq. (4.6) ensures that the learned keypoints follow a generalizable category-specific property – that they are a linear combination of common basis learned specifically for the category. To additionally model symmetry, Eq. (4.3) or (4.4) is directly used in Eq. (4.6). Therefore, two different Chamfer losses are possible modeling two different types of symmetries.

**Coverage and inclusivity loss.** The Chamfer loss in Eq. (4.6) does not ensure that the keypoints follow the object shape. However, one can add the following conditions: a) the keypoints cover the whole category shape (coverage loss), b) the keypoints are not far from the point cloud (inclusivity loss). The coverage loss can be defined as a Huber loss between the volume of the nodes $\mathsf{X}_i$ and that of the input shape $\mathsf{S}_i$, using the product of the singular values. However, we instead approximate the volume using the 3D bounding box defined by the points. This improves the training speed and, based on our initial evaluations, also does not harm performance. The coverage loss is thus given by:

$$\mathcal{L}_{cov} = \|\text{vol}(\mathsf{X}_i) - \text{vol}(\mathsf{S}_i)\| \tag{4.7}$$

The inclusivity loss is formulated as a single side Chamfer loss [13] which penalizes nodes in $\mathsf{X}_i$ that are far from the original shape $\mathsf{S}_i$, similarly to Eq. (4.6):

$$\mathcal{L}_{inc} = \sum_{k=1}^{N} \min_{\mathsf{s}_{ij} \in \mathsf{S}_i} \|\mathsf{x}_{ik} - \mathsf{s}_{ij}\|_2^2. \tag{4.8}$$

## 4.5 Experimental Results

We conduct experiments to evaluate the desired properties of the proposed category-specific keypoints and show their generalization over indoor/outdoor objects and

rigid/non-rigid objects with four different datasets in total (Sections 4.5.1 and 4.5.2). All these properties are also compared with a proposed baseline. We then evaluate the practical use of our keypoints for intra-category shapes registration (Section 4.5.3), analyzing the influence of symmetry, and for segmentation label transfer (Section 4.5.4). Furthermore, an experiment showing the generalization of our method on real data is included in Section 4.5.5. Additional qualitative results are shown in Section 4.5.6.

**Datasets.** We use four main datasets. They are ModelNet10 [156], ShapeNet parts [165], Dynamic FAUST [15] and Basel Face Model 2017 [49]. Since our method is category-specific, we require separate training data for each class in the datasets. For indoor rigid objects, we choose three categories from ModelNet10 [156]; chair, table and bed. Three outdoor rigid object categories: airplane, car and motorbike, are evaluated from ShapeNet parts [165]. For non-rigid objects, we randomly choose a sequence of the Dynamic Faust [15], that provides high-resolution 4D scans of human subjects in motion. Finally, we generate shape models of faces using the Basel Face Model 2017 [49] combining 50 different shapes and 20 different expressions. All models are normalized in the range $-1$ to 1 and are randomly misaligned within $\pm 45$ degrees.

**Baseline.** Since this is the first work computing category-specific keypoints from point sets, we construct our own baseline based on the recent work USIP [82]. The method detects stable interest points in 3D point clouds under arbitrary transformations and is also unsupervised, which makes it the closest method for comparison. The USIP detector is not category-based, so we train the network per category to create the baseline. Additionally, we adapt the number of predicted keypoints so that the results are directly comparable to ours. While training with some of the categories, specifically car and bed, we observe that predicting lower number of keypoints can lead to some degeneracies [82].

**Implementation details.** Input point clouds of dimension $3 \times 2000$ are used. We implement the network in Pytorch [102] and train it end-to-end from scratch using the Adam optimizer [72]. The initial learning rate is $10^{-3}$, which is exponentially decayed by a rate of 0.5 every 40 epochs. We use a batch size of 32 and train each model until convergence, for 200 epochs. The final loss function combines the three training losses, Eqs. (4.6), (4.7) and (4.8), and are weighted as follows: $w_{chf} = w_{cov} = 1$ and $w_{inc} = 2$. For ModelNet10 and ShapeNet parts, we use the training and testing split provided by the authors. For the Basel Face Model 2017, we follow the common practice and

split the 1000 generated faces in 85% training and 15% test. We use the same split strategy for the sequence '50009_jiggle_on_toes' of Dynamic Fuaust, which contains 244 examples.

| *Category* | *Coverage* | *Model Err* | *Correspondence* | *Inclusivity* | *Sym Err* | *Definition* |
|---|---|---|---|---|---|---|
| | % | % | % | % | ° | |
| chair | **88.83** | 0.72 | **100** | 90.46 | 0.40 | 10 |
| table | **93.33** | 0.99 | **100** | 93.38 | 2.86 | 6 |
| bed | **80.31** | 0.94 | **100** | **95.33** | 0.13 | 6 |
| airplane | **89.15** | 0.64 | **100** | **96.35** | 0.20 | 8 |
| car | **92.39** | 0.72 | **100** | **97.77** | 2.21 | 8 |
| motorbike | **96.13** | 0.79 | **100** | **90.53** | 1.42 | 8 |
| human body | **85.59** | 0.72 | **100** | 97.73 | 33.30 | 11 |
| faces | **97.93** | 0.41 | **100** | **100** | 0.15 | 9 |
| chair | 79.73 | – | 55.6 | **98.50** | – | 10 |
| table | 79.72 | – | 34.5 | **99.83** | – | 6 |
| bed | 42.18 | – | 49.33 | 70.00 | – | 6 |
| airplane | 69.24 | – | 47.5 | 87.13 | – | 8 |
| car | 26.87 | – | 32.18 | 74.0 | – | 8 |
| motorbike | 75.29 | – | 48.14 | 84.57 | – | 8 |
| human body | 72.66 | – | 50.45 | **100** | – | 11 |
| faces | 42.98 | – | 30.11 | **100** | – | 9 |

Table 4.1 *Properties Analysis:* Top (ours) and bottom (baseline [82]). For coverage, correspondence and inclusivity *higher is better*, and for model and symmetry error *lower is better*. We empirically show the desired properties of our keypoints, as well as the generalization of our method over indoor/outdoor and rigid/non-rigid objects. Best results are in bold.

### 4.5.1 Desired Properties Analysis

As described in Sec. 4.1 and 4.3, the category-specific keypoints satisfy certain desired properties. We propose six different metrics to evaluate the properties which are also used for comparison against the baseline. All the results are presented in Table 4.1, and are averaged across the test samples.

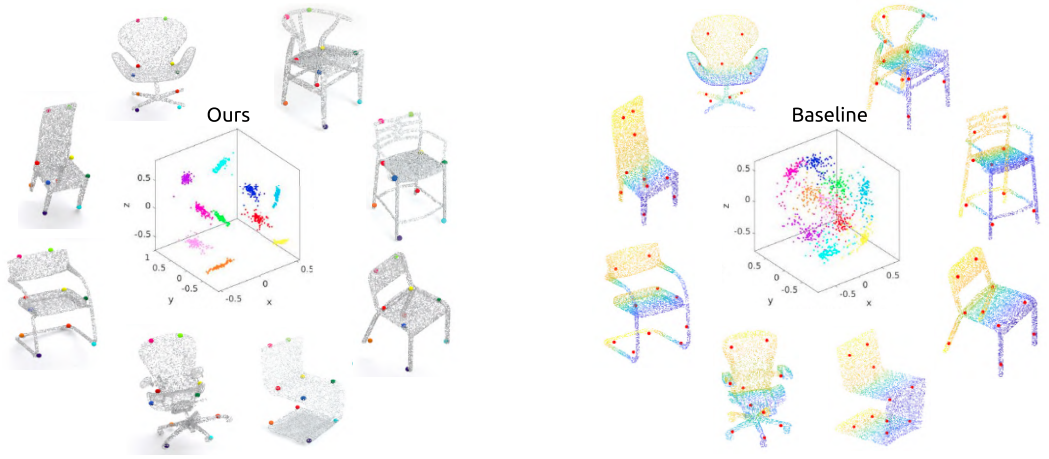



Fig. 4.3 **Keypoints correspondence/repeatability across instances**. We cluster the predicted keypoints for all the instances in the category to show their geometric consistency. Note how our keypoints are neatly clustered as they are consistently predicted in the corresponding geometric locations, unlike the baseline keypoints. (Note: cluster colors do not correspond to keypoint colors.)

*Coverage*: According to property *iii)*, we seek keypoints that are representative of each instance shape as well as of the category itself. To measure it, we calculate the percentage of the input shape covered by the keypoints' 3D bounding box. On average, we achieve a 29.4% more coverage than the baseline.

*Model Error*: This metric refers to the Chamfer distance between the estimated nodes and the learned category-specific keypoints, normalized by the model's scale. We obtain less than 1% of error in all the categories, meaning that the network satisfactorily manages to generalize, describing the nodes with the symmteric non-rigidity modeling (Properties *i)* and *iii)*).

*Correspondence/ Repeatability*: We measure the ability of the model to find the same set of keypoints on different instances of a given category (Property *ii)*). For our method, we cluster the keypoints using their inherent order whereas for the baseline, we use K-means clustering to evaluate and compare this property. We show a detailed evaluation of the chair category in Fig. 4.3, the rest of the categories are provided in Fig. 4.4. One can see at a glance how our keypoints are well clustered, unlike the baseline keypoints. Numerically, we show the % occurrence of each specific keypoint belonging to the same cluster across instances. Our keypoints satisfy 100% the correspondence/repeatability test thanks to our geometric non-rigidity modelling.

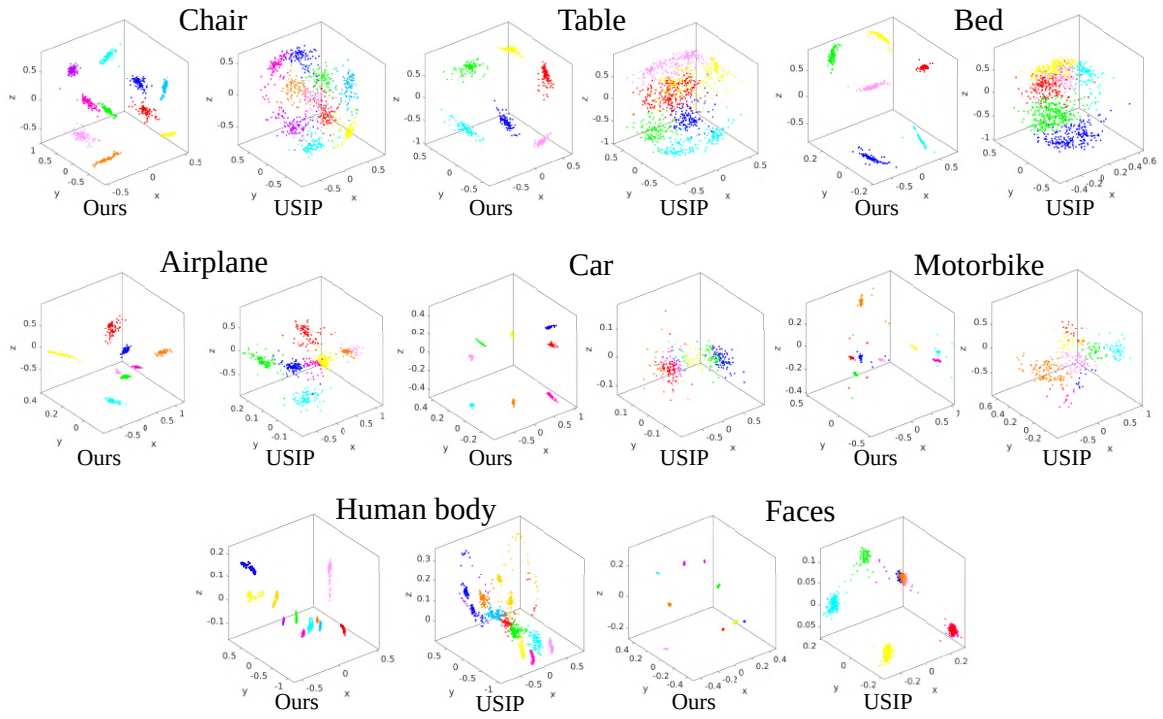



Fig. 4.4 **Keypoints correspondence across instances**. We cluster the keypoints predicted for all the instances of a category to show their geometric consistency. Note how our keypoints get neatly clustered creating a general 3D shape template.

*Inclusivity*: We measure the percentage of keypoints that lie inside the point cloud (of scale 2) within a chosen threshold of 0.015, which also proves property *iii)*. This is the only metric in which our method doesn't outperform the baseline in all cases. On average, our method achieves $\sim 95\%$ inclusivity compared to $\sim 89\%$ for the baseline.

*Symmetry:* The metric shows the angle error of the predicted reflective plane of symmetry. We obtain highly accurate prediction for rigid categories. In the non-rigid human body shape however, the ambiguities are severe. Despite that, the learned keypoints satisfy the other properties, particularly that of semantic correspondence.

*Definition:* final number of keypoints $N'$ predicted per category after the Non-Maximal Suppression.

### 4.5.2 Semantic Consistency

We use the ShapeNet part dataset [165] to show the semantic consistency of the proposed keypoints. Following the low-rank non-rigidity modelling, the keypoints lie

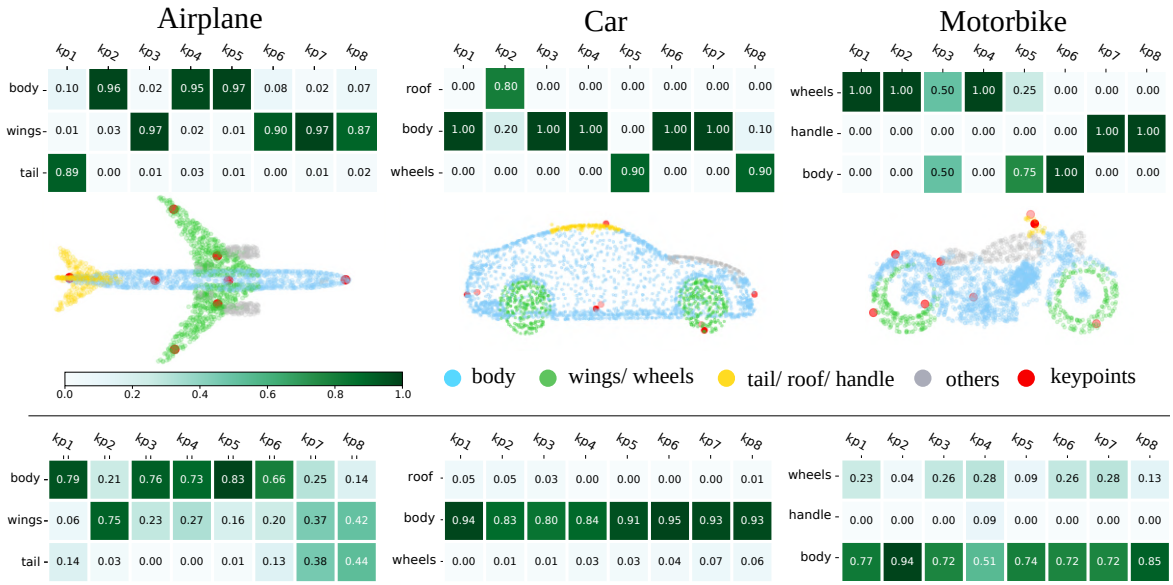



Fig. 4.5 **Semantic part correspondence.** Top to bottom: the semantic correspondence for the proposed keypoints, qualitative results and the baseline semantic correspondence. Our predicted keypoints show the correct semantic correspondence across the category.

on geometrically corresponding locations. The idea of the experiment is to measure keypoint-semantics relationship for every keypoint across instances of the category. The results are presented in Fig. 4.5 as covariance matrices, along with keypoint visualizations per category for our method. On average, the proposed keypoints have a high semantic consistency of 93% across instances, despite the large intra-category variability. The same experiment is performed for the baseline and presented in bottom of Fig. 4.5. Here, the degeneracy causes all the keypoints to approach the object centroid for 'Car'. Nonetheless, we observe no semantic consistency even for 'Airplane' without degeneracies. Our model, aiming for a common representation for all the instances of the category, avoids placing keypoints in less representative parts or unique parts, e.g., arm rests in chairs (in Fig. 4.9), engines in airplanes or gas tank in motorbikes. This highlights significant robustness achieved in modelling and learning the keypoints.

### 4.5.3 Objects Pose and Intra-category Registration

Previous methods do not handle misaligned data due to the obvious difficulty it poses to unsupervised learning. This deserves special attention since real data is never aligned. In this section we evaluate the intra-category registration performance of our

model and show the impact of the different symmetry models proposed. These results implicitly measure the object poses estimated as well.

**Rotation Ambiguities.** Recent unsupervised approaches for keypoint detection actually self-supervise rotation during training, e.g., [138, 82], and highlight that it is crucial for achieving a good performance. In our case, we do not directly supervise the rotations. Therefore, the different combination of basis shapes can result in different alignments. This implies that computing $\mathsf{P}_i$ with the deformation function $\Phi_{\mathcal{C}}$ will give the correct set of keypoints along with the correct plane of symmetry, but the predicted rotation alone is not meaningful for registration. As we show in Fig. 4.6, predicting the symmetry plane of the object category allows to have more control over the predicted instance poses. We came up with the idea of learning an additional common parameter, $\mathsf{R}_{\mathcal{C}}$, which is directly related to the symmetry plane. By adding this category-specific parameter, the network learns a common rotation for all the objects in the category. As a consequence, the instance-wise rotation, $\mathsf{R}_i$, can be thought like an offset from the reference basis alignment. Several evaluations confirmed that this strategy helps the learning process, reducing the rotation ambiguities.

**Experimental setup.** Despite the above ambiguity, an important characteristic of the proposed keypoints is that they are ordered, which empowers direct inter-instances registration since no extra descriptors are needed for matching. We perform experiments for the chair category, using 10 keypoints (Table 4.1) and a misalignment of $\pm 45$ degrees. Three different models are compared. The first one is trained without symmetry awareness following Eq. (4.2). A second one uses shape symmetry during training as shown in Eq. (4.3). The last model is trained with basis symmetry as in Eq. (4.4). We attempt to register keypoints in each instance to those of randomly chosen three aligned templates by computing a similarity transformation and observe the mean error. Fig. 4.6 shows that symmetry helps to have more control over the rotations and tackle higher misalignment.

### 4.5.4 Segmentation Label Transfer

Our predicted keypoints correspond to semantically meaningful locations. Therefore, here we explore the utility of the proposed category-specific keypoints for the segmentation label transfer task. In this experiment, for every point in the original shape $\mathsf{s}_{ij} \in \mathsf{S}_i$, we find its closest category-specific keypoint $\mathsf{p}_{ik} \in \mathsf{P}_i$, and transfer the corresponding

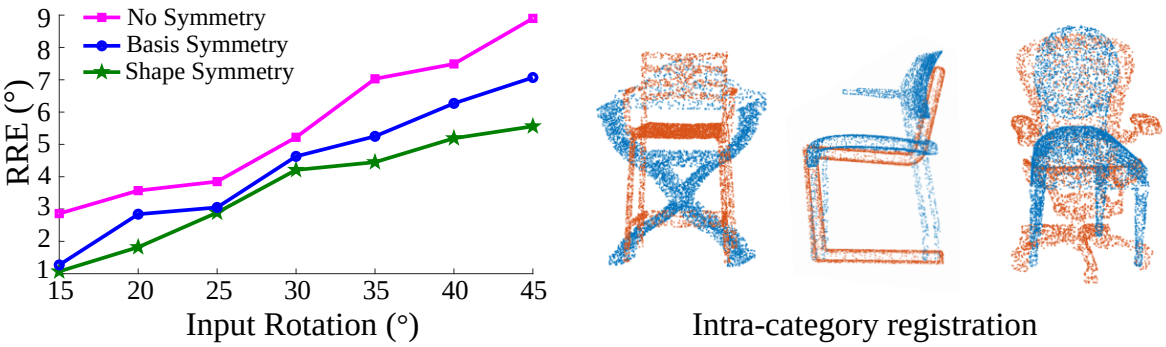



Fig. 4.6 Left: Relative rotation error for different symmetry modelings. Right: 3 examples of registration between different instances of the same category.

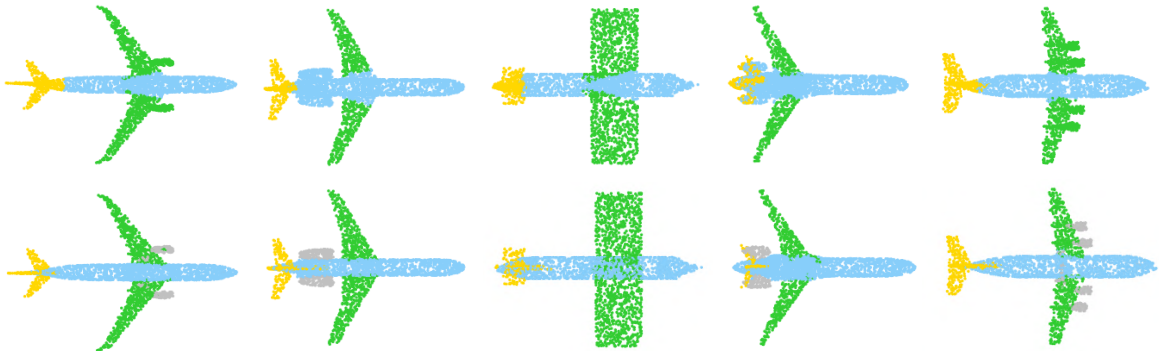


Fig. 4.7 **First row:** results of performing semantic label transfer with our keypoints. **Second row:** ground truth. This is evaluated in ShapeNet part dataset [165] using eight keypoints for the label transfer.

semantic label to it. We assume the keypoints labels are known and correspond to those in Figure 4.5.

Some qualitative results are shown in Fig. 4.7. Our method achieves full correspondence between instances, therefore avoiding placing keypoints in less representative parts. An example is the engine, in grey, in the case of airplanes. This is reflected in the label transfer since there is no distinction of these parts. Besides that, only with eight keypoints in the example, we achieve reasonable results, close to the ground truth data.

### 4.5.5 Real Data

In this section, we show the performance of our method for real data in Fig. 4.8. For this experiment, the network is trained on the chair category from the ModelNet10 dataset [156] and tested on real chairs from the SUNRGBD dataset [130]. To generate

Fig. 4.8 Results in real chairs from SUNRGBD dataset [130] training with CAD chairs from ModelNet10 dataset [156].

the real data dataset from [130], we crop the points inside the ground truth 3D bounding boxes provided by the authors. Real data entail additional challenges. This is not only because shapes appear incomplete and noisy, but also because other objects may cause occlusions, e.g. part of a table occluding a chair. As illustrated in Fig. 4.8, even though real data is fairly challenging, our network can still produce corresponding meaningful keypoints.

Being able to generalize to previously unseen real objects as demonstrated in Fig. 4.8 is crucial and really useful for many tasks such as guide for shape completion or shape generation.

### 4.5.6 Qualitative results

In this section, we provide additional qualitative results on various object categories from the datasets evaluated; ModelNet10 [156] in Fig. 4.9, ShapeNet parts [165] in Fig. 4.10, Dynamic FAUST [15] in Fig. 4.11 and Basel Face Model 2017 [49] in Fig. 4.12.

Again, we note that our network predicts corresponding keypoints between instances of the same category and consistently associates the same keypoint with the same semantic part. For instance, for the chair object category, the keypoint colored in pink

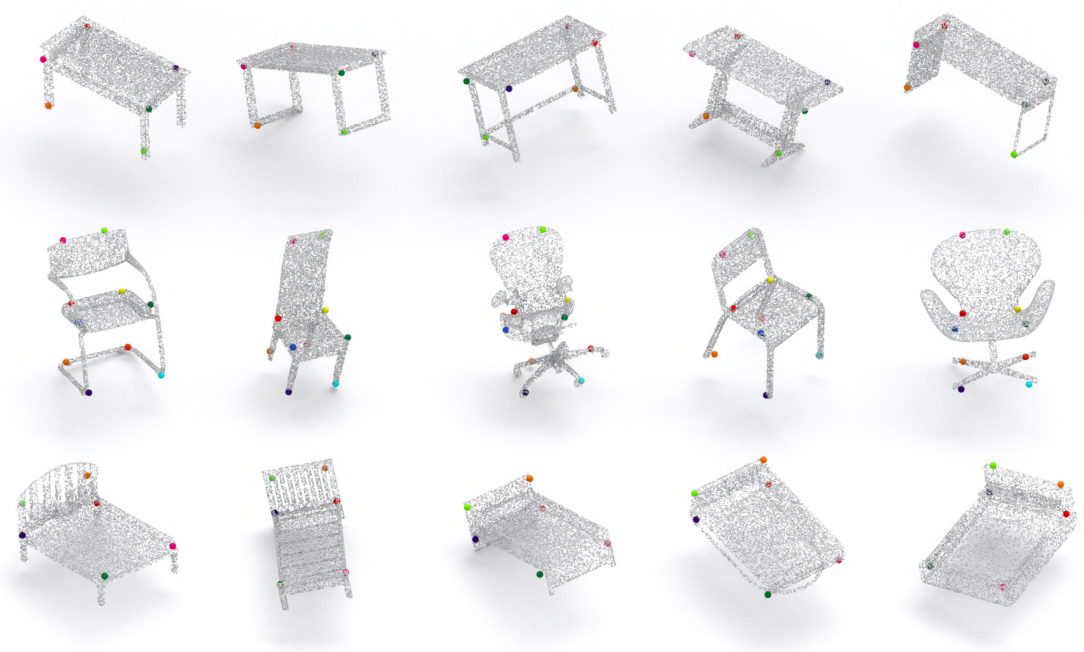


Fig. 4.9 Qualitative results in table, chair and bed categories from ModelNet10 dataset [156].

is always associated with the chair back, the keypoint colored in cyan is associated with the front left leg, etc.

## 4.6 Conclusions

This work investigates automatic discovery of kepoints in 3D misaligned point clouds that are consistent over inter-subject shape variations and intra-subject deformations in a category. We find that this can be solved, with unsupervised learning, by modeling keypoints with non-rigidity, based on symmetric linear basis shapes. Additionally, the proposed category-specific keypoints have one-to-one ordered correspondences and semantic consistency. Applications for the learned keypoints include registration, recognition, generation, shape completion and many more. Our experiments showed that high quality keypoints can be obtained using the proposed methods and that the method can be extended to complex non-rigid deformations. Future work could focus on better modeling complex deformations with non-linear approaches.

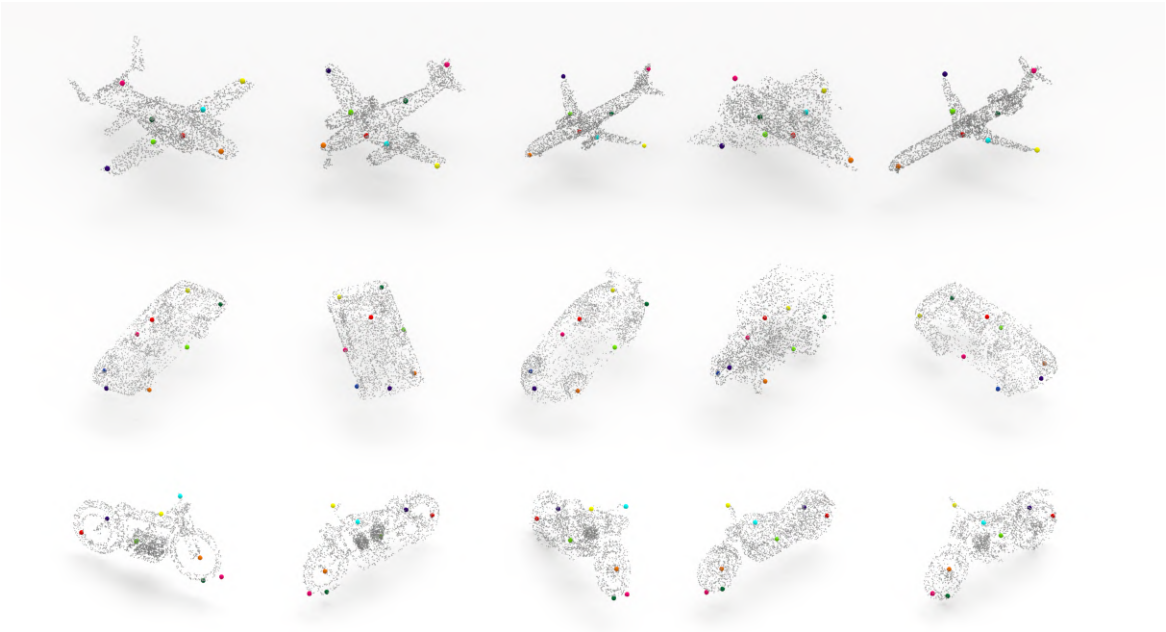


Fig. 4.10 Qualitative results in airplane, car and motorbike categories from ShapeNet parts dataset [165].

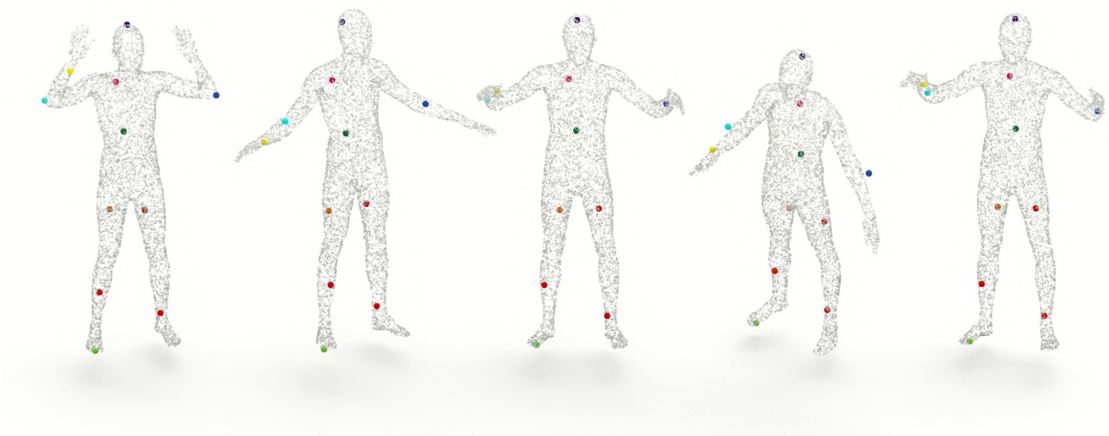


Fig. 4.11 Qualitative results in human bodies from Dynamic FAUST dataset [15].

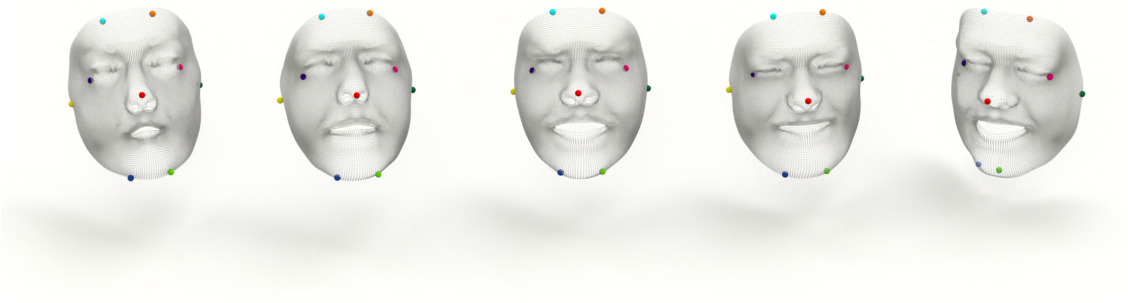


Fig. 4.12 Qualitative results in faces from Basel Face Model 2017 dataset [49].

# Chapter 5

# Discussion and Conclusions

*"We can only see a short distance ahead, but we can see plenty there that needs to be done."*

— Alan Turing

In this thesis we presented novel methods for indoor scene understanding using non-conventional cameras. We developed several methods to recover the 3D room layout from single 360 images, combining geometric constraints and learning techniques. Furthermore, we leveraged 360 images to perform object localization and segmentation and showed the advantages of using the layout prior for 2D-3D lifting of the objects in the room. Finally, we presented an unsupervised learning method to predict category-specific keypoints in a collection of 3D objects of some category, so that they are in geometric and semantic correspondence. As it typically happens in research, every novel idea or discovery not only raises answers, but also more questions. In the following, we will review the principal findings of each chapter and discuss the limitations of the presented work. We end with possible directions for further work.

## 5.1 Room Layout Estimation

Chapter 2 presents our contributions on the problem of 3D room layout estimation, as well as on the leverage of 360 images. We observed that the camera field of view is a limiting factor to predict closed layouts that satisfy the original geometry of the room. For this reason, we propose to use 360 cameras, that provide a full view of the room at once.

**Layouts with Geometry and Deep Learning.** In Section 2.4, we propose a new model that generalizes to cuboid and non-cuboid room configurations, while reducing the computation time with respect to some typical pre and post-processing operations. The main piece of information we use to estimate the room layout are lines and corners that represent the intersections between walls, ceiling and floor. First, we propose a RANSAC-based algorithm to detect the lines and vanishing points of the scene. The method works directly on the panoramic image, improving the overall efficiency compared to previous approaches. In cluttered scenes however, it is not trivial to know whether the lines come from actual wall intersections or from other elements of the scene. We show how one can filter most of the lines belonging to clutter using a deep learning model that returns probabilities, for each image pixel, to be part of an structural edge. Working directly with potential structural primitives, leads to a meaningful reduction of the number of hypotheses needed to recover the room layout and consequently, to a reduction in the post-processing computation time. An important contribution for generalizing to non-cuboid layouts is presented in the hypotheses generation process. We observed that traditional and deep learning algorithms struggle to detect structural corners that are non-visible in the image, which actually happens many often since floor corners are usually occluded by objects and, depending on the camera viewpoint, a wall may occlude an entire part of the room. We propose a post-processing algorithm based on a geometric reasoning about the 3D corners of the room, allowing the inclusion of new corners in order to get Manhattan layouts. This idea proved to be very powerful since it allows to localize a priori, invisible corners. A limitation of this method resides in how we leverage the aforementioned deep learning model. Conventional methods are not suitable for equirectangular images. Consequently, we need to split the equirectangular image into several perspective images, run the deep learning framework for each of them, and then combine the local results. This procedure not only entails a bottleneck in our proposal, but also limits the accuracy of the probability maps, as we cannot truly benefit from the whole context provided by the panoramic images.

**CFL: Corners for Layout.** We present CFL in Section 2.5 with several contributions. We note that using traditional methods with 360 images represents a major bottleneck in most recent methods, and the estimated results are not as good as we would desire. This is not surprising as panoramic images contain scene parts, such as the ceiling or visually less conspicuous regions, that generally do not appear in conventional images. With this motivation, we demonstrate how to create CNNs that work directly with 360 images, exploiting all their context and reducing processing time. Additionally, we

make the observation that the use of standard convolutions with equirectangular images can lead to a loss of performance for several reasons. First, pre-training on conventional images is critical due to the lack of training 360 data. However, the pre-trained feature space is non-distorted, which makes the pre-training less effective on the distorted space of the equirectangular images. Moreover, in omnidirectional images, the right and the left side of the images are the same spot in reality. Therefore, if we apply standard convolutions on these images, the network simply does not understand the continuity of the scene on the image borders. In order to overcome the problem, we propose an special type of convolution, named EquiConv, that adapts the size and shape of the kernel to the equirectangular image distortions. EquiConvs can directly substitute the standard convolutions and present several advantages to work with 360 images. First, aware of the distortion model, EquiConvs allow training models with 360 images using pre-training on conventional images, which is crucial to mitigate the lack of training 360 data. Additionally, EquiConvs demonstrate a better generalization to camera pose variations, since the model does not rely on characteristic patterns generated by the image distortions. This means that EquiConvs make the model more robust and generalizable to real life situations, not covered in the datasets, e.g. panoramas taken by hand, drones or small robots. Moreover, EquiConvs, convolving the image in the spherical domain, achieve a continuous understanding of the scene. This is essential to correctly predict structural information that lie on the image borders, avoiding duplicates, as would happen with standard convolutions. As a consequence, our model obtains more accurate and robust predictions without the need of additional expensive post-processing, becoming up to 100 times faster than previous methods. A limitation of CFL is that we directly join corners from left to right on the image. This means that our model could not be able to infer the correct order of the corners if any wall is occluded because of the convexity of the scene, even if it is detected. By choosing a camera viewpoint from which all the walls are visible, this problem could be avoided. However, for more complicated rooms, the problem is still unsolved. One possible solution would be to use the post-processing proposed in Section 2.4. This post-processing will lead to an increase of the computing time, but also will help fill the non-visible corners. A second option to avoid increasing the computing time, could be to predict directly the order of the room corners inside the network. However, the method could still fail if any corner is not predicted.

We have come a long way in the task of layout recovery, but we are still far from achieving human robustness and generalization on this problem. The next step should

focus on solving increasingly complex room geometries and aim for real-time algorithms. There are definitely many exciting ideas to be tried like; exploiting symmetry to predict more meaningful room corners, relaxing the Manhattan World assumption or adding more geometric constraints inside the deep learning algorithms. All these ideas would not only help to achieve more accurate results, but also to head towards unsupervised learning, providing more scalable frameworks.

## 5.2 Object Recognition

In Chapter 3 we present, up to our knowledge, the first object detection system working directly on 360 images, focused on indoor scene understanding. We study how to adapt existing CNNs, in this case designed for the task of object detection, to match the nature of the equirectangular image input. We adapt the anchor box proposals and substitute standard convolutions by EquiConvs. We already demonstrated in Section 2.5, how EquiConvs can help to generalize to different camera pose variations. For the task of object recognition, we observed that the use of EquiConvs is convenient even if the camera is always at the same place, since objects can be at many different locations inside the scene, e.g., objects closer to the camera will have greater distortions than objects around the horizon line. Additionally, since the distortion depends only on the polar angle, objects that appear orthogonal to the viewpoint will be symmetrically distorted, whereas objects at different poses will not. Therefore, EquiConvs here play an important role since they can learn object appearance by ignoring spherical distortion patterns. Additionally, we show the potential of exploiting the 2D room layout to improve the instance segmentation masks, and how to leverage the 3D layout to generate 3D object bounding boxes, directly from the improved segmentation masks. A limitation is that there is no learning on the 3D structure for object detection. We strongly believe that including the layout prior and the 2D-3D lifting inside the network, would improve our results.

## 5.3 Object Category Shape Modelling

In Chapter 4 we propose to learn 3D keypoints from a collection of objects of some category, so that they meaningfully represent objects' shape and their correspondences can be simply established order-wise across all objects. Our motivation is that keypoints-based methods are crucial to the success of many vision applications like 3D reconstruction, registration, human body pose, recognition, or generation. The

challenges we consider are the following: input shapes are misaligned 3D point clouds, 3D objects go through shape variations and a given semantic part may not be present in all objects in a category. We demonstrate that this problem can be solved, in an unsupervised manner, by modeling keypoints with non-rigidity, based on symmetric linear basis shapes. We do not assume the plane of symmetry to be known and consider two different priors: instance-wise symmetry (rigid objects) and symmetric deformation space (non-rigid objects). We show that the keypoints discovered by our method have one-to-one ordered correspondences and are geometrically and semantically consistent. A limitation of this work is related to the model performance on organic shapes like the human body but also probably on animals, organs and non-rigid shapes in general. Correspondences between pairs of such deformable shapes has been tackled using shape similarities, functional maps [99], etc. These methods usually rely on ground truth correspondences, pre-computed descriptors, shape templates or aligned data. However, tackling correspondences between a collections of non-rigid objects is considerably more challenging. Previous methods have explored techniques such us path invariance or cycle consistency. The challenges add further when no labels are available. We believe that it would be interesting to combine low rank constraints with functional maps and cycle consistency techniques. Another option that seems interesting, would be to explore multilinear (bilinear) models to analyze shape and pose variations independently [56].

## 5.4 Future Work

Improvements to the individual proposed works have been discussed above. To conclude this thesis, we would like to highlight some aspects for future research, which are particularly exciting for us, with the goal of developing intelligent systems that match the performance of human vision.

In this thesis we advance state of the art in several topics related to indoor scene understanding but of course, there is much more out there. Aiming for a complete scene understanding, complementary tasks as well as ways of connecting several tasks together are desired. In this regard, once we get to automatically understand the geometry of an individual room, we might be interested in estimating how the room is connected to other rooms, or how is the building distributed. Similarly, once we identify the objects inside a room and know their location inside the room, we care about the relationships between the entities or about the actions we can perform with them, e.g. chairs are usually around a table, we can sit on a chair or lay on a bed. In

the same way, once we are aware of the shape model and geometry of a particular object, it is interesting to know its color, material and physical properties in general. A recent work [5] demonstrates how to host these diverse types of semantics in a unified structure. They generate a 3D scene graph using a 3D mesh and registered panoramic images of the building, and combine existing detection methods in order to collect all the information. The proposed method is still semi-automatic and leaves room to many exciting improvements and novel ideas. A similar recent work [115] models the scene dynamics as well, e.g. traversability between places or rooms: "agent A is in room B at time t". Still, if we want to mimic the human visual perception, we should aspire to estimate such scene graphs incrementally and in real-time.

There are many obvious applications of getting such unified understanding which have been already discussed in this thesis, such as indoor navigation or virtual or augmented reality. One less obvious but very exciting application is to transfer the indoor space information to the Building Information Modelling (BIM) methodology, to model the existing building stock, either for facility management purposes, heritage conservation, building research projects or structural stability analyses. This technology is important as it increases the interoperability between multiple heterogeneous disciplines such as architecture, construction, plumbing, lighting/electrical, mechanical or engineering. However, this line of work still needs a lot of effort, as we need to consistently model the outdoor and indoor parts of the building, get as much details as possible from the structural components, and find a new data representation that facilitates the image (or point cloud) to BIM model conversion. Computer Vision applications in general, related to indoor scene understanding, are already very present in our society, and although some sectors remain skeptical, many have already embraced this technology. We have just experienced an unprecedented event in the last 100 years, the coronavirus. This disease, known as COVID-19, has hit the entire world population, making us wonder how we can change the future, and more specifically, how we can create a society that suffers less exposure to this type of diseases. From the Computer Vision side, it is more urgent than ever before that we specially contribute creating intelligent systems that are able to assist and interact with humans, making the discipline more powerful and ubiquitous. This will not only have a direct impact to help on these critical situations, where face-to-face interactions should be reduced, but also in the daily life of humans, seeking an improvement in their quality of life.

# Appendix A

# Layout hypotheses generation

Here we give a more detailed explanation of the layout hypotheses generation from Section 2.4.2 and summarize it in algorithm 1. The algorithm combines a series of geometric rules that a Manhattan layout should satisfy, while being flexible enough to draw new corners that might be occluded in the image or not properly retrieved during previous steps.

The algorithm receives as input the room layout candidate corners (See 2.4.2 - candidate corners extraction). We define the candidate corners as an unordered set of corners, $C = \{c_1, c_2, \ldots, c_M\},\ c_j \in \mathbb{R}^3,\ j \in \{1, 2, \ldots, M\}$. Each corner represents the 3D ray that goes from the center of the spherical image through its position on the surface of the sphere. We also input the vanishing points, i.e. Manhattan directions of the scene, $vp = (vp_x, vp_y, vp_z) \in \mathbb{R}^{3\times3}$, the assumed distance from the camera to the ceiling plane $d^c$, and a maximum number of iterations, $maxIter$.

The proposed algorithm starts by randomly sampling a group of corners among the candidate ones, represented with the tuple $\mathcal{G}_c = (c_1, c_2, \ldots, c_{N_{\mathcal{G}_c}}),\ c_i \in \mathbb{R}^3,\ i \in \{1, 2, \ldots, N_{\mathcal{G}_c}\}$, with the corners ordered clockwise in the $XY$-plane. The number of sampled corners $N_{\mathcal{G}_c}$ may vary at each iteration and is directly related to the maximum number of walls that our algorithm can handle, $N_w^{max} = 2(N_{\mathcal{G}_c} - 1)$. As an example, this means that we can draw room layouts with six walls from a minimum number of four corners, allowing the algorithm to introduce two new corners that may be occluded in the image. Additionally, for each corner $c_i$ we know the quadrant it belongs to $q_i$ according to its $x$ and $y$ coordinates. Although the sampling is random, it must satisfy a couple of rules. First, we observe that the proposed quadrant division provides a convenient way to sample corners as Manhattan World rooms always have an even number of walls and an odd number of corners at each quadrant. Therefore, the corner sampling must include corners from at least three quadrants, so that the corner in the

remaining quadrant can be estimated assuming closed Manhattan layouts. This also allows us to find new occluded corners. Second, there must be at least one corner of each hemisphere, so that the total room height can be estimated, $h = d^c + d^f$. Before starting the layout hypothesis generation, the 3D rays of the sampled ceiling corners $\mathcal{G}_c^c$ are intersected into a reference ceiling plane at an assumed distance $d^c$, obtaining already potential 3D ceiling corners $\hat{\mathcal{G}}_c^c$. For the sampled floor corners, we keep their 3D ray $\mathcal{G}_c^f$, as the distance to the floor $d^f$ is yet unknown.

We proceed with the geometric reasoning in 2D, considering a top view of the 3D scene. Please note here the assumption b) in the text, whereby we assume ceiling-floor parallelism. By this assumption, the output ceiling corners are on the same ceiling plane and the output floor corners are directly below the ceiling ones, meaning that corresponding ceiling and floor corners share $x$ and $y$ coordinates. The $z$ coordinate refers to the distance from the camera center to the corresponding ceiling $d^c$ or floor $d^f$ planes, whose normal direction is the vertical vanishing direction $vp_z$. In this way, we work in $\mathbb{R}^2$, reducing the dimensionality of the problem. See right side of Figures 2.4 and 2.5. Following a clockwise order, we always compare consecutive corners, i.e. $c_i, c_{i+1}$. The overall intuition is that, whenever we have corners that belong to the ceiling $\hat{\mathcal{G}}_c^c$, we try to generate consecutive perpendicular walls following the Manhattan World assumption. This is possible as they all belong to the same ceiling plane. To check whether a Manhattan union (wall) is possible or not, we check the angle between consecutive unions, which should be as perpendicular as possible ($90° \pm 5°$), keeping always track of the last wall direction in $dir\_walls$. There are only two cases where we draw new corners: i) when a quadrant $q_i$ is empty, ii) when a quadrant $q_i$ has an even number of corners. The new corners are computed on the fly when i) or ii) are detected. To solve i), a new corner belonging to the empty quadrant is easily computed with its nearest neighbours as $c' = (c_{i+1,x}, c_{i,y})$ if $q_i = 1, 3$, or as $c' = (c_{i,x}, c_{i+1,y})$ if $q_i = 2, 4$. See $c_6$ in Figure 2.5 top. To solve ii), we find the remaining corner similarly, computing in this case the two aforementioned options for $c'$, and selecting the one that generates a perpendicular union according to $dir\_walls$. See $c_4$ in Figure 2.5 top. Every time we compute a new corner, we add it to the tuple of corners $\mathcal{G}_c$, we add its corresponding quadrant to $\mathcal{Q}$, and update $dir\_walls$ with the last wall direction, namely $x$ or $y$.

When the first corner that belongs to the floor $\mathcal{G}_c^f$ appears, the algorithm computes the actual floor corner position along its ray by intersecting walls coming from the nearest corners and taking the average of the intersections. The algorithm proposes walls in the $x$ direction whenever the union is between quadrants $q_1$ and $q_4$ or between

$q_2$ and $q_3$, and walls in the $y$ direction otherwise. See $c_2$ in Figure 2.5 top and bottom. This operation provides the actual $x$ and $y$ coordinates of the corner that could satisfy the Manhattan assumption and allows to recover the distance from the camera center to the floor plane $d^f$ that verifies the ray equation, being the last unknown variable to complete the 3D room hypothesis. See Figure 2.4.

Mathematically, we define the layout by the tuple of corners of the structure of the room up to a scale. Consequently, the output of the proposed algorithm is a list of layout hypotheses $\mathcal{L}_H$ containing the corresponding layout corners, which we parameterize as $(\mathcal{G}_c, d^f, d^c)$, where $d^f$ and $d^c$ are the corresponding distances from the camera to the floor and ceiling planes and $\mathcal{G}_c = (c_1, c_2, \ldots, c_{Nw}),\ c_k \in \mathbb{R}^2,\ k \in \{1, 2, \ldots, N_w\}$, being $N_w$ the number of walls.

**Algorithm 1** Hypotheses Generation

**Input:** candidate corners $C$, $vp$, $d^c$ and $maxIter$
**Output:** layout hypotheses $\mathcal{L}_H$

Initialize $\mathcal{L}_H \leftarrow \emptyset$
**while** iter $\leq maxIter$ **do**
  $\mathcal{G}_c, Q \leftarrow$ sample group of corners and get corresponding quadrants $(C, vp)$
  $\mathcal{G}_c^c, \mathcal{G}_c^f \leftarrow$ differentiate between ceiling and floor corners $(\mathcal{G}_c)$
  $\hat{\mathcal{G}}_c^c \leftarrow$ intersect $\mathcal{G}_c^c$ with reference ceiling plane $(vp_z, d^c)$
  $\mathcal{G}_c \leftarrow$ order clockwise and $\mathbb{R}^3 \rightarrow \mathbb{R}^2$
  Initialize $dir\_walls \leftarrow \emptyset$
  **for** $c_i \in \mathcal{G}_c$ **do**
    **if** $c_i, c_{i+1} \in \hat{\mathcal{G}}_c^c$ **then**
      **if** empty quadrant between $c_i$ and $c_{i+1}$ **then**
        $c' \leftarrow (c_{i+1,x}, c_{i,y})$ **if** $q_i = 1, 3$, $(c_{i,x}, c_{i+1,y})$ **if** $q_i = 2, 4$
        $\mathcal{G}_c, \mathcal{Q}, dir\_walls \leftarrow$ update with $c'$
      **else if** $q_i$ and $q_{i+1}$ are equal or consecutive **then**
        **if** wall from $c_i$ to $c_{i+1}$ goes in $x$ or $y$ direction **then**
          **if** consecutive perpendicular walls **then** $dir\_walls \leftarrow$ update
          **else** break
        **else**
          $c' \leftarrow (c_{i+1,x}, c_{i,y})$ or $(c_{i,x}, c_{i+1,y})$ according to $dir\_walls$
          $\mathcal{G}_c, \mathcal{Q}, dir\_walls \leftarrow$ update with $c'$
    **else if** $c_{i+1} \in \mathcal{G}_c^f$ **then**
      **if** empty quadrant between $c_i$ and $c_{i+1}$ **then** break
      **else if** $q_i$ and $q_{i+1}$ are equal or consecutive **then**
        $c' \leftarrow avg(int(wall\_from\_c_i, c_{i+1}), int(wall\_from\_c_{i+2}, c_{i+1}))$
        **if** Manhattan assumption is satisfied **then**
          $d^f \leftarrow$ get distance from camera to floor plane
          $c_{i+1} \leftarrow c'$
          $\mathcal{G}_c, dir\_walls \leftarrow$ update with $c_{i+1}$
        **else** break
    **else if** $c_i \in \mathcal{G}_c^f$ **then**
      continue
  $\mathcal{L}_{Hi} \leftarrow (\mathcal{G}_c, d^f, d^c)$
  $\mathcal{L}_H \leftarrow$ add hypothesis $\mathcal{L}_{Hi}$

# References


[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

[2] Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. (2018). Learning representations and generative models for 3d point clouds. In *International Conference on Machine Learning*, pages 40–49.

[3] Akhter, I., Sheikh, Y., Khan, S., and Kanade, T. (2008). Nonrigid structure from motion in trajectory space. In *NIPS*.

[4] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. Ieee.

[5] Armeni, I., He, Z.-Y., Gwak, J., Zamir, A. R., Fischer, M., Malik, J., and Savarese, S. (2019). 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5664–5673.

[6] Armeni, I., Sax, A., Zamir, A. R., and Savarese, S. (2017). Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv:1702.01105*.

[7] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

[8] Bao, S. Y., Sun, M., and Savarese, S. (2011). Toward coherent object detection and scene layout understanding. *Image and Vision Computing*, 29(9):569–579.

[9] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346 – 359.

[10] Bazin, J.-C. and Pollefeys, M. (2012). 3-line ransac for orthogonal vanishing point detection. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4282–4287. IEEE.

[11] Bazin, J.-C., Seo, Y., Demonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., and Pollefeys, M. (2012a). Globally optimal line clustering and vanishing point estimation in manhattan world. In *IEE CVPR*, pages 638–645.

[12] Bazin, J.-C., Seo, Y., and Pollefeys, M. (2012b). Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision*, pages 539–551. Springer.

[13] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics.

[14] Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., and Black, M. J. (2016). Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer.

[15] Bogo, F., Romero, J., Pons-Moll, G., and Black, M. J. (2017). Dynamic faust: Registering human bodies in motion. In *CVPR*, pages 6233–6242.

[16] Bregler, C., Hertzmann, A., and Biermann, H. (2000). Recovering non-rigid 3D shape from image streams. In *CVPR*.

[17] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

[18] Brown, M. and Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73.

[19] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7291–7299.

[20] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *transactions on pattern analysis and machine intelligence*, 40(4):834–848.

[21] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*.

[22] Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. (2018). Spherical cnns. *arXiv:1801.10130*.

[23] Concha, A., Hussain, M. W., Montano, L., and Civera, J. (2014). Manhattan and Piecewise-Planar Constraints for Dense Monocular Mapping. In *Robotics: Science and systems*.

[24] Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 941–947. IEEE.

[25] Coughlan, J. M. and Yuille, A. L. (2003). Manhattan world: Orientation and outlier detection by bayesian inference. *Neural computation*, 15(5):1063–1088.

[26] Creusot, C., Pears, N., and Austin, J. (2012). 3d landmark model discovery from a registered set of organic shapes. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 57–64. IEEE.

[27] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3.

[28] Dai, Y., Li, H., and He, M. (2012). A simple prior-free method for non-rigid structure-from-motion factorization. In *CVPR*.

[29] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.

[30] Dasgupta, S., Fang, K., Chen, K., and Savarese, S. (2016). Delay: Robust spatial layout estimation for cluttered indoor scenes. In *IEEE CVPR*, pages 616–624.

[31] Delage, E., Lee, H., and Ng, A. Y. (2006). A dynamic bayesian network model for autonomous 3D reconstruction from a single indoor image. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2418–2428. IEEE.

[32] Deng, F., Zhu, X., and Ren, J. (2017). Object detection on panoramic images based on deep learning. In *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, pages 375–380. IEEE.

[33] Dong, X., Yan, Y., Ouyang, W., and Yang, Y. (2018). Style aggregated network for facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 379–388.

[34] Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. In *IEEE ICCV*, pages 2758–2766.

[35] Dvornik, N., Shmelkov, K., Mairal, J., and Schmid, C. (2017). Blitznet: A real-time deep network for scene understanding. In *ICCV*, pages 4154–4162.

[36] Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, pages 2650–2658.

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

[38] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[39] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

[40] Fernandez-Labrador, C., Chhatkuli, A., Paudel, D. P., Guerrero, J. J., Demonceaux, C., and Van Gool, L. (2020a). Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[41] Fernandez-Labrador, C., Facil, J. M., Perez-Yus, A., Demonceaux, C., Civera, J., and Guerrero, J. J. (April 2020b). Corners for layout: End-to-end layout recovery from 360 images. *IEEE Robotics and Automation Letters, 5 (2), pp: 1255-1262*.

[42] Fernandez-Labrador, C., Facil, J. M., Perez-Yus, A., Demonceaux, C., and Guerrero, J. J. (2018a). Panoroom: From the sphere to the 3d layout. *ECCV Workshop*.

[43] Fernandez-Labrador, C., Perez-Yus, A., Lopez-Nicolas, G., and Guerrero, J. J. (2018b). Layouts from panoramic images with geometry and deep learning. *IEEE Robotics and Automation Letters*, 3(4):3153–3160.

[44] Flint, A., Murray, D., and Reid, I. (2011). Manhattan scene understanding using monocular, stereo, and 3d features. In *Computer Vision (ICCV), 2011 International Conference on*, pages 2228–2235. IEEE.

[45] Forsyth, D. A. (2014). Object detection with discriminatively trained part-based models. *IEEE Computer*, 47:6–7.

[46] Fouhey, D. F., Delaitre, V., Gupta, A., Efros, A. A., Laptev, I., and Sivic, J. (2014). People watching: Human actions as a cue for single view geometry. *International journal of computer vision*, 110(3):259–274.

[47] Furlan *et al* (2013). Free your camera: 3d indoor scene understanding from arbitrary camera motion. *BMVC*.

[48] Gao, Y. and Yuille, A. L. (2016). Symmetric non-rigid structure from motion for category-specific object structure estimation. In *European Conference on Computer Vision*, pages 408–424. Springer.

[49] Gerig, T., Morel-Forster, A., Blumer, C., Egger, B., Luthi, M., Schönborn, S., and Vetter, T. (2018). Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE.

[50] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[51] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

[52] González, Á. (2010). Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49.

[53] Guerrero-Viu, J., Fernandez-Labrador, C., Demonceaux, C., and Guerrero, J. J. (2020). What's in my room? object recognition on indoor panoramic images. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 567–573.

[54] Gupta, S., Arbeláez, P., Girshick, R., and Malik, J. (2015). Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4731–4740.

[55] Gutiérrez-Gómez, D., Mayol-Cuevas, W., and Guerrero, J. J. (2015). What should i landmark? entropy of normals in depth juts for place recognition in changing environments using rgb-d data. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5468–5474. IEEE.

[56] Hasler, N., Ackermann, H., Rosenhahn, B., Thormählen, T., and Seidel, H.-P. (2010). Multilinear pose and body shape estimation of dressed subjects from image sets. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1823–1830. IEEE.

[57] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[58] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[59] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *IEEE CVPR*, pages 770–778.

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

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

[62] Hejrati, M. and Ramanan, D. (2012). Analyzing 3d objects in cluttered images. In *Advances in Neural Information Processing Systems*, pages 593–601.

[63] Hoiem, D., Efros, A. A., and Hebert, M. (2005). Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661. IEEE.

[64] Huang, S., Gong, M., and Tao, D. (2017). A coarse-fine network for keypoint localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3028–3037.

[65] Hussain, W., Civera, J., Montano, L., and Hebert, M. (2016). Dealing with small data and training blind spots in the Manhattan world. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE.

[66] Izadinia, H., Shan, Q., and Seitz, S. M. (2017). Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5134–5143.

[67] J. Xiao, K. A. Ehinger, A. O. and Torralba, A. (2012). Recognizing scene viewpoint using panoramic place representation. *Proceedings of 25th IEEE Conference on Computer Vision and Pattern Recognition*.

[68] Jahromi, A. B. and Sohn, G. (2016). Geometric context and orientation map combination for indoor corridor modeling using a single image. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 41.

[69] Jia, H. and Li, S. (2015). Estimating structure of indoor scene from a single full-view image. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4851–4858. IEEE.

[70] Joo *et al* (2018). Globally optimal inlier set maximization for atlanta frame estimation. *CVPR*.

[71] Karsch, K., Hedau, V., Forsyth, D., and Hoiem, D. (2011). Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)*, 30(6):157.

[72] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[73] Klepeis, N. E., Nelson, W. C., Ott, W. R., Robinson, J. P., Tsang, A. M., Switzer, P., Behar, J. V., Hern, S. C., and Engelmann, W. H. (2001). The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants. *Journal of Exposure Science & Environmental Epidemiology*, 11(3):231–252.

[74] Kneip, L., Li, H., and Seo, Y. (2014). Upnp: An optimal o(n) solution to the absolute pose problem with universal applicability. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*.

[75] Kong, C. and Lucey, S. (2019). Deep non-rigid structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1558–1567.

[76] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[77] Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE.

[78] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[79] Lee, C., Badrinarayanan, V., Malisiewicz, T., and Rabinovich, A. (2017). RoomNet: End-to-end room layout estimation. In *IEEE ICCV*.

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

[81] Li, J., Chen, B. M., and Hee Lee, G. (2018). So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406.

[82] Li, J. and Lee, G. H. (2019). Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 361–370.

[83] Li, Y. (2019). A novel fast retina keypoint extraction algorithm for multispectral images using geometric algebra. *IEEE Access*, 7:167895–167903.

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

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

[86] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16.

[87] Lopez-Nicolas, G., Omedes, J., and J.J. Guerrero (2014). Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. *Robotics and Autonomous Systems*, 62(9):1271–1281.

[88] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[89] Lukierski, R., Leutenegger, S., and Davison, A. J. (2017). Room layout estimation from rapid omnidirectional exploration. In *IEEE International Conference on Robotics and Automation*, pages 6315–6322. IEEE.

[90] Luong, Q.-T. and Faugeras, O. (1995). The fundamental matrix: theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17:43–75.

[91] Mallya, A. and Lazebnik, S. (2015). Learning informative edge maps for indoor scene layout prediction. In *IEEE ICCV*, pages 936–944.

[92] Meng, X., Zhang, X., Yan, K., and Zhang, H. (2018). Real-time detection and recognition of live panoramic traffic signs based on deep learning. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pages 584–588. IEEE.

[93] Mitra, N. J., Wand, M., Zhang, H., Cohen-Or, D., Kim, V., and Huang, Q.-X. (2014). Structure-aware shape processing. In *ACM SIGGRAPH 2014 Courses*, pages 1–21.

[94] Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., and Guibas, L. (2019). Structurenet: hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics*, 38(6).

[95] Moreno-Noguer, F. (2017). 3d human pose estimation from a single image via distance matrix regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2823–2832.

[96] Novotny, D., Ravi, N., Graham, B., Neverova, N., and Vedaldi, A. (2019). C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7688–7697.

[97] Omedes, J., López-Nicolás, G., and Guerrero, J. J. (2013). Omnidirectional vision for indoor spatial layout recovery. In *Frontiers of Intelligent Autonomous Systems*, pages 95–104. Springer.

[98] Organization, W. H. et al. (2011). *World report on disability 2011*. World Health Organization.

[99] Ovsjanikov, M., Corman, E., Bronstein, M., Rodolà, E., Ben-Chen, M., Guibas, L., Chazal, F., and Bronstein, A. (2016). Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, pages 1–60.

[100] Papert, S. A. (1966). The summer vision project.

[101] Parashar, S., Pizarro, D., and Bartoli, A. (2016). Isometric non-rigid shape-from-motion in linear time. In *CVPR*.

[102] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NIPS*.

[103] Pavlakos, G., Zhou, X., Chan, A., Derpanis, K. G., and Daniilidis, K. (2017). 6-dof object pose from semantic keypoints. In *ICRA*.

[104] Perez-Yus, A., Lopez-Nicolas, G., and J.J. Guerrero (2016). Peripheral expansion of depth information via layout estimation with fisheye camera. In *European Conference on Computer Vision*, pages 396–412.

[105] Persad, R. A. and Armenakis, C. (2017). Automatic 3d surface co-registration using keypoint matching. *Photogrammetric engineering & remote sensing*, 83(2):137–151.

[106] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.

[107] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*.

[108] Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE.

[109] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

[110] Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.

[111] Reed, M. P. (2013). Modeling body shape from surface landmark configurations. In *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*, pages 376–383. Springer.

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

[113] Ren, Y., Li, S., Chen, C., and Kuo, C.-C. J. (2016). A coarse-to-fine indoor layout estimation (cfile) method. In *ACCV*, pages 36–51.

[114] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241.

[115] Rosinol, A., Gupta, A., Abate, M., Shi, J., and Carlone, L. (2020). 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289*.

[116] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

[117] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015a). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[118] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015b). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

[119] S. Song, S. L. and Xiao., J. (2015). SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. *Proceedings of 28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[120] Salas, M., Hussain, W., Concha, A., Montano, L., Civera, J., and Montiel, J. (2015). Layout aware visual tracking and mapping. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 149–156. IEEE.

[121] Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947.

[122] Sattler, T., Leibe, B., and Kobbelt, L. (2011). Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE.

[123] Schindler, G. and Dellaert, F. (2004). Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE.

[124] Schwing, A. G., Fidler, S., Pollefeys, M., and Urtasun, R. (2013). Box in the box: Joint 3D layout and object reasoning from single images. In *IEEE International Conference on Computer Vision*, pages 353–360.

[125] Schwing, A. G. and Urtasun, R. (2012). Efficient exact inference for 3d indoor scene understanding. In *European Conference on Computer Vision*, pages 299–313.

[126] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304. Ieee.

[127] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

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

[129] Snavely, N., Seitz, S. M., and Szeliski, R. (2007). Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210.

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

[131] Song, S. and Xiao, J. (2016). Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816.

[132] Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017). Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*.

[133] Spelke, E. S. (1990). Principles of object perception. *Cognitive science*, 14(1):29–56.

[134] Sridhar, S., Rempe, D., Valentin, J., Sofien, B., and Guibas, L. J. (2019). Multi-view aggregation for learning category-specific shape reconstruction. In *Advances in Neural Information Processing Systems*, pages 2348–2359.

[135] Su, Y.-C. and Grauman, K. (2017). Learning spherical convolution for fast features from 360 imagery. In *Advances in Neural Information Processing Systems*, pages 529–539.

[136] Sun, C., Hsiao, C.-W., Sun, M., and Chen, H.-T. (2019). Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1047–1056.

[137] Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (2018). Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943.

[138] Suwajanakorn, S., Snavely, N., Tompson, J. J., and Norouzi, M. (2018). Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems*, pages 2059–2070.

[139] Tang, H., Xu, D., Liu, G., Wang, W., Sebe, N., and Yan, Y. (2019). Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2052–2060.

[140] Tateno, K., Navab, N., and Tombari, F. (2018). Distortion-aware convolutional filters for dense prediction in panoramic images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 707–722.

[141] Taylor, J., Jepson, A. D., and Kutulakos, K. N. (2010). Non-rigid structure from locally-rigid motion. In *CVPR*.

[142] Tola, E., Lepetit, V., and Fua, P. (2009). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830.

[143] Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154.

[144] Torresani, L., Hertzmann, A., and Bregler, C. (2008). Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):878–892.

[145] Tsai, G., Xu, C., Liu, J., and Kuipers, B. (2011). Real-time indoor scene understanding using bayesian filtering with motion cues. In *Computer Vision (ICCV), 2011 International Conference on*, pages 121–128. IEEE.

[146] Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

[147] United Nations, D. o. E. and Social Affairs, P. D. (2019). World population ageing 2019: Highlights (st/esa/ser. a/430).

[148] Verma, N., Boyer, E., and Verbeek, J. (2018). Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *CVPR*.

[149] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE.

[150] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

[151] von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G. (2012). Lsd: a line segment detector, image processing on line,(2012). *URL: http://dx. doi. org/10.5201/ipol*.

[152] Wang, C., Wang, Y., Lin, Z., Yuille, A. L., and Gao, W. (2014). Robust estimation of 3d human poses from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2361–2368.

[153] Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., and Guibas, L. J. (2019). Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651.

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

[155] Wu, S., Rupprecht, C., and Vedaldi, A. (2020). Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*.

[156] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920.

[157] Xiao, J., Ehinger, K., Oliva, A., and Torralba, A. (2012). Recognizing scene viewpoint using panoramic place representation. In *IEEE CVPR*, pages 2695–2702.

[158] Xu, J., Stenger, B., Kerola, T., and Tung, T. (2017). Pano2CAD: Room layout from a single panorama image. In *IEEE WACV*, pages 354–362.

[159] Yang, H. and Carlone, L. (2019). In perfect shape: Certifiably optimal 3d shape reconstruction from 2d landmarks. *arXiv preprint arXiv:1911.11924*.

[160] Yang, H. and Zhang, H. (2016). Efficient 3D room shape recovery from a single panorama. In *IEEE CVPR*, pages 5422–5430.

[161] Yang, S.-T., Wang, F.-E., Peng, C.-H., Wonka, P., Sun, M., and Chu, H.-K. (2018a). Dula-net: A dual-projection network for estimating room layouts from a single rgb panorama. *arXiv:1811.11977*.

[162] Yang, W., Qian, Y., Kämäräinen, J.-K., Cricri, F., and Fan, L. (2018b). Object detection in equirectangular panorama. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2190–2195. IEEE.

[163] Yang, Y., Jin, S., Liu, R., Bing Kang, S., and Yu, J. (2018c). Automatic 3d indoor scene modeling from single panorama. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[164] Yew, Z. J. and Lee, G. H. (2018). 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *European Conference on Computer Vision*, pages 630–646. Springer.

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

[166] Yu, X., Zhou, F., and Chandraker, M. (2016). Deep deformation network for object landmark localization. In *European Conference on Computer Vision*, pages 52–70. Springer.

[167] Zafeiriou, S., Chrysos, G. G., Roussos, A., Ververas, E., Deng, J., and Trigeorgis, G. (2017). The 3d menpo facial landmark tracking challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2503–2511.

[168] Zhang, J., Kan, C., Schwing, A. G., and Urtasun, R. (2013). Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *2013 International Conference on Computer Vision*, pages 1273–1280. IEEE.

[169] Zhang, W., Zhang, W., Liu, K., and Gu, J. (2017). Learning to predict high-quality edge maps for room layout estimation. *Transactions on Multimedia*, 19(5):935–943.

[170] Zhang, Y., Song, S., Tan, P., and Xiao, J. (2014a). PanoContext: A whole-room 3D context model for panoramic scene understanding. In *IEEE ECCV*, pages 668–686.

[171] Zhang, Z., Luo, P., Loy, C. C., and Tang, X. (2014b). Facial landmark detection by deep multi-task learning. In *European conference on computer vision*, pages 94–108. Springer.

[172] Zhang, Z., Rebecq, H., Forster, C., and Scaramuzza, D. (2016). Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 801–808. IEEE.

[173] Zhao, H., Lu, M., Yao, A., Guo, Y., Chen, Y., and Zhang, L. (2017). Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. *arXiv:1707.00383*.

[174] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464.

[175] Zou, C., Colburn, A., Shan, Q., and Hoiem, D. (2018). Layoutnet: Reconstructing the 3d room layout from a single rgb image. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2059.