

# **Automatic Reconstruction of Parametric, Volumetric Building Models from 3D Point Clouds**

Dissertation  
zur  
Erlangung des Doktorgrades (Dr. rer. nat.)  
der  
Mathematisch-Naturwissenschaftlichen Fakultät  
der  
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von  
Sebastian Klaus Ochmann  
aus  
Leverkusen

Bonn 2018

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen  
Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Reinhard Klein

2. Gutachter: Prof. Dr. Renato Pajarola

Tag der Promotion: 20.03.2019

Erscheinungsjahr: 2019

# Acknowledgments

---

I would like to thank my parents for supporting me (not only) throughout my studies and ultimately enabling me to write this thesis. Thank you for always believing in me, especially in more difficult times. I would also like to thank all of my colleagues at the computer graphics department of the University of Bonn. It has always been great to work together with so many nice people in a pleasant environment, no matter where our department was physically located over the years. In particular, I want to express my gratitude to Prof. Dr. Reinhard Klein for accompanying my studies, my diploma and now my thesis, to my fellow student, colleague, and friend Richard Vock for many years of support, honesty and good laughs, and to Raoul Wessel who was my greatest mentor when I started working at the department. I would not have arrived where I am now without all of you guys, thank you.



# Contents

---

<b>Abstract</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Historical Background and State of Research . . . . .	2
1.2 Contributions . . . . .	5
<b>2 Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans</b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 Related Work . . . . .	12
2.2.1 Point-cloud-based approaches . . . . .	13
2.2.2 Approaches based on alternative representations . . . . .	13
2.3 Preprocessing . . . . .	14
2.4 Probabilistic Point-To-Room Labeling . . . . .	14
2.5 Visibility Estimation . . . . .	15
2.6 Door Detection . . . . .	16
2.7 Graph Generation . . . . .	17
2.8 Evaluation . . . . .	18
2.9 Implementation Details . . . . .	19
2.10 Conclusion and Future Work . . . . .	19
<b>3 Towards the Extraction of Hierarchical Building Descriptions from 3D Indoor Scans</b>	<b>23</b>
3.1 Introduction . . . . .	23
3.2 Related Work . . . . .	24
3.2.1 Segmentation and Geometric Structure Extraction . . . . .	24
3.2.2 Object Recognition and Scene Understanding . . . . .	25
3.2.3 Topology Extraction . . . . .	25
3.3 Method Overview . . . . .	26
3.4 Segmentation Into Rooms . . . . .	26
3.5 Detection of Room Neighbors and Connections . . . . .	30
3.5.1 Room Neighborhood . . . . .	30
3.5.2 Room Connectivity . . . . .	31
3.6 Assignment of Objects to Rooms . . . . .	32
3.6.1 Extraction of Objects . . . . .	32
3.6.2 Object Shape Descriptors . . . . .	32
3.7 Graph Matching, Pruning and Scoring . . . . .	33
3.8 Evaluation . . . . .	34

3.9	Conclusion . . . . .	34
<b>4</b>	<b>Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds</b>	<b>37</b>
4.1	Introduction . . . . .	37
4.2	Related Work . . . . .	39
4.3	Approach . . . . .	40
4.4	Point Cloud Segmentation . . . . .	41
4.5	Generation of Wall Candidates . . . . .	42
4.6	Determination of an Optimal Room and Wall Layout . . . . .	43
4.7	Model Generation and Opening Detection . . . . .	46
4.8	Evaluation . . . . .	47
4.9	Conclusion and Future Work . . . . .	51
<b>5</b>	<b>Automatic Reconstruction of Fully Volumetric 3D Building Models from Point Clouds</b>	<b>53</b>
5.1	Introduction . . . . .	53
5.2	Related Work . . . . .	54
5.2.1	Indoor building reconstruction . . . . .	55
5.2.2	Abstraction, segmentation, and reconstruction . . . . .	56
5.2.3	Applications . . . . .	57
5.3	Overview . . . . .	58
5.4	Method . . . . .	58
5.4.1	Plane detection . . . . .	59
5.4.2	Point cloud cleaning . . . . .	59
5.4.3	Point cloud labeling . . . . .	59
5.4.4	Surface candidates . . . . .	60
5.4.5	Wall and slab candidates . . . . .	61
5.4.6	Arrangement of planes . . . . .	61
5.4.7	Volume and surface priors . . . . .	62
5.4.8	Cell complex optimization . . . . .	64
5.4.9	Optimization result . . . . .	66
5.5	Implementation details . . . . .	66
5.6	Evaluation . . . . .	66
5.7	Conclusion and future work . . . . .	72
<b>6</b>	<b>Conclusions and Future Work</b>	<b>73</b>
6.1	Future Work . . . . .	75
<b>A</b>	<b>Curriculum vitae</b>	<b>77</b>
	<b>Bibliography</b>	<b>79</b>
	<b>List of Figures</b>	<b>85</b>
	<b>List of Tables</b>	<b>91</b>

# Abstract

---

Planning, construction, modification, and analysis of buildings requires means of representing a building's physical structure and related semantics in a meaningful way. With the rise of novel technologies and increasing requirements in the architecture, engineering and construction (AEC) domain, two general concepts for representing buildings have gained particular attention in recent years. First, the concept of Building Information Modeling (BIM) is increasingly used as a modern means for representing and managing a building's as-planned state digitally, including not only a geometric model but also various additional semantic properties. Second, point cloud measurements are now widely used for capturing a building's as-built condition by means of laser scanning techniques. A particular challenge and topic of current research are methods for combining the strengths of both point cloud measurements and Building Information Modeling concepts to quickly obtain accurate building models from measured data. In this thesis, we present our recent approaches to tackle the intermeshed challenges of automated indoor point cloud interpretation using targeted segmentation methods, and the automatic reconstruction of high-level, parametric and volumetric building models as the basis for further usage in BIM scenarios. In contrast to most reconstruction methods available at the time, we fundamentally base our approaches on BIM principles and standards, and overcome critical limitations of previous approaches in order to reconstruct globally plausible, volumetric, and parametric models.



# Zusammenfassung

---

Für die Planung, Konstruktion, Modifikation und Analyse von Gebäuden werden Möglichkeiten zur sinnvollen Repräsentation der physischen Gebäudestruktur sowie dazugehöriger Semantik benötigt. Mit dem Aufkommen neuer Technologien und steigenden Anforderungen im Bereich von Architecture, Engineering and Construction (AEC) haben zwei Konzepte für die Repräsentation von Gebäuden in den letzten Jahren besondere Aufmerksamkeit erlangt. Erstens wird das Konzept des Building Information Modeling (BIM) zunehmend als ein modernes Mittel zur digitalen Abbildung und Verwaltung des „As-Planned“-Zustands von Gebäuden verwendet, welches nicht nur ein geometrisches Modell sondern auch verschiedene zusätzliche semantische Eigenschaften beinhaltet. Zweitens werden Punktwolkenmessungen inzwischen häufig zur Aufnahme des „As-Built“-Zustands mittels Laser-Scan-Techniken eingesetzt. Eine besondere Herausforderung und Thema aktueller Forschung ist die Entwicklung von Methoden zur Vereinigung der Stärken von Punktwolken und Konzepten des Building Information Modeling um schnell akkurate Gebäudemodelle aus den gemessenen Daten zu erzeugen. In dieser Dissertation präsentieren wir unsere aktuellen Ansätze um die miteinander verwobenen Herausforderungen anzugehen, Punktwolken mithilfe geeigneter Segmentierungsmethoden automatisiert zu interpretieren, sowie hochwertige, parametrische und volumetrische Gebäudemodelle als Basis für die Verwendung im BIM-Umfeld zu rekonstruieren. Im Gegensatz zu den meisten derzeit verfügbaren Rekonstruktionsverfahren basieren unsere Ansätze grundlegend auf Prinzipien und Standards aus dem BIM-Umfeld und überwinden kritische Einschränkungen bisheriger Ansätze um vollständig plausible, volumetrische und parametrische Modelle zu erzeugen.



## Introduction

---

Planning, construction, modification, and analysis of buildings requires means of representing a building's physical structure and related semantic information in a meaningful way. Depending on the respective requirements, such a representation may be abstract and idealizing, or realistic and detailed in its nature. With the rise of novel technologies and increasing requirements in the architecture, engineering and construction (AEC) domain, two general concepts for representing buildings have gained particular attention in recent years. First, the concept of *Building Information Modeling* (BIM) is increasingly used as a modern concept for representing and managing a building's *as-planned* state digitally, including not only a geometric model but also various additional semantic properties. Second, point cloud measurements are now widely used for capturing a building's *as-built* condition by means of e.g. laser scanning techniques. Applications for point clouds in the AEC domain include fast acquisition of measurements of the already existing, legacy building stock, architectural heritage, surveillance, and comparing the as-planned to as-built states of buildings.

One particular challenge and topic of current research are methods for combining the strengths of both point cloud measurements and Building Information Modeling concepts to quickly obtain accurate building models from measured data. While techniques and devices for the acquisition of high quality point clouds have become fast and affordable, this initial advantage is often lost due to largely manual and cumbersome methods for working with the measured data. First, point cloud scans are mostly unstructured, i.e. they contain millions of measured points without any annotations or relations to higher-level entities like floors, walls, or rooms. Since point clouds are measurements of the real world, they usually also contain large amounts of clutter and outlier points that are not part of the building. These aspects complicate their usage by making the measured data difficult to navigate and interpret. This also hinders automated means of assisting the user by e.g. highlighting or hiding parts of the data on a semantically meaningful level of rooms or other architectural elements. Second, even using commercial software products, generating BIM models based on point cloud data still is a laborious task where elements of the BIM model are often fitted manually to the measured data. The previously mentioned lack of semantic annotation or segmentation of the point cloud data further aggravates this issue.

In this thesis, we present our recent approaches to tackle the intermeshed challenges of automated indoor point cloud interpretation using targeted segmentation methods, and the generation of high-level, parametric and volumetric building models as the basis for further usage in BIM scenarios. In the following, we first give an overview of the historical background and the state of research dealing with approaches for tackling the aforementioned challenges. Afterwards, we give an overview of our contributions to the tasks of automated point cloud interpretation and building model reconstruction, and present each of our methods in detail.

## 1.1 Historical Background and State of Research

Planning and design of buildings was traditionally conducted by means of drawings, reflecting the ideas of the architect in the form of either accurate, realistic images, or by means of abstracted, idealized technical drawings. While this allows the designer to formulate his or her ideas in a quite efficient way, the end result is a rather static representation which is hard to modify, even less in a collaborative manner. The first step towards digital drawing systems, which were usable for persons without programming skills, was the now legendary Sketchpad by Ivan Sutherland [1]. It arguably paved the way for generally available computer aided design (CAD) applications which allowed the user to easily revise drawings, include parametric constraints which were automatically resolved, create reusable libraries of drawing elements, and easily make precise copies.

Visions of the applicability of computer-based descriptions with a particular focus on buildings date back to at least 1974, when Eastman et al. [2] described their idea of the Building Description System (BDS). They first identify crucial shortcomings of drawings with respect to the architecture domain: Redundancy due to the necessity to depict areas of the building multiple times at e.g. different scales or from different angles, the difficulty to keep information current over the course of a building's life cycle, and cumbersome augmentation of plans with additional information beyond the geometric representation. While they merely provide an outline of their vision and ideas for an implementation for PDP-11 computers at this point, their ideas strikingly foreshadow fundamental concepts which would later be returned to by modern approaches: Libraries of building parts which could even be provided by companies as a service, a common data basis for generating various views and plans, automated quantitative analysis, volumetric spaces and solids together with operations to work with them, and connectivity between individual building components.

In the following decades, several attempts were made at defining formats and concepts for describing so-called *product models* for (but not necessarily limited to) architecture, engineering and design (AEC) applications. One approach to formulate AEC-specific requirements was presented as the General AEC Reference Model (GARM) [3] in 1988. Being part of the "Standard for the exchange of product model data" (STEP) that was in development at that time, it was a relatively small piece of the much larger and ambitious ISO/STEP standardization effort. Other related approaches proposed during that time were the "Computer Models for the Building Industry in Europe" (COMBINE) project [4], which aimed at leveraging the upcoming STEP standard for analyzing buildings under energy and heating, ventilation and air conditioning (HVAC) aspects, and the RATAS project [5], which aimed at developing concepts and formats for product model data exchange and databases. The STEP standard was in development from 1984 to 2002 until all parts of it were released as ISO 10303. The standard consists of a large number of individual parts dealing with various aspects of "electronic exchange of product data between computer-based product life-cycle systems" [6].

In 1995, while the STEP standard was still in development, an alliance of twelve companies started the development of standards for data exchange tailored specifically to the architecture domain. Their goal was to make interoperability and information exchange available to the industry and fundamentally base their developments on open standards. This approach was further developed under the name International Alliance for Interoperability (IAI) founded in 1996, which was later renamed to buildingSMART in 2008 [7]. It was during that time that the term *Building Information Modeling* (BIM) was coined by Laiserin in 2001 [8] to describe "the next generation of design software" which deals with "more than just geometry". One main argument for this novel concept of dealing with information was that classical CAD principles – let alone drawings – were no longer sufficient to encompass the increasingly complex processes around design, planning and management of buildings. A concrete realization of these principles is the Industry Foundation Classes (IFC) [9] standard developed by buildingSMART

which integrates two tightly related aspects into a common standard. First, IFC not only describes a building's geometry but also integrates additional, possibly abstract information such as time schedules, usage annotations, materials, participating actors, and regulatory aspects. Ideally, this enables the usage of an IFC BIM model as a central data basis throughout a building's life cycle which can be used collaboratively by various participating parties which are interested in different aspects of an edifice. Second, the building's geometry is described as parametric, interrelated elements which strongly resembles the ideas envisioned by Eastman in 1974 [2]. In particular, elements such as walls, floors, doors, and windows are represented by volumetric, parametric entities together with additional annotations how these elements are interconnected and related to each other. In contrast to simple boundary mesh models, this not only more closely resembles the physical structure of a building, but also inherently provides information about e.g. room topology.

A complementing counterpart to the often idealized *as-planned* state represented by BIM models is the measurement of the physical *as-built* state of already existing buildings or other objects. Advancements in terrestrial laser scanning technology [10] led to rapid adoption of 3D point clouds as a standard means to efficiently obtain detailed, high-resolution measurements in various domains such as architecture, heritage conservation, and archaeology [11]. With respect to the AEC community and modern Building Information Modeling, 3D point cloud measurements have become a valuable and increasingly cost-effective tool for performing digital measurements, verification of construction or renovation processes, and synchronization of as-planned models with as-built structures. One particular application is the usage of point clouds as a template or stencil for creating or modifying models of already existing buildings. However, there exists a critical semantic gap between largely unstructured point cloud data and highly structured BIM models which consequently makes interpreting the point cloud measurements and generating a model a highly manual and time-consuming process, even using current, commercial software tools. For this reason, several research communities ranging from computer graphics and computer vision to photogrammetry and robotics aim at the development of methods for automated reconstruction of digital building models from 3D point cloud data. The general approach usually consists of the intertwined challenges of interpreting the unstructured point cloud data and deriving semantic information about e.g. room topology, object classes, outliers, etc., and subsequently generating a model that fits to the measured data and derived information, possibly adhering to additional constraints defining the plausibility of a reconstructed model.

Depending on the targeted use case, interpretation of point cloud data may deal with very different levels of detail and semantics. In the simplest case, detecting primitive shapes such as planes may already give insight into the structure of man-made objects, and the main architectural elements of buildings are usually representable by a set of simple shapes. The result of shape detection usually is a segmentation of the point cloud into subsets belonging to the respective detected shapes, and parametric definitions for each primitive. A variety of approaches for detecting primitive shapes in point clouds such as region growing [12] and RANSAC-based methods [13] have been proposed. In addition to planes, the latter method also supports the detection of other kinds of shapes such as cylinders and spheres which may also be relevant in an architecture context (e.g. columns). Such detected shapes are already valuable since they allow for an extraction of candidates for e.g. wall surfaces while filtering out unstructured clutter. They are also a prerequisite for many reconstruction approaches which use them as a first step to detect surfaces for further analysis.

Going a step further, a classification of detected shapes as walls, floors, and ceilings, as well as an automated segmentation into e.g. rooms and stories can help to understand the structure of a building on a higher semantic level. Some examples for this level of semantics include the methods by Adán et al. [14] and Ambruş et al. [15] which perform a classification of detected surfaces in order to reconstruct rooms

from the point cloud data. In contrast to low-level primitive shapes, such classifications and segmentations may take into account semantics which are specific to the particular use-case or domain. For instance, a segmentation of an indoor point cloud into separate rooms can support navigation tasks and improve understandability of complex datasets by enabling visualization of separate rooms or stories. Certain reconstruction approaches, such as the ones presented in this thesis, perform a high-level segmentation of the point cloud data as a prerequisite for reconstructing a building model.

More detailed segmentation and classification approaches may detect and annotate finer details and more object classes which go beyond the coarse wall, floor and ceiling surfaces of a building. Entities like door and window openings, beams, columns, and other objects such as fire suppression equipment are also defined in the IFC standard for BIM models, making approaches for detecting such elements a valuable addition for further enrichment of models. Recent examples dealing with more fine-grained structural elements are the approaches by Son et al. [16] and Bassier et al. [17] which consider additional entities such as columns and beams in point cloud data. A method for detecting openings in wall surfaces is presented by Michailidis et al. [18] who perform a graph-cut-based segmentation in two-dimensional arrangements of lines on detected planar surfaces. They also explicitly take into account occluders in order to detect and recover occluded areas on surfaces which are very common in real-world, indoor scenarios. The approach by Adán et al. [19] aims to detect a large variety of different kinds of objects relevant to the BIM domain in RGB-D data by means of segmentation and classification in the color and depth images.

All of the above mentioned goals aim at the interpretation of the point cloud data itself which can support the reconstruction of higher-level building models. In turn, these reconstruction approaches may also aim at different levels of detail. Detected planes, as mentioned above, may already be interpreted as a simple model of the building under the often valid assumption that the geometry of buildings may be sufficiently approximated by piecewise planar shapes. For example, Sanchez et al. [20] fit a planar model to the input point cloud data by classifying detected planes as ceiling, floor, or wall surfaces, and approximating their shapes as concave hulls using alpha-shapes. In addition, they propose further detection of parametric stair models in the remaining set of unclassified points. The approach by Monszpart et al. [21] also uses sets of planes to represent scenes of man-made objects such as indoor and outdoor building scans. Their focus lies on the regularization of the resulting set of planes with respect to certain relationships between planes such as mutual parallelism or orthogonality constraints which are common in man-made objects such as buildings. However, these approaches do not take into account important, domain-specific semantics such as wall connectivity, or volumetric entities.

Analogously to higher-level segmentations of point clouds, some approaches model separate rooms as watertight volumes, thus integrating surface connectivity into the reconstruction process. For example, the method by Mura et al. [22] is among the state-of-the-art methods for reconstructing room boundary representations. The approach is based on a volumetric cell complex induced by an arrangement of planes which is also used by other promising approaches [23–26], and by our own reconstruction methods presented in this thesis. The goal is to find plausible locations of different rooms within the volumetric cell complex by means of an optimization problem steered by prior information derived from the input point cloud data. In contrast to many other approaches, which only support vertical and horizontal surfaces, Mura et al. support the more general case of slanted ceiling surfaces in their reconstruction. While volumetric descriptions of the interiors of rooms are also part of the IFC standard, these approaches do not explicitly model the volumetric floor, ceiling and wall elements and their relations. However, in order to be usable in the envisaged Building Information Modeling scenario, a globally plausible arrangement of these volumetric elements is required.

Apart from our contributions described in this thesis, few other approaches have tackled the challenge

of reconstructing buildings based on volumetric wall elements. For example, Stambler et al. [27] generate an over-complete set of wall candidates by a volumetric growing approach and subsequently regularize and prune elements by means of an optimization approach. Macher et al. [28] propose a semi-automatic approach for the reconstruction of IFC models. They first perform an automated detection of planes and classification of relevant points, resulting in a temporary 3D model in OBJ format. This model is subsequently loaded into an open source modeling software and supports manual generation of the final volumetric elements. The method by Murali et al. [29] aims at automatically reconstructing semantically labeled BIM models. However, they assume walls to be oriented in a Manhattan world manner, i.e. walls have two principal, orthogonal directions, and they do not support multiple stories. While these approaches are closely related to our contributions to the reconstruction challenge in that they base their models on volumetric wall entities, they have several drawbacks such as assuming specifically oriented geometry, lack of support for multiple stories, mandatory manual intervention, and no enforcement of critical constraints such as plausible enclosing of rooms by wall elements.

## 1.2 Contributions

In contrast to most reconstruction methods available at the time, our goal was to fundamentally base our reconstruction approaches on BIM principles and entities defined in the IFC standard, and to overcome limitations of previous approaches. Guided by an initial, semantically high-level and automatic interpretation of the input point clouds, our reconstruction methods find a globally plausible arrangement of volumetric rooms, floors, ceilings, and walls by means of global optimization without restriction to the number of stories or wall orientation. In this thesis, we tackle these two strongly related challenges of automated interpretation of indoor building point clouds, and the automatic reconstruction of volumetric, parametric building models.

The first two papers deal with the automated segmentation of indoor laser scans into semantic entities of the building on different levels of detail, including stories, rooms, and objects. Furthermore, the detection of door openings allows for the generation of a lightweight, graph-based representation of a building's room topology which can be used e.g. for navigation tasks. In the second part, which also consists of two publications, we present our contributions to the challenge of automatic generation of building models from point cloud data which are usable in a BIM setting. To this end, the reconstruction process is based on parametric and volumetric building elements that are closely related to established BIM model formats such as the Industry Foundation Classes (IFC). As part of the reconstruction, the automated segmentation of the input point cloud into rooms serves as an important prerequisite for estimating where rooms and surfaces are likely located.

### Semantic Interpretation of Point Clouds

The first two papers presented in this thesis dealing with automated interpretation of point clouds are:

- *Automatic generation of structural building descriptions from 3D point cloud scans*, Sebastian Ochmann, Richard Vock, Raoul Wessel, Martin Tamke and Reinhard Klein, in proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP) 2014. (Chapter 2)
- *Towards the Extraction of Hierarchical Building Descriptions from 3D Indoor Scans*, Sebastian Ochmann, Richard Vock, Raoul Wessel and Reinhard Klein, Eurographics Workshop on 3D Object Retrieval (3DOR) 2014. (Chapter 3)

Our first contribution tackles the challenge of performing an automatic structuring and decomposition of 3D indoor point clouds based on the architectural unit of *rooms* of a building. The result is a decomposition of the point cloud on a higher semantic level than e.g. low-level primitive shapes such as planes. In addition, openings between neighboring rooms are detected and the respective room pairs are connected in a graph structure, yielding a graph-based, lightweight building descriptor.

The basic idea of our approach is that we start with a labeling of the point cloud (i.e. each point has one of  $n$  labels assigned to it) such that the labeling roughly corresponds to the rooms of the building. We then correct this initial, coarse labeling by iteratively changing labels of points where appropriate. The main questions are how to obtain a suitable initial labeling, and how to refine this labeling to correspond to rooms.

Since the notion of rooms is often vague such that clear heuristic rules are hard to define, we devised an intuitive definition on which we base a probabilistic point clustering approach for determining a suitable room labeling. As a starting point for the labeling, we assume that each room has been scanned from one (or only few) scan positions such that separate scans roughly correspond to the individual rooms. We thus use this given segmentation into scans as initial labeling of the point cloud. Overlaps between different scans result in areas where the labeling does not correctly correspond to rooms, necessitating a refinement of the labeling. To this end, we make the intuitive assumption that a particular point  $p$ , which is currently labeled as being part of a room  $r$ , is likely to be labeled correctly if and only if many other points that are labeled as room  $r$  are *visible* from  $p$ , i.e. the line of sight is not obstructed by permanent structures. In particular, if  $p$  lies outside of room  $r$ , only few other points within  $r$  will be visible from  $p$  (e.g. through a door opening). This intuition is implemented as a probabilistic clustering which takes into account mutual visibility between point pairs where visibility is estimated by means of ray casting performed against planar structures detected in the point cloud data. For each point, the most likely room label with respect to this probabilistic estimation is selected and the process is iterated until convergence.

After the labeling process, openings between neighboring rooms are detected. To this end, simulated laser scanning from the original scan positions is employed to detect rays which originate in a different room than the points measured by the respective ray. In this case, a transition between rooms must take place along the ray. By intersecting the rays with planes detected in the point cloud and clustering of the intersection points, locations and extents of openings are determined.

Finally, the room segmentation together with connections between rooms are combined in a graph-based descriptor with node and edge attributes for estimated room area, point cloud subsets per room, and opening extents. Such representations have previously been used for e.g. graph-based retrieval and classification of buildings [30], and can also facilitate navigation of the point cloud data by targeted selection and visualization of point cloud subsets. A semantic segmentation of the point cloud may also be used as a starting point for further analysis and processing as later shown in our contributions to the building model reconstruction problem.

Improving on the room segmentation described above, we propose the concept of a hierarchical building descriptor using a more fine-grained decomposition of the point cloud into stories, rooms and objects. Based on the idea of a graph-based abstraction of a building, we construct a descriptor in which the elements are not only connected to entities on the same level of detail (e.g. neighboring rooms) but also across different levels (e.g. a room and objects contained therein). As an example application for the devised descriptor we demonstrate graph-based retrieval of room and object constellations within buildings.

Similar to the room segmentation performed in the approach described above, the initial labeling is based on separate scans in the input point cloud data. In contrast to the first contribution, the refinement of the room labeling is modeled as a diffusion process based on mutual visibility between point pairs.

After the point cloud has been segmented into rooms, floor heights are estimated separately for each room in order to subsequently cluster rooms into stories by means of a greedy binning process. Additionally, horizontal as well as vertical connections between rooms are detected in a similar manner as described in the first contribution. Point subsets corresponding to the individual rooms are further segmented into more fine-grained elements to extract individual objects within rooms. First, permanent structures such as wall and floor surfaces are filtered out by removing points on detected planes. Subsequently, separate objects in the remaining point set are extracted by means of connected component segmentation. To enable queries on the building descriptor by means of object retrieval, a global shape descriptor is computed for each of the detected objects.

The extracted stories, rooms, and objects are combined into a holistic graph-based building descriptor in which each entity is represented by a node, and edges connect stories to rooms, neighboring rooms to each other, and rooms to objects contained within. The different nodes are attributed with properties of the respective entities such as room areas, opening extents and object descriptors. Finally, we demonstrate applicability of our approach for performing graph-based queries by means of finding subgraph monomorphisms of a query graph within the building descriptor.

## **Reconstruction of Parametric, Volumetric Building Models**

The latter two papers presented in this thesis deal with the reconstruction of digital models from indoor point cloud scans:

- *Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds*, Sebastian Ochmann, Richard Vock, Raoul Wessel and Reinhard Klein, Computers & Graphics special issue on CAD/Graphics 2015. (Chapter 4)
- *Automatic reconstruction of fully volumetric 3D building models from point clouds*, Sebastian Ochmann, Richard Vock and Reinhard Klein, submitted to ISPRS Journal of Photogrammetry and Remote Sensing 2018. (Chapter 5)

Automatic interpretation of indoor point clouds as described above is a first step towards automated understanding of building structure from measured data, and is an important prerequisite for the reconstruction of digital building models described in the following. Analogously to the varying levels of detail and semantics of point cloud decompositions, the reconstruction of geometric models from point cloud scans may target different goals, resulting in models with vastly different granularity, semantics, and structuredness. While most previous reconstruction approaches generate models based on boundary surface representations, our main goal is to base our resulting models on the notion of volumetric elements for walls, floors and ceilings which not only closely resembles the physical structure of buildings, but also corresponds to concepts and file formats used in Building Information Modeling.

In this thesis, we present two approaches for tackling this challenge. While the first method is restricted to single-story buildings and makes assumptions about the input data regarding the availability of separate scans within each room, the second method overcomes these limitations and is able to reconstruct models for multi-story buildings from completely unstructured point cloud scans. The latter method also introduces a new formulation of the reconstruction task as an integer linear programming problem which offers flexible means of steering the reconstruction using hard constraints.

Our first method automatically constructs parametric, globally interconnected walls from indoor point clouds by means of solving a global energy minimization problem. The input data is a registered point cloud of a single building story consisting of separate scans. The first step is to perform a room segmentation of the point cloud data for which we employ an extended version of the segmentation

approach described in the previous Section. In addition to assigning each point to one room of the building as described above, outlier points are also removed as part of this process. This removes large amounts of irrelevant points outside of the building which are very common in real-world scans due to scanning through windows.

Using the obtained segmentation of the point cloud into individual rooms, our goal is to find a constellation of volumetric wall elements which fits to the measured data with respect to two criteria. First, walls should fit to the observed vertical surfaces in the point cloud data. Second, the areas enclosed by walls (i.e. rooms) should match the previously estimated point cloud segmentation. In particular, different segments of the point cloud should result in different rooms in the resulting model, separated by suitably placed walls. We formulate this task as an optimization problem which requires two parts, namely a geometric data structure representing candidate locations for the placement of walls and room areas, and a means to steer the selection of suitable wall and room locations.

Similar to promising related approaches, the first part is realized as a two-dimensional *arrangement of lines* as follows. Detected vertical planes in the point cloud are considered as candidates for wall surfaces. In some previous methods, these surfaces were projected into the horizontal plane as infinitely long lines and intersected with each other to obtain an arrangement of lines. This induces a cell complex whose faces represent bounded 2D areas which can be part of a room or the outside area, and the separating edges are candidates for wall surfaces. A problem of this approach is that wall surfaces are considered independently of each other, without taking into account the relationship between opposing surface pairs which are necessary to reconstruct volumetric wall elements, each consisting of a pair of surfaces. We thus introduce the notion of *wall centerlines* which are used instead of single wall surfaces in our line arrangement. Matching pairs of nearby, almost parallel wall surfaces with opposing normals are determined, and the centerline in between each pair is inserted into the arrangement, attributed with a wall thickness estimated from the respective surface pair. Since walls between rooms and the outside area are usually scanned from only one side, additional virtual centerlines are added to account for these cases.

The second part, i.e. the selection of suitable wall and room locations, is formulated as a graph multi-label problem. Each of the faces of the cell complex may be assigned one of multiple possible room labels, or a special label for the outside area. By defining a suitable cost function which penalizes the placement of rooms and walls at implausible locations, we obtain a cost minimization problem which results in a room labeling of all cells of the arrangement. For estimating plausible locations of rooms and walls, the previously determined point cloud segmentation and detection of vertical surfaces is used as priors, i.e. the respective room labels should be used where the different segments of the point clouds are located, and walls should be supported by planes detected in the scans.

The resulting room labeling can then be interpreted as a constellation of wall elements, i.e. wall centerlines that separate differently labeled cells are reconstructed as walls with a thickness associated with the respective line segments of the arrangement. An important property of this model is that, by definition, walls always properly enclose rooms, and additional information how walls are connected to each other is directly available from the data structure. Finally, openings in the reconstructed walls are detected by means of ray casting against the walls, similar to the detection performed in our first contribution. However, we extend this approach by a classification of detected openings as doors and windows, which are then added to the model and related to the reconstructed wall elements.

We improve on this method in various ways in our second reconstruction approach. First, for the initial point cloud segmentation into rooms, we do not rely on the availability of separate scans in the input data, and we do not impose certain rules for scanning such as having one or only few scans within each room. Instead, we devise a fully automatic and unsupervised clustering method based on graph

flow simulation between point pairs, steered by mutual visibility between points. The intuitive idea is that point pairs within rooms have higher mutual visibility and thus natural clusters of points with high visibility can be determined. In contrast, point sets in separate rooms are only weakly connected, i.e. through openings between rooms. This fully automatic clustering enables usage of the reconstruction algorithm on a broader variety of datasets captured using scanning devices which may not even deliver discrete, separate scans but e.g. a continuous stream of points.

Second, while the method is also based on the idea of cell complexes for finding a plausible constellation of building elements, we employ a 3D cell complex instead of projecting vertical surfaces down to the horizontal plane. By not only considering vertical walls but also horizontal slabs, we enable the reconstruction of buildings consisting of multiple stories, including complex cases of individual rooms spanning multiple stories of the building. Also, instead of representing walls as centerlines between surfaces with a scalar thickness, we represent walls as true volumes in 3D space, and the placement of walls is formulated as part of the cell labeling problem, similar to room interiors and the outside area. As demonstrated in detail in Chapter 5, this improves on the regularization of walls compared to our previous approach, and also directly results in a fully volumetric model of interconnected wall elements instead of converting the centerline structure to volumetric entities in a post-processing step.

Third, we propose a new formulation of the labeling task as a 0-1 integer linear programming problem. In this setting, binary variables corresponding to rooms, the outside area, or walls are assigned to each cell of the 3D cell complex, and the value of each variable (i.e. 0 or 1) determines whether the respective label is assigned to the cell. The determination of an optimal assignment is steered by an objective function as well as a set of hard constraints which define a set of rules that any valid labeling needs to fulfill. Examples for constraints include that each cell may only be assigned to at most one room, or that a wall must exist at the boundary of rooms. In contrast to previous graph-cut based optimization approaches – including our first reconstruction method – this allows for the formulation of complex rules directly as hard constraints of the optimization problem.



# Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans

---

**Abstract.** We present a new method for automatic semantic structuring of 3D point clouds representing buildings. In contrast to existing approaches which either target the outside appearance like the facade structure or rather low-level geometric structures, we focus on the building’s interior using indoor scans to derive high-level architectural entities like rooms and doors. Starting with a registered 3D point cloud, we probabilistically model the affiliation of each measured point to a certain room in the building. We solve the resulting clustering problem using an iterative algorithm that relies on the estimated visibilities between any two locations within the point cloud. With the segmentation into rooms at hand, we subsequently determine the locations and extents of doors between adjacent rooms. In our experiments, we demonstrate the feasibility of our method by applying it to synthetic as well as to real-world data.

This chapter corresponds to the paper *Automatic generation of structural building descriptions from 3D point cloud scans*, Sebastian Ochmann, Richard Vock, Raoul Wessel, Martin Tamke and Reinhard Klein, in proceedings of the International Conference on Computer Graphics Theory and Applications (GRAPP) 2014.

## 2.1 Introduction

Digital 3D representations of buildings highly differ regarding the amount of inherent structuring. On the one hand, new building drafts are mainly created using state-of-the-art Building Information Modeling (BIM) approaches that naturally ensure a high amount of structuring into semantic entities ranging e.g. from storeys and rooms over walls down to doors and windows. On the other hand, 3D point cloud scans, which today are increasingly used as a documentation tool for already existing, older buildings, are almost completely unstructured. The semantic gap between such low- and high-level representations hinders the efficient usage of 3D point cloud scans for various purposes including retrofitting, renovation, and semantic analysis of the legacy building stock, for which no structured digital representations exist. As a solution, architects and construction companies usually use the measured point cloud data to manually generate a 3D BIM overlay that provides enough structuring to allow easy navigation and computation of key properties like room or window area. However, this task is both cumbersome and time-consuming.

While existing approaches for building structuring either require 3D CAD models [30, 31], consider primarily outdoor (airborne) scans [32–34], or target the recognition of objects like furniture inside single

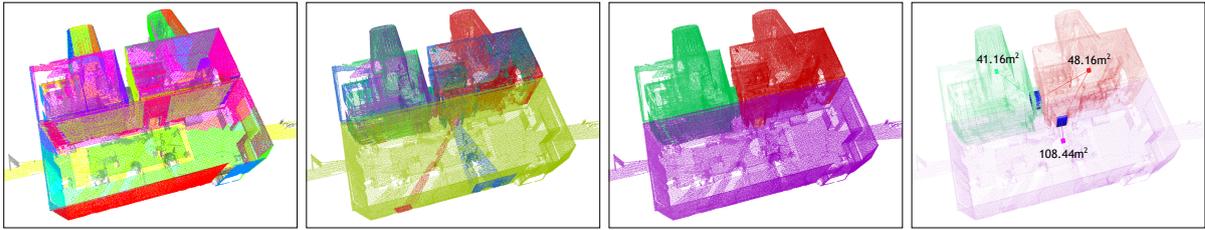


Figure 2.1: Overview of our method. From left to right: Detected planar structures; colour-coded indices of the individual scans; final point assignments after the re-labeling process; extracted room topology graph.

rooms [35, 36], our method focuses on the extraction of the overall room structure from indoor 3D point cloud scans. We aim at first decomposing the point cloud such that each point is assigned to a room. To this end, we propose a probabilistic clustering algorithm that exploits knowledge about the belonging of each measured point to a particular scan. The resulting segmentation constitutes the fundamental building block for computing important key properties like e.g. room area. In a second step we detect doors connecting the rooms and construct a graph that encodes the building topology.

Our method may be summarized as follows (see also Figure 2.1): We start with multiple indoor 3D point clouds scans which have been registered beforehand (this step is not in the scope of this paper). Each point is initially labeled with an index of the scan from which it originates. In a preprocessing step, normals are estimated for each point (if necessary) and planar structures are detected using the algorithm by [13]. This yields a set of planes together with subsets of points corresponding to each plane. Next, point sets belonging to scanning positions that are located in the same room are merged; this affects those rooms in which scans from multiple positions were performed. This step may be either performed automatically or semi-automatically using a graphical user interface for correcting automatically generated results. Following the merging step, an iterative re-labeling process is performed to update assignments of points to rooms. This process uses a probabilistic model to estimate updated soft assignments of points to room labels. After convergence, a new (hard) assignment of points to rooms is obtained.

Finally, the original and the new labeling of the points are used to determine the locations and extents of doors between pairs of adjacent rooms. Together with the set of room points, this constitutes a room connectivity graph in which rooms are represented by attributed nodes and doors are represented by edges. Such representations are an important tool for semantic building analysis or graph-based building retrieval [30, 31].

The contributions of this paper are:

1. Automatic decomposition of a 3D indoor building scan into rooms.
2. A method for detecting connections between adjacent rooms using the computed decomposition.
3. Construction of a graph encoding the building’s topology purely from point clouds.

## 2.2 Related Work

In this Section we briefly review the related work on the extraction of structural building information from digital representations.

### 2.2.1 Point-cloud-based approaches

In [32] a parametric shape grammar is used to detect common roof shapes in LIDAR data. They initially extract connected components of planar point sets and filter out small components in order to get separated subsets of points for ground and roof structures. Subsequently arbitrary polygons are fitted to the roof and by analyzing their topological relations, simple, parameterized roof shapes are matched to the scan data. This approach is rather specific to roof shape detection, where a concise, parametric model for detected objects can be derived. By representing point clouds as a topology graph of interrelated basic shape primitives [33] reduce the problem of finding shapes in point clouds to a subgraph matching problem. The scene as well as the query object are decomposed into primitive shapes using a RANSAC (random sample consensus) approach. A topology graph for each cloud is constructed capturing the spatial relationships between these primitive shapes. A lot of research has recently been conducted in exploiting symmetries in order to understand scan data. For example [34] detect partial symmetries in point cloud data by matching line features detected using slippage analysis of point neighbourhoods in a moving least squares scheme. Matching these line features and extracting symmetric components yields a decomposition of the scanned scene into cliques of symmetric parts. These parts however do not necessarily carry a semantic meaning in architectural terms and are therefore not suited for semantic analysis of building structures.

While the above methods are used to structure scans showing the outside of a building, the interior of buildings has recently gained attention, especially in the field of robotics, where segmentation of furniture is an important topic for indoor navigation, see e.g. [37, 38]. An iterative search-and-classify method is used by [36] for segmentation of indoor scenes. They start with an over-segmented scene and iteratively grow regions while maximizing classification likelihood. A similar, supervised learning approach is used by [35]. Their method is divided into a learning phase, in which multiple scans of single objects are processed in order to obtain a decomposition into simple parts as well as to derive their spatial relations, and a recognition phase. Both of these methods focus on the recognition of objects like furniture that can be well represented by certain templates and do not take the overall room structure of a building into account.

### 2.2.2 Approaches based on alternative representations

[30] extract topological information from low-level 3D CAD representations of buildings. They find floor planes in the input polygon soup and extract 2D plans by cutting each storey at different heights. These cuts are then analyzed in order to extract rooms and inconsistencies between different cut heights yield candidates for doors and windows. In [39], high-level BIM models are used for the extraction of a building's topology. They derive information about the topological relationship between rooms from IFC (Industry Foundation Classes) files by analysis of certain entity constellations which can be used to perform sketch-based queries in a database. [40] use 2D floor plans to extract the structure and semantics of buildings. The input drawing is first segmented into graphical parts of differing line thickness used to extract the geometry of rooms by means of contour detection and symbol matching, and a textual part used for semantic enrichment by means of OCR (optical character recognition) and subsequent matching with predefined room labels. A hybrid approach is presented in [41]. The authors use 3D point cloud data together with ground-level photographs to reconstruct a CSG representation based on fitted rectangle primitives. Note that this work rather focuses on indoor scene reconstruction than on the actual extraction of a semantic structuring.

## 2.3 Preprocessing

The input data for our method consists of multiple 3D point clouds which have been registered beforehand in a common world coordinate system. The registration step is usually done automatically or semi-automatically by the scanner software and is not in the scope of this paper. Each individual scan also contains the scanner position.

If normals are not part of the input data, they are approximated by means of local PCA (principal component analysis) of point patches around the individual data points. To determine the correct orientation of the normal, it is checked whether the determined normal vector points into the hemisphere in which the scanner position associated with the point is located.

Subsequently, planar structures are detected in the whole point cloud using a RANSAC algorithm by [13]. This yields a set of planes defined by their normal and distance to the origin. Additionally, the association between each plane  $p \in \mathcal{P}$  and the point set  $P_p$  that constitutes this plane is stored.

The known assignments of points to the individual scans will be used in the next section as an initial guess which points belong to a room. As one particular room may have been scanned from more than one scanner location, scans belonging to the same room are merged. This step is performed in a semi-automatic manner. A conservative measure whether two scans belong to the same room is used to give the user suggestions which scans should be merged. This measure takes into account the ratio of overlapping points between two scans to the total number of involved points. If the ratio is above a certain threshold, the rooms are suggested to be merged. The user may choose to accept these suggestions or make manual changes as desired. Note that the original scanner positions are stored for later usage, together with the information which scans were merged.

## 2.4 Probabilistic Point-To-Room Labeling

After the merging step described in Section 2.3, we are left with  $m$  different room labels and their associated points. In the following, we describe the task of point-to-room assignment as a probabilistic clustering problem [42]. Let  $\omega_j, j = 1, \dots, m$  denote the random variable that indicates a particular room, and furthermore, let  $x$  denote a random variable corresponding to a point in  $\mathbb{R}^3$ . Then the class-conditional densities  $p(x|\omega_j)$  describe the probability for observing a point  $x$  given the existence of the  $j$ th room. By that, we can describe the problem of segmenting a point cloud into single rooms by computing the conditional probability  $p(\omega_j|x)$  that an observed point  $x$  belongs to the  $j$ th room. Given a particular observation  $x_k$  from the point cloud scan, we can determine its probabilistic room assignment by computing

$$p(\omega_j|x_k) = \frac{p(x_k|\omega_j)p(\omega_j)}{\sum_{j'=1}^m p(x_k|\omega_{j'})p(\omega_{j'})}. \quad (2.1)$$

From the above equation it becomes clear that determining the clustering actually boils down to modeling and computing the class-conditional probabilities. Intuitively speaking, for a particular room  $p(x_k|\omega_j)$  should provide a high value if the observed measurement  $x_k$  lies inside it. Suppose we already knew a set of points  $\bar{X}_j$  that belongs to the room. A good choice to decide whether an observation lies inside would be to consider the visibility  $v_j(x_k) \in [0, 1]$  of this particular point from all the other points in  $\bar{X}_j$  that we know belong to the room (we will describe how to estimate the visibility in detail in Section 2.5).

In order to steer the impact of varying visibilities we formulate the class-conditional probability in

terms of a normal distribution over the visibilities:

$$p(x_k|\omega_j, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(1-v_j(x_k))^2}{2\sigma^2}\right). \quad (2.2)$$

Apparently, computing the visibility  $v_j(x_k)$  requires knowledge about the exact dimensions of the room, rendering the underlying clustering a chicken-egg problem. We alleviate by starting with an educated guess about the point-to-room assignments, which is given by the information which points originate from the individual (merged) scans. This prior knowledge is used to compute  $v_j(x_k)$  in the first iteration. We then iterate between the cluster assignment and visibility computation until our algorithm converges. In Algorithm 1, we show an overview of the iterative assignment process.

**Data :** A set of observations  $x_k \in \mathbb{R}^3, k = 1, \dots, n$

**Data :** A set of initial room assignments:  $\forall k \exists r_{init}(x_k) \in \{1, \dots, m\}$

**Data :** A set of planes  $p_i \in \mathcal{P}, i = 1, \dots, |\mathcal{P}|$

**Result :** A new point-to-room assignment:  $\forall k \exists r_{final}(x_k) \in \{1, \dots, m\}$

Initialize prior room probabilities according to number of measured points:

$$p(\omega_j) := \frac{|r_{init}(x_k)=j, k=1, \dots, n|}{\sum_{j'=1}^m |r_{init}(x_k)=j', k=1, \dots, n|};$$

Initialize  $r_{curr}(x_k) := r_{init}(x_k)$ ;

**while not converged do**

    Compute the set of points belonging to the  $j$ th room:  $\bar{X}_j := \{x | r_{curr}(x) = j\}$ ;

    Compute class-conditional probabilities:  $p(x_k|\omega_j, \sigma) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(1-v_j(x_k))^2}{2\sigma^2}\right)$ ;

    Compute probabilistic cluster assignments:  $p(\omega_j|x_k) = \frac{p(x_k|\omega_j)p(\omega_j)}{\sum_{j'=1}^m p(x_k|\omega_{j'})p(\omega_{j'})}$ ;

    Compute hard cluster assignments:  $r_{curr}(x_k) = \arg \max_{j=1, \dots, m} p(\omega_j|x_k)$ ;

    Renormalize room priors:  $p(\omega_j) := \frac{1}{n} \sum_{k=1}^n p(\omega_j|x_k)$

**end**

$r_{final}(x_k) := r_{curr}(x_k)$ ;

**Algorithm 1 :** Iterative room-to-point assignment.

In our tests we use a fixed  $\sigma$  throughout the iterations and for all rooms (see Section 2.9). Note however that it might improve results if for each room individual  $\sigma_j$  would be used. Estimation of  $\sigma_j$  and computation of the soft assignments could be carried out using an expectation-maximization (EM) algorithm [43].

## 2.5 Visibility Estimation

We construct the visibility function  $v_j(x_k)$  to estimate how unobstructed the line-of-sights between a point  $x_k$  to all (or a sampled subset of) points  $\bar{X}_j$  of the  $j$ th room are on average.

The intuition behind this functional is depicted in Figure 2.2. If we wish to estimate the visibility between the points  $x_1$  and  $x_2$ , we take into account the set of planes  $p_i$  that are intersected by the line segment  $s$  between the two points. For intersection testing, we consider the complete infinitely large plane. Because a plane  $p_i$  is only constituted by a subset of the measured points, the line segment  $s$

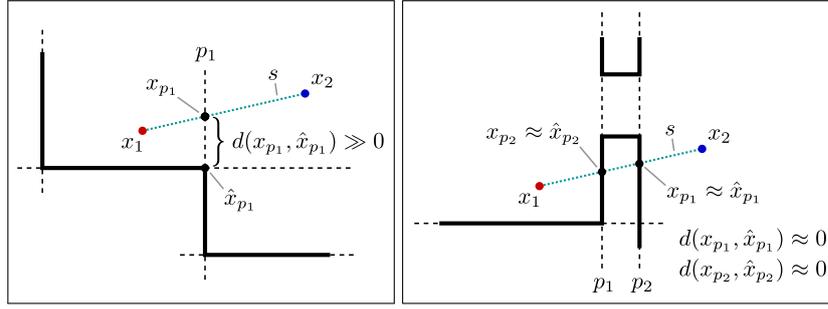


Figure 2.2: Different cases of visibility estimation. Left: Visibility between two points  $x_1, x_2$  located within a room. The line-of-sight is unobstructed since line segment  $s$  does not intersect any building structure. Right: If a wall lies between  $x_1, x_2$ ,  $s$  intersects multiple planes and visibility is estimated to be low.

passing through that plane shall only be considered to be highly obstructed if measured points exist in the vicinity of the intersection point. Let  $x_{p_i}$  be the intersection point of  $s$  with plane  $p_i$  and let  $\hat{x}_{p_i}$  be the point belonging to the plane  $p_i$  which is closest to  $x_{p_i}$ :

$$\hat{x}_{p_i} := \operatorname{argmin}_{x \in P_p} \|x - x_{p_i}\|_2. \quad (2.3)$$

The left half of Figure 2.2 shows the situation that  $s$  is located completely inside a single room. Note that, in this case,  $d(x_{p_1}, \hat{x}_{p_1})$  is large and thus the visibility regarding plane  $p_1$  is high. In the right half, the situation of  $s$  crossing through a wall is shown. In this case, the distances  $d(x_{p_1}, \hat{x}_{p_1})$  and  $d(x_{p_2}, \hat{x}_{p_2})$  to the nearest points in the planes are (almost) zero and thus the estimate of the visibility between the points is low.

Let  $p$  be a plane and  $x_p$  the intersection of  $s$  with  $p$ . Furthermore, for the set of planes  $\mathcal{P}$  and the set of all measured points  $X$ , let

$$v_p(x_1, x_2, p) : X \times X \times \mathcal{P} \rightarrow [0, 1] \quad (2.4)$$

be a function estimating the visibility between  $x_1$  and  $x_2$  with respect to a plane  $p$  following the above intuition which yields a high value iff the line-of-sight is unobstructed (see Section 2.9 for implementation details).

The visibility term over all planes  $\bar{\mathcal{P}}$  which  $s$  intersects is defined as

$$v_{\bar{\mathcal{P}}}(x_1, x_2) := \frac{1}{1 + \sum_{p \in \bar{\mathcal{P}}} (1 - v_p(x_1, x_2, p))}. \quad (2.5)$$

Intuitively speaking, the more planes with low visibility values are encountered along the line segment, the more the total visibility value decreases. Finally, we use Equation (2.5) to define the average visibility from a point  $x_k$  to a point set  $\bar{X}_j$  containing points labeled to belong to the  $j$ th room:

$$v_j(x_k) := \frac{\sum_{x \in \bar{X}_j} v_{\bar{\mathcal{P}}}(x_k, x)}{|\bar{X}_j|}. \quad (2.6)$$

## 2.6 Door Detection

The presence of a door causes the points from one single scan to belong to at least two different semantic rooms. As a consequence, when considering a point that changed the room assignment during our

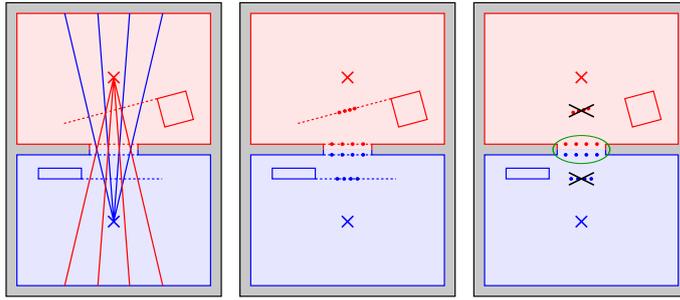


Figure 2.3: For door detection, the rays whose target point’s labeling has changed during segmentation are intersected with the planes belonging to the room in which the scanner is located (left); the intersection point sets on each plane are potential door candidates (middle); pairs of candidates that fulfill certain constraints are extracted (right).

probabilistic relabeling, the associated ray must have been shot through a connecting door. We exploit this observation for our door localization method. Note that this requires the doors to stand open during the scanning process. This assumption is justified as overlaps between scans are usually required for registration.

In preparation for door detection, we create a mapping of planes to rooms by iterating through the set of planes and counting the number of points on that plane that are labeled to be part of the individual rooms. If the number of points belonging to room  $r$  exceeds a threshold, the plane as well as the subset of points is assigned to  $r$ . This process may also assign one plane to multiple rooms but its constituting point set is split among the rooms sharing this plane.

The next step is the generation of candidate point sets that may constitute the coarse shape of one side of a door opening. Consider a room  $r$  consisting of scanner positions  $s_0, \dots, s_n$ . We are interested in those rays going from  $s_i$  through the position of the associated point  $p_i$  where the labeling of  $p_i$  has changed during the segmentation step. For each such ray, we check for intersections with each plane belonging to room  $r$  (Figure 2.3, left). This yields co-planar point sets which are stored together with the information which room  $s_i$  belongs to, which room the point  $p_i$  belongs to and which plane was intersected (Figure 2.3, middle).

After the candidates have been constructed, we split the point sets of each candidate into connected components. This is necessary if a room pair is connected through multiple doors lying in the same plane.

In a final step, pairs of candidates fulfilling certain constraints are extracted. The first constraint forces the normals of the two involved planes to point away from each other. The second constraint forces the distance between the planes to be within a given range. Lastly, the point set of one of the candidates is projected onto the plane of the other candidate and the intersection between the projected point sets is computed. If the ratio of the points lying in the intersection is high enough, the candidate pair is considered to be a door (encircled points in Figure 2.3, right).

## 2.7 Graph Generation

To construct a graph encoding the building’s room topology, each room is represented by a node. An edge is added for each detected door, connecting the two associated rooms. We enrich this representation with node and edge attributes. Each node is assigned a position which is set to the location of one of the involved scanners. Alternatively, the mean of the positions of all points belonging to the respective room could be used. However, this does not guarantee that the position lies within the room. Furthermore,

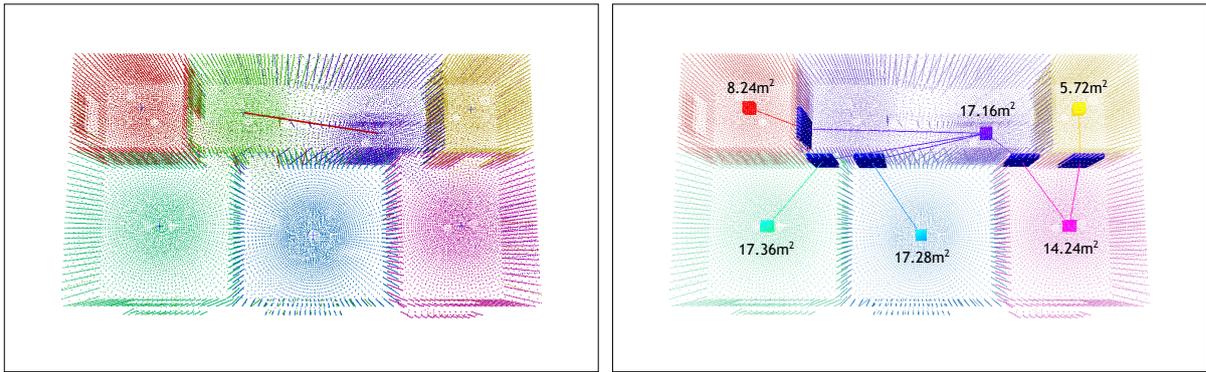


Figure 2.4: Results on synthetic data; the first picture also shows an automatic merging suggestion (red line between two scanner positions).

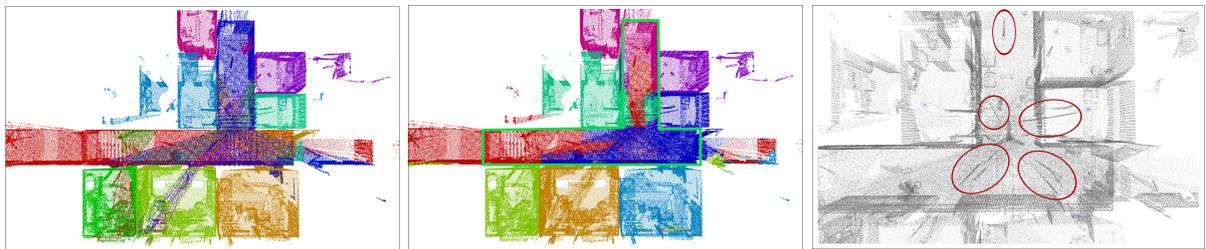


Figure 2.5: A failure case of our method due to a strongly non-convex room and planar artifacts in the scan data.

each node is assigned an approximate room area which is computed by projecting all points into a two-dimensional grid that is aligned with the x-y-plane. The approximate room area is then estimated by computing the area of cells which contain at least one point. For each door edge, we compute the bounding box of the previously generated intersection points. The edge is then assigned this bounding box which constitutes an approximation of the door’s position and shape.

## 2.8 Evaluation

We tested our method on synthetic as well as real-world datasets. Figure 2.4 shows a result on synthetic data. In the upper half, the initial labeling of scan positions is shown, together with an automatically computed suggestion which scan locations should be merged (indicated by the red line). The lower half shows the final topology graph as well as the estimated room areas. Figure 2.6 shows the results on real-world data measured in two different storeys of the same building. The upper-left pictures show the association of data points to the individual scans before merging. On the upper-right, the associations of points to merged scan positions are shown. The lower-left part shows the final labeling of the points after segmentation. The lower-right pictures show the extracted topology graph including the estimated room areas as well as the door locations and extents (blue boxes).

A failure case of our method is shown in Figure 2.5. The left part shows the initial association to the scanner positions. In the middle, the labeling after re-labeling is shown. Many points in the T-shaped corridor (marked with the green line) were erroneously labeled to be part of adjacent rooms. The explanation for this behaviour is twofold. Firstly, the assumption that most of the points belonging to a room can be seen from an arbitrary point within that room is violated for (strongly) non-convex rooms like the T-shaped corridor in the depicted dataset. Secondly, large planar artifacts in the input data (Figure

2.5, right) may lead to low visibility values in regions which should be free space.

Computation time ranges from almost instantaneous (for the lower-resolution synthetic dataset shown in Figure 2.4 consisting of 70000 points) to about 30 seconds (for larger datasets as shown in Figure 2.6 consisting of about 500000 and 600000 points, respectively).

## 2.9 Implementation Details

For fast determination whether measured points exist near a point  $x$  on a plane  $p$  and to enable efficient computations on the GPU, we pre-compute bitmaps for each plane, providing a discretized occupancy map. Each pixel may take a continuous value in  $[0, 1]$  and is initially set to 1 if at least one pixel lies within the boundary of that pixel, and to 0 otherwise. The bitmaps are subsequently smoothed using a box filter such that pixels in the vicinity of filled pixels (and thus points) are assigned values which are negatively proportional to the distance to the nearest points in the plane (the distances labeled  $d(x_{p_i}, \hat{x}_{p_i})$  in Figure 2.2).

Using the bitmaps, we approximate the visibility function  $v_p(x_1, x_2, p)$  of Section 2.5 for two points  $x_1, x_2$  and a plane  $p$ . Let  $\text{bitmap\_value}_p(x)$  be a function of the plane  $p$ 's bitmap value which determines the pixel value at the projection of  $x$  onto  $p$ . Then the visibility function is defined as

$$v_p(x_1, x_2, p) := 1 - \text{bitmap\_value}_p(x_p). \quad (2.7)$$

For our experiments, we set  $\sigma$  in Equation 2.2 to 0.05. For the plane detection, the minimum count of points needed to constitute a plane was set to a value between 200 and 2000, depending on the density of the dataset. The radius of the box filter for bitmap smoothing was chosen such that it corresponds to approximately 20 cm in point cloud coordinates. Visibility tests between pairs of points were computed on the GPU using OpenCL. The experiments were conducted on a 3.5 GHz Intel Core i7 CPU and a GeForce GTX 670 GPU with 4 GB of memory. To obtain synthetic data, we implemented a virtual laser scanner which simulates the scanning process within 3D CAD building models.

## 2.10 Conclusion and Future Work

We presented a method for the extraction of structural building descriptions using 3D point cloud scans as the input data. Our method was evaluated using synthetic and real-world data, showing the feasibility of our approach. In most cases, the algorithm produced satisfactory results, yielding useful semantic representations of buildings for applications like navigation in point clouds or structural queries.

The usage of different visibility functionals as well as an EM-formulation to determine the parameters of the used normal distributions separately for each room label could help to improve the robustness of our method. Performance in the presence of non-convex rooms could possibly be improved by using a measure for (potentially indirect) *reachability* between points instead of visibility tests. Also, limiting the tests to a more local scope may help to overcome the identified problems. The output of our approach could be further enriched by more attributes, for instance by applying methods for the analysis of the individual room point sets after segmentation.

## Acknowledgements

We would like to thank Henrik Leander Evers for the scans of Kronborg Castle, Denmark, and the Faculty of Architecture and Landscape Sciences of Leibniz University Hannover for providing the 3D building

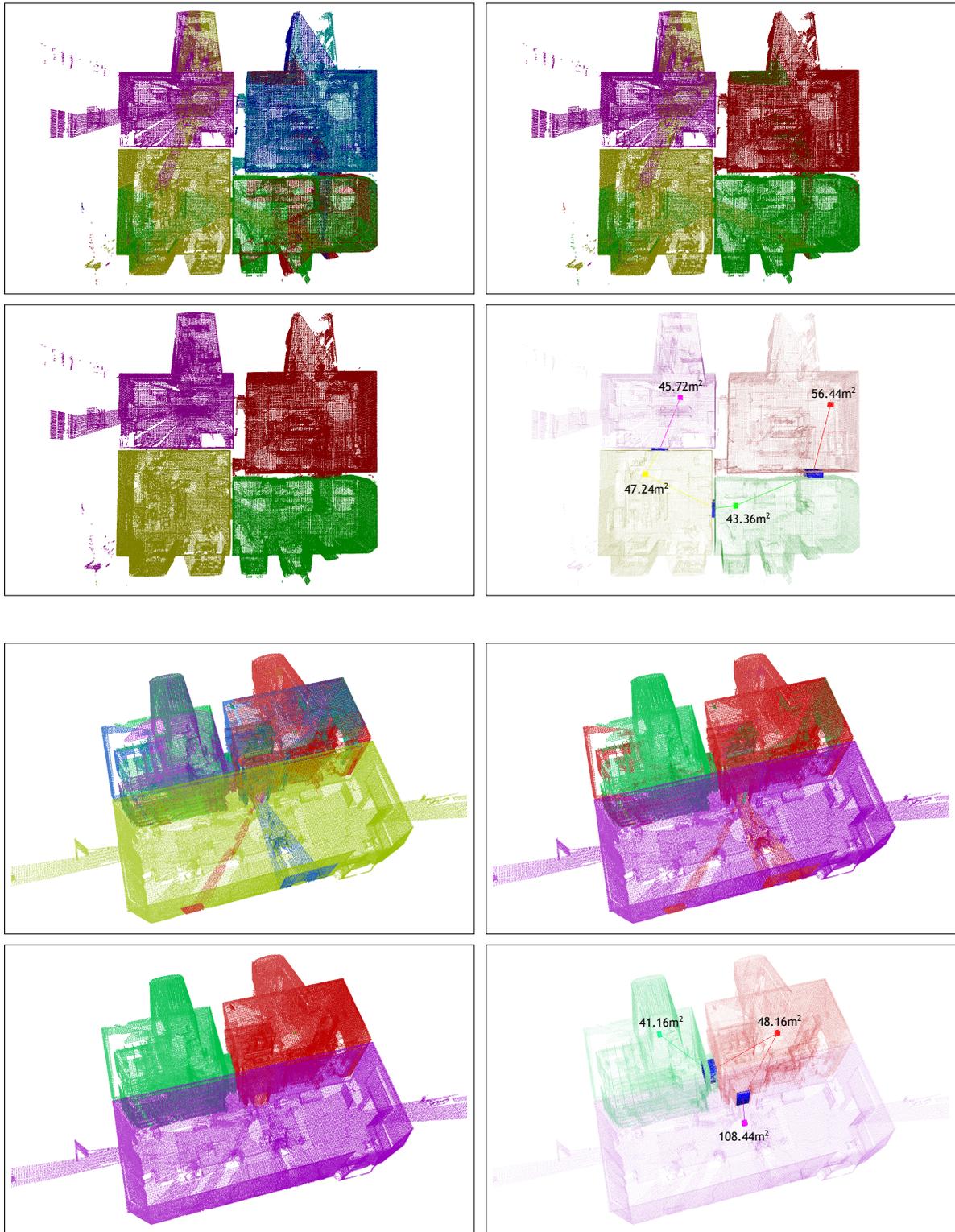


Figure 2.6: Results on two real-world datasets. The upper-left pictures show the room associations before merging of scans belonging to the same rooms. Upper-right: Associations after the merging step. Lower-left: The room assignments after segmentation. Lower-right: The extracted graphs including detected doors and annotations for the estimated room areas.

models that were used for generating the synthetic data. This work was partially funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600908 (DURAARK) 2013-2016.



# Towards the Extraction of Hierarchical Building Descriptions from 3D Indoor Scans

---

**Abstract.** We present a new method for the hierarchical decomposition of 3D indoor scans and the subsequent generation of an according hierarchical graph-based building descriptor. The hierarchy consists of four basic levels with according entities, building - storey - room - object. All entities are represented as attributed nodes in a graph and are linked to the upper level entity they are located in. Additionally, nodes of the same level are linked depending on their spatial and topological relationship. The hierarchical description enables easy navigation in the formerly unstructured data, measurement takings, as well as carrying out retrieval tasks that incorporate geometric, topological, and also functional building properties describing e.g. the designated use of single rooms according to the objects it contains. In contrast to previous methods which either focus on the segmentation into rooms or on the recognition of indoor objects, our holistic approach incorporates a rather large spectrum of entities on different semantic levels that are inherent to 3D building representations. In our evaluation we show the feasibility of our method for extraction of hierarchical building descriptions for various tasks using synthetic as well as real world data.

This chapter corresponds to the paper *Towards the Extraction of Hierarchical Building Descriptions from 3D Indoor Scans*, Sebastian Ochmann, Richard Vock, Raoul Wessel and Reinhard Klein, Eurographics Workshop on 3D Object Retrieval (3DOR) 2014.

## 3.1 Introduction

With the availability of fast and cheap 3D acquisition devices, digital point clouds have replaced analog methods of site measuring as the favored means for documenting the as-is state of buildings, especially regarding their interior. However, due to their inherent lack of structure, point clouds only serve as a starting point for tasks like retrofitting or renovation. The need for easy navigation, targeted (textual) search, manipulation, taking of measurements, and efficient rendering usually forces architects and construction companies to manually generate additional metadata information or even 3D Building Information Modeling (BIM) overlays of the point cloud. By that, the advantages that come along with the digital capturing devices are partially lost again as a large amount of manual postprocessing is still required.

The first step to a better usability of indoor measurements for architectural, engineering, and construc-

tion purposes is a proper segmentation into semantically meaningful parts including storeys and rooms. Several most recently introduced methods have taken on this task [44–46]. While the main goal in most of these publications is to reconstruct a per-room boundary representation of the captured building which facilitates rendering, such methods also partially improve navigating and manipulating the data, e.g. by hiding or removing certain rooms. Additionally, although not explicitly mentioned in the publications, such methods would theoretically allow for basic measurement tasks like e.g. determining the area or height of a room.

The second step to better usability of 3D indoor scans is to equip segmented rooms with an appropriate set of metadata that allows targeted retrieval of information pointing to the function of a room according to the objects it contains. For example, if a room contains objects like a sink or a shower, its designated function is likely to be that of a bathroom. The above-mentioned methods are restricted to geometric properties of the room; they are not able to derive function-related information. Apart from the field of robotics where recognition of shapes in indoor scenes is a very important task (see e.g. [38, 47]), identifying objects in rooms has been addressed by several more architecture and construction-related approaches lately, see e.g. [35, 36]. However, current methods for indoor analysis relying on point cloud representations treat the two semantic levels of segmentation into rooms and the interior of a room as being rather isolated from each other. Thereby, the full potential of an integrated building representation is not used, especially for tasks like targeted retrieval. In this work we try to overcome the aforementioned drawbacks. Our holistic hierarchical graph-based building description incorporates rather coarse levels including storeys and rooms, but it also captures the objects that are located in each room. By that, we are able to combine search queries relying on the topological arrangement of rooms fulfilling certain constraints (e.g. area) with queries that target the semantically high-level function of a room. In our evaluation we show several exemplary queries and retrieval results in non-trivial real-world data. Summarizing the contributions of our work, they are:

- A method for segmenting buildings into storeys & rooms
- Extraction of room neighborhood and room connectivity
- A holistic concept for a hierarchical building descriptor

## 3.2 Related Work

The analysis of architectural data is a wide field with different problem statements and applications. Our method combines methods for *Scene Segmentation* (Section 3.2.1), *Scene Understanding and Object Recognition* (Section 3.2.2), and *Topological Structuring* (Section 3.2.3) of a building.

### 3.2.1 Segmentation and Geometric Structure Extraction

Since point cloud data is inherently unstructured it is mandatory to structure and segment the scene before attempting to extract the structure of a building. [44] aim to extract a building model which is segmented on a per-room level. They construct a polyhedral model by projecting, clustering and finally intersecting wall candidates in a two-dimensional cell complex. In [48] the authors propose extracting and projecting floor and ceiling structures and finding a segmentation of the plane into cells in order to derive the building’s ground plan. A method combining 3D point cloud data and ground-level photographs is presented in [49]. The authors reconstruct a CSG (constructive solid geometry) representation and use an “Inverse-Constructive-Solid-Geometry” approach to determine the *observed empty* space. [46] present a method for creating 2D floor plans and 2.5D building models including a segmentation into rooms by

reducing the room-labeling to a *Graph-Cut* problem on a 2D Delaunay triangulation where triangles are labeled as *interior* or *exterior*. A similar approach is suggested in [50] for the reconstruction of permanent structures by using a *Graph-Cut* approach to decompose and label the space into interior and exterior. Probabilistic clustering of points based on their mutual visibility is used in [45] for obtaining a segmentation of the point cloud into rooms, followed by a detection of openings. In contrast to [48–50] our method does not aim to reconstruct purely geometric properties of the scanned scene, but to perform a hierarchical, semantically meaningful segmentation. While [44, 46] perform a segmentation into rooms, they do not determine connections between them. Also, our method works directly on the point cloud without reconstructing a mesh model. In comparison to [45] our method is able to cope with highly non-convex rooms, determines room neighborhoods and provides a more robust opening detection.

### 3.2.2 Object Recognition and Scene Understanding

In contrast to focusing on the coarse structure of rooms and their connectivity a lot of research went into the recognition of objects and their relations within a room. Regarding the overwhelming amount of publications targeting this task especially in the field of robotics, we restrict ourselves on approaches that focus on architecture and construction using static scenes. [47] propose a hybrid approach to understand indoor scenes by using geometric, as well as surface models in order to segment the scene into objects. A method based on an oversegmentation of the scene based on the smoothness and continuity of surfaces with subsequent labeling based on Markov Random Fields is described in [38]. Template learning using the large amount of freely available synthetic 3D models to enable object recognition is described in [51]. In [35], the authors use learned graph-based models for objects in a *learning phase* by matching stable primitive parts across measurements, and then try to fit these models in a *recognition phase*. A similar approach that tries to avoid the problem of prior segmentation is presented in [36]. The authors propose a *Search-Classify* method where an oversegmented scene is iteratively simplified while simultaneously maximizing classification likelihood for previously computed feature descriptors. [52] propose an interactive method for the segmentation of indoor scenes into semantic entities (e.g. furniture elements) from RGBD images. Having obtained semantic labels in each image using a Conditional Random Field model, they reconstruct the scene with objects from a model database. While these methods provide tools for the segmentation and understanding of scenes on a room level, they do not take into account the overall building topology as proposed in this paper.

### 3.2.3 Topology Extraction

There is a wide variety of research work that focuses on extraction and applications of a building’s topology (i.e. the structure, connectivity and accessibility of rooms), although almost all of the proposed methods work with representations different from 3D point clouds. [40] use image segmentation and OCR (optical character recognition) techniques to extract room structure and semantics in 2D floor plans. In [53] the authors enhance this approach by adding a semantic analysis based on SURF (speeded up robust features) which yields a graph representation of rooms used for retrieval of room configurations. Another method based on image representations of floor-plans is proposed in [54], which is based on a recursive decomposition of the image to retrieve nearly convex regions. [30] extract topological information from low-level 3D CAD representations of buildings by comparing 2D plans extracted at different cut-heights for each storey. In [39], a building’s topology is derived from high-level BIM models by analyzing certain entity constellations in the model. In contrast to these approaches, our method does not rely on the availability of 2D or 3D models of the building but works purely on point cloud scans.

### 3.3 Method Overview

This section provides a concise overview of our approach for generating a hierarchical building descriptor from indoor point cloud scans. Starting with registered point cloud scans of a building, the steps of our approach are as follows.

- Planar structures are detected. This yields an initial, coarse structuring of the point cloud.
- Using the given initial assignment of points to scanners, a semi-automatic method for segmenting the point cloud into rooms is performed (Section 3.4). This assigns each point to exactly one room and resolves ambiguities in regions where multiple scans overlap.
- Using this segmentation, the neighborhood relation of rooms (Section 3.5.1) and their connectivity (Section 3.5.2) is determined and encoded in a graph structure.
- The point subset of each room is further segmented into objects like furniture elements (Section 3.6.1). For each object, shape descriptors are computed and connected to the respective room node (Section 3.6.2).
- After the hierarchical descriptor for a building has been built, structural queries may be performed by means of matching attributed query graphs representing room and object constellations (Section 3.7).

Note that the presented processing chain is very modular in that different algorithms may be plugged in for performing the individual steps.

### 3.4 Segmentation Into Rooms

The first step of our approach is the segmentation of the point cloud into subsets corresponding to rooms. The approach uses a coarse initial guess for point-to-room assignments which is subsequently refined. We initially generate a preliminary room label  $r'_i$  for each scanner location  $s_i$  and assign all points belonging to  $s_i$  the label  $r'_i$ . Because a room may have been scanned from multiple positions, these scan positions must be merged in the first phase. In the scope of this paper, this step is done manually using an intuitive graphical interface. In the second phase, an automatic relabeling procedure is carried out which is based on the following assumption. Let  $x$  be a point which belongs to room  $r$  (even though the initial room assignment of  $x$  may be different from  $r$ ). We hypothesize that most points which are (directly or “almost” directly) *visible* from the position of  $x$  belong to the same room as  $x$  and that those points tend to be labeled correctly because only a relatively small fraction of points is located in regions where scans overlap.

The relabeling procedure is formulated as a diffusion process in which the “transfer” of point-to-room labels between points is governed by the mutual *visibility* between point pairs. The rationale behind this formulation is that it not only allows transfers between points which are *directly* visible but also allows *indirect* connections via a few ray “bounces”. The importance of this is that occlusion effects (either due to non-convex rooms or clutter) are mitigated. We model the transfer probability between points as a Markov chain with the row-stochastic transition matrix

$$T := \begin{pmatrix} \frac{v_{11}}{k_1} & \dots & \frac{v_{1n}}{k_1} \\ \vdots & \ddots & \vdots \\ \frac{v_{n1}}{k_n} & \dots & \frac{v_{nn}}{k_n} \end{pmatrix}, \quad (3.1)$$

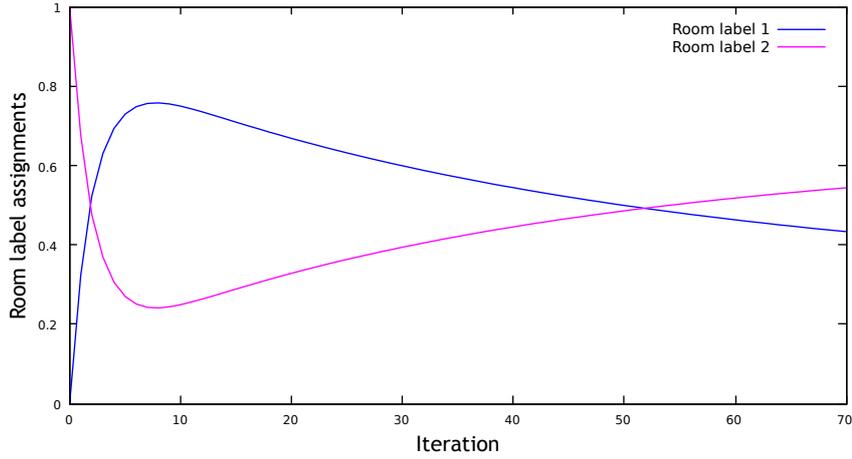


Figure 3.1: Label diffusion for a point which is initially labeled incorrectly (the correct label is 1). The correct label is assumed after a few iterations, however it may once again become incorrect with increasing number of iterations.

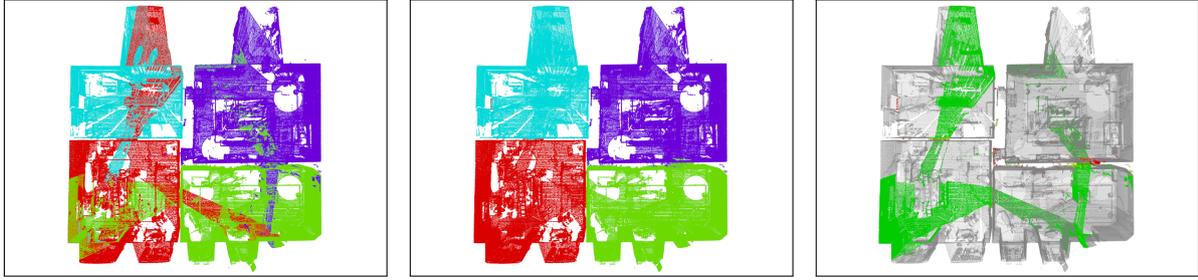


Figure 3.2: Room labeling before (left) and after (middle) relabeling. Right: Points which were relabeled are highlighted.

where  $n$  is the number of points,  $v_{ij} = 1$  iff  $x_j$  is visible from  $x_i$  and 0 otherwise, and  $k_i = \sum_{j=1}^n v_{ij}$ . In addition, we define  $v_{ii} = 1$  for all  $i$ . The value  $T_{i,j}^k$  yields the probability of “moving” from point  $x_i$  to  $x_j$  via line-of-sight rays in exactly  $k$  steps. In addition, let an initial (hard) point-to-room assignment be given as the label matrix

$$L := \begin{pmatrix} l_{11} & \cdots & l_{1m} \\ \vdots & \ddots & \vdots \\ l_{n1} & \cdots & l_{nm} \end{pmatrix}, \quad (3.2)$$

where  $m$  is the number of rooms,  $l_{ij}$  equals 1 iff point  $x_i$  is associated with room label  $j$  and 0 otherwise. The product

$$L_k := T^k L \quad (3.3)$$

yields a new label distribution which takes into account the average labels of points encountered after  $k$  bounces of line-of-sight rays between points. Consider the progression

$$L_k, \quad k = 1, \dots, \quad (3.4)$$

whose limit for  $k \rightarrow \infty$  yields the distribution of point-to-room assignments after an infinite number of ray bounces. Figure 3.1 shows the labeling progression for a particular point. Two effects can be observed in the plot. Firstly, it is sometimes necessary to run a few iterations until the point assumes its correct

labeling (in this case, three iterations were necessary). The main reason for this are occlusions within the building such that the majority of the room the point belongs to may only be seen *indirectly* after a few ray bounces. Secondly, the label assignments in the limit of the progression may once again become incorrect as the diffusion spreads throughout building. In the extreme case, if there exists a path between all pairs of points (and thus  $T$  is an irreducible transition matrix), all points will assume the same label distribution in the limit. As a compromise between allowing multiple iterations and avoiding the limit case, we decide for a room label  $room(x_i)$  for point  $x_i$  by integrating over the first  $N$  iterations for each label:

$$room(x_i) := \operatorname{argmax}_{j \in \{1, \dots, m\}} \sum_{k=1}^N (L_k)_{i,j}. \quad (3.5)$$

In our experiments, a value of  $N = 10$  yielded satisfactory results. For the practical implementation, a set of scans is initially given, together with the respective scan origins. The scans are assumed to be registered in a common coordinate system (this step is usually done by the scanner software). As a prerequisite for the relabeling, point normals are estimated by means of local PCA (principal component analysis) of point patches and planar structures are detected using a RANSAC (random sample consensus) implementation by Schnabel et al. [13]. Each detected plane is also assigned the set of points which constitute it, as well as an occupancy bitmap which is used for performing fast, approximate intersection tests with the building structure. Each bitmap pixel may take a continuous value in  $[0, 1]$  and is initially set to 1 iff the projection of at least one point lies within the boundary of that pixel and 0 otherwise. The bitmaps are subsequently smoothed using a box filter in order to fill small holes.

For approximating  $L_k$ , a stochastic, iterative ray voting scheme is used. Instead of averaging the labels of *all* points that are visible from  $x$ ,  $k$  sample rays are generated whose directions are randomly sampled on the hemisphere around the normal of  $x$ . For each sample ray  $r_i$ , the nearest intersection  $p_{isect}$  with the set of planes is determined (taking into account the respective occupancy bitmaps). If the nearest intersection is with a plane whose normal points into the same hemisphere as the ray direction, the sample is not counted. In each iteration, the new label soft assignment vector  $l(x)$  of  $x$  is determined by averaging the label vectors of points located within the area of each occupancy bitmap pixel intersected by a ray as well as averaging over all sample rays:

$$l_{new}(x) := \frac{1}{h+1} \left( l(x) + \sum_{i=1}^h \left( \frac{1}{b_i} \sum_{j=1}^{b_i} l(y_{i,j}) \right) \right), \quad (3.6)$$

where  $h$  is the number of sample rays which intersected some plane,  $b_i$  is the number of points within the bitmap pixel  $b$  hit by ray  $r_i$ , and  $y_{i,j}$  is the  $j$ th point within  $b$ . Note that if not a single sample ray successfully intersected a plane, the definition yields  $l_{new}(x) = l(x)$ . Figure 3.2 shows part of the point cloud before and after relabeling as well as an overview of which parts of the point cloud have been relabeled. For each room  $r$ , the approximate room area is determined which may later be used as a constraint when querying for room constellations. Let  $P_h$  be the set of points belonging to (approximately) horizontal planes, and let  $P_r$  be the set of points of room  $r$ . The point set  $P_r \cap P_h$  is projected into a regular grid in the x-y-plane with cell size  $c$ . The number of cells  $n$  containing at least one projected point yields the approximate room area  $area_r := nc^2$ . We also estimate the floor elevation of each room which is later used for aligning the  $z$  position of object descriptors with the floor elevation. Let  $\mathcal{P} = \{p_1, \dots, p_n\}$  be the set of (approximately) horizontal planes whose normals are pointing upwards, let  $P_{p_i}$  be the respective

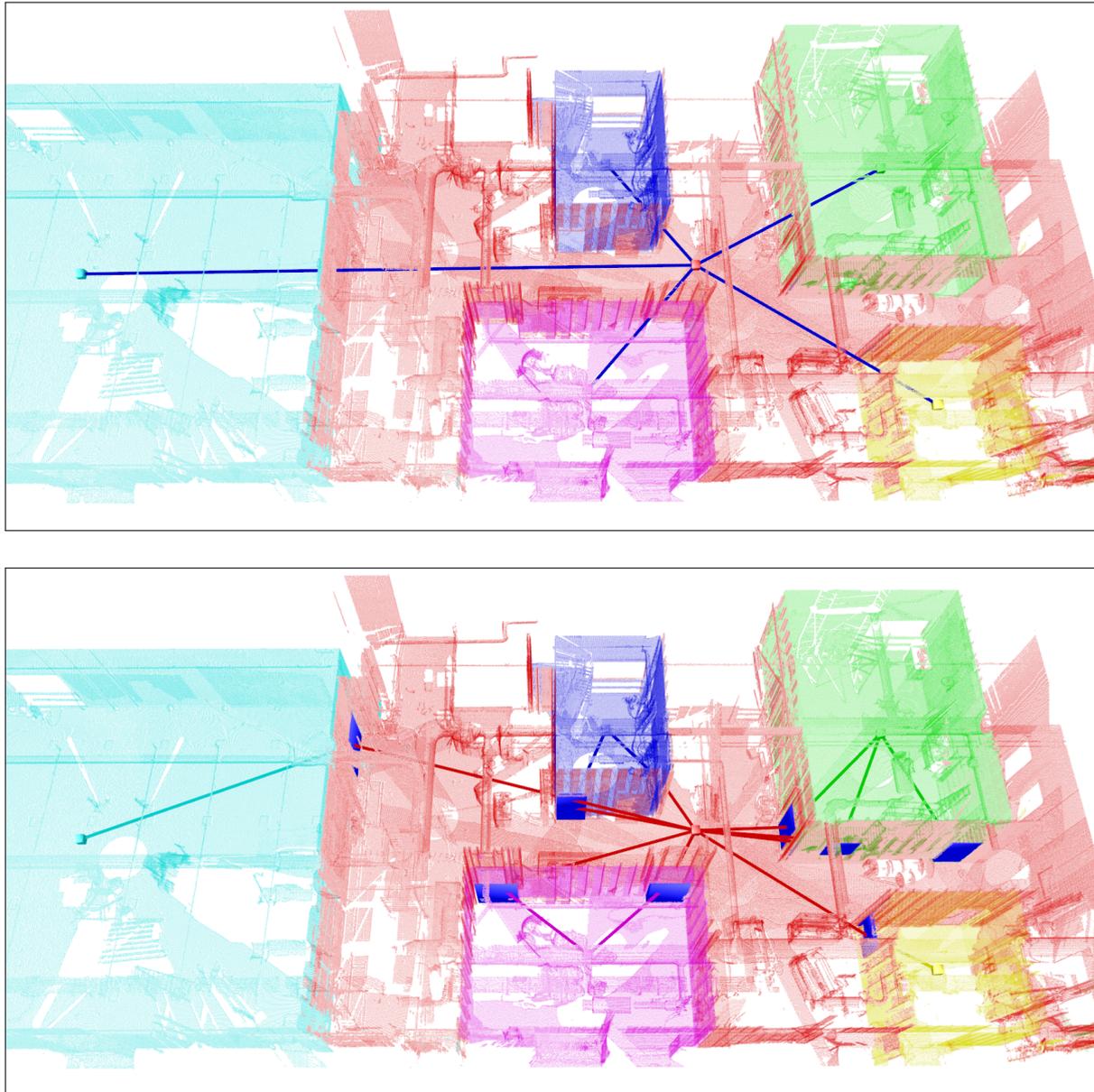


Figure 3.3: Room neighbors (left) and connections extracted from a real-world dataset.

point sets belonging to plane  $p_i$ , and  $P_r$  as above. We define the best floor candidate plane as

$$p_{floor} := \operatorname{argmax}_{i \in \{1, \dots, n\}} |P_r \cap P_{p_i}|. \quad (3.7)$$

The mean  $z$  position of the point set  $P_r \cap P_{p_{floor}}$  is chosen as the floor elevation  $elev_r$  of room  $r$ . Having estimated each room's floor elevation, a simple binning procedure is used to group rooms into storeys. We start with an empty set of bins. For each room  $r$ , it is checked whether there exists a bin  $b$  in which all rooms have a floor elevation which is close enough to  $elev_r$  with respect to a threshold. If  $b$  exists, the room is inserted into  $b$ ; otherwise a new bin containing  $r$  is created. The result up to this point is a graph in which a *storey node* is inserted for each group of rooms which share approximately the same floor elevation, each connected to a (root) *building node*. For each room label, a *room node* is inserted, connected to the respective storey node, and assigned its point subset, area, and floor elevation.

### 3.5 Detection of Room Neighbors and Connections

This Section describes our method for determining relations between rooms. Examples for the extraction of the room neighborhood relation and room connectivity in a real-world dataset are shown in Figure 3.3.

#### 3.5.1 Room Neighborhood

For determining which rooms are adjacent and shall thus be connected by a *room neighbor edge*, we assume that two rooms are adjacent iff they share at least one wall. The task is to find those walls together with the information which pairs of rooms are separated by them. The top-left image in Figure 3.4 shows the room labeling after the relabeling step. The process starts with the extraction of plane pairs which are close enough regarding a given threshold and whose normals approximately point away from each other (top-right). A particular plane might belong to more than one room (i.e. the point set associated with the plane contains points associated with different rooms; bottom-left). Therefore the point set of each plane is segmented into point sets belonging to the individual rooms. For each candidate pair of point sets  $A$  and  $B$ , the associated points are projected into a common plane (either of the involved planes may be used) and an approximate intersection point set is computed by testing for each point  $a \in A$  whether there exists a point  $b \in B$  with  $\|a - b\|_2 < thresh$  (and vice versa). If the number of points in the approximate intersection set exceeds a given threshold, an edge is inserted and attributed with the information which plane primitive pair was involved. Note that in practice, more than two detected planes may constitute a wall due to noise and clutter such that a binning approach is used and a set of planes is assigned to the room neighbor edge; technical details have been omitted here for brevity. The bottom-right image shows

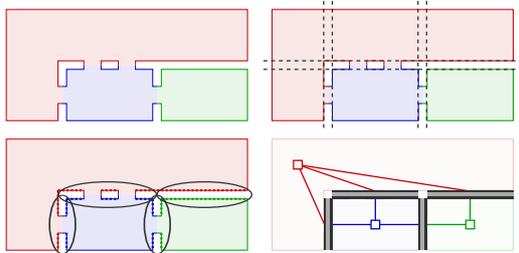


Figure 3.4: Steps of the room neighborhood graph generation.

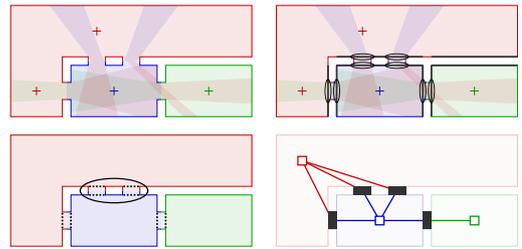


Figure 3.5: Steps of the room connectivity graph generation.

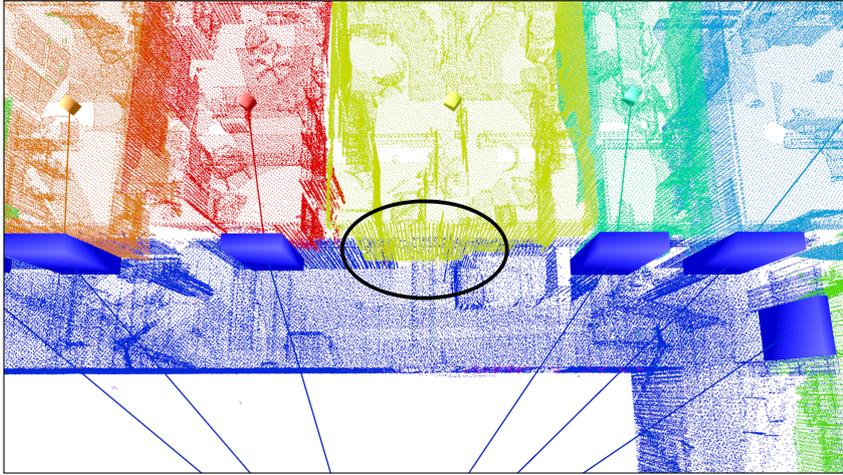


Figure 3.6: A failure case of the door detection.

the resulting graph for the example.

### 3.5.2 Room Connectivity

The opening detection is based on the observation that certain rays that were cast during the scanning process indicate the existence of openings in the building’s structure. In particular, if the origin of a ray (scanner position) is located in another room than the measured point, there must be an opening located inbetween the two points. This set of rays is extracted by considering the point-to-room labels *before* and *after* the relabeling step. If the label of a point  $x$  was *changed* by the relabeling procedure, the ray which captured  $x$  is assumed to pass through an opening. The top-left image in Figure 3.5 shows the regions observed by the individual scanners as well as their positions. Note that the “red” room consists of multiple scans which have been merged, however the original scanner positions are used for ray generation. Let  $r$  be a (laser) ray which measured point  $x$  and let  $room_{old}(x)$  and  $room_{new}(x)$  denote the room label of  $x$  before and after the relabeling step. If  $room_{old}(x) \neq room_{new}(x)$  and there exists a room neighbor edge  $e = (room_{old}(x), room_{new}(x))$ , the intersection of  $r$  with all planes associated with  $e$  is computed. The top-right image shows the positions where these intersections are located in the example. Because a pair of rooms may in general be connected by more than one door (bottom-left), each point set is split into connected components using a point distance threshold. Finally, the extracted point sets are used for approximating the positions and sizes of the openings. The resulting openings are shown in the bottom-right image. Note that the aforementioned methods do not make assumptions on the orientation of planes and thus not only allow horizontal connections (e.g. doors) but also vertical connections (e.g. stairways) as long as overlaps between the respective scans exist.

A failure case of the door detection is shown in Figure 3.6. The reasons for the missing door are twofold. Firstly, the door was closed when the scan inside of the “yellow” room was performed. Because the algorithm only requires rays to be shot through an opening from one side, it would normally be able to cope with this situation. However, the other side of the opening was only scanned from sharp angles and thus almost no rays were shot through the door.

## 3.6 Assignment of Objects to Rooms

We now extend the graph descriptor by information about objects contained within the individual rooms. By assigning each extracted object a shape descriptor, we enable example-based queries for objects in combination with queries for topological constellations of rooms.

### 3.6.1 Extraction of Objects

We now separate objects from broader building structures. Let  $R$  be the set of points associated with room  $r$  and let  $P = \bigcup_{p \in \mathcal{P}} p$  be the set of points associated with detected planes. Points belonging to planes are removed from the room's set of points,  $R' := R \setminus P$ . Subsequently, connected components in  $R'$  are determined, that is two points are assumed to belong to the same component iff their distance is below a threshold. The point cloud is filtered beforehand by considering the mean  $\mu$  and standard deviation  $\sigma$  of the distance between a point and its  $k$  nearest neighbors and filtering out points for which the average distance to its neighbors lies above  $\mu + \alpha\sigma$  where  $\alpha$  is a user-defined constant (see [55]). The obtained segmentation tends to over-segment the point cloud. However, the object descriptor as described in the next Section also takes into account combinations of nearby segments and thus mitigates this problem.

### 3.6.2 Object Shape Descriptors

A global shape descriptor is constructed for each of the extracted object components. Note that we restrict ourselves to a relatively simple object descriptor in the scope of this paper, but it may easily be exchanged with other kinds of descriptors. The three-dimensional space around an extracted point subset is segmented into  $\Theta$  horizontal slices,  $\Phi$  concentric shells and  $\Psi$  sectors. Each descriptor is built around a local, vertical axis whose x-y-position is centered at the mean position of all points associated with the segment. The z position of the bottom end of the descriptor is set to the previously determined floor elevation  $elev_r$  of room  $r$  in which the object is located. The (unnormalized) descriptor  $D''_o(\theta, \phi, \psi)$  for an object  $o$  is defined as

$$D''_o(\theta, \phi, \psi) := \sum_{x \in (\theta, \phi, \psi)} \frac{1}{2\phi}, \quad (3.8)$$

where  $x \in (\theta, \phi, \psi)$  are the points located within the respective bin defined by slice  $\theta$ , shell  $\phi$ , and sector  $\psi$ . The normalization factor within the sum accounts for the increase of volume of shells located farther away from the center. For values of  $\theta$  outside of the range  $[0, \Theta - 1]$  or  $\phi$  outside of the range  $[0, \Phi - 1]$ , the respective parameters are set to the nearest valid value. Values of  $\psi$  outside of the range  $[0, \Psi - 1]$  are repeated periodically (modulo  $\Psi$ ). The bin values are subsequently smoothed using a box filter according to

$$D'_o(\theta, \phi, \psi) := \sum_{\alpha, \beta, \gamma \in \{-1, 0, 1\}} D''_o(\theta + \alpha, \phi + \beta, \psi + \gamma). \quad (3.9)$$

The descriptor is normalized according to

$$D_o(\theta, \phi, \psi) := \frac{D'_o(\theta, \phi, \psi)}{\sum_{\theta', \phi', \psi'} D'_o(\theta', \phi', \psi')}. \quad (3.10)$$

For comparing two object descriptors  $D_q$  and  $D_o$ , a symmetric version of the  $\chi^2$  distance is used. In order to enable rotation invariance along the z-axis, all possible shifts of the sectors of one of the descriptors

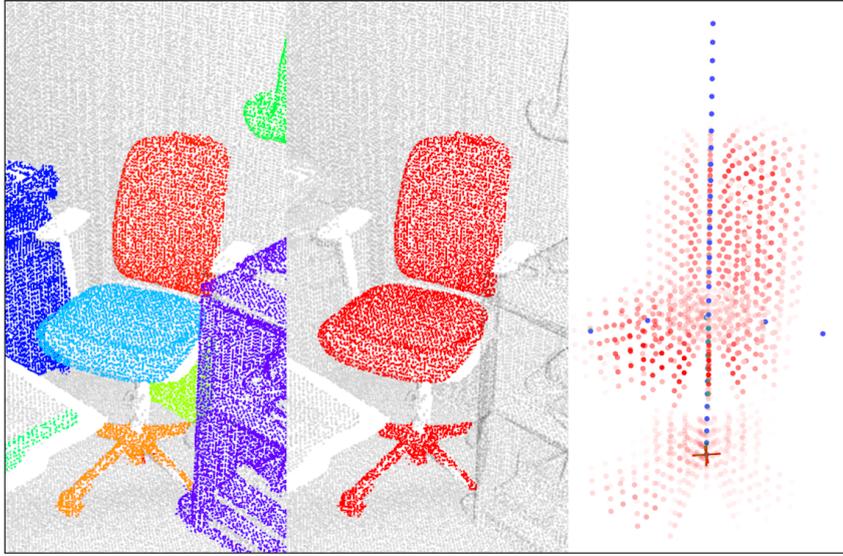


Figure 3.7: Left: A chair consisting of multiple segments. Middle: One of the segment combinations added to the graph descriptor. Right: Visualization of descriptor coefficients.

are evaluated which yields the descriptor distance

$$d(D_q, D_o) := \operatorname{argmin}_{\delta \in \{0, \dots, \Psi-1\}} \left( \sum_{\theta, \phi, \psi} \frac{(D_q(\theta, \phi, \psi) - D_o(\theta, \phi, \psi + \delta))^2}{D_q(\theta, \phi, \psi) + D_o(\theta, \phi, \psi + \delta)} \right). \quad (3.11)$$

Note that summands with a zero denominator are set to zero. As pointed out in the previous section, one object may be separated into multiple components; see Figure 3.7 for an example. To allow matching of complete query objects against objects which are comprised of multiple segments, additional object nodes consisting of combinations of *up to* three nearby segments are added to the graph (this approach is loosely based on ideas from [36]). For each combination of segments, an *object node* is added to the graph, assigned its shape descriptor and connected to the respective room node.

### 3.7 Graph Matching, Pruning and Scoring

This Section describes how combined topological and example-based object queries are performed. A query is given in form of an attributed graph  $G = (V, E)$  consisting of a set of nodes  $V$  (of type storey, room, or object, possibly attributed with room areas or object descriptors), and a set of edges  $E$  (horizontal/vertical opening, or object edge). For determining subgraph matches, *subgraph monomorphisms* are sought (using the VF2 implementation of the Boost Graph Library). In general, the types of nodes and edges must be the same in order to match. In addition, room nodes in the query graph may be attributed with a minimum and/or maximum area which must match the target room's area, if given. When matching two object nodes, the *dissimilarity* of the associated shape descriptors is determined. Other kinds of hard and soft constraints for nodes and edges are possible, however for the experiments in this paper we restrict ourselves to the aforementioned constraints. Apart from binary compatibility decisions for nodes and edges, we used the object dissimilarities for scoring each match. For a match  $m$ , let  $q_i$  be the object nodes in the query graph and  $t_i$  the respective matching nodes in the target graph, then the score is defined as  $\operatorname{score}(m) := -\sum_i d(D_{q_i}, D_{t_i})$ . Because we include pairs and triples of object

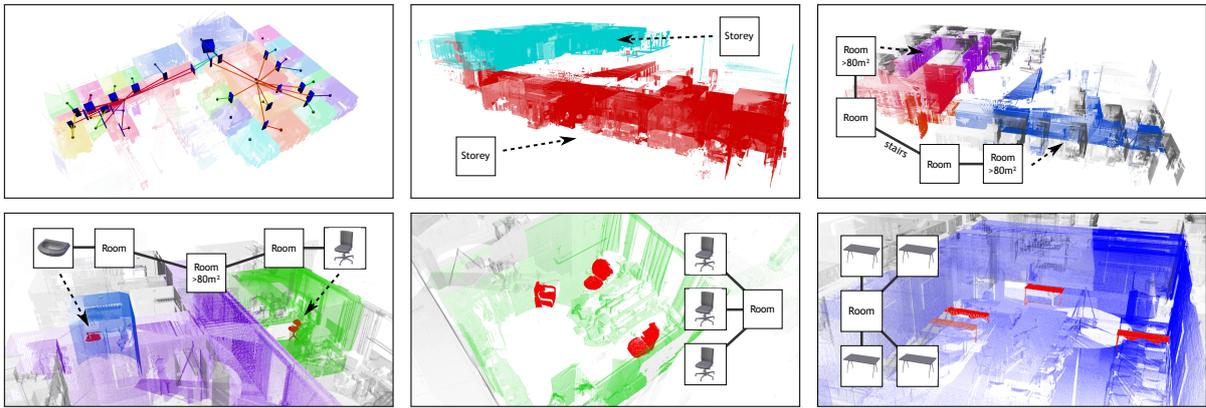


Figure 3.8: Results of subgraph queries. Top-left: The room connections extracted from the dataset. Our approach not only allows queries for topological constellations (top middle, top right), but also for combined queries including objects (bottom row).

segments as object nodes in the graph, an additional pruning step is performed to avoid matches in which a particular segment is used multiple times.

### 3.8 Evaluation

In this section, we present results on part of a real-world laser scan of the *Risløkka Trafikkstasjon* (Oslo). The dataset consists of 33 scans which were merged to 28 room labels. The cloud was coarsely cropped in order to remove some surrounding clutter like trees, and subsampled such that there is at most one point within a voxel of  $1 \text{ cm}^3$ , resulting in a total number of 25.7 million points. Apart from intuitively defining arbitrary attributed room and storey configurations, our approach allows to attach certain objects to the room nodes that are incorporated in the query. To this end, the user may either select an object that was identified during our segmentation process, or he may also include external mesh models. In the latter case, the mesh model is uniformly sampled upon loading in order to obtain a point cloud for which a shape descriptor is computed as described in Section 3.6.2. Figure 3.8 shows exemplary results of subgraph queries on the dataset. Mesh models were used as the input representation for the query objects. As can be seen from our preliminary results, our method on the one hand allows for improved navigation of the formerly unstructured point cloud data by segmenting it into storeys and rooms. On the other hand, it also allows to get a hint on the intended usage of single rooms by identifying function-related fixtures, see e.g. the detected basin, which can be used to constitute the base for further generation of high-level textual room attribution.

### 3.9 Conclusion

We presented a holistic approach for the extraction of hierarchical building descriptors purely from 3D indoor point clouds which incorporate topological and functional properties of a building. The outlined processing chain is very modular such that individual parts may easily be exchanged and improved. The current chain comprises a segmentation of the point cloud into storeys and rooms which is subsequently used for determining room neighbors and connections. A segmentation of the rooms' point subsets into objects contained within each room further enriches the graph structure with object shape descriptors. After the building descriptor has been built, combined queries for room constellations and contained

objects may be performed. The current implementation allows to include constraints like room area, example-based object shape and connection type in the queries. Our approach has been demonstrated on a large-scale real-world dataset. In the future we want to investigate the relation between our diffusion-based segmentation and the approach recently suggested by Mura et al. [44] who used a GPS embedding suggested by Rustamov [56] which is also closely related to a diffusion process on a mesh.

## **Acknowledgements**

We would like to thank Dag Fjeld Edvardsen from Catenda, Norway, for providing scans of Risløkka trafikkstasjon, Oslo as well as Henrik Leander Evers and Martin Tamke for scans of the Technical University of Denmark, and scans of Kronborg Castle, Denmark. This work was partially funded by the German Research Foundation (DFG) under grant KL 1142/9-1 (Mapping on Demand), and by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600908 (DURAARK) 2013-2016.

## Errata

This is the errata for the paper “Towards the Extraction of Hierarchical Building Descriptions from 3D Indoor Scans” presented in Chapter 3.

In Equation 3.11, the descriptor distance is mistakenly computed using argmin instead of min. The corrected definition thus is

$$d(D_q, D_o) := \min_{\delta \in \{0, \dots, \Psi-1\}} \left( \sum_{\theta, \phi, \psi} \frac{(D_q(\theta, \phi, \psi) - D_o(\theta, \phi, \psi + \delta))^2}{D_q(\theta, \phi, \psi) + D_o(\theta, \phi, \psi + \delta)} \right).$$

# Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds

---

**Abstract.** We present an automatic approach for the reconstruction of parametric 3D building models from indoor point clouds. While recently developed methods in this domain focus on mere local surface reconstructions which enable e.g. efficient visualization, our approach aims for a volumetric, parametric building model that additionally incorporates contextual information such as global wall connectivity. In contrast to pure surface reconstructions, our representation thereby allows more comprehensive use: First, it enables efficient high-level editing operations in terms of e.g. wall removal or room reshaping which always result in a topologically consistent representation. Second, it enables easy taking of measurements like e.g. determining wall thickness or room areas. These properties render our reconstruction method especially beneficial to architects or engineers for planning renovation or retrofitting. Following the idea of previous approaches, the reconstruction task is cast as a labeling problem which is solved by an energy minimization. This global optimization approach allows for the reconstruction of wall elements shared between rooms while simultaneously maintaining plausible connectivity between all wall elements. An automatic prior segmentation of the point cloud into rooms and outside area filters large-scale outliers and yields priors for the definition of labeling costs for the energy minimization. The reconstructed model is further enriched by detected doors and windows. We demonstrate the applicability and reconstruction power of our new approach on a variety of complex real-world datasets requiring little or no parameter adjustment.

This chapter corresponds to the paper *Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds*, Sebastian Ochmann, Richard Vock, Raoul Wessel and Reinhard Klein, Computers & Graphics special issue on CAD/Graphics 2015.

## 4.1 Introduction

Digital 3D building models are increasingly used for diverse tasks in architecture and design such as construction planning, visualization, navigation, simulation, facility management, renovation, and retrofitting. Especially for legacy buildings, suitable models are usually not available from the initial planning. Point cloud measurements are often used as a starting point for generating 3D models in architectural software. But despite fast scanning devices and modern software, the generation of models from scratch still are largely manual and time-consuming tasks which makes automatic reconstruction methods highly desirable.

Reconstruction of indoor environments poses specific challenges due to complex room layouts, clutter

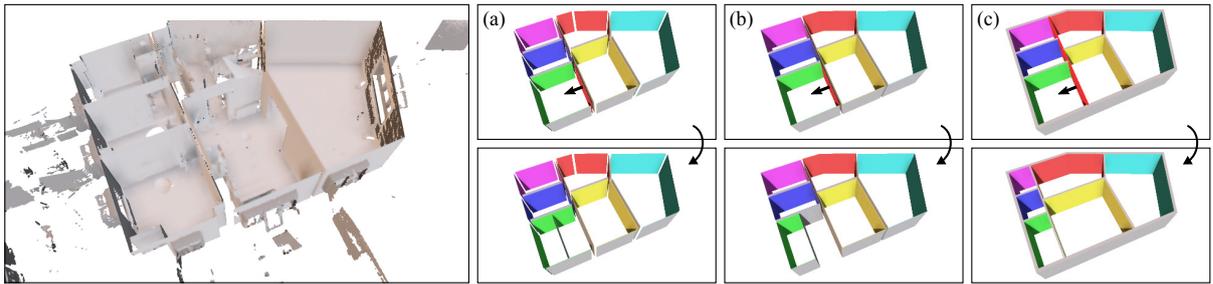


Figure 4.1: Schematic of editing capabilities of different kinds of reconstructions. The input point cloud is shown on the left. The remaining columns exemplify editing operations, i.e. elements are moved in the directions of the arrows. Surface representations without (column (a)) or with (column (b)) connectivity information do not allow intuitive editing on the level of wall elements. Our reconstruction (column (c)) maintains room topology and global wall connectivity.

and occlusions. Furthermore, planning and maintenance tasks often require models which give deeper insight into a building’s structure on the level of building elements such as walls, and their relations like wall connectivity. This enables high-level editing for prototyping planned changes and simulations requiring information like room neighborhood or wall thickness. While previous reconstruction methods are able to faithfully recover partially observed surfaces from indoor point clouds and generate accurate boundary representations in the form of mesh models, a plausible decomposition into parametric, globally interrelated, volumetric building elements yet remained an open challenge. Existing approaches either represent walls, floors and ceilings as sets of unconnected planar structures detected in the point cloud [14, 20, 21, 57, 58] (Fig. 4.1 (a)), or as collections of closed 3D boundaries of either the whole building [24], or separate rooms [46, 48, 59, 60] (Fig. 4.1 (b)). While the method in [41] reconstructs volumetric walls, their thickness is defined manually instead of being estimated from the input data.

To overcome the limitations of previous approaches, we propose a novel reconstruction method in which the representation of buildings using parametric, interrelated, volumetric elements (Fig. 4.1 (c)) is an integral component. Our approach automatically reconstructs walls between adjacent rooms from opposite wall surfaces observed in the input data while simultaneously taking into account globally plausible connectivity of all elements. Together with a faithful estimation of wall thickness, the result is a high-level editable model of volumetric wall elements. The reconstruction is formulated as an energy minimization problem which simultaneously optimizes costs for assigning rooms to areal regions of the building, and costs for separating adjacent rooms by volumetric wall elements. In contrast to previous approaches, this has the advantage that reasonable binary costs for the assignment of pairs of room labels to adjacent areal regions of the building – and thus the selection of suitable wall elements – is directly incorporated into the global optimization. To make our method robust against large-scale clutter outside the building, outliers are automatically filtered prior to reconstruction. Finally, doors and windows are detected, classified and assigned to the respective wall elements to further enrich the model. Our evaluation using various real-world indoor scans shows that our method rapidly provides models which can be used for e.g. planning of retrofitting, especially since our method requires little or no parameter adjustment.

**Applications** The distinguishing feature of our approach is that it directly captures important properties and relations of building elements. Since architectural Building Information Modeling (BIM) formats (e.g. Industry Foundation Classes, IFC) are based on similar relational paradigms, exporting our results to architectural software is straight forward. This enables a whole range of processing and analysis tasks

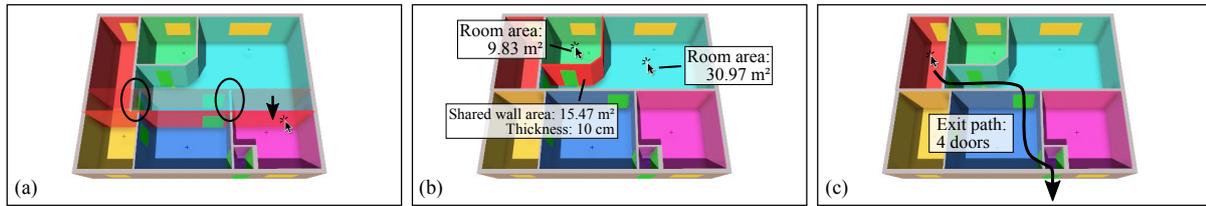


Figure 4.2: Example operations which are easily implemented using our results. (a) Relations between walls and rooms enable editing while maintaining room topology. Note how incident walls are adjusted automatically. (b) Automatic determination of wall elements shared between rooms together with automatic measurements enable e.g. acoustic or thermal simulations. (c) Global connectivity enables pathfinding for e.g. simulation and optimization of escape routes.

in industry-standard software. We exemplify some applications for e.g. planning of retrofitting in Fig. 4.2 which can directly be implemented using our results: Since the incidence and adjacency relations of walls and rooms are inherently known, selecting e.g. all walls enclosing a room or manipulating whole walls while maintaining overall room topology is easily possible (Fig. 4.2 (a)). This allows for quick, high-level prototyping of changes on the level of semantically meaningful construction element groups. The available information also enables more complex queries for e.g. the subset of wall elements that are simultaneously incident to two adjacent rooms (Fig. 4.2 (b)). Together with directly available properties like wall thickness, openings, room and wall areas, this provides important information for performing acoustic or thermal simulations. The global connectivity information further allows to perform pathfinding in the whole building story (Fig. 4.2 (c)) for e.g. simulating and optimizing escape routes.

## 4.2 Related Work

Okorn et al. [57] generate 2D floor plans from 3D point clouds. A histogram of the vertical positions of all measured points is built. Peaks in this histogram are considered to be large horizontal planar structures (i.e. floor and ceiling surfaces). After removing points belonging to the detected horizontal structures, a line fitting on the remaining points is performed. The resulting line segments constituting the floor plan are not connected and do not provide e.g. closed boundaries of rooms. Budroni and Boehm [48] extract planar structures for floors, ceilings and walls by conducting a plane sweep. Using a piecewise linear partitioning of the x-y-plane, they classify cells of this partitioning as inside and outside by determining the occupancy of the cells by measured points and considering densely occupied cells as inside. The result is a 2.5D extrusion of the determined room boundary. In the approach by Sanchez and Zakhor [20], points are classified into floor, ceiling, wall, and remaining points using the point normal orientations. For floor, ceiling and wall points, planar patches are fitted and their extents are estimated using alpha shapes. Parametric staircase models are fitted to the set of remaining points. The resulting mesh models consist of unconnected planar surfaces. Monszpart et al. [21] propose a method for extracting planar structures in point clouds which follow regularity constraints. Their optimization approach balances data fitting and simplicity of the resulting arrangement of planes. A method for generating visually appealing indoor models is proposed by Xiao and Furukawa [41]. An “inverse-CSG” approach is used for reconstructing the building’s geometry by detecting planar structures and then fitting cuboid primitives. These primitives are combined using CSG operations; the quality of the resulting model is tested using an energy functional. Finally, the resulting mesh model is textured using captured images. A drawback is that the building needs to be sufficiently well approximated by the used cuboid primitives. Adan and Huber [14] reconstruct planar floor, ceiling, and wall surfaces from multi-story point clouds by first

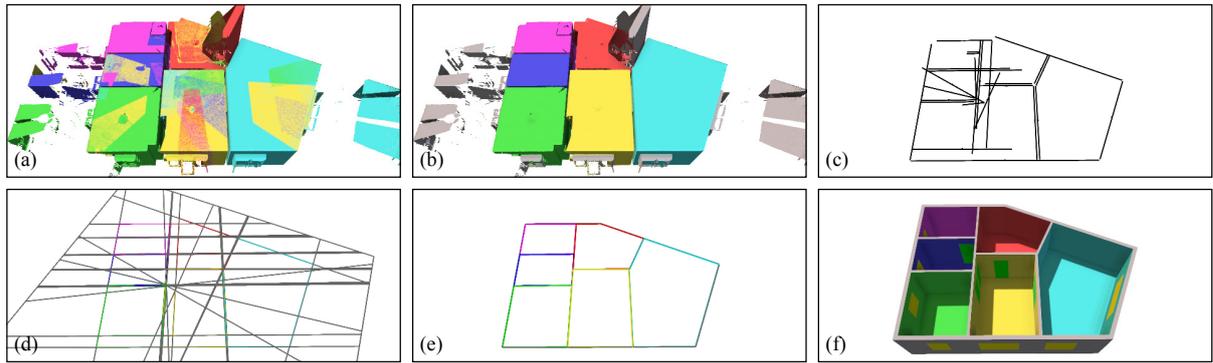


Figure 4.3: Overview of our approach (see also Section 4.3). **(a)** Input point cloud; assignment of points to scans shown in different colors. **(b)** Refined assignment after automatic segmentation. **(c)** Detected vertical planes transferred to the horizontal plane. **(d)** Candidates for walls are derived from single and pairs of projected planes. Intersecting their centerlines yields a planar graph whose faces are subsequently assigned labels for rooms or outside area. **(e)** Only edges separating differently labeled faces are retained. **(f)** The final model with detected and classified wall openings, e.g. doors (green) and windows (yellow).

detecting the modes of a histogram of point height values to find horizontal planes, and then detecting vertical planes by means of Hough transform. They recover occluded parts of reconstructed surfaces, and perform an opening detection by means of Support Vector Machine (SVM) learning. Xiong et al. [58] extend this approach by classifying detected planar patches as floor, ceiling, wall or clutter using a stacked learning approach, also taking into account contextual information of neighboring patches. Mura et al. [60] reconstruct indoor scenes with arbitrary wall orientations by building a 3D Delaunay tetrahedralization of the input dataset and partitioning inside and outside using a diffusion process governed by affinities of tetrahedron pairs. A binary space partitioning is also done by Oesau et al. [24] by first splitting the input dataset horizontally at height levels of high point densities and then constructing 2D arrangements of projections of detected wall surfaces. The space partitioning into inside and outside is performed by means of Graph-Cut. Other approaches not only perform binary space partitioning but label different rooms: Turner and Zakhor [46] generate 2.5D watertight meshes by first computing an inside/outside labeling of a triangulation of wall points and a subsequent partitioning into separate rooms using a Graph-Cut approach. This method is further developed by Turner et al. in [59], improving the texture mapping capabilities of the algorithm. The results are well-regularized, watertight, textured mesh models. Mura et al. [23] first extract candidate wall elements while taking into account possibly occluded parts of the surfaces to determine the real wall heights for filtering out invalid candidates. After constructing a 2D line arrangement, they use a diffusion embedding to establish a global affinity measure between faces of the arrangement, and determine clusters of faces constituting rooms. The result is a labeled boundary representation of the building's rooms. Many of these methods build upon a spatial partitioning defined by detected wall surfaces and a subsequent classification of regions of this partitioning. Although the resulting models have applications like visualization, navigation or energy monitoring [61], they do not realize a reconstruction of volumetric, interconnected building elements like walls.

### 4.3 Approach

The starting point of our approach is a registered point cloud of one building story consisting of multiple indoor scans including scanner positions. Registration is usually done using the scanner software and

is outside the scope of this paper. The unit of measurement and up direction are assumed to be known. Surface normals for each point are estimated.

We argue that the wall structure of most building stories can be represented as a piecewise-linear, planar graph in which edges represent wall elements and vertices are locations where walls are incident (Fig. 4.3 (e)). Wall thickness is a scalar edge attribute. Conversely, faces of this graph represent the spatial room layout. There obviously exists a duality between the story’s room layout and its wall constellation, i.e. one representation can directly be derived from the other. The main idea of our approach is that – while both representations are essentially equally hard to reconstruct – we can derive important hints (priors) for the room layout from indoor point cloud scans since they are a sampling of the inner surfaces of room volumes. It is therefore meaningful to base our reconstruction on the derivation of a suitable room layout from which the constellation of walls is immediately obtained due to the duality.

We extract priors for the room layout as follows: assuming that each room was scanned from one position (or few positions), separate scans yield a coarse segmentation of the point cloud into separate rooms (Fig. 4.3 (a)). We improve this segmentation using a diffusion process which eliminates most overlapping regions between scans (Fig. 4.3 (b)) and automatically filters out clutter outside of the building. As further described below, the determination of a suitable room layout is then formulated as a labeling problem of the regions of a suitable partitioning of the horizontal plane (using labels for different rooms and the outside area). This directly follows the aforementioned duality principle: after determining a suitable labeling, connected components of identically labeled cells are rooms, and edges separating differently labeled regions are wall elements.

Since our goal is to extract a piecewise-linear graph of walls, we construct a partitioning based on potential wall surfaces: We first detect vertical planes as candidates for wall surfaces and project them to the horizontal plane (Fig. 4.3 (c)). Similar to previous approaches [23, 24, 50] we then construct an arrangement of (infinitely long) lines from the set of possible wall surfaces (Fig. 4.3 (d)). In contrast to previous approaches, edges of this arrangement represent wall *centerlines* instead of wall *surfaces*. Furthermore, arrangement lines are not only constructed from *single* wall surfaces but also from *pairs* of parallel surfaces which yield candidates for walls separating adjacent rooms. This subtle but crucial difference allows us to go beyond the reconstruction of *separate* room volumes as done in previous works (Fig. 4.1 (b)) by enabling the algorithm to reconstruct room-separating wall elements directly. In order to guide the selection of adequate wall elements, we retain the information from which supporting measured points each edge originates. This yields wall selection priors encouraging the reconstruction of wall elements which were constructed from surfaces belonging to the *same* pair of rooms that the wall separates.

The determination of a globally plausible labeling is then formulated as an energy minimization problem. This allows us to incorporate room layout priors and wall selection priors as unary and binary costs into one optimization. After an optimal labeling has been determined, only retaining edges separating differently labeled regions are the sought wall structures (Fig. 4.3 (e)). Extruding walls according to estimated room heights and a detection and classification of openings yields the final parametric model (Fig. 4.3 (f)).

## 4.4 Point Cloud Segmentation

To obtain priors for the localization of rooms in subsequent steps, each point of the input point cloud is automatically assigned a label for a room or the outside area. Our approach is based on the method by Ochmann et al. [45] which we will briefly summarize before describing our modifications: the original method assumes at least one scan within each room; multiple scans per room are merged manually such

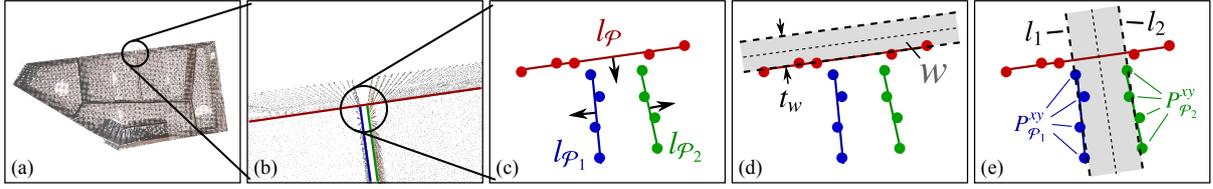


Figure 4.4: Wall candidate generation. **(a)-(b)** Detected vertical planes in the 3D point cloud are projected into the horizontal plane. **(c)** Different wall surface lines including the respective (projected) support points and surface normals. **(d)** For each single wall surface, an infinitely long wall candidate  $w$  for a wall separating a room from outside area is generated. In this case, the thickness  $t_w$  is user-specified. **(e)** For each pair of approximately parallel wall surfaces, a candidate for separating adjacent rooms is generated. In this case, wall thickness is estimated from the data.

that a one-to-one mapping between (merged) scans and rooms is obtained. The initial assignment of each point to one of the (merged) scans (Fig. 4.3 (a)) provides a coarse segmentation of the point cloud into rooms. However, openings such as open doors lead to severe overlaps between scans, causing large areas of the point cloud to contain a mix of differently labeled points. To obtain a point labeling that roughly corresponds to the building’s room layout and is homogeneous within each room (Fig. 4.3 (b)), an automatic labeling refinement is performed. The process is based on the assumption that most points that are *visible* from the position of a point  $p$  are already labeled correctly. By determining which points are visible from the position of  $p$  and averaging the observed labels, a new (soft) labeling of  $p$  is obtained. After iterating this procedure, the label with the highest confidence is assigned to  $p$ . This process can be interpreted as a diffusion of point labels between points governed by mutual visibility. In practice, a stochastic ray casting from the position of  $p$  into the hemisphere around the normal of  $p$  is performed.

We extend this method in two ways: first, we automatically filter out clutter outside of the building which is often caused by windows or mirrors. We argue that for a point  $p$  that is part of clutter outside of the building, most rays cast from  $p$  into the hemisphere around the normal of  $p$  do not hit any interior wall surfaces. In this case we assign a high value for an additional “outside” label to  $p$ . This modification proves to be highly effective in our experiments as demonstrated in Fig. 4.3 (b) (gray points have been assigned the outside label). Second, we do not require that multiple scans per room are merged manually. Instead, we run the reconstruction using all scans as separate labels. In case of multiple scans in a room, this leads to implausible walls within rooms which are subsequently removed as described in Section 4.7.

## 4.5 Generation of Wall Candidates

Candidates for wall elements are derived from vertical surfaces observed in the scans. They constitute possible locations of walls for the optimization in Section 4.6. Since wall heights and lengths are not regarded in this step, the following 2D representation is used: Each wall candidate  $w = (t_w, n_w, d_w)$  is defined by a thickness  $t_w \in \mathbb{R}^{\geq 0}$  and an infinite centerline in the horizontal plane given in Hesse normal form  $\langle n_w, x \rangle - d_w = 0$ . Wall heights and lengths will be determined later.

In a first step, planes in the 3D point cloud are detected using a RANSAC implementation by Schnabel et al. [13]. Nearly vertical planes ( $\pm 1^\circ$ ) with a sufficiently large approximate area ( $\geq 1.5\text{m}^2$ ) are considered as potential wall surfaces. For a plane  $\mathcal{P}$  fulfilling these constraints, let  $n_{\mathcal{P}} \in \mathbb{R}^3$  be the plane normal and  $P_{\mathcal{P}}$  the set of measured points supporting  $\mathcal{P}$ . Each extracted plane  $\mathcal{P}$  is transferred to the horizontal plane as a *wall surface line*  $l_{\mathcal{P}}$  defined by  $\langle n_{l_{\mathcal{P}}}, x \rangle - d_{l_{\mathcal{P}}} = 0$ . A schematic example for the extraction of wall surface lines is shown in Figures 4.4 (a)-(c). The normal  $n_{l_{\mathcal{P}}}$  is approximated by the

projection of  $n_{\varphi}$  into the horizontal plane,  $n_{l_{\varphi}} := \frac{\langle (n_{\varphi})_x, (n_{\varphi})_y \rangle}{\| \langle (n_{\varphi})_x, (n_{\varphi})_y \rangle \|_2}$ . The distance to the origin  $d_{l_{\varphi}}$  is determined by least squares fitting to the set  $P_{\varphi}^{xy}$  of support points projected to the horizontal plane using the fixed normal  $n_{l_{\varphi}}$  such that  $\sum_{p \in P_{\varphi}^{xy}} (\langle n_{l_{\varphi}}, p \rangle - d_{l_{\varphi}})^2$  is minimized. From the wall surface lines, we then generate two kinds of wall candidates as we do not know at this point which types of candidates will yield a globally plausible reconstruction:

**Outside Walls** For each *single* wall surface line  $l_{\varphi}$ , we construct a candidate for a wall separating a room from the outside area (Fig. 4.4 (d)). Since the real wall thickness cannot be determined automatically from a single surface, a user-specified thickness is used (in our experiments,  $t_w = 20\text{cm}$ ). The centerline of the candidate is constructed such that the side of the wall candidate that points towards the inside of the room is identical to  $l_{\varphi}$ , i.e. the centerline is defined by  $\langle n_{l_{\varphi}}, x \rangle - d_w = -\frac{t_w}{2}$ .

**Room-Separating Walls** To generate candidates for walls separating adjacent rooms, each *pair* of wall surface lines fulfilling certain constraints is considered as two opposite surfaces of a wall separating adjacent rooms (Fig. 4.4 (e)). Let  $l_{\varphi_1}, l_{\varphi_2}$  be two wall surface lines that are approximately parallel ( $\pm 1^\circ$ ) and have opposing normal orientations. To prune invalid pairs, a coarse check is performed whether the projected support pointsets of the originating planes  $P_{\varphi_1}^{xy}, P_{\varphi_2}^{xy}$  (partially) overlap. To this end, the support pointsets are projected onto the respective opposite line. If support points are present near the projected points, their support is considered overlapping. For each pair fulfilling these constraints, a wall candidate is generated by fitting to  $l_{\varphi_1}$  and  $l_{\varphi_2}$  simultaneously: The candidate's normal  $n_w$  is first determined as the average of the normals  $n_1, n_2$  of  $l_{\varphi_1}, l_{\varphi_2}$ , weighted with the cardinality of the support pointsets,

$$n_w := \frac{|P_{\varphi_1}^{xy}|n_1 + |P_{\varphi_2}^{xy}|(-n_2)}{\| (|P_{\varphi_1}^{xy}|n_1 + |P_{\varphi_2}^{xy}|(-n_2)) \|_2}.$$

Using the common normal  $n_w$ , two parallel lines  $l_i, i \in \{1, 2\}$  defined by  $\langle n_w, x \rangle - d_i = 0$  are fitted to the respective support pointsets such that  $\sum_{p \in P_{\varphi_i}^{xy}} (\langle n_w, p \rangle - d_i)^2$  is minimized. The centerline of the wall candidate is constructed midway between the parallel lines,  $\langle n_w, x \rangle - \frac{1}{2}(d_1 + d_2) = 0$ , and the candidate's thickness is defined as the distance between them,  $t_w = |d_1 - d_2|$ . Candidates with a thickness above a threshold are discarded (in our experiments,  $t_w > 60\text{cm}$ ).

## 4.6 Determination of an Optimal Room and Wall Layout

From the infinitely long wall candidates, we determine a set of wall segments which yields a plausible reconstruction of the building's walls. To this end, we consider the intersection of all wall candidate centerlines in the horizontal plane which yields a planar graph  $W' = (V', E')$  (Fig. 4.5 (b)). Faces of  $W'$  are regions of the building's layout (i.e. parts of rooms or outside area), edges  $E'$  are segments of possible walls, and vertices  $V'$  are possible locations where walls are incident. We follow the intuition that walls separate different regions, i.e. adjacent rooms, or rooms and the outside world. Consequently, a classification of the faces of  $W'$  implies locations of walls in the following sense: connected components of identically-labeled faces are rooms (or outside areas), and edges between differently-labeled faces are walls. Fig. 4.5 (c) shows an example for a face labeling from which connected wall elements as shown in Fig. 4.5 (d) are extracted. Wall thickness of an edge  $e$  is set to the thickness of the wall candidate from which  $e$  originates.

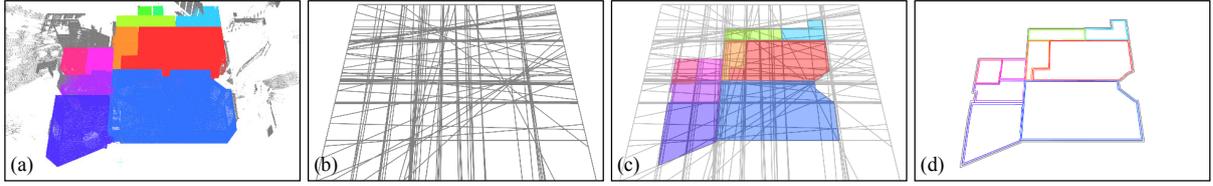


Figure 4.5: Determination of suitable wall candidate segments. **(a)** Input point cloud after segmentation. **(b)** Intersecting all wall candidate centerlines yields a planar graph. We determine an assignment of all faces to rooms or outside area such that connected components of identically-labeled faces are rooms and edges between differently-labeled faces are wall elements. **(c)** Resulting labeling of faces after optimization; colors indicate room labels. **(d)** Retaining only edges separating differently-labeled faces yields a subgraph representing the sought wall elements and their connectivity.

We formulate the face classification as a labeling problem which is solved using an energy minimization approach. The target functional has two terms: unary costs for the assignment of labels to faces of  $W'$  and binary costs for the assignment of label pairs to adjacent faces. Unary costs provide hints where rooms (or outside areas) are located and binary costs guide the selection of adequate edges for separating differently-labeled faces. In particular, if two adjacent rooms share a common wall, a wall candidate constructed from wall surfaces of *these* rooms should separate them. We will now formalize the problem. Let  $W = (V, E)$  be the dual graph of  $W'$  and let  $\{l_1, \dots, l_k, l_o\}$  be the set of labels where  $l_i, i \in \{1, \dots, k\}$  are labels for each scan and  $l_o$  is the outside label. For clarity, we assume for now that each room was scanned from exactly one position and thus  $k$  equals the number of rooms; the more general case of multiple scans per room will be discussed later. Given a unary cost function  $U_v(l_v)$  yielding the cost for assigning label  $l_v$  to a vertex  $v \in V$ , and a binary cost function  $B_{v,w}(l_v, l_w)$  yielding the cost for assigning the (unordered) pair of labels  $l_v, l_w$  to  $v, w \in V$ , we minimize the total cost for a labeling  $l$ , i.e.

$$E(l) = \sum_{v \in V} U_v(l_v) + \sum_{(v,w) \in E} B_{v,w}(l_v, l_w) \rightarrow \min. \quad (4.1)$$

Applying the minimization algorithm to the dual graph  $W$  of  $W'$  allows us to determine a labeling of the faces of  $W'$  by finding an optimal labeling of the vertices of  $W$ . The problem stated in Eq. 4.1 is solved using the algorithm by Boykov et al. [62–64]. We now define unary and binary cost functions for label assignments. In the following, the notation for *label vectors*

$$\mathcal{L}(\cdot) = (c_1, \dots, c_k, c_o), \quad \forall i : c_i \geq 0, \quad \|\mathcal{L}(\cdot)\|_1 = 1$$

will be used for soft label assignments to different entities, e.g. points, faces and edges. The coefficient  $c_i$  of  $\mathcal{L}(\cdot)$  corresponding to label  $l_i$  will be denoted  $\mathcal{L}_i(\cdot)$ . As a shorthand, let  $\mathcal{I}_i$  denote a hard label vector with  $c_i = 1$ , and let  $\mathcal{I}_{ij} := \frac{1}{2}(\mathcal{I}_i + \mathcal{I}_j)$ .

**Unary Costs** Intuitively, the cost  $U_v(l_v)$  shall be low iff the area spanned by face  $f$  in  $W'$  is likely to belong to  $l_v$ . We first estimate a label vector  $\mathcal{L}(f)$  whose coefficients reflect the probabilities that the area covered by  $f$  belongs to each room or the outside area. A naive approach would be to project all measured points into the horizontal plane and to determine how many points of each room (with respect to the point labels obtained in Section 4.4) are located within  $f$ . The first problem is that non-uniform distributions of measured points (Fig. 4.6, left) yield a similar probability estimate like a uniform distribution (Fig. 4.6, right) although the latter provides stronger evidence that the whole face belongs to a certain room. The second problem is that we need to estimate the probability that  $f$  is located in the outside area which

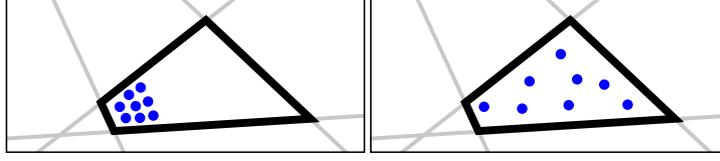


Figure 4.6: Considering only the number of projected points within a face for unary costs does not take into account their spatial distribution.

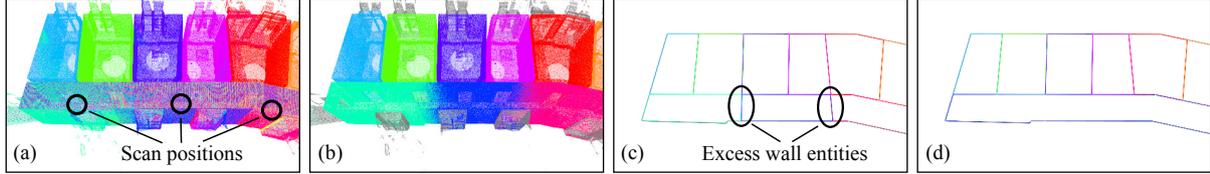


Figure 4.7: Multiple scans within a single room. **(a)** The hallway has been scanned from three positions; room labels are mixed within that room. **(b)** After segmentation (Section 4.4), the hallway is still split in multiple sections. **(c)** The labeling algorithm separates these regions by wall elements that are not part of the building’s true walls. **(d)** By detecting and removing excess wall elements, faces are merged to larger rooms.

is not represented by measured points.

We therefore propose a stratified sampling method which takes the spatial distribution of projected measured points into account and yields an estimate for the outside label. All measured points are projected into a uniform 2D grid in the horizontal plane. The side length of the grid cells is chosen as twice the point cloud subsampling density (see Section 4.8). The label vectors of all points within a grid cell are averaged and empty cells are assigned the outside label vector  $\mathcal{I}_o$ . Subsequently, the label vector  $\mathcal{L}(f)$  of  $f$  is estimated by picking in the grid at uniformly sampled positions within  $f$  and averaging the resulting label vectors. The number of samples within  $f$  is chosen proportionally to the face area (at least one sample is enforced). The unary cost function is then defined as

$$U_v(l_v) := \alpha \cdot \text{area}(f) \cdot \|\mathcal{L}(f) - \mathcal{I}_v\|_1, \quad (4.2)$$

where  $v$  is the vertex of  $W$  corresponding to face  $f$  in  $W'$ , and  $\alpha$  is a weighting factor (see Section 4.8).  $\mathcal{L}(f)$  is the *estimated* labeling of face  $f$ , and  $\mathcal{I}_v$  is the ideal *expected* label vector for label  $l_v$ . The distance between these label vectors is weighted proportionally to the area of  $f$  in order to mitigate the impact of differently sized faces in the sum of total labeling costs.

**Binary Costs** For the binary cost  $B_{v,w}(l_v, l_w)$ , consider edge  $e$  in  $W'$  to which the edge  $(v, w)$  in  $W$  corresponds. Intuitively, the cost for assigning labels  $l_v, l_w$  to  $v \in V$  and  $w \in V$  shall be low iff the surfaces of the wall represented by  $e$  are supported by measured points with labels  $l_v, l_w$  (in the case of a wall bordering the outside area, there should be no support on the exterior side). In other words, for the separation of faces with different labels  $l_v, l_w$ , wall elements whose surfaces are supported by points with labels  $l_v, l_w$  shall be preferred. For estimating the label vector for an edge  $e$ , a sampling strategy similar to the face label vectors is used. Consider edge  $e$  originating from up to two wall surface lines  $l_{\varphi_1}, l_{\varphi_2}$  (see Section 4.5) with according projected support points  $P_{\varphi_1}^{xy}, P_{\varphi_2}^{xy}$ . If  $e$  originates from a single wall surface line  $l_{\varphi_1}$ , we set  $P_{\varphi_2}^{xy} = \emptyset$ . Analogously to the 2D grid in the horizontal plane, we construct a one-dimensional grid on  $e$ . The support points  $P_{\varphi_1}^{xy} \cup P_{\varphi_2}^{xy}$  are projected into the grid and their point labels are averaged per cell. Empty cells are assigned the outside label. The label vector  $\mathcal{L}(e)$  is now estimated

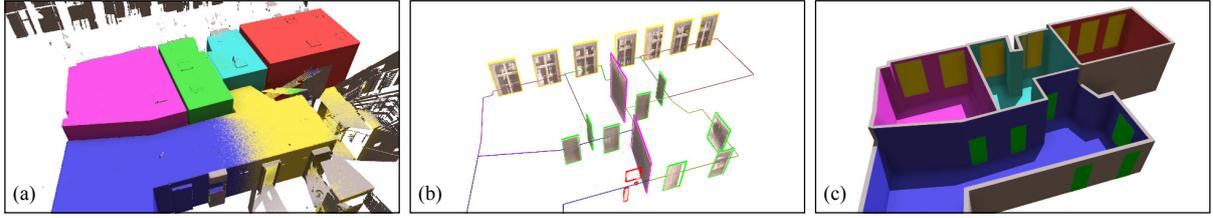


Figure 4.8: Opening detection and classification. **(a)** The input point cloud after segmentation. **(b)** Detected clusters of intersections between reconstructed walls and simulated laser rays between scanner positions and measured points. Clusters are classified as doors (green), windows (yellow), “virtual” clusters indicating walls to be removed for merging multiple scans within a room (magenta), and “invalid” (red). **(c)** The final model after removal of walls containing “virtual” openings.

by sampling uniformly distributed points on  $e$  and averaging the label vectors obtained by picking in the grid at the sample positions. We then define the binary costs as

$$B_{v,w}(l_v, l_w) := \begin{cases} \beta \cdot \text{len}(e) \cdot (\|\mathcal{L}(e) - \mathcal{I}_{vw}\|_1 + \gamma \mathcal{L}_o(e)), & \text{if } l_v \neq l_w, \\ 0, & \text{otherwise,} \end{cases} \quad (4.3)$$

where  $v, w$  are the vertices of  $W$  corresponding to faces  $f, g$  in  $W'$  that are separated by edge  $e$ ,  $\text{len}(e)$  is the Euclidean length of edge  $e$ , and  $\beta, \gamma$  are weighting factors (see Section 4.8) respectively. Similar to the unary costs, weighting the distance between the observed and ideal label vectors by edge length mitigates the influence of different edge lengths. The additional term  $\mathcal{L}_o(e)$  penalizes usage of edges with a high outside prior. We found that this term helps to select correct edges with support points on both sides for separating adjacent rooms. After the face labeling is determined, only edges which separate differently labeled faces are retained. The resulting subgraph  $\overline{W}$  of  $W'$  (Fig. 4.5 (d)) is used in Section 4.7 for reconstructing connected wall elements.

**Multiple scans within one room** We previously assumed that each room was scanned from exactly one position within that room. In the case of more than one scan, one room is represented by a set of different labels. Fig. 4.7 (a) shows an example of a hallway scanned from three positions. After segmentation (Section 4.4), the hallway is split into multiple regions represented by differently labeled points (Fig. 4.7 (b)). The graph labeling optimization separates these sections by implausible walls (Fig. 4.7 (c)). We remove such walls (Fig. 4.7 (d)) as part of the opening detection in the next section.

## 4.7 Model Generation and Opening Detection

From the determined graph, the final model can now be derived in a straight forward manner. The model is further enriched by detected window and door openings.

**Walls** For each edge  $\bar{e} = (\bar{v}, \bar{w})$  of  $\overline{W}$ , a wall element  $\mathcal{W}$  is constructed with centerline endpoints located at  $\bar{v}$  and  $\bar{w}$ . The thickness of  $\mathcal{W}$  is determined by the thickness of the wall candidate from which  $\bar{e}$  originates. Endpoints of wall elements are connected iff the corresponding edges are incident to a common vertex. For vertical extrusion, we first estimate floor and ceiling heights for each face  $\bar{f}$  in  $\overline{W}$  separately using the following heuristic: Consider all approximately *horizontal* planes detected during wall candidate generation (Sec. 4.5). For each plane, the number of support points located within  $\bar{f}$  is determined. The elevation of the plane with the largest support within  $\bar{f}$  and upwards- (resp. downwards-)

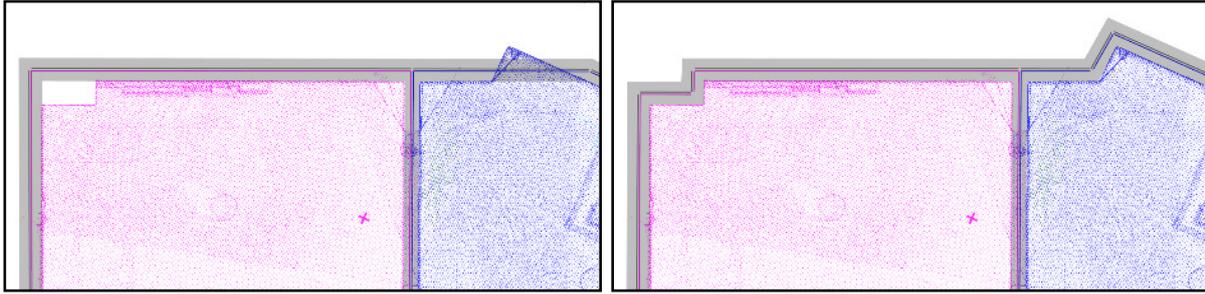


Figure 4.9: Different plane detection options: Allowing smaller planes as potential wall surfaces allows for more detailed structures (right-hand side) at the cost of possibly detecting incorrect candidates in clutter.

facing normal is chosen as the floor height  $h_{fl}(\bar{f})$  (resp. ceiling height  $h_{cl}(\bar{f})$ ) of  $\bar{f}$ . The vertical extent of a wall represented by edge  $\bar{e}$  separating faces  $\bar{f}_1, \bar{f}_2$  is then defined to span the heights of both adjacent rooms:  $[\min(h_{fl}(\bar{f}_1), h_{fl}(\bar{f}_2)), \max(h_{cl}(\bar{f}_1), h_{cl}(\bar{f}_2))]$ .

**Opening Detection** Openings in walls either arise from doors and windows, or because a reconstructed wall was artificially introduced due to multiple scans within one room as described in Section 4.6. By classifying detected openings accordingly, we further enrich the model by doors and windows, and determine which walls to remove for handling multiple scans within rooms. To locate potential openings, we determine intersection points between reconstructed walls and simulated laser rays from the scan positions to the measured points. The intersection points are clustered in the 2D domain of the wall surfaces (a simple greedy, single-linkage clustering based on distances between intersection points yielded satisfactory results); see Fig. 4.8 (b) for an example. The clusters are then classified as doors, windows, “virtual” (i.e. openings due to excess walls) or “invalid” (i.e. clutter) by means of supervised learning using libsvm [65]. Six-dimensional feature vectors with the following features are used to characterize openings: cluster bounding box width and height, distance from lower and upper wall bounds, approximate coverage by intersection points, and a binary feature indicating whether the associated wall is adjacent to outside area. Clusters recognized as doors or windows are assigned to the respective wall elements. Adjacent faces of  $\bar{W}$  separated by wall elements containing at least one “virtual” opening (magenta clusters in Fig. 4.8 (b)) are merged by removing all edges to which both faces are incident. To account for changes after a wall removal, the determination of room heights, intersection points, clusters and opening classes is performed iteratively until no more virtual openings exist.

## 4.8 Evaluation

We tested our approach on real-world point clouds of 14 stories from 5 different buildings; statistics are given in Table 4.1. The shown number of points is after subsampling with the Point Cloud Library [66] using a resolution of  $\varepsilon = 0.02\text{cm}$  (i.e. in a voxel grid with a resolution of  $\varepsilon$ , at most one point in each voxel is retained). Normals are estimated by means of local PCA using point patches of 48 nearest neighbors. Normals are flipped towards the respective scanner position.

**Parameter selection** The first set of crucial parameters affects plane detection in the extraction of wall surfaces (Section 4.5). For classifying planes as vertical (wall surfaces) or horizontal (floor and ceiling surfaces) we chose a threshold on the angular deviation of  $\pm 1^\circ$  from the ideal orientations. We ignore planes with less than 500 support points or an approximate areal coverage by support points



Figure 4.10: Different choices for  $\alpha, \beta, \gamma$  in Eqs. 4.2 and 4.3. **(a)-(b)** Perspective and orthographic view of an example situation. **(c)** Parameters chosen as described in Section 4.8. Wall centerlines are well-regularized and common wall elements have been reconstructed between rooms. **(d)** Without unary costs ( $\alpha := 0$ ). While the resulting walls are well-regularized, parts of rooms are missing despite high areal support by measured points. **(e)** Without binary costs ( $\beta := 0$ ). Walls are located similar to (c) but are overly complex due to missing regularization and preference for correctly labeled edges. **(f)** Without penalty for high outside labeling ( $\gamma := 0$ ). The algorithm does not prefer common walls for separating adjacent rooms.

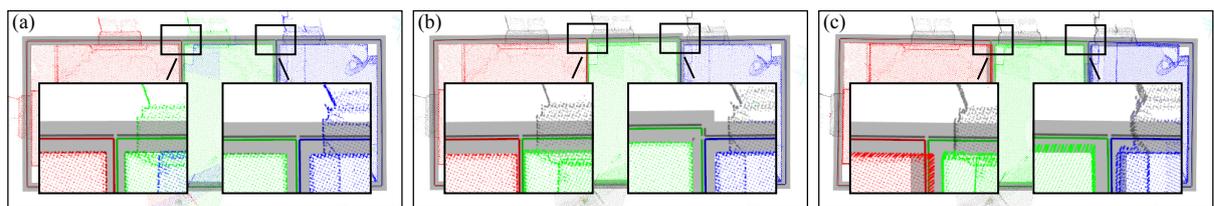


Figure 4.11: Effects of registration errors. **(a)** Result without alignment errors. Wall volumes are shown in gray together with the respective wall centerlines. **(b)** Translational registration errors may result in offset walls to which the algorithm adapts accordingly (right detail view). Wall thickness may also change (wall separating the red and green rooms in the left detail view). **(c)** Rotational registration errors may lead to wall surface pairs not to be associated to common walls. Wall thickness is incorrect since the wall candidates do not originate from wall surfaces pairs.

below  $1.5\text{m}^2$ . Also, vertical planes resulting in line segments below 0.5m are ignored. These parameters control a tradeoff between avoiding clutter and ignoring small details: high thresholds only consider larger (but potentially more stable) planes as candidates for wall surfaces. Conversely, low thresholds may introduce clutter due to incorrectly detected planes. Figure 4.9 demonstrates different choices. The second important set of parameters consists of the weights  $\alpha, \beta, \gamma$  in Equations 4.2 and 4.3. In our experiments, we found that a ratio of  $\alpha/\beta$  of 4/1, and  $\gamma = 4$  yielded good results (Fig. 4.10 (c)). The effects of setting either  $\alpha, \beta$ , or  $\gamma$  to zero are shown in Figures 4.10 (d)-(f). We also found that smoothing the 2D and 1D grids used for the determination of face and edge label vectors in Section 4.6 using a large Gaussian kernel usually improves results.

**Robustness** Quality and robustness of our reconstruction depend on plane detection quality which is influenced by e.g. scanner noise, point density, registration accuracy, and clutter outside and inside of the building. As our datasets were captured using professional laser scanners, noise level and sampling density were no issue. Registration errors directly influence the position of detected planes and thus the generated wall candidates. Our algorithm adapts well to small misalignment; stronger translational or rotational alignment errors have specific effects as exemplified in Fig. 4.11. Clutter outside of the building is effectively eliminated by our automatic filtering method. Clutter inside of rooms and scan holes pose big challenges when working with indoor scans. Except for extreme cases (e.g. completely unobserved wall surfaces or objects which span the whole story height and thus yield planes that are indistinguishable from walls), our algorithm proves to be robust against e.g. furniture within rooms as shown in Fig. 4.12: Despite the presence of large scan holes, using all available points (from e.g. furniture) as priors for room localization closes holes, and the smoothness property of the used graph-cut-based optimization yields well-regularized walls. Furthermore, as our approach uses infinitely long wall candidates, small or medium sized holes in the support pointsets of wall surfaces caused by occlusions are automatically bridged in a plausible manner (Fig. 4.13). We also found that the algorithm is very robust against errors in the segmentation step (Section 4.4), especially in the interior of the building, i.e. if overlaps between scans of adjacent rooms still exist. However, filtering out large-scale clutter outside the building is

Dataset	Points	Scans	Time (s)	Figures
Building A, storey 1	9524724	23	84.9	4.12*
Building A, storey 2	19365622	33	215.7	4.16 (a)
Building B, storey 1	826229	5	6.0	—
Building B, storey 2	1676486	6	11.9	4.9*
Building B, storey 3	1673919	6	12.2	4.1, 4.3
Building B, storey 4	2203670	8	16.0	4.2
Building B, storey 5	2470678	11	17.7	4.5
Building C, storey 1	4749565	9	39.7	4.14*
Building C, storey 2	22757718	67	486.3	4.7*, 4.8*, 4.16 (c)
Building C, storey 3	23883396	63	449.6	—
Building D, storey 1	17712659	34	252.2	4.15
Building E, storey 1	14399907	37	189.3	4.13*
Building E, storey 2	19769647	51	319.9	—
Building E, storey 3	17104101	43	241.5	4.16 (b)

Table 4.1: Datasets used in our experiments. Figure references marked with “\*” indicate that only a subset of the dataset is shown.

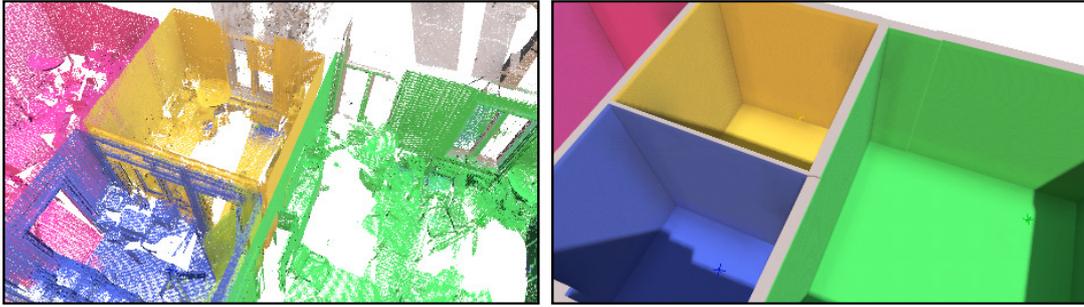


Figure 4.12: Highly cluttered rooms. Left: Clutter and transparent surfaces (windows) cause large scan holes; wall surfaces are only partially scanned. Right: Reconstructed walls still are well regularized and separate rooms correctly.

important in order to avoid erroneous classification of outside area as rooms.

**Opening detection accuracy** Our method for opening detection consists of three parts (see Section 4.7): (1) Determination of intersection points, (2) clustering of intersection points, and (3) classification of these clusters using supervised learning. Regarding (1), we found that the determined intersection points indicate locations and extents of openings well, with a low number of false positives. For (2), a simple single-linkage clustering based on point distances already yielded good results due to the low number of clutter intersection points which may cause chaining effects. However, a more sophisticated clustering method could improve results in some cases, e.g. multiple neighboring windows are sometimes recognized as a single cluster. Concerning (3), our training examples comprise 269 doors, 306 windows, 118 virtual, and 415 invalid clusters which were obtained by manually correcting a heuristic classification. During training, all stories originating from the building being classified were removed from the training set. Average cross validation rate of the training sets was 90.34%, average classification accuracy was 85.02%. This small yet significant gap indicates a generalization performance below optimum which we believe is caused by systematic differences between e.g. the used windows in different buildings, causing the feature vectors to not be i.i.d. Given the limited number of test data, we think that our approach is promising, especially since newly obtained examples can be fed back into the algorithm.

**Comparison to manually generated models** A visual comparison between our reconstruction and a professional, manually generated model is shown in Fig. 4.15. Locations and thickness of wall elements, and locations of doors are generally good; a few walls are missing either due to the fact that (small) rooms were not scanned separately and thus room labels are missing, or because openings were misclassified as “virtual” clusters.

**Time and memory requirements** Our experiments were run on a 6-core Intel Core i7-4930K (32 GB RAM) with a NVIDIA GeForce GTX 780 (3 GB RAM). Processing times of our prototypical implementation are shown in Table 4.1. Peak RAM usage (incl. visualization) for the largest dataset was about 16 GB.

**Limitations** If rooms are not completely enclosed by walls (e.g. balconies or partially scanned staircases), points might erroneously be classified as outside area during the segmentation step which may lead to missing parts in the reconstruction. Due to the current formulation of our approach, wall elements which are not connected to other walls at *both* ends cannot be represented. As a consequence, they

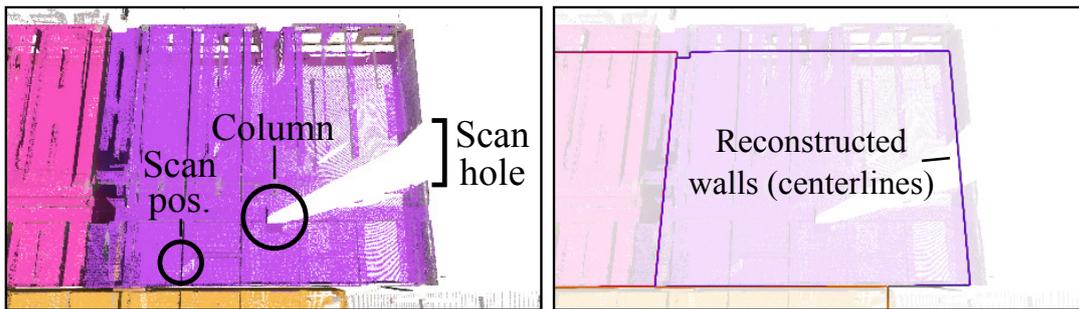


Figure 4.13: Our cost minimization approach and infinitely long wall candidates automatically bridge scan holes in a plausible manner.

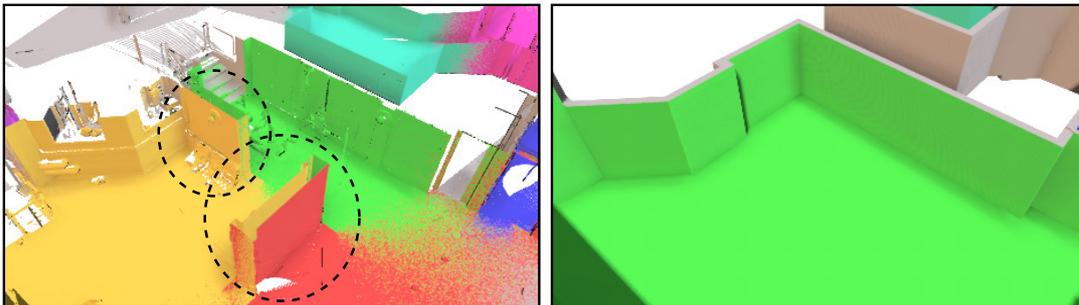


Figure 4.14: Wall elements which are not connected at both ends to other walls are currently not representable by our reconstruction.

are either missing (Fig. 4.14), or erroneously connected to other wall elements. Also, since we only consider planar wall surfaces and linear wall candidates, only piecewise linear wall structures can be reconstructed.

## 4.9 Conclusion and Future Work

We presented the first automatic method for the reconstruction of high-level parametric building models from indoor point clouds. The feasibility of our approach was demonstrated on a variety of complex real-world datasets which could be processed with little or no parameter adjustments. In the future, a more thorough comparison of reconstruction results with existing, manually generated models would help to analyze reconstruction results quantitatively. A generalization to multiple building stories poses specific challenges but would enable the reconstruction of multi-story models without the need to process stories separately. Also, the usage of different capturing devices (e.g. mobile devices) and real-time handling of streamed data are topics for future investigation.

## Acknowledgements

This work was partially funded by the German Research Foundation (DFG) under grant KL 1142/9-2 (MoD), as well as by the European Community under FP7 grant agreements 600908 (DURAARK) and 323567 (Harvest4D).

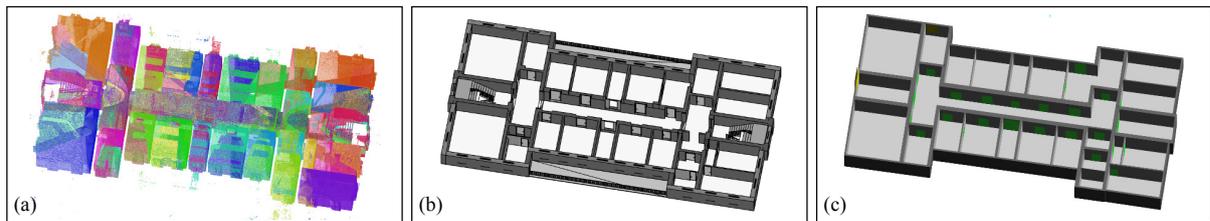


Figure 4.15: Visual comparison of reconstruction and manually generated model. (a) Input point cloud; scans are shown in different colors. (b) Professionally, manually generated model. (c) Reconstructed model. Locations and thickness of walls, and locations of doors are generally good.

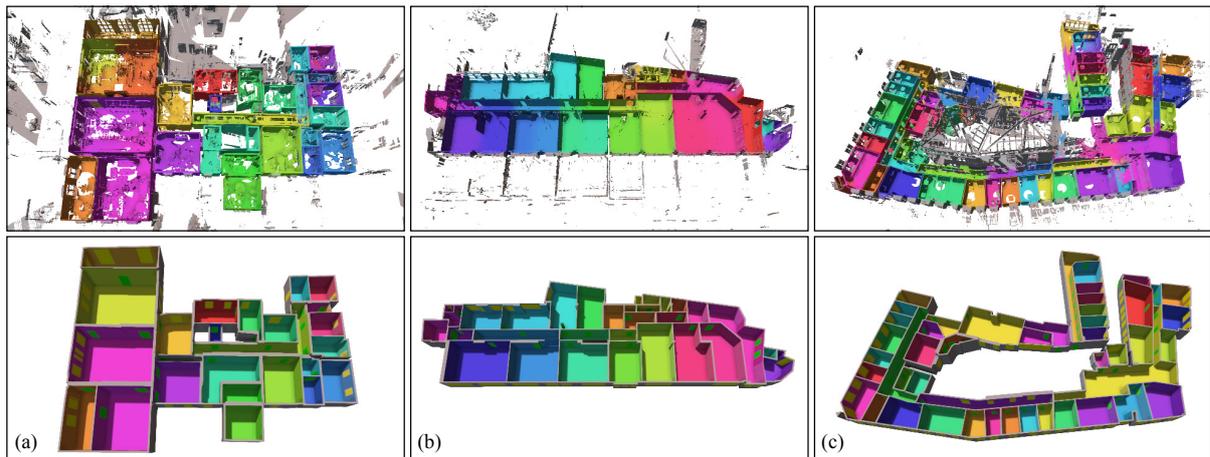


Figure 4.16: Example results on point clouds with 33, 43, and 67 scans. Upper row: Point clouds after segmentation step; most ceiling points (i.e. points with downwards-facing normals) are removed for visualization. Lower row: Reconstructed models; detected windows are shown in yellow, doors are shown in green. Most wall elements are faithfully reconstructed; some excess walls have not been removed (see e.g. the large room in the lower-right corner of the second column).

## Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.cag.2015.07.008>.

# Automatic Reconstruction of Fully Volumetric 3D Building Models from Point Clouds

---

**Abstract.** We present a novel method for reconstructing parametric, volumetric, multi-story building models from unstructured, unfiltered indoor point clouds by means of solving an integer linear optimization problem. Our approach overcomes limitations of previous methods in several ways: First, we drop assumptions about the input data such as the availability of separate scans as an initial room segmentation. Instead, a fully automatic room segmentation and outlier removal is performed on the unstructured point clouds. Second, restricting the solution space of our optimization approach to arrangements of volumetric wall entities representing the structure of a building enforces a consistent model of volumetric, interconnected walls fitted to the observed data instead of unconnected, paper-thin surfaces. Third, we formulate the optimization as an integer linear programming problem which allows for an exact solution instead of the approximations achieved with most previous techniques. Lastly, our optimization approach is designed to incorporate hard constraints which were difficult or even impossible to integrate before. We evaluate and demonstrate the capabilities of our proposed approach on a variety of complex real-world point clouds.

This chapter corresponds to the paper *Automatic reconstruction of fully volumetric 3D building models from point clouds*, Sebastian Ochmann, Richard Vock and Reinhard Klein, submitted to ISPRS Journal of Photogrammetry and Remote Sensing 2018.

## 5.1 Introduction

The challenging problem of generating high-quality, three-dimensional building models from point cloud scans has been approached in a variety of ways in recent years by the computer graphics and remote sensing communities as well as in the architecture domain. Especially for various applications in Computer Aided Design (CAD) and emerging fields such as Building Information Modeling (BIM), the reconstructed models are usually required to adhere to industry-standard specifications such as the Industry Foundation Classes (IFC). In contrast to the representation of a building in the form of e.g. an unordered point cloud, a set of unconnected surfaces, or boundary meshes, a BIM/IFC model closely resembles the physical building structure by defining buildings as semantically annotated, volumetric building entities such as walls and floor slabs, usually including additional information regarding how these elements are interconnected.

Most previous approaches focus on the reconstruction of completely separate, planar surfaces without additional information regarding how they relate to each other [20], or on representing buildings as watertight boundaries of either the whole building [24] or separate rooms [22, 59] and are thereby lacking in providing insights into the building structure. Also, assumptions such as the one that stories can be globally separated by horizontal planes are very limiting in practice. None of these approaches yields a representation which enables unhindered usage in the aforementioned scenario. While one recent approach [67] does model the measured point cloud data using volumetric building entities, the method is restricted to single-story buildings which limits its usability without laborious manual separation of the point cloud data into separate stories. Additionally, generation of resulting wall and floor slab elements is done in a post-processing step without being integrated into the used optimization framework which may result in locally implausible results. Other methods [29, 68] aiming at reconstructing true BIM models make the severe assumption that walls are positioned in a Manhattan world constellation which is often violated by real-world buildings.

Our proposed method overcomes limitations of previous approaches by alleviating the requirements on the input data and by providing a flexible optimization framework for indoor building reconstruction. Some prior methods (e.g. [22, 67]) require separate scans and scan positions to derive an initial, coarse segmentation into rooms. In contrast, our fully automatic room segmentation approach, by design, does not depend on the availability of such information and does not impose particular rules for scanning (e.g. one scan per room). Furthermore, our novel integer linear programming formulation for the reconstruction problem provides flexible means to steer the reconstruction process while globally constraining the solution space to feasible solutions, thus guaranteeing a plausible model. Additional information such as manually augmented hints may optionally be incorporated by means of hard constraints in order to further guide the reconstruction process. While some previous approaches regularize the resulting model based on room boundary complexity, they fail to account for dependencies between surfaces related by volumetric wall elements, e.g. opposing surfaces between neighboring rooms. Our formulation of the solution space based on volumetric entities enables better regularization of the model with respect to the actual volumetric walls and slabs used to represent the building. In contrast to any previous approach, the result of our optimization immediately yields the complete geometry of all reconstructed walls and slabs as well as their volumetric intersections which allows for a direct generation of plausible BIM/IFC models.

In summary, the main features of our approach are:

1. Fully automatic, volumetric reconstruction including volumetric intersections between elements.
2. Flexible integration of constraints to enforce global and local properties of the resulting model.

Our main technical contributions are:

1. Automatic filtering of outliers and room segmentation of unstructured, multi-story 3D point clouds.
2. A new formulation of the indoor reconstruction task as a linear integer programming problem that can be efficiently solved using off-the-shelf software.

## 5.2 Related Work

Research on scan-to-BIM and related approaches led to a wide range of developments in recent years and still is a current topic of ongoing work. We first provide a comprehensive overview of methods dealing specifically with indoor building reconstruction which we then complement with a summary of more loosely related but complementary abstraction approaches and applications.

	No scan positions	Non-Manhattan	Multiple rooms	Full 3D recons. <sup>1</sup>	Slanted ceilings	Volumetric walls
Budroni '10 [48]	✓	✗	✗	✗	✗	✗
Adán '11 [14]	✗	✓	✓	✗	✗	✗
Xiong '13 [58]	✗	✓	✓	✗	✗	✗
Mura '14 [23]	✗	✓	✓	✗	✗	✗
Oesau '14 [24]	✓	✓	✗	✗	✗	✗
Previtali '14 [25]	✗	✓	✗	✗	✗	✗
Turner '15 [59]	✗	✓	✓	✗	✗	✗
Mura '16 [22]	✗	✓	✓	✓	✓	✗
Ochmann '16 [67]	✗	✓	✓	✗	✗	✓
Ambruş '17 [15]	✓ <sup>2</sup>	✓	✓	✗	✗	✗
Macher '17 [28]	✓	✓	✓	✗	✗	✓ <sup>3</sup>
Murali '17 [29]	✓	✗	✓	✗	✗	✓
Wang '17 [26]	✗ <sup>4</sup>	✓	✓	✗	✗	✗
<b>Ours</b>	✓	✓	✓	✓	✗	✓

Table 5.1: Feature comparison of recent indoor reconstruction approaches. Notes: <sup>1</sup>Full 3D reconstruction means that arbitrary vertical room constellations including rooms spanning multiple floors are handled correctly. <sup>2</sup>Virtual scan positions are automatically estimated. <sup>3</sup>Volumetric walls are generated manually in post-processing. <sup>4</sup>Scanner trajectories are used.

### 5.2.1 Indoor building reconstruction

The works presented in this section are closely related to our goal of indoor building reconstruction. Table 5.1 summarizes and compares key features of different approaches.

Some methods aim at the generation of 2D floor plans. Okorn et al. [57] model 2D floor plans by projecting detected structures into the horizontal plane and performing wall segment detection based on the Hough transform. Ambruş et al. [15] reconstruct floor plans including a room labeling obtained using an energy minimization approach. A deep neural architecture for automatic floor plan generation from RGBD video has been presented by Liu et al. [68]. Using pixel-wise predictions of floor plan geometry and semantics, integer programming [69] is used to recover a vector graphics reconstruction.

Some approaches perform a reconstruction of individual rooms. Budroni et al. [48] reconstruct closed boundary representations of single rooms using plane sweep surface detection and a 2D line arrangement with a split-and-merge approach. The methods by Adán et al. [14] and Xiong et al. [58] focus on recovering detailed surface labelings, explicitly reasoning about occlusions using a ray-tracing approach. In a similar spirit, Previtali et al. [25] perform a reconstruction of single rooms as polyhedral models including ray-tracing based reasoning about occlusions and opening detection.

Certain methods aim at the reconstruction of the building as a whole without explicitly considering room topology or segmentation. Sanchez et al. [20] represent buildings as polygonal surface models including detection of smaller-scale structures such as parametric staircases. Oesau et al. [50] use a 2D cell decomposition to perform binary inside/outside labeling using a Graph-Cut based optimization. The detail level of this approach is enhanced by Oesau et al. in [24] by means of an improved line detection strategy. With a similar goal of providing simplified environment maps for e.g. navigation, Xiao et al. [41] employ constructive solid geometry (CSG) operations to generate a volumetric wall model. Room topology is not explicitly modeled.

Many recent methods approach the reconstruction problem in a 2.5D setting, including a segmentation into separate rooms. Mura et al. [23, 44] model buildings as 2.5D polyhedral meshes by means of constructing a 2D line arrangement and performing  $k$ -medoid clustering based on diffusion embeddings. Mura et al. [60] also propose a related approach which allows arbitrary wall orientations and performs

recursive clustering on a constrained Delaunay tetrahedralization. The method by Turner et al. [46] provides efficient means to generate 2.5D, textured meshes for e.g. navigation purposes including a room segmentation obtained by Graph-Cut in a triangulated environment map. An extension providing enhanced texture mapping has been presented in [59]. The reconstruction method by Wang et al. [26] models outer and inner walls by means of 2D line arrangements labeled using diffusion embeddings similar to [23]. They also reconstruct doors using a simulated ray casting approach. Murali et al. [29] present a system to quickly generate BIM models from mobile devices such as Google Project Tango, Microsoft Kinect or Microsoft HoloLens, including semantic annotations and relations between reconstructed elements. The approach is currently limited to single-story, Manhattan world buildings.

Few approaches consider the more general case of slanted walls or ceilings. Mura et al. [22] reconstruct polyhedral room boundaries with arbitrary wall and ceiling orientations. Early rule-based classification of detected elements helps pruning invalid parts. Room segmentation is performed by clustering steered by visible surface overlap. Mura et al. [70] propose an extension using automatically clustered synthetic viewpoints and show applicability on complex multi-story buildings.

None of the aforementioned methods reconstruct volumetric wall and slab elements which are directly usable in a BIM setting. Few methods have approached this problem before. Stambler et al. [27] aim to generate volumetric 3D building models using learning approaches for the classification and scoring of detected elements, and simulated annealing for optimizing the overall model. The approach makes strong assumptions about the input data, requiring both interior and exterior scans, as well as scanner positions. Thomson et al. [71] generate volumetric walls from point clouds by detecting planes using a RANSAC approach and fitting suitable IFC wall entities to the detected surfaces; room volumes and topology are not explicitly modeled. They also propose a point cloud segmentation scheme based on a corresponding IFC model. A method which explicitly represents buildings as interconnected volumetric wall elements has been presented by Ochmann et al. [67]. They construct a 2D line arrangement of wall center lines representing pairs of opposing wall surfaces and perform a room labeling of the arrangement faces by means of a Graph-Cut based multi-label energy minimization. Multi-story buildings are not supported. Macher et al. [28] propose a semi-automatic reconstruction approach by first segmenting the input data automatically and exporting the result in an interim OBJ format, and subsequently constructing an IFC file with manual intervention in a post-processing step.

To our knowledge, our approach is the first to combine general multi-story, multi-room reconstruction with fully volumetric room and wall entities.

### 5.2.2 Abstraction, segmentation, and reconstruction

We now highlight some loosely related approaches which pursue more general or complementary goals which may be beneficial for tackling the reconstruction problem on different levels. Monszpart et al. [21] represent man-made scenes (e.g. buildings) by a regular arrangement of planes, taking into account non-local inter-primitive symmetry relations. Such a regularization may be useful for various arrangement-based reconstruction approaches. A method for reconstructing lightweight, manifold, polygonal boundary models from point clouds has been presented by Nan et al. [72]. They employ an inside/outside labeling approach using binary linear programming. Jung et al. [73] generate watertight floor maps by means of skeletonization in a 2D binary occupancy map with subsequent labeling of separate rooms. A 3D room partitioning approach using anisotropic potential fields with subsequent unsupervised clustering has been presented by Bobkov et al. [74]. Pursuing a similar goal, Ochmann et al. [45] perform a segmentation of indoor point clouds into separate rooms using a visibility-based approach. Openings between neighboring rooms are detected to obtain a room connectivity graph. Bassier et al. [17] employ a machine learning approach to classify structural elements such as walls,

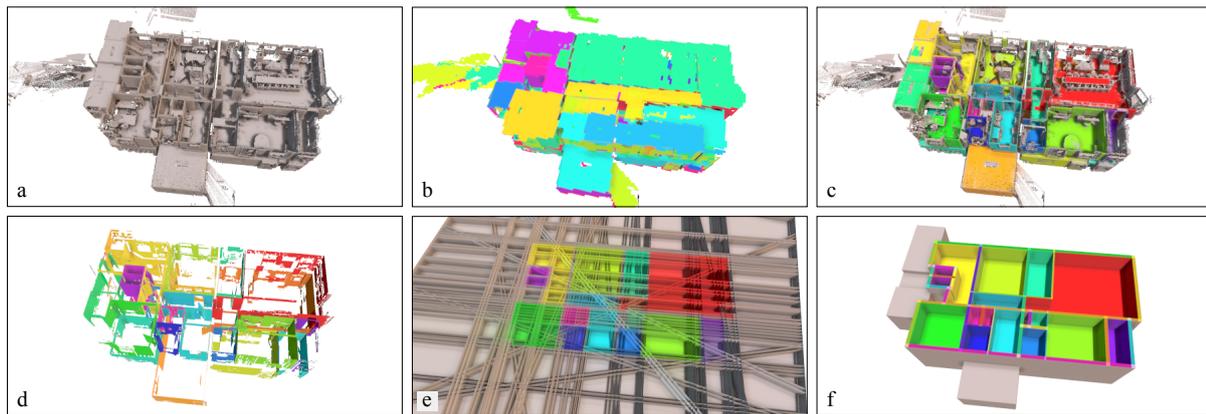


Figure 5.1: Overview of the main steps. Ceiling of upper floor is hidden in (a), (c), and (f) for visualization purposes. (a) The input is a registered but otherwise unstructured and unfiltered indoor point cloud. (b) Planes are detected by means of RANSAC shape detection. (c) Outliers are automatically removed and rooms are segmented using an unsupervised clustering approach based on mutual visibility between point patches. (d) Detected planes are classified as horizontal slab surfaces and vertical wall surfaces (only latter shown). Surfaces are assigned multi-label support bitmaps. (e) A 3D plane arrangement is constructed by intersecting all planes, yielding a cell complex. Priors for rooms, outside area and surface support are estimated. (f) The final model consisting of interrelated room and wall volumes is obtained by solving an integer linear program in which cell labels are binary variables.

floors, ceilings, and beams in point cloud data. The method by Liu et al. [69] generates topologically and geometrically consistent floor plans from 2D raster images using an integer programming approach. While the approach is designed to work on 2D data and assumes Manhattan world geometry, the idea to enforce global properties of the resulting model using integer programming is related to our work. Focusing on non-structural elements relevant to BIM models, Adán et al. [19] present an approach for detecting various important entities such as sockets, switches, signs, and safety-related items. While the method by Son et al. [16] does not explicitly model a building's room topology, they detect various important volumetric elements such as walls, slabs, columns and beams, also taking into account material properties and relations between elements.

### 5.2.3 Applications

Automated scan-to-BIM methods facilitate a range of diverse applications in different areas such as construction surveillance, facility management, or energy simulations. Garwood et al. [75] propose a framework for storing building geometry in a format suitable for e.g. energy simulation and verification tasks, and highlight the importance of fast, automated methods for obtaining suitable models. Hyland et al. [76] propose the usage of open standards and automatically derived BIM models from measurements for performing automated compliance control by comparing the as-built and as-designed states of buildings. In a similar spirit, O'Keeffe et al. [77] have developed validation approaches for determining and analyzing differences between scans and BIM models. A prototypical approach has been presented by Brodie et al. [78] who propose a cloud-based platform integrating tools for generating models from and validating models against point clouds. Krispel et al. [79] developed a method for automatic detection of power sockets and for the generation of hypotheses for electrical lines based on automatically generated building models. An approach for integrating IFC BIM models and point cloud data in a common file

format has been presented by Krijnen et al. [80]. They highlight the semantically meaningful association of both worlds for documentation, structuring, annotation, synchronization and retrieval tasks.

### 5.3 Overview

The input of our approach is a 3D indoor point cloud (Figure 5.1 a) with oriented normals whose “up” direction is assumed to be the  $z$ -axis. If normals are not yet available, they are estimated by local Principal Component Analysis (PCA).

We first detect planes using an efficient RANSAC implementation [13] (Figure 5.1 b) and compute occupancy bitmaps for each detected plane from the respective supporting points.

The detected planes are used to automatically eliminate outlier points, and to determine point clusters corresponding to individual rooms. This clustering is performed by means of Markov Clustering [81] which does not require prior information about the number of rooms and results in a labeling of the point cloud (Figure 5.1 c).

The resulting point labels are projected to the previously detected planes and discretized into multi-label bitmaps. Planes are pruned, rectified, clustered, and classified as candidates for vertical wall or horizontal slab surfaces (Figure 5.1 d; only vertical surfaces shown for visualization purposes). Since we base our reconstruction on volumetric walls and slabs instead of single surfaces, pairs of nearby, approximately parallel surfaces are grouped to wall and slab candidates.

Based on promising previous approaches (e.g. [15, 22, 67, 70]), we then derive a three-dimensional *arrangement of planes* from the set of wall and slab candidates (Figure 5.1 e). To this end, all surfaces are interpreted as infinite planes and intersected with each other which results in a segmentation of 3D space into convex polyhedral cells. In particular, each wall and slab candidate is represented by a set of cells located between the respective two candidate surfaces. Priors for the existence of different rooms and wall surfaces are estimated for each 3D cell and 2D face using the labeled surface candidates.

The main step of our approach is to find a *labeling* of all cells such that each cell is either assigned to a *room*, or *outside space*. Additionally, volumetric *walls* must be placed wherever a transition between inside and outside space takes place which is also modeled as part of the labeling problem. The labeling should faithfully conform to the measured data and simultaneously fulfill certain constraints (e.g. wall connectivity) to ensure a plausible resulting model (Figure 5.1 f).

Formulating this task as an optimization problem requires three parts: First, we define a space of possible solutions with meaningful priors to guide the solver. The geometry of this space is given by the arrangement of planes. Priors for locations of rooms and walls in the cell complex are derived from the measured data. Second, we need to define constraints to restrict the feasibility of a solution. They enforce that any solution satisfies predefined rules, e.g. a room and outside space must be separated by a wall. Third, an objective function for assessing the quality of a solution is formulated as a cost function which is minimized under the given constraints.

After a solution is found it can easily be converted into a format suitable for rendering or exporting, e.g. an IFC file or a mesh, by considering the cell labeling and the boundaries between differently labeled cells.

### 5.4 Method

In this Section, we provide details regarding each of the steps involved in our approach with a focus on the formulation as an optimization problem.

### 5.4.1 Plane detection

Based on the widely used assumption that the coarse geometry of most buildings can be represented (or sufficiently approximated) by piecewise planar surfaces, a crucial first step of our approach is the detection of planes in the point cloud data. To this end, an efficient RANSAC approach [13] implemented in CGAL [82] is used. The most important parameters are maximum point-to-plane distance, normal angle threshold, minimum number of supporting points per plane, and the probability to miss the largest plane candidate. These can usually be chosen depending on point cloud data quality and used for a wide variety of datasets with similar characteristics (e.g. scanner type, density, noise level). The supporting points of each plane are projected into occupancy bitmaps on the respective plane (Figure 5.2 b), yielding a discretized approximation of support by measured points. Planes with low support area (estimated using the occupancy bitmaps) are pruned later (Section 5.4.4). Since the relatively coarse occupancy bitmaps are independent of the point cloud density, the minimum number of points for detecting a plane of the RANSAC algorithm may be set relatively low to cope with lower-resolution point clouds.

### 5.4.2 Point cloud cleaning

Real-world point clouds often contain large amounts of outlier points, often due to outside areas scanned through openings. In order to prune outlier points early in the process, we employ a simple but very effective ray casting approach similar to [67]. From each point  $p$ ,  $n$  stochastically sampled rays  $r_i$ ,  $i = 1, \dots, n$ , are cast into the hemisphere oriented into the direction of the normal at point  $p$ . Ray casting is performed against the occupancy bitmaps of the previously detected primitives. Let  $h(r_i)$  be a *hit* function which is 1 if some surface was hit, and 0 otherwise. We approximate the probability that  $p$  lies inside of the building as  $in(p) = \frac{1}{n} \sum_{i=1}^n h(r_i)$ . If  $in(p)$  is below a given threshold (in our experiments 0.5),  $p$  is removed from the point cloud and the occupancy bitmaps of the planes are updated. This process is iterated a small number of times.

### 5.4.3 Point cloud labeling

Priors for the locations of rooms and outside area in three-dimensional space are vital for the later optimization step, even if they are coarse estimations. We formulate the estimation of priors as a point cloud labeling problem where each label represents either a room, or the outside area.

Our proposed automatic labeling approach is based on the idea that regions of the point cloud with high mutual visibility form clusters which correspond to rooms of the building. We implement this by performing visibility tests by means of ray casting between point patches on detected surfaces which yields a visibility graph. Nodes of this graph are then clustered by means of the Markov Clustering algorithm [81] which determines natural clusters within the graph by flow simulation.

Point patches are constructed by generating coarse occupancy bitmaps for each plane and considering each occupied pixel as a patch with a normal identical to the respective plane normal. In our experiments, a patch size of  $40\text{cm} \times 40\text{cm}$  was used. We use patches instead of all points to drastically reduce the number of nodes in the visibility graph which makes the computation feasible. Let  $p_i$  be the  $i$ -th patch with center position  $c_i$  and normal  $n_i$ . For each pair  $p_i, p_j$ ,  $i \neq j$ , ray casting between the points  $c_i + \varepsilon n_i$  and  $c_j + \varepsilon n_j$ , with  $\varepsilon := 10\text{cm}$  in our experiments, is performed. If no surface is hit, the visibility between  $p_i, p_j$  is set to 1, otherwise it is set to 0. This yields a visibility graph whose nodes are clustered using the Markov Clustering algorithm. The computed visibility is interpreted as flow between node pairs corresponding to the respective point patches. The main advantage of this method is that it is unsupervised and thus does not require a manual specification of the number of occurring labels.

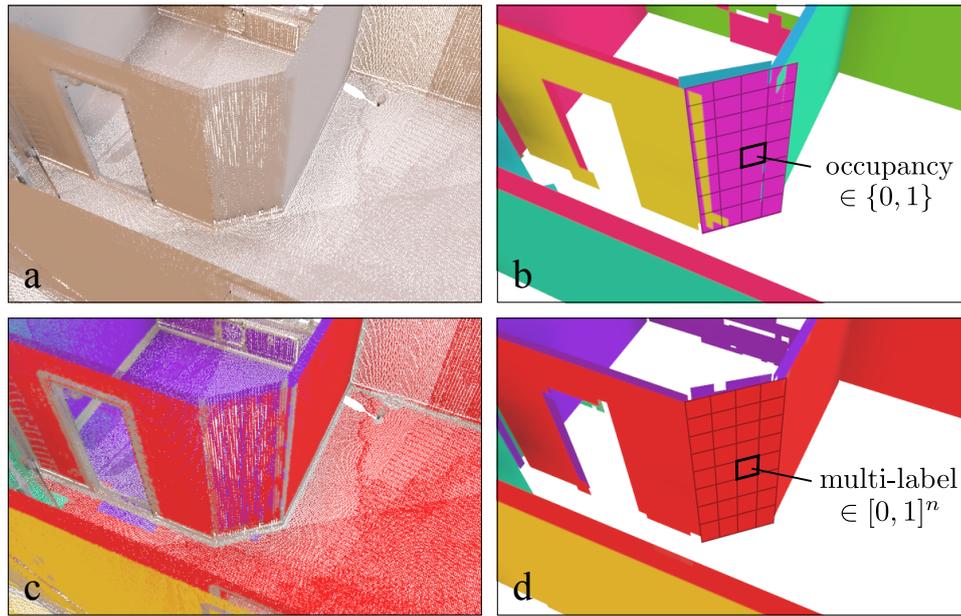


Figure 5.2: Detected planes are the basis for our reconstruction. Different kinds of bitmaps (i.e. grids) on the planes are used throughout the approach. (a) Unlabeled input point cloud. (b) Binary occupancy bitmaps on the detected planes (Section 5.4.1) are used as a lightweight representation of support by the point cloud. Different planes are shown in different colors. (c) Ray casting against the occupancy bitmaps and clustering yields a segmentation of the point cloud into rooms (Section 5.4.3). Different room labels are shown in different colors. (d) The point cloud labeling is projected into multi-label bitmaps where each pixel contains a soft assignment  $[0, 1]^n$  to the  $n$  different room labels as defined in Section 5.4.3. This is used for estimating locations of rooms in 3D space as described in Section 5.4.7. The multi-label bitmaps are shown using the same label colors as in (c). In (b) and (d), only vertical planes are shown for clarity.

As a result, we obtain  $n$  disjoint clusters of patches which belong to different rooms and define the set of room labels  $\mathcal{R} := \{r_1, \dots, r_n\}$  which will be used throughout the remainder of the reconstruction process. Each point of the point cloud is assigned the room label of the respective point patch. Note that the number of room labels  $n$  may be larger than the number of rooms that will actually be contained in the final reconstruction.

#### 5.4.4 Surface candidates

The detected planes usually include many surfaces which are not part of walls, floors and ceilings. Even correctly detected surfaces will generally not be perfectly vertical or horizontal. We thus apply a pruning, classification and rectification step to extract two sets of candidates for wall and slab surfaces. The occupancy bitmaps are used to estimate the support area of each surface independently of point cloud density. Planes with support below an area threshold as well as planes which are not approximately vertical or horizontal are discarded. The remaining planes are classified as wall or slab surface candidates depending on their normal direction, and adjusted to be perfectly horizontal or vertical.

As a prerequisite for later room prior estimation (Section 5.4.7), each surface is also assigned a multi-label support bitmap with continuous values in  $[0, 1]$  for each room label in  $\mathcal{R}$  (Figure 5.2 d). This provides a soft-assignment of different regions of each surface to different room labels. The label bitmaps are generated by projecting all supporting points onto the respective surface and averaging the previously determined point labels within each pixel.

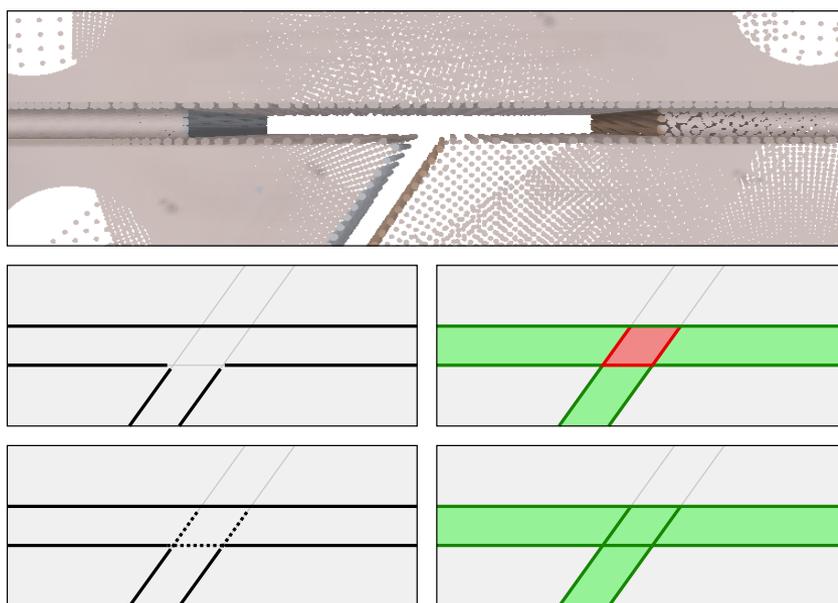


Figure 5.3: Dilation of surface support. Top: Example point cloud viewed from above. Middle: Surface support (left, thick lines) is naturally restricted to parts visible to the scanner. This leads to high costs for the wall intersection since wall surfaces are not supported by points (right). Bottom: Dilating the surface support (left, dotted lines) extends support into the interiors of walls, encouraging placement of volumetric intersections (right).

Furthermore, we *dilate* the support bitmaps. The rationale is that reconstructed walls with no surface support by the point cloud data are penalized by a cost function defined later in Section 5.4.8. Since we reconstruct wall intersections volumetrically, placing the respective wall entities in between rooms would cause high costs since surface support is naturally restricted to regions that are visible to the scanner (Figure 5.3, middle row). By slightly extending the surface support, we encourage construction of intersecting wall entities in regions with nearby surface support (Figure 5.3, bottom row).

#### 5.4.5 Wall and slab candidates

Since our approach is based on the notion of *volumetric* walls and slabs instead of single surfaces, the next step is to determine pairs of opposing surfaces forming potential building elements. To this end, a simple pairing procedure is employed. For each surface, we search a matching, approximately parallel surface with opposing normal orientation within a user-defined distance and angle threshold. If a match is found, the two surfaces are paired to form a wall or slab candidate. It should be noted that a single surface may thus be part of multiple pairs. For surfaces without any matching counterpart, *virtual* surfaces with a user-defined distance are added to the set of surfaces. This is usually the case for outside walls for which only the inner side has been scanned. This augmentation is important since interior spaces are required to be bounded by volumetric walls or slabs. The  $m$  generated candidates constitute the set of wall/slab labels  $\mathcal{W} := \{w_1, \dots, w_m\}$ . Which of these walls and slabs are contained in the final model is decided by the optimization described in Section 5.4.8.

#### 5.4.6 Arrangement of planes

The geometry of the search space for finding an optimal constellation of rooms, walls and slabs is modeled as an *arrangement of planes* and the *3D cell complex* induced thereby. It is constructed by

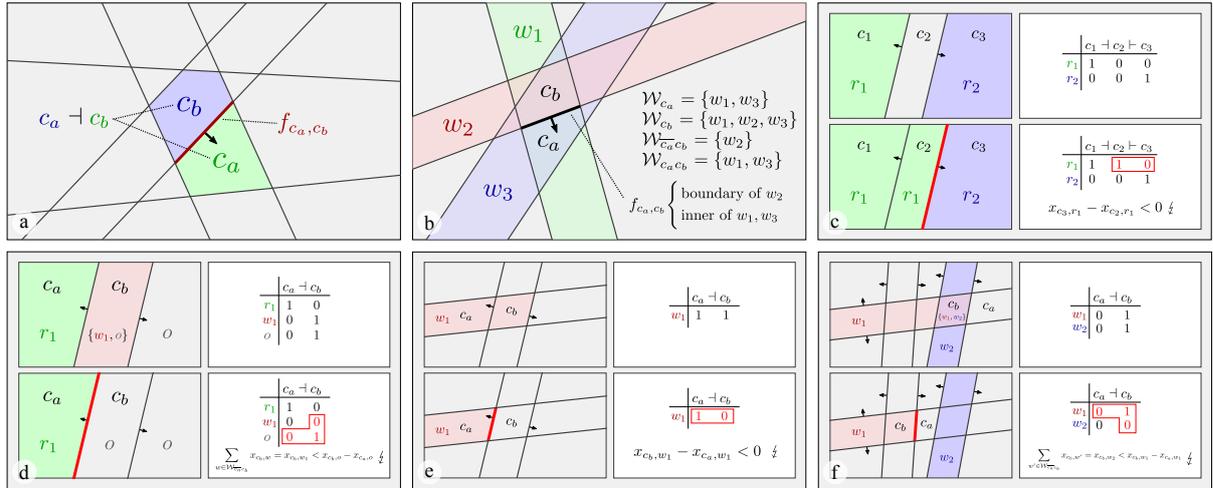


Figure 5.4: Explanation of notation and constraints. (a) Neighboring cells are considered as ordered pairs  $c_a \dashv c_b$  with respect to normal orientation of the separating face  $f_{c_a, c_b}$ . (b) Considering a cell  $c$ ,  $\mathcal{W}_c$  is the set of wall candidates enclosing cell  $c$ . A face  $f_{c_a, c_b}$  may be a boundary or an inner face of a set of walls. The set of boundary walls with respect to that face is  $\mathcal{W}_{\bar{c}_a c_b}$ , the set of inner walls is  $\mathcal{W}_{c_a c_b}$ . (c) Transitions between interior and exterior area at a face  $f$  may only occur with the room label being on the positive side of  $f$  (Constraint 2). (d) If a face  $f$  is the boundary of a room, some wall needs to be active on the negative side of  $f$  (Constraint 4). (e) A wall may end at an inner face  $f$  only if the wall label is on the negative side of  $f$  (Constraint 5). (f) If a wall  $w_1$  ends at an inner face, this face must be a boundary face of at least one other active wall  $w_2$ . This enforces connectedness between walls.

intersecting all (infinite) planes of the wall and slab candidate surfaces with each other. Since vertical walls and horizontal slabs are treated identically, we will hereafter refer to both simply as *walls*.

Cells of the arrangement are convex, three-dimensional subsets of the space inside and outside of the building. Each cell belongs either to a room, or the outside area. Additionally, walls may be placed in cells that are part of the outside area. Constraints such as that a cell may belong to at most *one* room, or that walls may *only* occur in the outside area (e.g. between rooms) are formulated as constraints for the optimization problem in Section 5.4.8.

Faces between neighboring cells are convex, two-dimensional subsets of regions on the planes of wall surfaces. Each face may separate different regions (e.g. a room and a wall) from each other.

### 5.4.7 Volume and surface priors

For guiding the optimization, two kinds of priors are estimated from the data. First, volumetric priors for the existence of different rooms as well as outside area are estimated for each 3D cell of the arrangement. Second, support by the point cloud data is estimated for each 2D face between neighboring cells.

**Preparations** The arrangement consists of cells  $C := \{c_1, \dots, c_p\}$ . For two cells, the notation  $c_a \dashv c_b$  means that  $c_a, c_b$  are neighboring and the normal of the separating oriented face  $f_{c_a, c_b}$  points towards  $c_a$  (Figure 5.4 a). The set of all oriented faces is denoted as  $\mathcal{F} := \{f_{c_a, c_b} \mid c_a \dashv c_b\}$ . For brevity, we write  $f$  instead of  $f_{c_a, c_b}$  if the specific incident cells are irrelevant.

The set of room labels is  $\mathcal{R} := \{r_1, \dots, r_n\}$  with  $n$  being the number of room clusters as introduced in Section 5.4.3, and the set of wall labels is  $\mathcal{W} := \{w_1, \dots, w_m\}$  with  $m$  being the number of generated wall candidates as introduced in Section 5.4.5. We furthermore define an additional *outside* label  $O := \{o\}$ . As

detailed later, the outside label is used for cells that are not the interior space of a room. The union of rooms and outside labels is denoted  $\mathcal{R}_o := \mathcal{R} \cup \mathcal{O}$ ; the set of all labels is  $\mathcal{L} := \mathcal{R} \cup \mathcal{O} \cup \mathcal{W}$ .

Let  $C_w \subseteq C$  be the set of cells that are *contained* in wall candidate  $w$ , i.e. all cells that are located between the two surfaces of  $w$ . Conversely,  $\mathcal{W}_c := \{w \in \mathcal{W} \mid c \in C_w\}$  is the set of walls that *contain* cell  $c$ .

For a particular cell pair  $c_a, c_b$ , we define the set  $\mathcal{W}_{\overline{c_a c_b}}$  of walls that are contained in cell  $c_b$  but *not* in  $c_a$ , i.e.

$$\mathcal{W}_{\overline{c_a c_b}} := \mathcal{W}_{c_b} \setminus \mathcal{W}_{c_a}.$$

The separating face  $f_{c_a, c_b}$  is called a *boundary* face of the walls in  $\mathcal{W}_{\overline{c_a c_b}}$ . Analogously, we define the set  $\mathcal{W}_{c_a c_b}$  of walls that are contained in *both*  $c_a$  and  $c_b$ , i.e.

$$\mathcal{W}_{c_a c_b} := \mathcal{W}_{c_a} \cap \mathcal{W}_{c_b}.$$

The separating face  $f_{c_a, c_b}$  is called an *inner* face of the walls in  $\mathcal{W}_{c_a c_b}$ . These definitions are exemplified in Figure 5.4 b.

It should be noted that an inner face of a wall  $w$  is always the boundary face of another wall (which is often approximately perpendicular to  $w$ ). As an example, in Figure 5.4 b, face  $f_{c_a, c_b}$  is an inner face of wall  $w_1$  and a boundary face of the intersecting wall  $w_2$ . This will become important for the definition of the optimization constraints in Section 5.4.8.

**Room and outside priors** To estimate probabilities where different rooms and outside area are located in 3D space, we estimate a volumetric prior function  $p_C(c, l) : C \times \mathcal{R}_o \rightarrow [0, 1]$  which returns a high value iff a label  $l$  is likely to occur within a cell  $c$ . To this end, we perform stochastic ray casting from sampled points in 3D space and average previously computed room labels on surfaces visible from each point. For each cell  $c$ ,  $k$  random points are sampled within  $c$ . To draw enough samples for narrow cells, which are very common due to parallel surfaces,  $k$  is chosen proportional to

$$\max(\text{volume}(c), \text{diameter}(c)).$$

Centered at each sampled point,  $d$  rays are cast into random directions.  $p_C(c, l_i), i = 1, \dots, n$ , is then set to the average over all observed room labels. Rays hitting the back side of surfaces, as well as rays without surface intersections, are counted as *outside*.

**Face support priors** In addition to the volumetric room and outside prior function, we estimate a face support function  $p_{\mathcal{F}}(f) : \mathcal{F} \rightarrow [0, 1]$  which returns a high value iff a face  $f$  is supported by the point cloud. This function is later used for selecting probable wall candidates and regularizing the optimization result. To estimate  $p_{\mathcal{F}}(f)$  for a face  $f$ , we first sample  $k$  random points within  $f$  where  $k$  is proportional to

$$\max(\text{area}(f), \text{diameter}(f)).$$

Subsequently, all sampled points are projected onto the surface from which face  $f$  was generated in the arrangement.  $p_{\mathcal{F}}(f)$  is then set to the ratio between the number of sampled points lying within the support approximated by the occupancy bitmap of the respective surface to the total number of sampled points.

### 5.4.8 Cell complex optimization

For finding an optimal labeling of all cells, we employ a 0-1 integer linear programming approach in which binary variables for each cell are interpreted as room, outside, and wall label assignments to cells. This approach has the advantage that a set of rules to be fulfilled by any feasible solution can be formulated as hard constraints. Approximate multi-label methods based on e.g. Graph Cuts [62] are more restricted regarding the family of objective functions and constraints that can be used and may fail to find good solutions if the objective is not sufficiently smooth. We first discuss the set of constraints imposed on our model before defining the objective function.

**Preparations** Each of the binary variables

$$x_{c,l} \in \{0, 1\}, \quad c \in \mathcal{C}, \quad l \in \mathcal{L},$$

of our optimization is a binary assignment of a label  $l$  to a cell  $c$ . A value of 1 means that the label is *assigned* or *active*. It should be noted that a cell is not necessarily assigned only a single label. In particular, a cell can be assigned the outside label *and* a nonempty set of wall labels at the same time as defined by the constraints below. Also, cells where walls intersect are assigned *all* labels of the intersecting walls. We also use the notion of *inner* and *boundary* faces as defined in Section 5.4.7.

*Constraint 1.* Each cell  $c$  must be assigned exactly one label from  $\mathcal{R}_o$ , i.e.

$$\forall c \in \mathcal{C} : \sum_{r \in \mathcal{R}_o} x_{c,r} = 1. \quad (5.1)$$

*Constraint 2.* At boundary faces of room interiors, the room label may only occur on the *positive* side of the separating face, i.e.

$$\forall f_{c_a, c_b} \in \mathcal{F} \quad \forall r \in \mathcal{R} : x_{c_a, r} - x_{c_b, r} \geq 0, \quad (5.2)$$

as shown in Figure 5.4 c. Note that this constraint implies that two different room labels  $r_p \neq r_q$  cannot be directly neighboring since this would violate the constraint for one of the room labels. As a consequence, this avoids “paper thin” walls between rooms since they must be separated by outside area, thereby following the physical nature inherent to walls.

*Constraint 3.* Wall labels may only occur in cells which are assigned the outside label, i.e.

$$\forall c \in \mathcal{C} \quad \forall w \in \mathcal{W}_c : x_{c,w} \leq x_{c,o}. \quad (5.3)$$

*Constraint 4.* The boundary faces of room interiors must also be the boundary faces of an active wall, i.e.

$$\forall f_{c_a, c_b} \in \mathcal{F} : \sum_{w \in \mathcal{W}_{\overline{c_a c_b}}} x_{c_b, w} \geq x_{c_b, o} - x_{c_a, o}, \quad (5.4)$$

as illustrated Figure 5.4 d. This constraint implies that there cannot be a transition between room interior and outside area without activating a wall at all faces where the transition occurs.

*Constraint 5.* At wall boundaries which occur at inner faces, the wall label must be on the negative side of the respective faces, i.e.

$$\forall f_{c_a, c_b} \in \mathcal{F} \quad \forall w \in \mathcal{W}_{c_a c_b} : x_{c_b, w} - x_{c_a, w} \geq 0, \quad (5.5)$$

as exemplified Figure 5.4 e. This constraint is a prerequisite for Constraint 6 as well as the objective function which require the left-hand side expression to be nonnegative.

*Constraint 6.* A wall may end at an inner face only if this face is a boundary face of at least one other active wall, i.e.

$$\forall f_{c_a, c_b} \in \mathcal{F} \quad \forall w \in \mathcal{W}_{c_a, c_b} : \sum_{w' \in \mathcal{W}_{c_a, c_b}^-} x_{c_b, w'} \geq x_{c_b, w} - x_{c_a, w}, \quad (5.6)$$

as depicted Figure 5.4 f. This constraint enforces that walls are interconnected at their endpoints since it disallows that a wall ends at an inner face without it coinciding with a boundary face of an active wall.

**Objective function** To determine the optimal labeling, we define a cost function  $F_C$  for a solution over the for cell complex  $C$  of the form

$$F_C := -R_C + \alpha(W_{\mathcal{F}_b} + W_{\mathcal{F}_i}), \quad (5.7)$$

consisting of the following terms. The volumetric room and outside area fitness term  $R_C$  rewards the assignment of the most likely labels for each cell  $c \in C$  and is defined as

$$R_C := \sum_{c \in C} \sum_{r \in \mathcal{R}_o} x_{c, r} \cdot p_C(c, r) \cdot \text{volume}(c), \quad (5.8)$$

where  $x_{c, r}$  denotes the binary variable for the assignment of label  $r$  to cell  $c$  and  $p_C(c, r)$  represents the volumetric room and outside prior (Section 5.4.7), weighted by the volume of cell  $c$ . Note that this term is included with a negative sign within  $F_C$  such that its value is being maximized. The wall face cost terms  $W_{\mathcal{F}_b}$  and  $W_{\mathcal{F}_i}$  penalize placement of walls in terms of the required boundary and inner face areas, respectively. This penalty is attenuated for faces with high support. The terms are defined as

$$W_{\mathcal{F}_b} := \sum_{f_{c_a, c_b} \in \mathcal{F}} \sum_{w \in \mathcal{W}_{c_a, c_b}^-} x_{c_b, w} \cdot (1 - p_{\mathcal{F}}(f_{c_a, c_b})) \cdot \text{area}(f_{c_a, c_b}), \quad (5.9)$$

and

$$W_{\mathcal{F}_i} := \sum_{f_{c_a, c_b} \in \mathcal{F}} \sum_{w \in \mathcal{W}_{c_a, c_b}} (x_{c_b, w} - x_{c_a, w}) \cdot (1 - p_{\mathcal{F}}(f_{c_a, c_b})) \cdot \text{area}(f_{c_a, c_b}), \quad (5.10)$$

where  $x_{c_a, w}, x_{c_b, w}$  are the binary variables for the assignment of the wall label  $w$  to the cells  $c_a, c_b$  respectively,  $p_{\mathcal{F}}(f_{c_a, c_b})$  is the face support prior (Section 5.4.7), and  $\text{area}(f_{c_a, c_b})$  is the area of face  $f_{c_a, c_b}$ . It should be noted that  $(x_{c_b, w} - x_{c_a, w}) \in \{0, 1\}$  due to Constraint 5. Also note that in Equation 5.9, it suffices to consider  $x_{c_b, w}$  since for a boundary face of wall  $w$ ,  $x_{c_a, w}$  does not exist (i.e.  $x_{c_a, w}$  can be considered to be zero).

We then minimize  $F_C$  s.t. Constraints 1-6 using the Gurobi Optimizer [83].

Note that in our experiments, we added the following constraint which gave a small performance improvement although it is already implied by Constraints 1-2. At boundary faces of outside area, the outside label may only occur on the *negative* side of the separating face, i.e.

$$\forall f_{c_a, c_b} \in \mathcal{F} : x_{c_a, o} - x_{c_b, o} \leq 0. \quad (5.11)$$

We attribute this slight performance improvement to heuristics used by the particular optimizer implementation.

### 5.4.9 Optimization result

The result is an assignment of each cell to either one room, or the outside area. Cells which are assigned the outside area may also be assigned a nonempty set of walls. On the one hand this provides a dense segmentation of space into rooms and outside space. Volumes to which multiple walls are assigned are (volumetric) intersections of the respective walls. Since the underlying data structure provides adjacency information between all cells, semantic information like room adjacency and wall incidence is immediately available, e.g. for navigation or simulation purposes. On the other hand this information is closely related to the definition of building elements in BIM formats like IFC. This enables immediate transfer of the results into standard architecture software and integration into existing BIM pipelines.

## 5.5 Implementation details

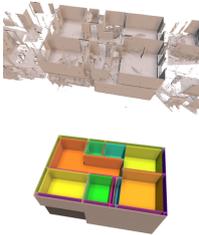
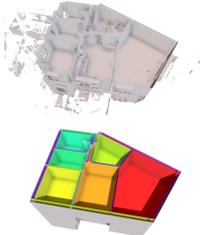
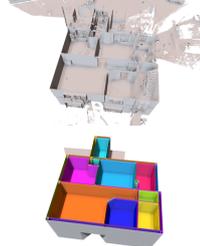
Input point clouds were subsampled to a minimum point distance of 2 cm. Plane detection was performed using a plane distance threshold of 1 cm, a point cluster epsilon of 20 cm, a normal threshold of about  $6^\circ$  ( $18^\circ$  for the “Case study 2” dataset), minimum support of 1000 points and miss probability of 0.001. Multi-label bitmaps had a resolution (pixel size) of 10 cm, occupancy bitmaps had a resolution of 20 cm. Three ray casting iterations were performed for point cloud cleaning. For automatic labeling, MCL was used with default parameters (inflation set to 2.0) in multi-threaded mode. The surface cost weight  $\alpha$  in Equation 5.7 was empirically chosen as 0.04. We used PCL 1.8.1 [66], CGAL 4.12 [82, 84], MCL 14-137 [81], Gurobi 8.0.1 [83], and NVIDIA OptiX 5.0 for GPU-based ray casting under Linux on a 6-core Intel i7 CPU and a NVIDIA GeForce GTX 980 GPU.

## 5.6 Evaluation

We evaluate the reconstruction quality and performance of our approach on a variety of datasets and show comparisons with groundtruth IFC and related work. Furthermore, we exemplify the flexibility of our integer linear programming approach by specifying additional constraints to modify and guide the resulting reconstruction in an intuitive manner.

**Datasets** We used a variety of real-world datasets and one synthetic dataset for our evaluation. Table 5.2 shows six multi-story point clouds measured using terrestrial laser scanners. These datasets were provided by The Royal Danish Academy of Fine Arts Schools of Architecture, Design and Conservation (CITA). The Table lists properties of the input data including the number of points and scans, as well as quantities derived during reconstruction such as the number of room labels, extracted surfaces, wall candidates, etc. It also shows runtime measurements of the main processing steps. We also tested our approach on publicly available datasets provided by other research groups. Figure 5.5 shows the dataset “synth3” by the Visualization and MultiMedia Lab at University of Zurich, Figure 5.10 depicts the dataset “Case study 2” from the ISPRS Benchmark on Indoor Modeling [85], and Figure 5.9 shows the dataset “Area 3” from the Stanford 3D Large-Scale Indoor Spaces Dataset [86]. We used the latter two for demonstrating different parameters and interactive modification as described below.

**Reconstruction quality** Our reconstruction approach generally worked well on the test datasets without any dataset-specific tuning. Automatic outlier removal reliably ignored even large-scale clutter scanned through windows in e.g. Datasets 1, 3, and 6. In some cases, particularly thick walls (bottom region of Dataset 1, top region of Dataset 2) were reconstructed as two thinner, parallel wall elements which

	Dataset 1	Dataset 2	Dataset 3
<b>Input</b>			
#scans / #points / #pts. cleaned	12 / 3168600 / 2702813	21 / 5151388 / 4723219	29 / 7688111 / 5874557
<b>#Entities</b>			
Room labels / Surfaces <sup>1</sup> / Walls <sup>1</sup>	27 / 42+5 / 37+5	30 / 34+7 / 28+5	39 / 51+5 / 39+4
Cells / Variables / Constraints	17666 / 594748 / 1775298	12749 / 459373 / 1334699	17334 / 781794 / 2261980
Nonzeros	4174634	3134503	5315578
<b>Runtime (seconds)</b>			
Plane detection	18.2	20.1	73.9
Cleaning (3 iterations)	14.1	21.9	32.7
Auto labeling	6.3	7.2	11.4
Arrangement + Priors	14.6	11.5	15.5
Optimization	20.9	7.4	9.8
			

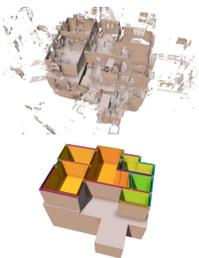
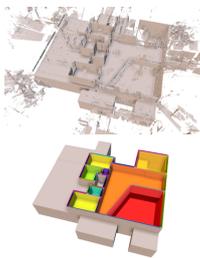
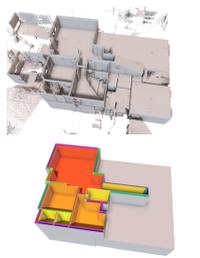
	Dataset 4	Dataset 5	Dataset 6
<b>Input</b>			
#scans / #points / #pts. cleaned	21 / 6452193 / 5627781	13 / 12409443 / 10688132	39 / 34964707 / 33687751
<b>#Entities</b>			
Room labels / Surfaces <sup>1</sup> / Walls <sup>1</sup>	49 / 61+7 / 58+5	29 / 48+8 / 46+7	108 / 72+5 / 66+5
Cells / Variables / Constraints	42262 / 2366866 / 7035794	35196 / 1254848 / 3757832	52701 / 6041984 / 17610275
Nonzeros	16535885	8837356	41350862
<b>Runtime (seconds)</b>			
Plane detection	51.1	82.6	218.6
Cleaning (3 iterations)	28.2	59.0	170.9
Auto labeling	16.1	56.8	84.5
Arrangement + Priors	38.9	29.1	53.6
Solving	85.4	42.9	182.8
			

Table 5.2: Evaluation results on various real-world datasets. The number of scans in the point cloud is shown for reference only and is not used in our approach. The number of points is the point count after subsampling. <sup>1</sup>The number of surfaces and walls is given as *vertical+horizontal*.

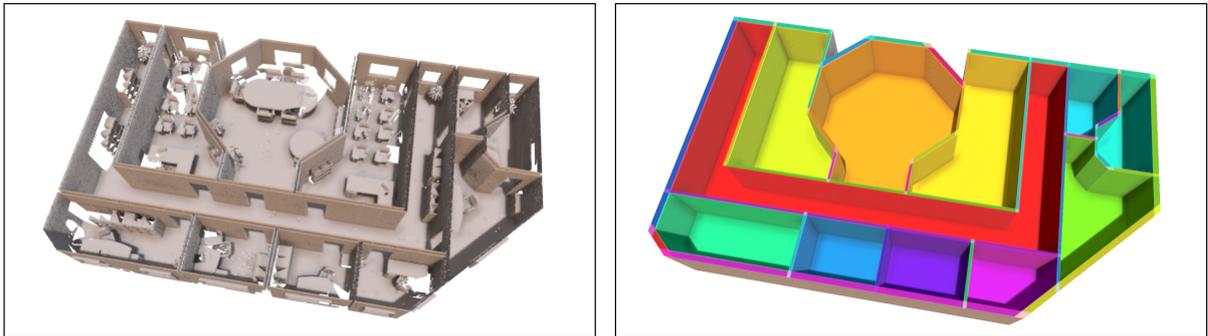


Figure 5.5: Our reconstruction result on the synthetic dataset “synth3” by the Visualization and MultiMedia Lab at University of Zurich. Rooms and walls are accurately reconstructed.

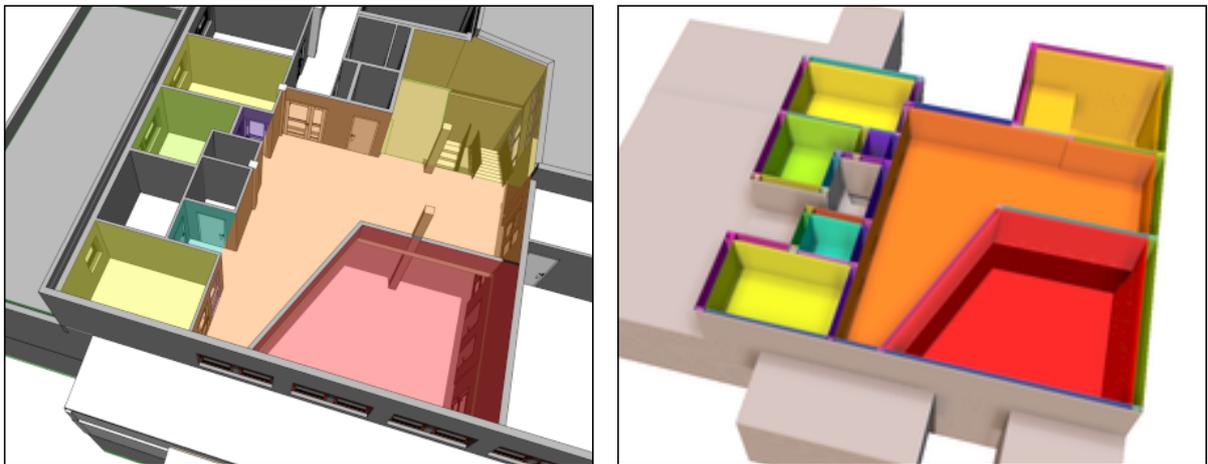


Figure 5.6: Comparison of a hand-crafted BIM model (left) and our reconstruction (right) of Dataset 5 (see Table 5.2). Reconstructed room labels were manually overlaid on the BIM model for reference.

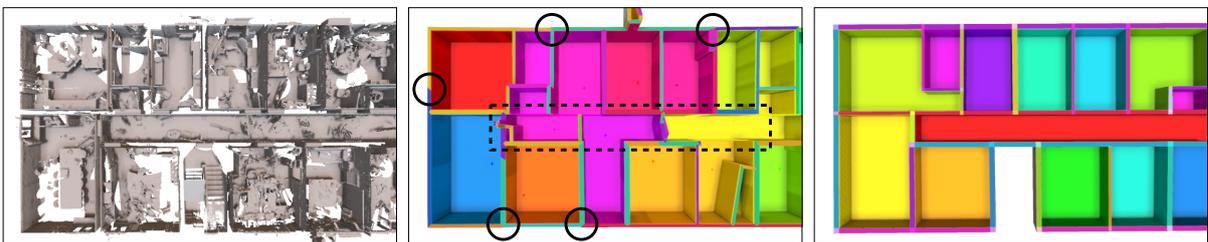


Figure 5.7: Comparison between different reconstruction approaches. Left: Input point cloud viewed from above. Center: The method described in [67] may fail to regularize chains of almost coplanar walls, leading to segmented walls (circles). Also, reliance on separate scans as initial room labeling may lead to oversegmented rooms (dashed rectangle). Right: Our approach overcomes these issues by incorporating costs for all surfaces of volumetric wall elements, and room segmentation that is independent of scan positions.

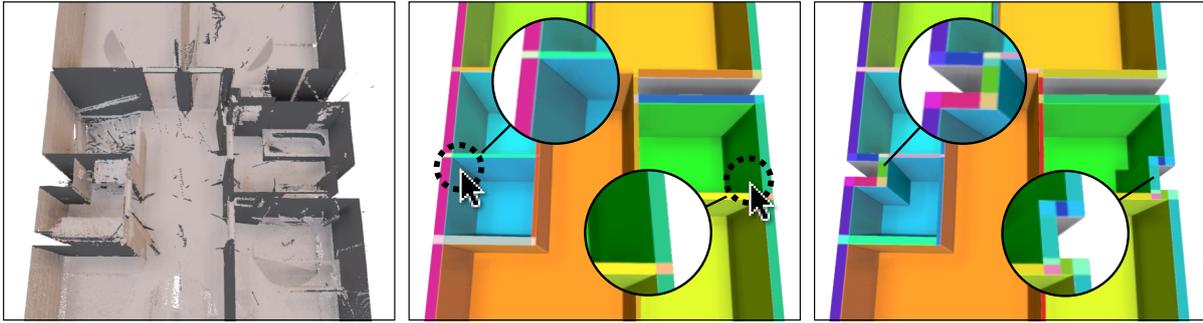


Figure 5.8: Additional constraints may be added to interactively steer the reconstruction. Small indentations of wall surfaces (left) are initially lost in the reconstruction. By forcing regions to be outside area (center), our method finds an alternative wall placement under these constraints (right).

may be a matter of interpretation. Increasing the maximum thickness of generated wall candidates in these cases can help recognizing such cases as single walls. A few cases of room-oversegmentation can be observed. In Dataset 4, the large central room is split into a larger L-shaped part (orange) and a smaller room (green, to the right of the building) without a real wall separating the reconstructed rooms in the point cloud data. In Dataset 5, indentations of the central room (orange) were reconstructed as small, separate rooms (cyan, purple). Since our approach currently only considers horizontal ceilings, the slanted ceiling of the staircase in Dataset 6 (yellow, elongated room) is reconstructed as a horizontal structure (see also Limitations below).

**Runtime** Total runtime for the reconstruction of the test datasets lies in the range of one minute (Datasets 1, 2) to 10 minutes (Dataset 6). The runtime of primitive detection is mainly dependent on the CGAL implementation, and the time for solving the optimization problem is the runtime of the Gurobi optimizer. The runtime for auto labeling contains the time for our raycasting and clustering using the Markov Cluster Algorithm. Runtime of the optimization mainly depends on the complexity of the plane arrangement, which in turn depends on the number of detected surfaces since every surface introduces global splits in the cell complex. Therefore a tradeoff between reconstructing details (i.e. small surfaces) and computational feasibility must be made. In our experiments, we thus chose a minimum estimated area of  $2 \text{ m}^2$  for vertical surfaces and  $5 \text{ m}^2$  for horizontal surfaces.

**Comparison to IFC** For Dataset 5 a corresponding, professionally made BIM model in IFC format was available. Figure 5.6 shows a comparison between our reconstruction and the BIM model. Colors of the reconstructed rooms were manually overlaid on the IFC model on the left-hand side. All rooms that were part of the scans mostly match the groundtruth BIM. The upper story of the building is connected to the lower story through a large horizontal opening. These areas were reconstructed as two separate rooms (red and orange) and the railing at the edge of the gallery was reconstructed as walls. The small cyan and purple rooms are an oversegmentation of the upper floor, probably due to the dilated surface support. However, this error can easily be fixed manually.

**Comparison to related work** A comparison between reconstructions by the approach described in [67] and our method is shown in Figure 5.7. In addition to fundamental advantages of our approach such as reconstruction of multiple stories, two crucial differences are particularly notable. First, our approach results in stronger regularization of wall elements where using multiple different, similar walls

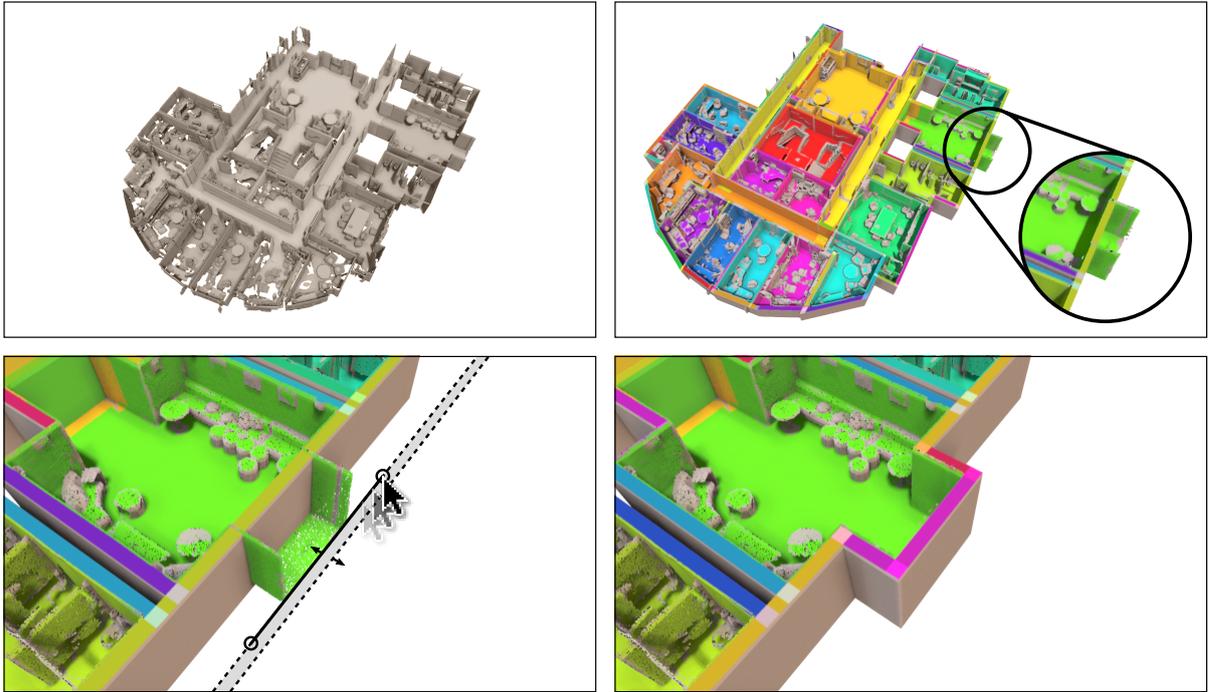


Figure 5.9: Manual addition of a wall demonstrated on the dataset “Area 3” from the Stanford 3D Large-Scale Indoor Spaces Dataset. Top right: A hallway ends without a terminating wall surface such that no wall candidate is available for enclosing the protrusion. Reconstruction and point cloud are shown overlaid. Bottom left: A wall candidate can easily be added by drawing a line, adding two opposing “virtual” wall surfaces. Bottom right: The algorithm now encloses the protrusion, using the perpendicular walls with real support in the point cloud.

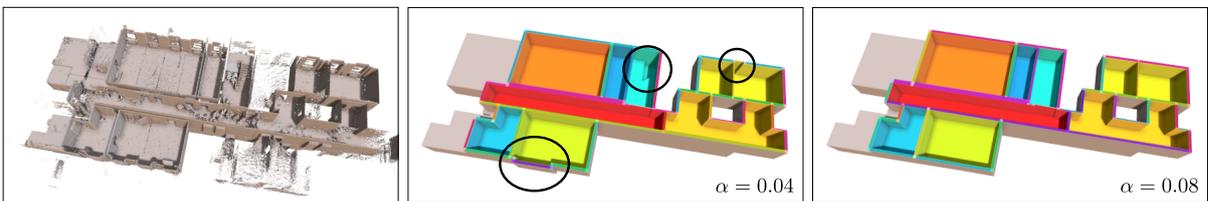


Figure 5.10: Different settings for the wall surface cost parameter  $\alpha$  in Equation 5.7 demonstrated on the dataset “Case study 2” from the ISPRS Benchmark on Indoor Modeling. Center: Our default setting of  $\alpha = 0.04$  results in some walls to be fitted to windows which have high point support in this dataset. Also, a slab has a hole since floor support in staircases is often complex. Right: Increasing to  $\alpha = 0.08$  leads to stronger regularization of walls and slabs.

to represent the building would be unnecessary. The approach in [67] leads to jumps between different, almost coplanar walls (Figure 5.7, center, black circles) instead of using longer, continuous walls. This can be explained by the principle that the approach separated rooms by wall center lines in 2D such that jumping from one wall to an almost coplanar wall resulted in almost no penalty in the cost function. In our case, a fully volumetric wall element would need to be added to the model to connect the parallel walls, resulting in relatively high costs. Second, the approach in [67] relies on given, separate scans and their positions for estimating an initial room segmentation. This leads to an oversegmentation of the hallway (Figure 5.7, center, dashed rectangle) since it tries to reconstruct one room per scan. Our method works independently of separate scans and estimates a room segmentation by unsupervised clustering.

**Interactive modification** Our linear programming approach allows for additional constraints to be easily added. One example for manual post-processing of the reconstruction results by interactively adding hard constraints is shown in Figure 5.8. In this case the indentation of the wall surfaces on the left and right sides of the building were lost by the regularization of the model as can be seen in Figure 5.8, center. The user has the option to add constraints such as forcing inside area, outside area, wall, no wall, etc. by clicking at the desired location. In this case, the highlighted locations were forced to be outside area. The algorithm then finds the next best option, placing new walls that fulfill all constraints as shown in Figure 5.8, right. Another example is shown in Figure 5.9 where a hallway ends without any terminating wall surface in the input data. Since the algorithm has no wall candidate available, it cannot enclose the protruding room area. By adding a virtual wall candidate by means of simply drawing a line, the algorithm is able to include the protrusion in the reconstructed model, automatically using the perpendicular wall surfaces that are present in the input data. Different choices of the wall surface penalty weight  $\alpha$  in Equation 5.7 control global regularization strength. Figure 5.10, center shows a reconstruction where some wall and slab elements are slightly misplaced due to relatively strong surface support at windows. Increasing  $\alpha$  from our default of 0.04 to 0.08 leads to stronger regularization as shown in Figure 5.10, right.

**Limitations** One technical limitation of our current implementation is that slanted walls, floors or ceilings are not taken into account although this is not an inherent limitation of our approach. The reason for our decision not to include these elements is that the construction of the 3D cell complex needs to be exact to guarantee the integrity of the data structure (e.g. cell neighborhood). Unfortunately, computing the cell complex in 3D induces numerical problems and currently we do not have a stable implementation for this task at our disposal. We thus opted to use the numerically stable implementation of 2D arrangements in CGAL [84] and extend it to 3D by stacking 2D arrangements separated by horizontal planes. A numerically stable extension of arrangements supporting general slanted planes would be an interesting direction for future research which we consider to be outside the scope of this paper. Processing of very large datasets may also require further optimizations to make them computationally feasible. In particular, using a global plane arrangement results in a large increase of cells and thus variables in the optimization model with every additional detected surface. More sophisticated selection of potential surfaces, and improved optimization methods, e.g. splitting the problem into smaller subproblems, are targets for further research. Last but not least, our current algorithm is not able to identify and include important architectural structures overarching the whole building like the pillars that are included in the hand-crafted model in Figure 5.6. Automatically identifying such structural elements and incorporating them into the automatic reconstruction is also an interesting direction for future research.

## **5.7 Conclusion and future work**

We have presented a novel approach to tackle the indoor building reconstruction problem from point clouds using integer linear programming. In contrast to previous methods, our approach reconstructs fully volumetric, interconnected wall entities and room topology on multi-story buildings with weak assumptions on the input data. The resulting models are very close to the requirements needed for Building Information Modeling tasks including volumetric representations of room spaces and wall entities, and their interrelations. Additional hard constraints such as forcing or avoiding certain entities at chosen locations may simply be added as constraints of the optimization problem. We demonstrated our approach on a variety of real-world datasets.

Future work for our proposed method includes the extension of the plane arrangement data structure to support slanted surfaces and possibly non-planar primitives. Strategies for reducing computational complexity by e.g. pruning invalid surface and wall candidates early in the process would improve applicability to larger-scale datasets. Also, connecting our reconstruction methodology with e.g. opening and object detection approaches would further enrich the resulting models.

## **Acknowledgments**

We acknowledge the Visualization and MultiMedia Lab at University of Zurich (UZH) and Claudio Mura for the acquisition of the 3D point clouds, and UZH as well as ETH Zürich for their support to scan the rooms represented in these datasets. Their datasets were used in our evaluation (Figure 5.5). We also used datasets provided by The Royal Danish Academy of Fine Arts Schools of Architecture, Design and Conservation (CITA) (Table 5.2), from The ISPRS Benchmark on Indoor Modeling [85] (Figure 5.10), and from the Stanford 3D Large-Scale Indoor Spaces Dataset [86] (Figure 5.9). This work was supported by the DFG projects KL 1142/11-1 (DFG Research Unit FOR 2535 Anticipating Human Behavior) and KL 1142/9-2 (DFG Research Unit FOR 1505 Mapping on Demand).

---

# Conclusions and Future Work

---

Automatic methods for the reconstruction of semantically high-level building models from point clouds and automated interpretation of unstructured point cloud scans of building interiors are still active topics of current research. In order for the reconstructed models to be usable in a Building Information Modeling (BIM) setting, they need to adhere to certain concepts specifically developed for the architecture domain. For instance, the widely adopted Industry Foundation Classes (IFC) standard defines the geometric representation of buildings as interrelated, volumetric, parametric elements for rooms, walls, floors, and ceilings. This kind of representation differs from classical boundary mesh models in that it more closely resembles the physical building structure and explicitly annotates individual elements with their type and connections with other entities. While several reconstruction approaches have recently been proposed, most of them do not use volumetric wall elements and instead represent buildings as either sets of unconnected primitive shapes, or boundaries of separate rooms. Few approaches aim at reconstructing volumetric walls, however they restrict themselves to simpler cases such as single stories or axis-aligned geometry.

In this thesis, we have presented new methods for tackling the two strongly related challenges of automated indoor point cloud interpretation, and automatic reconstruction of digital building models that are suitable for usage in BIM scenarios. In our first paper presented in Chapter 2, we devise an approach for segmenting unstructured indoor point cloud data into rooms which gives insight into a building's room topology. Since a clear definition of what constitutes a room is generally not available, we pursue an intuitive, ray-casting-based segmentation approach which determines a segmentation without relying on strict, heuristic rules which may fail in more complex real-world scenarios. An initial segmentation given by separate scans in the input data is iteratively refined by considering visibility between points in the measured data. The underlying assumption is that point sets are likely to belong to the same room if and only if mutual visibility between points from that set is high. We show that this idea can successfully be applied to complex real-world datasets, which is further proven in our later reconstruction methods which employ refined versions of this segmentation approach. In addition to the segmentation into rooms, openings between neighboring spaces are detected which yields a lightweight graph-based descriptor of the building's rooms and connections between them. Our second paper described in Chapter 3 extends the idea of graph-based buildings descriptor by incorporating multiple levels of detail in a hierarchical representation of stories, rooms, and objects within rooms. This extended representation enables different usage scenarios such as a classification of rooms based on the objects they contain. We demonstrate the applicability of the obtained hierarchical building descriptor for performing semantic queries for room and object constellations encoded as attributed subgraphs. In a BIM context, extracting additional semantic information from the measured data allows for further enrichment of reconstructed models by

e.g. annotating rooms with their type depending on the objects detected within.

The latter two papers presented in this thesis exploit the automatic structuring of the input point clouds to reconstruct building models from the measured data. Our first reconstruction approach described in Chapter 4 aims at reconstructing fully parametric building models from indoor point cloud scans. In contrast to previous methods, which either represent buildings as separate room boundaries or even completely unconnected surfaces, our method is fundamentally based on the idea of representing the building as an interconnected arrangement of volumetric, parametric walls. This not only closely resembles the physical building structure, but is also inspired by BIM standards such as the Industry Foundation Classes. Our approach builds upon the idea of performing a piecewise linear segmentation of space by means of a line arrangement which induces a cell complex whose cells are subsequently assigned to different rooms, or outside area. While previous approaches have used line arrangements constructed from separate, single surfaces in order to reconstruct boundaries of room interiors, our approach combines pairs of opposing surfaces as candidates for volumetric walls and constructs an arrangement from them. A constellation of rooms which matches the previously derived room segmentation of the point cloud is determined by solving the problem of assigning rooms to cells of the arrangement by means of a multi-label, graph-cut based cost minimization problem. Separating edges of the arrangement between differently labeled cells are then interpreted as wall elements between neighboring rooms, or a room and outside area. Since the edges were originally constructed from pairs of surfaces in the point cloud data, they can be reconstructed as volumetric wall elements with a thickness derived from the point cloud data. Furthermore, by construction, wall elements are guaranteed to fully enclose each room's interior area, and vertices of the arrangement where edges are incident can directly be interpreted as connections between walls. Together with a per-room estimation of floor and ceiling heights, this allows us to reconstruct a complete model of interconnected walls representing the measured point cloud data. A general limitation of this approach is that it employs a two-dimensional line arrangement into which wall candidates are projected, thus limiting the approach to datasets without vertically stacked rooms. We overcome this limitation in our latest contribution to the reconstruction challenge presented in Chapter 5, which further extends and refines this reconstruction approach in several ways. First, we do not rely on the availability of separate scans or scanner positions for obtaining an initial room segmentation. Instead, we devise a fully automatic, unsupervised clustering approach for room segmentation which works on completely unstructured point cloud datasets. Second, our approach supports multiple stories without simplifying assumptions such as that stories can be globally separated by horizontal cuts through the building. This allows for complex room layouts with varying floor and ceiling heights between neighboring rooms which is common in real-world buildings. Third, our new formulation of the optimization as integer linear programming problem enables us to integrate well-defined rulesets into the reconstruction as hard constraints. These rulesets guarantee e.g. plausible wall connectivity, allow incorporation of additional user-specified constraints, and improve regularization of the reconstructed model in comparison to our previous method. The resulting model is an arrangement of volumetric walls, floors, and ceilings, together with fully volumetric intersections between elements. Since this information closely corresponds to entity and relation types defined in IFC file formats, our resulting model can easily be transferred to off-the-shelf architecture software. The level of detail and semantics allow for usage as structural models of the scanned buildings for e.g. energy simulations, structural analysis, or comparisons with existing models.

In summary, we have presented approaches for automated interpretation of 3D indoor point cloud measurements of buildings, and reconstruction of parametric, volumetric building models from point clouds with a focus on suitability for modern Building Information Modeling workflows. Our methods successively improve on current state-of-the-art approaches by enabling reconstruction of real-world buildings with complex room constellations with lowered requirements on the input data. Nevertheless,

there still remain challenges and open questions in this active field of research which we outline in the following Section.

## 6.1 Future Work

Despite great advancements in the direction of fully automated reconstruction of accurate digital building models from point clouds, there still exist many opportunities and open challenges for further research. From our point of view, directions for future work include:

- Improving computational feasibility for reconstructing large-scale models while simultaneously offering at least the same level of versatility offered by our latest linear programming approach which enables integration of rulesets as hard constraints into the reconstruction process. To this end, intelligent prior segmentation of the reconstruction problem into smaller, possibly independent subproblems, as well as more intelligent preprocessing of the input data to prune and classify parts of the input data early in the process may prove advantageous. With respect to the core optimization problem, approaches aiming at exact inference of maximum a posteriori (MAP) problems by decomposition into a difficult combinatorial problem and an easier linear programming problem such as [87] may be an interesting direction for making larger-scale optimization problems feasible.
- Intricate room constellations with varying floor and ceiling heights, rooms spanning multiple stories, etc. can lead to fairly complex arrangements of wall, floor, and ceiling elements in order to represent the identified constellation of interior spaces. While it is in principle possible to generate an IFC model by simply generating corresponding entities for each element of our internal, volumetric data structure, more intelligent strategies for the placement of IFC entities could increase the level of quality of the resulting models. To reach a quality comparable to manually generated models, at least two general directions should be further investigated. First, experts from the architecture domain should be consulted on how to best transfer our internal representation to final IFC entities with respect to established modeling paradigms. Second, automated learning from existing, manually generated IFC models could help to find and successively extend suitable strategies without relying on hand-crafted heuristics.
- While our latest reconstruction approach is in theory not limited to perfectly vertical and horizontal walls and slabs, supporting arbitrarily oriented surfaces requires more work with respect to the underlying plane arrangement data structure to make it numerically stable in all possible edge cases. For example, an extension of the 2D arrangements with arbitrary precision number types provided by CGAL [84] to 3D arrangements of planes could enable more general and numerically stable 3D cell complexes.
- Integration of a greater variety of elements beyond the coarse building structure which are also defined in IFC, e.g. columns, beams, staircases, doors, windows, heating, ventilation and air conditioning (HVAC), safety-related objects, toilets, sinks, and power outlets would help to further enrich the strongly semantic building representation. While some previous approaches deal with subsets of these aspects, a holistic approach combining elements from different levels of detail into a common model would further help to ensure globally plausible models. As a long-range objective for the challenge of scan-to-BIM reconstruction, *simultaneous* optimization of e.g. positions of walls, stairs, openings, furniture, etc. could mutually enforce consistency between different elements and ultimately help to reconstruct a coherent model even in case of erroneous data.



## APPENDIX **A**

---

### **Curriculum vitae**

---



# Bibliography

---

- [1] I. E. Sutherland, “Sketch pad a man-machine graphical communication system”, *Proceedings of the SHARE design automation workshop*, ACM, 1964 6 (cit. on p. 2).
- [2] C. Eastman et al., *An Outline of the Building Description System. Research Report No. 50.*, (1974) (cit. on pp. 2, 3).
- [3] W. Gieling, *General AEC reference model (GARM)*, ISO TC184/SC4 (1988) (cit. on p. 2).
- [4] G. Augenbroe, “An overview of the COMBINE project”, *Proceedings of the First European Conference on Product and Process Modeling in the Building Industry (ECPPM’94), Dresden, Germany*, Citeseer, 1994 547 (cit. on p. 2).
- [5] B.-C. Björk, *RATAS project – Developing an infrastructure for computer-integrated construction*, *Journal of Computing in civil engineering* **8** (1994) 401 (cit. on p. 2).
- [6] M. J. Pratt, *Introduction to ISO 10303 – the STEP standard for product data exchange*, *Journal of Computing and Information Science in Engineering* **1** (2001) 102 (cit. on p. 2).
- [7] *History – buildingSMART*, <https://www.buildingsmart.org/about/about-buildingsmart/history/>, Accessed: 2018-10-08 (cit. on p. 2).
- [8] *The LaiserinLetter (tm)*, <http://www.laiserin.com/features/issue15/feature01.php>, Accessed: 2018-10-08 (cit. on p. 2).
- [9] *IFC overview summary*, <http://www.buildingsmart-tech.org/specifications/ifc-overview>, Accessed: 2018-10-08 (cit. on p. 2).
- [10] W. Boehler and A. Marbs, *3D scanning instruments*, *Proceedings of the CIPA WG* **6** (2002) 9 (cit. on p. 3).
- [11] D. Barber and J. Mills, *3D Laser Scanning for Heritage: Advice and guidance to users on laser scanning in archaeology and architecture*, English Heritage, 2007 (cit. on p. 3).
- [12] F. Lafarge and C. Mallet, *Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation*, *International journal of computer vision* **99** (2012) 69 (cit. on p. 3).
- [13] R. Schnabel, R. Wahl and R. Klein, *Efficient RANSAC for Point-Cloud Shape Detection*, *Computer Graphics Forum* **26** (2007) 214 (cit. on pp. 3, 12, 14, 28, 42, 58, 59).
- [14] A. Adan and D. Huber, “3D reconstruction of interior wall surfaces under occlusion and clutter”, *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, IEEE, 2011 275 (cit. on pp. 3, 38, 39, 55).
- [15] R. Ambruş, S. Claiici and A. Wendt, *Automatic room segmentation from unstructured 3-d data of indoor environments*, *IEEE Robotics and Automation Letters* **2** (2017) 749 (cit. on pp. 3, 55, 58).

- [16] H. Son and C. Kim, *Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity*, *Advanced Engineering Informatics* **34** (2017) 114 (cit. on pp. 4, 57).
- [17] M. Bassier, B. Van Genechten and M. Vergauwen, *Classification of sensor independent point cloud data of building objects using random forests*, *Journal of Building Engineering* (2018) (cit. on pp. 4, 56).
- [18] G.-T. Michailidis and R. Pajarola, *Bayesian graph-cut optimization for wall surfaces reconstruction in indoor environments*, *The Visual Computer* **33** (2017) 1347, ISSN: 1432-2315, URL: <https://doi.org/10.1007/s00371-016-1230-3> (cit. on p. 4).
- [19] A. Adán et al., *Scan-to-BIM for secondary building components*, *Advanced Engineering Informatics* **37** (2018) 119 (cit. on pp. 4, 57).
- [20] V. Sanchez and A. Zakhor, “Planar 3D modeling of building interiors from point cloud data”, *2012 19th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2012 1777 (cit. on pp. 4, 38, 39, 54, 55).
- [21] A. Monszpart et al., *RAPter: Rebuilding Man-made Scenes with Regular Arrangements of Planes*, *ACM SIGGRAPH 2015* (2015) (cit. on pp. 4, 38, 39, 56).
- [22] C. Mura, O. Mattausch and R. Pajarola, “Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements”, *Computer Graphics Forum*, vol. 35, 7, Wiley Online Library, 2016 179 (cit. on pp. 4, 54–56, 58).
- [23] C. Mura et al., *Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts*, en, *Computers & Graphics* **44** (2014), ISSN: 00978493 (cit. on pp. 4, 40, 41, 55, 56).
- [24] S. Oesau, F. Lafarge and P. Alliez, *Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut*, *ISPRS Journal of Photogrammetry and Remote Sensing* **90** (2014) 68, ISSN: 09242716 (cit. on pp. 4, 38, 40, 41, 54, 55).
- [25] M. Previtali et al., *Towards automatic indoor reconstruction of cluttered building rooms from point clouds*, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2** (2014) 281 (cit. on pp. 4, 55).
- [26] R. Wang, L. Xie and D. Chen, *Modeling Indoor Spaces Using Decomposition and Reconstruction of Structural Elements*, *Photogrammetric Engineering & Remote Sensing* **83** (2017) 827 (cit. on pp. 4, 55, 56).
- [27] A. Stambler and D. Huber, “Building modeling through enclosure reasoning”, *3D Vision (3DV), 2014 2nd International Conference on*, vol. 2, IEEE, 2014 118 (cit. on pp. 5, 56).
- [28] H. Macher, T. Landes and P. Grussenmeyer, *From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings*, *Applied Sciences* **7** (2017) 1030 (cit. on pp. 5, 55, 56).
- [29] S. Murali et al., “Indoor Scan2BIM: Building information models of house interiors”, *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017 6126 (cit. on pp. 5, 54–56).

- 
- [30] R. Wessel, I. Blümel and R. Klein, “The Room Connectivity Graph: Shape Retrieval in the Architectural Domain”, *WSCG 2008*, UNION Agency-Science Press, 2008, ISBN: 978-80-86943-15-2 (cit. on pp. 6, 11–13, 25).
- [31] C. Langenhan et al., “Topological Query on Semantic Building Models Using Ontology and Graph Theory”, *EG-ICE*, TUM, 2012 (cit. on pp. 11, 12).
- [32] V. Verma, R. Kumar and S. Hsu, *3D Building Detection and Modeling from Aerial LIDAR Data*, *CVPR 2012* **2** (2006) 2213, ISSN: 1063-6919 (cit. on pp. 11, 13).
- [33] R. Schnabel et al., “Shape Recognition in 3D Point-Clouds”, *WSCG 2008*, UNION Agency-Science Press, 2008 (cit. on pp. 11, 13).
- [34] M. Bokeloh et al., *Symmetry Detection Using Line Features*, *Computer Graphics Forum (EG 2009)*, *Computer Graphics Forum* **28** (2009) 697, ed. by P. Dutré and M. Stamminger (cit. on pp. 11, 13).
- [35] Y. M. Kim et al., *Acquiring 3D Indoor Environments with Variability and Repetition*, *ACM Transactions on Graphics (TOG)* **31** (2012) 138 (cit. on pp. 12, 13, 24, 25).
- [36] L. Nan, K. Xie and A. Sharf, *A Search-classify Approach for Cluttered Indoor Scene Understanding*, *ACM Transactions on Graphics* **31** (2012) 137:1 (cit. on pp. 12, 13, 24, 25, 33).
- [37] R. B. Rusu et al., *Towards 3D point cloud based object maps for household environments*, *Robotics and Autonomous Systems* **56** (2008) 927,  
URL: <http://www.sciencedirect.com/science/article/pii/S0921889008001140>  
(visited on 10/07/2013) (cit. on p. 13).
- [38] H. S. Koppula et al., “Semantic Labeling of 3D Point Clouds for Indoor Scenes”, *NIPS 2011*, 2011 244, URL:  
[http://machinelearning.wustl.edu/mlpapers/paper\\_files/NIPS2011\\_0186.pdf](http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2011_0186.pdf)  
(visited on 24/07/2013) (cit. on pp. 13, 24, 25).
- [39] C. Langenhan et al., *Graph-based retrieval of building information models for supporting the early design stages*, *Advanced Engineering Informatics* (2013), ISSN: 14740346,  
URL: <http://linkinghub.elsevier.com/retrieve/pii/S1474034613000499> (visited on 15/07/2013) (cit. on pp. 13, 25).
- [40] S. Ahmed et al., “Automatic Room Detection and Room Labeling from Architectural Floor Plans”, *10th IAPR International Workshop on Document Analysis Systems (DAS-2012)*, IEEE, 2012 339, ISBN: 978-0-7695-4661-2, 978-1-4673-0868-7,  
URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6195390>  
(visited on 15/07/2013) (cit. on pp. 13, 25).
- [41] J. Xiao and Y. Furukawa, *Reconstructing the world’s museums*, *International journal of computer vision* **110** (2014) 243 (cit. on pp. 13, 38, 39, 55).
- [42] R. Duda, P. Hart and D. Stork, *Pattern Classification*, Wiley, 2001 (cit. on p. 14).
- [43] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* **39** (1977) 1 (cit. on p. 15).

- [44] C. Mura et al., “Robust reconstruction of interior building structures with multiple rooms under clutter and occlusions”, *Computer-Aided Design and Computer Graphics (CAD/Graphics), 2013 International Conference on*, IEEE, 2013 52 (cit. on pp. 24, 25, 35, 55).
- [45] S. Ochmann et al.,  
“Automatic generation of structural building descriptions from 3D point cloud scans”,  
*2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*,  
IEEE, 2014 (cit. on pp. 24, 25, 41, 56).
- [46] E. Turner and A. Zakhor,  
“Floor plan generation and room labeling of indoor environments from laser range data”,  
*2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*,  
IEEE, 2014 (cit. on pp. 24, 25, 38, 40, 56).
- [47] R. B. Rusu et al., “Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments”,  
*Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*,  
IEEE, 2009 (cit. on pp. 24, 25).
- [48] A. Budroni and J. Boehm, *Automated 3D reconstruction of interiors from point clouds*,  
*International Journal of Architectural Computing* **8** (2010) 55 (cit. on pp. 24, 25, 38, 39, 55).
- [49] J. Xiao and Y. Furukawa, “Reconstructing the world’s museums”, *Computer Vision–ECCV 2012*,  
Springer, 2012 668 (cit. on pp. 24, 25).
- [50] S. Oesau, F. Lafarge and P. Alliez,  
“Indoor Scene Reconstruction using Primitive-driven Space Partitioning and Graph-cut”,  
*Eurographics Workshop on Urban Data Modelling and Visualisation*, 2013  
(cit. on pp. 25, 41, 55).
- [51] K. Lai and D. Fox,  
*Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation*,  
*International Journal of Robotics Research* **29** (2010) 1019, ISSN: 0278-3649 (cit. on p. 25).
- [52] T. Shao et al.,  
*An interactive approach to semantic modeling of indoor scenes with an RGBD camera*,  
*ACM Transactions on Graphics* **31** (2012) 12, ISSN: 07300301, (visited on 21/02/2014)  
(cit. on p. 25).
- [53] S. Ahmed et al., *Automatic Analysis and Sketch-Based Retrieval of Architectural Floor Plans*,  
*Pattern Recognition Letters* **35** (2014) 91 (cit. on p. 25).
- [54] S. Macé et al., “A System to Detect Rooms in Architectural Floor Plan Images”,  
*Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, DAS '10*,  
ACM, 2010 167, ISBN: 978-1-60558-773-8 (cit. on p. 25).
- [55] R. B. Rusu, *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*,  
PhD thesis: Computer Science department, Technische Universitaet Muenchen, Germany, 2009  
(cit. on p. 32).
- [56] R. M. Rustamov,  
“Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation”,  
*Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP '07*,  
Eurographics Association, 2007 225, ISBN: 978-3-905673-46-3 (cit. on p. 35).

- 
- [57] B. Okorn et al., “Toward automated modeling of floor plans”, *Proceedings of the symposium on 3D data processing, visualization and transmission*, vol. 2, 2010 (cit. on pp. 38, 39, 55).
- [58] X. Xiong et al., *Automatic creation of semantically rich 3D building models from laser scanner data*, *Automation in Construction* **31** (2013) 325 (cit. on pp. 38, 40, 55).
- [59] E. Turner, P. Cheng and A. Zakhor, *Fast, automated, scalable generation of textured 3D models of indoor environments*, *IEEE Journal of Selected Topics in Signal Processing* **9** (2015) 409 (cit. on pp. 38, 40, 54–56).
- [60] C. Mura et al., “Reconstructing Complex Indoor Environments with Arbitrary Wall Orientations”, *Eurographics (Posters)*, 2014 19 (cit. on pp. 38, 40, 55).
- [61] M. Tamke et al., “From Point Clouds to Definitions of Architectural Space”, *eCAADe 2014*, 2014 (cit. on p. 40).
- [62] Y. Boykov, O. Veksler and R. Zabih, *Fast approximate energy minimization via graph cuts*, *IEEE TPAMI* **23** (2001) 1222 (cit. on pp. 44, 64).
- [63] V. Kolmogorov and R. Zabih, *What energy functions can be minimized via graph cuts?*, *IEEE TPAMI* **26** (2004) 147 (cit. on p. 44).
- [64] Y. Boykov and V. Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*, *IEEE TPAMI* **26** (2004) 1124 (cit. on p. 44).
- [65] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, *ACM Transactions on Intelligent Systems and Technology* **2** (3 2011) 27:1, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (cit. on p. 47).
- [66] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL)”, *IEEE International Conference on Robotics and Automation (ICRA)*, 2011 (cit. on pp. 47, 66).
- [67] S. Ochmann et al., *Automatic reconstruction of parametric building models from indoor point clouds*, *Computers & Graphics* **54** (2016) 94 (cit. on pp. 54–56, 58, 59, 68, 69, 71).
- [68] C. Liu, J. Wu and Y. Furukawa, *FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans*, arXiv preprint arXiv:1804.00090 (2018) (cit. on pp. 54, 55).
- [69] C. Liu et al., “Raster-to-vector: Revisiting floorplan transformation”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017 2195 (cit. on pp. 55, 57).
- [70] C. Mura and R. Pajarola, “Exploiting the room structure of buildings for scalable architectural modeling of interiors”, *ACM SIGGRAPH 2017 Posters*, ACM, 2017 4 (cit. on pp. 56, 58).
- [71] C. Thomson and J. Boehm, *Automatic geometry generation from point clouds for BIM*, *Remote Sensing* **7** (2015) 11753 (cit. on p. 56).
- [72] L. Nan and P. Wonka, “PolyFit: Polygonal Surface Reconstruction From Point Clouds”, *The IEEE International Conference on Computer Vision (ICCV)*, 2017 (cit. on p. 56).

- [73] J. Jung, C. Stachniss and C. Kim, *Automatic Room Segmentation of 3D Laser Data Using Morphological Processing*, *ISPRS International Journal of Geo-Information* **6** (2017) 206 (cit. on p. 56).
- [74] D. Bobkov et al., “Room segmentation in 3D point clouds using anisotropic potential fields”, *Multimedia and Expo (ICME), 2017 IEEE International Conference on*, IEEE, 2017 727 (cit. on p. 56).
- [75] T. L. Garwood et al., *A framework for producing gbXML building geometry from Point Clouds for accurate and efficient Building Energy Modelling*, *Applied Energy* **224** (2018) 527 (cit. on p. 57).
- [76] N. Hyland et al., *Automatic Open Standard Reporting for Dimensional Control Compliance*, (2017) (cit. on p. 57).
- [77] S. O’Keeffe et al., *Automatic Validation of As-Is and As-Generated IFC BIMs for Advanced Scan-to-BIM Methods*, (2017) (cit. on p. 57).
- [78] S. Brodie et al., *The BIM & Scan® Platform: A Cloud-Based Cyber-Physical System for Automated Solutions Utilising Real & Virtual Worlds.*, (2017) (cit. on p. 57).
- [79] U. Krispel et al., *Data completion in building information management: electrical lines from range scans and photographs*, *Visualization in Engineering* **5** (2017) 4 (cit. on p. 57).
- [80] T. Krijnen and J. Beetz, *An IFC schema extension and binary serialization format to efficiently integrate point cloud data into building models*, *Advanced Engineering Informatics* **33** (2017) 473 (cit. on p. 58).
- [81] S. Dongen, *A Cluster Algorithm for Graphs*, tech. rep., 2000 (cit. on pp. 58, 59, 66).
- [82] Sven Oesau and Yannick Verdie and Clément Jamin and Pierre Alliez and Florent Lafarge and Simon Giraudot, “Point Set Shape Detection”, *CGAL User and Reference Manual*, 4.12, CGAL Editorial Board, 2018, URL: <https://doc.cgal.org/4.12/Manual/packages.html#PkgPointSetShapeDetection3Summary> (cit. on pp. 59, 66).
- [83] I. Gurobi Optimization, *Gurobi Optimizer Reference Manual*, 2016, URL: <http://www.gurobi.com> (cit. on pp. 65, 66).
- [84] R. Wein et al., “2D Arrangements”, *CGAL User and Reference Manual*, 4.12, CGAL Editorial Board, 2018, URL: <https://doc.cgal.org/4.12/Manual/packages.html#PkgArrangement2Summary> (cit. on pp. 66, 71, 75).
- [85] K. Khoshelham et al., *The ISPRS Benchmark on Indoor Modelling*, *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* **42** (2017) (cit. on pp. 66, 72).
- [86] I. Armeni et al., *Joint 2D-3D-Semantic Data for Indoor Scene Understanding*, ArXiv e-prints (2017), arXiv: 1702.01105 [cs.CV] (cit. on pp. 66, 72).
- [87] S. Haller, P. Swoboda and B. Savchynskyy, *Exact MAP-inference by Confining Combinatorial Search with LP Relaxation*, (2018) (cit. on p. 75).

# List of Figures

---

2.1	Overview of our method. From left to right: Detected planar structures; colour-coded indices of the individual scans; final point assignments after the re-labeling process; extracted room topology graph. . . . .	12
2.2	Different cases of visibility estimation. Left: Visibility between two points $x_1, x_2$ located within a room. The line-of-sight is unobstructed since line segment $s$ does not intersect any building structure. Right: If a wall lies between $x_1, x_2$ , $s$ intersects multiple planes and visibility is estimated to be low. . . . .	16
2.3	For door detection, the rays whose target point's labeling has changed during segmentation are intersected with the planes belonging to the room in which the scanner is located (left); the intersection point sets on each plane are potential door candidates (middle); pairs of candidates that fulfill certain constraints are extracted (right). . . . .	17
2.4	Results on synthetic data; the first picture also shows an automatic merging suggestion (red line between two scanner positions). . . . .	18
2.5	A failure case of our method due to a strongly non-convex room and planar artifacts in the scan data. . . . .	18
2.6	Results on two real-world datasets. The upper-left pictures show the room associations before merging of scans belonging to the same rooms. Upper-right: Associations after the merging step. Lower-left: The room assignments after segmentation. Lower-right: The extracted graphs including detected doors and annotations for the estimated room areas. . . . .	20
3.1	Label diffusion for a point which is initially labeled incorrectly (the correct label is 1). The correct label is assumed after a few iterations, however it may once again become incorrect with increasing number of iterations. . . . .	27
3.2	Room labeling before (left) and after (middle) relabeling. Right: Points which were relabeled are highlighted. . . . .	27
3.3	Room neighbors (left) and connections extracted from a real-world dataset. . . . .	29
3.4	Steps of the room neighborhood graph generation. . . . .	30
3.5	Steps of the room connectivity graph generation. . . . .	30
3.6	A failure case of the door detection. . . . .	31
3.7	Left: A chair consisting of multiple segments. Middle: One of the segment combinations added to the graph descriptor. Right: Visualization of descriptor coefficients. . . . .	33
3.8	Results of subgraph queries. Top-left: The room connections extracted from the dataset. Our approach not only allows queries for topological constellations (top middle, top right), but also for combined queries including objects (bottom row). . . . .	34

4.1	Schematic of editing capabilities of different kinds of reconstructions. The input point cloud is shown on the left. The remaining columns exemplify editing operations, i.e. elements are moved in the directions of the arrows. Surface representations without (column (a)) or with (column (b)) connectivity information do not allow intuitive editing on the level of wall elements. Our reconstruction (column (c)) maintains room topology and global wall connectivity. . . . .	38
4.2	Example operations which are easily implemented using our results. <b>(a)</b> Relations between walls and rooms enable editing while maintaining room topology. Note how incident walls are adjusted automatically. <b>(b)</b> Automatic determination of wall elements shared between rooms together with automatic measurements enable e.g. acoustic or thermal simulations. <b>(c)</b> Global connectivity enables pathfinding for e.g. simulation and optimization of escape routes. . . . .	39
4.3	Overview of our approach (see also Section 4.3). <b>(a)</b> Input point cloud; assignment of points to scans shown in different colors. <b>(b)</b> Refined assignment after automatic segmentation. <b>(c)</b> Detected vertical planes transferred to the horizontal plane. <b>(d)</b> Candidates for walls are derived from single and pairs of projected planes. Intersecting their centerlines yields a planar graph whose faces are subsequently assigned labels for rooms or outside area. <b>(e)</b> Only edges separating differently labeled faces are retained. <b>(f)</b> The final model with detected and classified wall openings, e.g. doors (green) and windows (yellow). . . . .	40
4.4	Wall candidate generation. <b>(a)-(b)</b> Detected vertical planes in the 3D point cloud are projected into the horizontal plane. <b>(c)</b> Different wall surface lines including the respective (projected) support points and surface normals. <b>(d)</b> For each single wall surface, an infinitely long wall candidate $w$ for a wall separating a room from outside area is generated. In this case, the thickness $t_w$ is user-specified. <b>(e)</b> For each pair of approximately parallel wall surfaces, a candidate for separating adjacent rooms is generated. In this case, wall thickness is estimated from the data. . . . .	42
4.5	Determination of suitable wall candidate segments. <b>(a)</b> Input point cloud after segmentation. <b>(b)</b> Intersecting all wall candidate centerlines yields a planar graph. We determine an assignment of all faces to rooms or outside area such that connected components of identically-labeled faces are rooms and edges between differently-labeled faces are wall elements. <b>(c)</b> Resulting labeling of faces after optimization; colors indicate room labels. <b>(d)</b> Retaining only edges separating differently-labeled faces yields a subgraph representing the sought wall elements and their connectivity. . . . .	44
4.6	Considering only the number of projected points within a face for unary costs does not take into account their spatial distribution. . . . .	45
4.7	Multiple scans within a single room. <b>(a)</b> The hallway has been scanned from three positions; room labels are mixed within that room. <b>(b)</b> After segmentation (Section 4.4), the hallway is still split in multiple sections. <b>(c)</b> The labeling algorithm separates these regions by wall elements that are not part of the building’s true walls. <b>(d)</b> By detecting and removing excess wall elements, faces are merged to larger rooms. . . . .	45
4.8	Opening detection and classification. <b>(a)</b> The input point cloud after segmentation. <b>(b)</b> Detected clusters of intersections between reconstructed walls and simulated laser rays between scanner positions and measured points. Clusters are classified as doors (green), windows (yellow), “virtual” clusters indicating walls to be removed for merging multiple scans within a room (magenta), and “invalid” (red). <b>(c)</b> The final model after removal of walls containing “virtual” openings. . . . .	46

4.9	Different plane detection options: Allowing smaller planes as potential wall surfaces allows for more detailed structures (right-hand side) at the cost of possibly detecting incorrect candidates in clutter. . . . .	47
4.10	Different choices for $\alpha, \beta, \gamma$ in Eqs. 4.2 and 4.3. <b>(a)-(b)</b> Perspective and orthographic view of an example situation. <b>(c)</b> Parameters chosen as described in Section 4.8. Wall centerlines are well-regularized and common wall elements have been reconstructed between rooms. <b>(d)</b> Without unary costs ( $\alpha := 0$ ). While the resulting walls are well-regularized, parts of rooms are missing despite high areal support by measured points. <b>(e)</b> Without binary costs ( $\beta := 0$ ). Walls are located similar to (c) but are overly complex due to missing regularization and preference for correctly labeled edges. <b>(f)</b> Without penalty for high outside labeling ( $\gamma := 0$ ). The algorithm does not prefer common walls for separating adjacent rooms. . . . .	48
4.11	Effects of registration errors. <b>(a)</b> Result without alignment errors. Wall volumes are shown in gray together with the respective wall centerlines. <b>(b)</b> Translational registration errors may result in offset walls to which the algorithm adapts accordingly (right detail view). Wall thickness may also change (wall separating the red and green rooms in the left detail view). <b>(c)</b> Rotational registration errors may lead to wall surface pairs not to be associated to common walls. Wall thickness is incorrect since the wall candidates do not originate from wall surfaces pairs. . . . .	48
4.12	Highly cluttered rooms. Left: Clutter and transparent surfaces (windows) cause large scan holes; wall surfaces are only partially scanned. Right: Reconstructed walls still are well regularized and separate rooms correctly. . . . .	50
4.13	Our cost minimization approach and infinitely long wall candidates automatically bridge scan holes in a plausible manner. . . . .	51
4.14	Wall elements which are not connected at both ends to other walls are currently not representable by our reconstruction. . . . .	51
4.15	Visual comparison of reconstruction and manually generated model. <b>(a)</b> Input point cloud; scans are shown in different colors. <b>(b)</b> Professionally, manually generated model. <b>(c)</b> Reconstructed model. Locations and thickness of walls, and locations of doors are generally good. . . . .	52
4.16	Example results on point clouds with 33, 43, and 67 scans. Upper row: Point clouds after segmentation step; most ceiling points (i.e. points with downwards-facing normals) are removed for visualization. Lower row: Reconstructed models; detected windows are shown in yellow, doors are shown in green. Most wall elements are faithfully reconstructed; some excess walls have not been removed (see e.g. the large room in the lower-right corner of the second column). . . . .	52

5.1	<p>Overview of the main steps. Ceiling of upper floor is hidden in (a), (c), and (f) for visualization purposes. (a) The input is a registered but otherwise unstructured and unfiltered indoor point cloud. (b) Planes are detected by means of RANSAC shape detection. (c) Outliers are automatically removed and rooms are segmented using an unsupervised clustering approach based on mutual visibility between point patches. (d) Detected planes are classified as horizontal slab surfaces and vertical wall surfaces (only latter shown). Surfaces are assigned multi-label support bitmaps. (e) A 3D plane arrangement is constructed by intersecting all planes, yielding a cell complex. Priors for rooms, outside area and surface support are estimated. (f) The final model consisting of interrelated room and wall volumes is obtained by solving a integer linear program in which cell labels are binary variables. . . . .</p>	57
5.2	<p>Detected planes are the basis for our reconstruction. Different kinds of bitmaps (i.e. grids) on the planes are used throughout the approach. (a) Unlabeled input point cloud. (b) Binary occupancy bitmaps on the detected planes (Section 5.4.1) are used as a lightweight representation of support by the point cloud. Different planes are shown in different colors. (c) Ray casting against the occupancy bitmaps and clustering yields a segmentation of the point cloud into rooms (Section 5.4.3). Different room labels are shown in different colors. (d) The point cloud labeling is projected into multi-label bitmaps where each pixel contains a soft assignment <math>[0, 1]^n</math> to the <math>n</math> different room labels as defined in Section 5.4.3. This is used for estimating locations of rooms in 3D space as described in Section 5.4.7. The multi-label bitmaps are shown using the same label colors as in (c). In (b) and (d), only vertical planes are shown for clarity. . . . .</p>	60
5.3	<p>Dilation of surface support. Top: Example point cloud viewed from above. Middle: Surface support (left, thick lines) is naturally restricted to parts visible to the scanner. This leads to high costs for the wall intersection since wall surfaces are not supported by points (right). Bottom: Dilating the surface support (left, dotted lines) extends support into the interiors of walls, encouraging placement of volumetric intersections (right). .</p>	61
5.4	<p>Explanation of notation and constraints. (a) Neighboring cells are considered as ordered pairs <math>c_a \dashv c_b</math> with respect to normal orientation of the separating face <math>f_{c_a, c_b}</math>. (b) Considering a cell <math>c</math>, <math>\mathcal{W}_c</math> is the set of wall candidates enclosing cell <math>c</math>. A face <math>f_{c_a, c_b}</math> may be a boundary or an inner face of a set of walls. The set of boundary walls with respect to that face is <math>\mathcal{W}_{c_a, c_b}^-</math>, the set of inner walls is <math>\mathcal{W}_{c_a, c_b}^+</math>. (c) Transitions between interior and exterior area at a face <math>f</math> may only occur with the room label being on the positive side of <math>f</math> (Constraint 2). (d) If a face <math>f</math> is the boundary of a room, some wall needs to be active on the negative side of <math>f</math> (Constraint 4). (e) A wall may end at an inner face <math>f</math> only if the wall label is on the negative side of <math>f</math> (Constraint 5). (f) If a wall <math>w_1</math> ends at an inner face, this face must be a boundary face of at least one other active wall <math>w_2</math>. This enforces connectedness between walls. . . . .</p>	62
5.5	<p>Our reconstruction result on the synthetic dataset “synth3” by the Visualization and MultiMedia Lab at University of Zurich. Rooms and walls are accurately reconstructed.</p>	68
5.6	<p>Comparison of a hand-crafted BIM model (left) and our reconstruction (right) of Dataset 5 (see Table 5.2). Reconstructed room labels were manually overlaid on the BIM model for reference. . . . .</p>	68

---

5.7	Comparison between different reconstruction approaches. Left: Input point cloud viewed from above. Center: The method described in [67] may fail to regularize chains of almost coplanar walls, leading to segmented walls (circles). Also, reliance on separate scans as initial room labeling may lead to oversegmented rooms (dashed rectangle). Right: Our approach overcomes these issues by incorporating costs for all surfaces of volumetric wall elements, and room segmentation that is independent of scan positions. . . . .	68
5.8	Additional constraints may be added to interactively steer the reconstruction. Small indentations of wall surfaces (left) are initially lost in the reconstruction. By forcing regions to be outside area (center), our method finds an alternative wall placement under these constraints (right). . . . .	69
5.9	Manual addition of a wall demonstrated on the dataset “Area 3” from the Stanford 3D Large-Scale Indoor Spaces Dataset. Top right: A hallway ends without a terminating wall surface such that no wall candidate is available for enclosing the protrusion. Reconstruction and point cloud are shown overlaid. Bottom left: A wall candidate can easily be added by drawing a line, adding two opposing “virtual” wall surfaces. Bottom right: The algorithm now encloses the protrusion, using the perpendicular walls with real support in the point cloud. . . . .	70
5.10	Different settings for the wall surface cost parameter $\alpha$ in Equation 5.7 demonstrated on the dataset “Case study 2” from the ISPRS Benchmark on Indoor Modeling. Center: Our default setting of $\alpha = 0.04$ results in some walls to be fitted to windows which have high point support in this dataset. Also, a slab has a hole since floor support in staircases is often complex. Right: Increasing to $\alpha = 0.08$ leads to stronger regularization of walls and slabs. . . . .	70



# List of Tables

---

4.1	Datasets used in our experiments. Figure references marked with “*” indicate that only a subset of the dataset is shown. . . . .	49
5.1	Feature comparison of recent indoor reconstruction approaches. Notes: <sup>1</sup> Full 3D reconstruction means that arbitrary vertical room constellations including rooms spanning multiple floors are handled correctly. <sup>2</sup> Virtual scan positions are automatically estimated. <sup>3</sup> Volumetric walls are generated manually in post-processing. <sup>4</sup> Scanner trajectories are used. . . . .	55
5.2	Evaluation results on various real-world datasets. The number of scans in the point cloud is shown for reference only and is not used in our approach. The number of points is the point count after subsampling. <sup>1</sup> The number of surfaces and walls is given as <i>vertical+horizontal</i> . . . . .	67