




Article

Feasibility of Using Grammars to Infer Room Semantics

Xuke Hu ¹, Hongchao Fan ^{2,*}, Alexey Noskov ¹, Alexander Zipf ¹, Zhiyong Wang ³
and Jianga Shang ^{4,5}

¹ GIScience Research Group, Institute of Geography, Heidelberg University, 69120 Heidelberg, Germany

² Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway

³ Department of Human Geography and Spatial Planning, Faculty of Geosciences, Utrecht University, 3584 Utrecht, The Netherlands

⁴ School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China

⁵ National Engineering Research Center for Geographic Information System, Wuhan 430074, China

* Correspondence: hongchao.fan@ntnu.no

Received: 13 May 2019; Accepted: 26 June 2019; Published: 28 June 2019



Abstract: Current indoor mapping approaches can detect accurate geometric information but are incapable of detecting the room type or dismiss this issue. This work investigates the feasibility of inferring the room type by using grammars based on geometric maps. Specifically, we take the research buildings at universities as examples and create a constrained attribute grammar to represent the spatial distribution characteristics of different room types as well as the topological relations among them. Based on the grammar, we propose a bottom-up approach to construct a parse forest and to infer the room type. During this process, Bayesian inference method is used to calculate the initial probability of belonging an enclosed room to a certain type given its geometric properties (e.g., area, length, and width) that are extracted from the geometric map. The approach was tested on 15 maps with 408 rooms. In 84% of cases, room types were defined correctly. It, to a certain degree, proves that grammars can benefit semantic enrichment (in particular, room type tagging).

Keywords: indoor mapping; room type tagging; semantic enrichment; grammar; Bayesian inference

1. Introduction

People spend most of their time indoors, such as offices, houses, and shopping malls [1]. New indoor mobile applications are being developed at a phenomenal rate, covering a wide range of indoor social scenarios, such as indoor navigation and location-enabled advertisement [2]. Semantically-rich indoor maps that contain the usage of rooms (e.g., office, restaurant, or book shop) are indispensable parts of indoor location-based services [3,4]. The floor plans modelled in computer-aided design (CAD), building information modeling (BIM)/industrial foundation classes (IFC), and GIS systems (e.g., ArcGIS and Google Maps) contain rich semantic information, including the type or function of rooms. However, only a small fraction of millions of indoor environments is mapped [5], let alone the type of rooms.

Currently, two mainstream indoor mapping methods include digitalization-based and measurement-based. The first provides digitized geometric maps comprising rooms, corridors, and doors extracted automatically from existing scanned maps [6–9]. Normally, it is incapable of extracting the room type information from the scanned map. According to the type of measurements, we can further divide the second group of approaches into three categories: LIDAR point cloud [10–13], image [14–16], and volunteers' trace [5,17]. LIDAR point cloud and image-based approaches can

reconstruct accurate 3-D scenes that contain rich semantics, such as walls, windows, ceilings, doors, floors, and even the type of furniture in rooms (e.g., sofa, chairs, and desks) [11,18] but ignore the type of rooms. Utilizing volunteers' traces to reconstruct indoor maps has received much attention due to its low requirement on hardware and low computational complexity compared to LIDAR point cloud and image-based approaches. With the help of abundant traces, it can detect accurate geometric information (e.g., the dimension of rooms and corridors) and simple semantic information (e.g., stairs and doors). However, it is difficult to infer the type of rooms based on traces. Briefly, current indoor mapping approaches can detect accurate geometric information (e.g., dimension of rooms) and partial semantics (e.g., doors and corridors) but are incapable of detecting the room type (i.e., for digitalization-based and trace-based approaches) or ignore this issue (i.e., image and LIDAR point cloud-based approaches). To solve this problem, [19] proposed using a statistical relational learning approach to reason the type of rooms as well as buildings at schools and office buildings. [3,20] used check-in information to automatically identify the semantic labels of indoor venues in malls, i.e., business names (e.g., Starbucks) or categories (e.g., restaurant). However, it is problematic when check-in information is unavailable in indoor venues.

This work takes research buildings (e.g., laboratories and office buildings at universities) as examples, investigating the feasibility of using grammars to infer the type of rooms based on the geometric maps that can be obtained through the aforementioned reconstruction approaches. The geometric map we use contains the geometric information of rooms and simple semantic information (i.e., corridors and doors). We must reckon that it is impossible to manually construct a complete and reliable grammar that can represent the layout of the research buildings all over the world, which is not the aim of this work. Our goal is to prove that to a certain extent, grammars can benefit current indoor mapping approaches, at least the digitalization-based and traces-based methods, by providing the room type information. As for the creation of complete and reliable grammars, we plan to use grammatical inference techniques [21,22] to automatically learn a probabilistic grammar based on abundant training data in the future.

In this work, we use grammars to represent the topological and spatial distribution characteristics of different room types and use the Gaussian distribution to model the geometric characteristics of different room types. They are combined to infer the type of rooms. Grammar rules are mainly derived from guidebooks [23–26] about the design principles of research buildings. The idea is based on two assumptions: (1) Different room types follow certain spatial distribution characteristics and topological principles. For instance, offices are normally adjacent to external walls. Two offices are connected through doors. Multiple adjacent labs are clustered without being separated by other room types, such as toilets and copy rooms. (2) Different room types vary in geometric properties. For instance, a lecture room is normally much larger than a private office. We assume that the geometric properties (e.g., area, length, and width) of each room type follow the Gaussian distribution.

The input of the proposed approach is the geometric map of a single floor of a building without the room type information. The procedure of the proposed approach is as follows: We first obtain the frequency and the parameters of the multivariate Gaussian distribution of each room type from training rooms. Then, we improve the rules defined in the previous work [27] by removing a couple of useless rules and adding a couple of useful rules in semantic inference and changing the format of rules for the purpose of generating models to the format of rules for semantic inference. The next step is to partition these rules into multiple layers based on their dependency relationship. Then, we apply rules in primitive objects of each test floor-plan from the lowest layer to the highest layer to construct a parse forest. When applying rules at the lowest layer, Bayesian inference is used to calculate the initial probability of assigning an enclosed room with a certain type based on its geometric properties (e.g., area, length, and width) that are extracted from the geometric map. Low-ranking candidate types are removed to avoid the exponential growth of the parse trees. The constructed forest includes multiple parse trees with each corresponding to a complete semantic interpretation of the entire primitive rooms. The main contributions of this work include two parts:

- (1) To the best of our knowledge, this is the first time inferring the types of rooms by using grammars given geometric maps.
- (2) To a certain degree, we prove that grammars can benefit semantic enrichment.

The remainder of this paper is structured as follows: In Section 2, we present a relevant literature review. We introduce the semantic division of research buildings and the defined rules of the constraint attribute grammar in Section 3. In Section 4, we present the workflow and the details of each step of the proposed approach. In Section 5, we evaluate our approach using 15 floor-plans and discuss some issues in Section 6. We conclude the paper in Section 7.

2. Related Works

Models for indoor space. Currently, the mainstream geospatial standards that may cover indoor space and describe the spatial structure and semantics include CAD, BIM/IFC, city geography markup language (CityGML), and IndoorGML. CAD is normally used in the process of building construction, representing the geometric size and orientation of buildings' indoor entities. It uses the color and thickness of lines to distinguish different spatial entities. Apart from the notes related to indoor spaces, no further semantic information is represented in CAD. Compared to CAD, BIM is capable of restoring both geometric and rich semantic information of building components as well as their relationships [28]. It enables multi-schema representations of 3D geometry for indoor entities. The IFC is a major data exchange standard in BIM. It aims to facilitate the information exchange among stakeholders in AEC (architecture, construction, and engineering) industry [29]. Different from IFC, CityGML [30] is developed from a geospatial perspective. It defines the classes and relations for the most relevant topographic objects such as buildings, transportation, and vegetation in cities with respect to their geometrical, topological, semantic, and appearance properties [31]. CityGML has five levels of detail (LoDs), each for a different purpose. Particularly, the level of detail 4 (LoD 4) is defined to support the interior objects in buildings, such as doors, windows, ceiling, and walls. The only type for indoor space is room, which is surrounded by surfaces. However, they lack features related to indoor space model, navigation network, and semantics for indoor space, which are critical requirements of most applications of indoor spatial information [32]. In order to meet the requirements, IndoorGML is published by OGC (open geospatial consortium) as a standard data model and XML-based exchange format. It includes geometry, symbolic space, and network topology [33]. The basic goals of IndoorGML are to provide a common framework of semantic, topological, and geometric models for indoor spatial information, allowing for locating stationary or mobile features in indoor space and to provide spatial information services referring their positions in indoor space, instead of representing building architectural components.

Digitalization-based indoor modeling. The classical approach of parsing the scanned map or the image of floor-plans consists of two stages: Primitive detection and semantics recognition [6,34–37]. It starts from low-level image processing: Extracting the geometric primitives (i.e., segments and arcs) and vectorizing these primitives. Then, to identify the semantic classes of indoor spatial elements (e.g., walls, doors, windows, and furniture). In recent years, machine-learning techniques have been applied to detect the semantic classes (e.g., room, doors, and walls). For instance, de las Heras et al. [7,8,38] presented a segmentation-based approach that merges the vectorization and identification of indoor elements into one procedure. Specifically, it first tiles the image of floor plans into small patches. Then, specific feature descriptors are extracted to represent each patch in the feature space. Based on the extracted features, classifiers such as SVM can be trained and then used to predict the class of each patch. With the rapid development of deep learning in computer vision, deep neural networks have also been applied in parsing the image of floor plans. For instance, Dodge et al. [9] adopted the segmentation-based approach and fully convolutional network (FCN) to segment the pixels of walls. The approach achieves a high identification accuracy without adjusting parameters for different styles. Overall, the digitalization-based approach is useful considering the existence of substantial images of

floor-plans. However, it is incapable of identifying the type of rooms if the image contains no text information that indicates the type of rooms.

Image-based indoor modeling. Image-based indoor modeling approaches can capture accurate geometric information by using smartphones. The advent of depth camera (RGB-D) further improves the accuracy and enables capturing rich semantics in indoor scenes. For instance, Sankar and Seitz [39] proposed an approach for modeling the indoor scenes including offices and houses by using cameras and inertial sensors equipped on smartphones. It allows users to create an accurate 2D and 3D model based on simple interactive photogrammetric modeling. Similarly, Pintore and Gobbetti [40] proposed generating metric scale floor plans based on individual room measurements by using commodity smartphones equipped with accelerometers, magnetometers, and cameras. Furukawa et al. [14] presented a Manhattan-world fusion technique for the purpose of generating floor plans for indoor scenes. It uses structure-from-motion, multiview stereo (MVS), and a stereo algorithm to generate an axis-aligned depth map, which is then merged with MVS points to generate a final 3D model. The experimental results of different indoor scenes are promising. Tsai et al. [41] proposed using motion cues to compute likelihoods of indoor structure hypotheses, based on simple geometric knowledge about points, lines, planes, and motion. Specifically, a Bayesian filtering algorithm is used to automatically discover 3-D lines from point features. Then, they are used to detect planar structures that forms the final model. Ikehata et al. [16] presented a novel 3D modeling framework that reconstructs an indoor scene from panorama RGB-D images and structure grammar that represents the semantic relation between different scene parts and the structure of the rooms. In the grammar, a scene geometry is represented as a graph, where nodes correspond to structural elements such as rooms, walls, and objects. However, these works focused on capturing mainly the geometric layout of rooms without semantic representation. To enrich the semantics of reconstructed indoor scenes, Zhang et al. [18] proposed an approach to estimate both the layout of rooms as well as the clutter (e.g., furniture) that compose the scene by using both appearance and depth features from RGB-D Sensors. Briefly, image-based approaches can accurately reconstruct the geometric model and even the objects in the scene, but they normally dismiss the estimation of room types.

Trace-based indoor modeling. Trace-based solutions assume that users' traces reflect accessible spaces, including unoccupied internal spaces, corridors, and halls. With enough traces, they can infer the shape of rooms, corridors, and halls. For instance, Alzantot and Youssef [4], Jiang et al. [42], and Gao et al. [5] used volunteers' motion traces and the location of landmarks derived from inertial sensor data or Wi-Fi to determine the accurate shape of rooms and corridors. However, the edge of a room sometimes is blocked by furniture or other obstacles. Users' traces could not cover these places, leading to inaccurate detection of room shapes. To resolve this problem, Chen et al. [43] proposed a CrowdMap system that combines crowdsourced sensory and images to track volunteers. Based on images and estimated motion traces, it can create an accurate floor plan. Recently, Gao et al. [44] proposed a Knitter system that can fast construct the indoor floor plan of a large building by a single random user's one-hour data collection efforts. The core part of the system is a map fusion framework. It combines the localization result from images, the traces from inertial sensors, and the recognition of landmarks by using a dynamic Bayesian network. Trace-based approaches can recognize partial semantic information, such as corridors, stairs, and elevators, but without the definition of room types.

LIDAR point cloud-based indoor modeling. These methods achieve a higher geometric accuracy of rooms than trace-based methods. For instance, Mura [45] et al. proposed reconstructing a clean architectural model for a complex indoor environment with a set of cluttered 3D input scans. It is able to reconstruct a room graph and accurate polyhedral representation of each room. Another work concerned mainly recovering semantically rich 3D models. For instance, Xiong et al. [10] proposed a method to automatically converting the raw 3D point data into a semantically rich information model. The points are derived from a laser scanner located at multiple locations throughout a building. It mainly models the structural components of an indoor environment, such as walls, floors, ceilings, windows, and doorways. Ambruş et al. [13] proposed an automatic approach to reconstructing a 2-D

floor plan from raw point cloud data using 3D point information. They can achieve accurate and robust detection of building structural elements (e.g., wall and opening) by using energy minimization. One of the novelties of the approach is that it does not rely on viewpoint information and Manhattan frame assumption. Nikoohemat et al. [46] proposed using mobile laser scanners for data collection. It can detect openings (e.g., windows and doors) in cluttered indoor environments by using occlusion reasoning and the trajectories from the mobile laser scanners. The outcomes show that using structured learning methods for semantic classification is promising. Armeni et al. [11] proposed a new approach to semantic parsing of large-scale colored point clouds of an entire building using a hierarchical approach: Parsing point clouds into semantic spaces and then parsing those spaces into their structural (e.g., floor, walls, etc.) and building (e.g., furniture) elements. It can capture rich semantic information that includes not only walls, floors, and rooms, but also the furniture in the room, such as chairs, desks, and sofas. Qi et al. [12] proposed a multilayer perceptron (MLP) architecture named PointNet on point clouds for 3D classification and segmentation. It extracts a global feature vector from a 3D point and processes each point using the extracted feature vector and additional point level transformations. PointNet is a unified architecture that directly takes point clouds as input and outputs either class labels for the entire input or per point segment/part labels for each point of the input. Their method operates at the point level and, thus, inherently provides a fine-grained segmentation and highly accurate semantic scene understanding. Similar to the image-based approaches, they normally dismissed the estimation of room types.

Rule-based indoor modeling. This group of approaches uses the structural rules or knowledge of a certain building type to assist the reconstruction of maps. The rules can be gained through manual definitions [27,47–49] or machine learning techniques [19,50–52]. Yue et al. [48] proposed using a shape grammar that represents the style of Queen Anne House to reason the interior layout of residential houses with the help of a few observations, such as footprints and the location of windows. The work in [49] used split grammars to describe the spatial structures of rooms. The grammar rules of one floor can be learned automatically from reconstructed maps and then be used to derive the layout of the other floors. In this way, fewer sensor data are needed to reconstruct the indoor map of a building. However, the defined grammar consists of mainly splitting rules, producing geometric structure of rooms rather than rooms with semantic information. Similarly, Khoshelham and Díaz-Vilariño [53] used a shape grammar [54] to reconstruct indoor maps that contain walls, doors, and windows. The collected point clouds can be used to learn the parameters of rules. Dehbi et al. [55] proposed learning weighted attributed context-free grammar rules for 3D building reconstruction. They used support vector machines to generate a weighted context-free grammar and predict structured outputs such as parse trees. Then, based on a statistical relational learning method using Markov logic networks, the parameters and constraints for the grammar can be obtained. Rosser et al. [50] proposed learning the dimension, orientation, and occurrence of rooms from true floor plans of residential houses. Based on this, a Bayesian network is built to estimate room dimensions and orientations, which achieves a promising result. Luperto et al. [52] proposed a semantic mapping system that classifies rooms of indoor environments considering typology of buildings where a robot is operating. More precisely, they assume that a robot is moving in a building with a known typology, and the proposed system employs classifiers specific for that typology to semantically label rooms (small room, medium room, big room, corridor, hall.) identified from data acquired by laser range scanners. Furthermore, Luperto et al. [19] proposed using a statistical relational learning approach for global reasoning on the whole structure of buildings (e.g., office and school buildings). They assessed the potential of the proposed approach in three applications: Classification of rooms, classification of buildings, and validation of simulated worlds. Liu et al. [56] proposed a novel approach for automatically extracting semantic information (e.g., rooms and corridors) from more or less preprocessed sensor data. They propose to do this by means of a probabilistic generative model and MCMC-based reasoning techniques. The novelty of the approach is that they construct an abstracted semantic and top-down representation of the domain under consideration: A classical indoor environment consisting of

several rooms, that are connected by doorways. Similarly, Liu et al. [57] proposed a generalizable knowledge framework for data abstraction. Based on the framework, the robot can reconstruct a semantic indoor model. Specifically, the framework is implemented by combining Markov logic networks and data-driven MCMC sampling. Based on MLNs, we formulate task-specific context knowledge as descriptive soft rules. Experiments on real world data and simulated data confirm the usefulness of the proposed framework.

3. Formal Representation of Layout Principles of Research Buildings

3.1. Definition of Research Buildings

Research buildings are the core buildings at universities, including laboratories [26] and office buildings. Specifically, laboratories refer to the academic laboratories of physical, biological, chemical, and medical institutes. They have a strict requirement on the configuration of labs [21]. According to [21], we categorize the enclosed rooms in research buildings into 11 types: Labs, lab support spaces, offices, seminar/lecture rooms, computer rooms, libraries, toilets, copy/print rooms, storage rooms, lounges, and kitchens. Labs refer to the standard labs that normally follow the module design principle [24,25] and are continuously occupied throughout working days. Thus, they are located in naturally lit perimeter areas. Lab support spaces consist of two parts. One is the specialist or non-standard laboratories, which do not adopt standard modules and are generally not continuously occupied. Therefore, they may be planned in internal areas. The other is ancillary spaces that support labs, such as equipment rooms, instrument rooms, cold rooms, glassware storage, and chemical storage [25].

This work focuses on the typical research building, which refers to those research buildings whose layouts are corridor based. Figure 1 shows the three types of layouts of typical research buildings based on the layout of corridors: Single-loaded, double-loaded, and triple-loaded [24], as shown in Figure 1. Most research buildings are the variations of them.

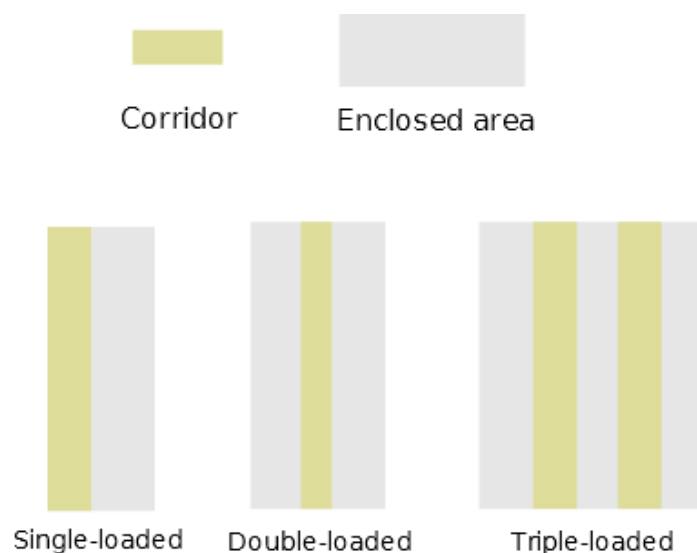


Figure 1. Three typical plans of research buildings.

3.2. Hierarchical Semantic Division of Research Buildings

We use a UML class diagram to represent the hierarchical semantic division of research buildings, as shown in Figure 2. Note that all the defined objects in the diagram are based on one floor of a building. We ignore the multi-level objects that cross multiple floors (e.g., atrium). A building consists of one or more building units that are adjacent or connected through overpasses. Each building unit has a core function, including lab-centered (e.g., laboratories), office-centered (e.g., office buildings),

and academic-centered (e.g., lectures and libraries). Physically, a building unit contains freely accessible spaces (e.g., corridors and halls) and enclosed areas. Enclosed areas can be categorized into two types according to the physical location: The perimeter area on external walls and the central dark zone for lab support spaces and ancillary spaces [24]. A central dark zone does not mean the area is dark without light but refers to the area that is located in the center of a building and cannot readily receive natural light. A perimeter area is divided into a primary zone and optional ancillary spaces at the two ends of the primary zone. A primary zone has three variations: Lab zone, office zone, and academic zone. A primary zone is further divided into single room types: Labs, offices, lab support spaces, seminar rooms, and libraries.

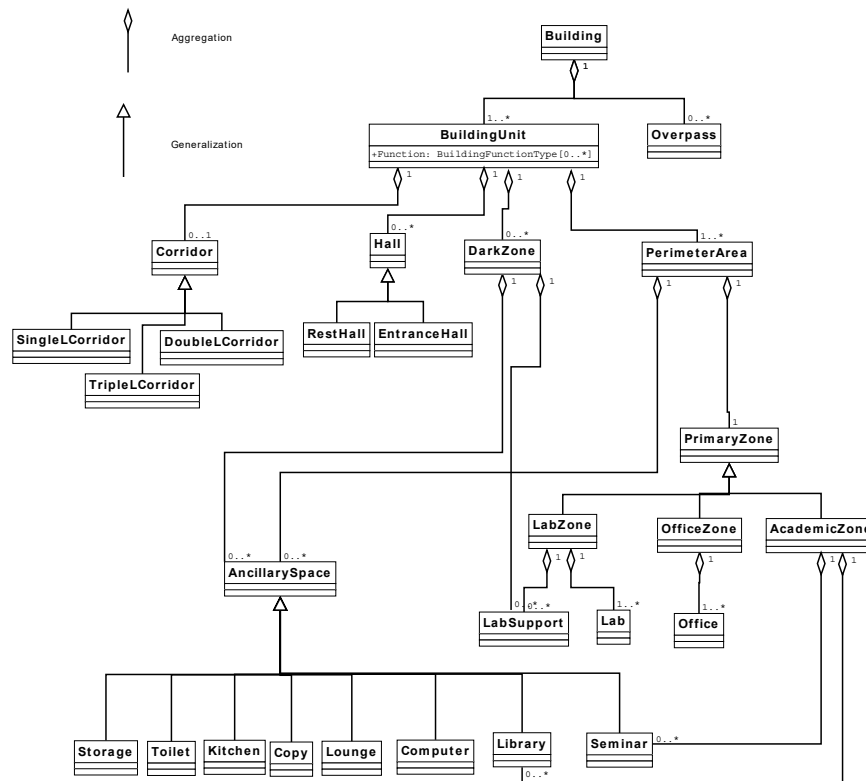


Figure 2. Semantic division of research buildings.

We take an example in Figure 3 to explain the semantic division of research buildings. Figure 3a shows a building consists of two adjacent building units. In Figure 3b, the left building unit is divided into three perimeter areas (1, 4, and 5), two central dark zones (2 and 3), one triple-loaded corridor, and one entrance hall. The right one is divided into one perimeter area and one single-loaded corridor. In Figure 3c, perimeter areas 1, 4, 5, and 6 are divided into a lab zone without ancillary spaces, an office zone without ancillary spaces, an office zone with ancillary spaces, and an academic zone without ancillary spaces, respectively. In Figure 3d, the lab zone is divided into single labs and lab support spaces. The two office zones are divided into multiple offices. The ancillary spaces are divided into a computer room and a seminar room. The academic zone is divided into two seminar rooms. The two central dark zones are divided into multiple lab supported spaces and toilets.

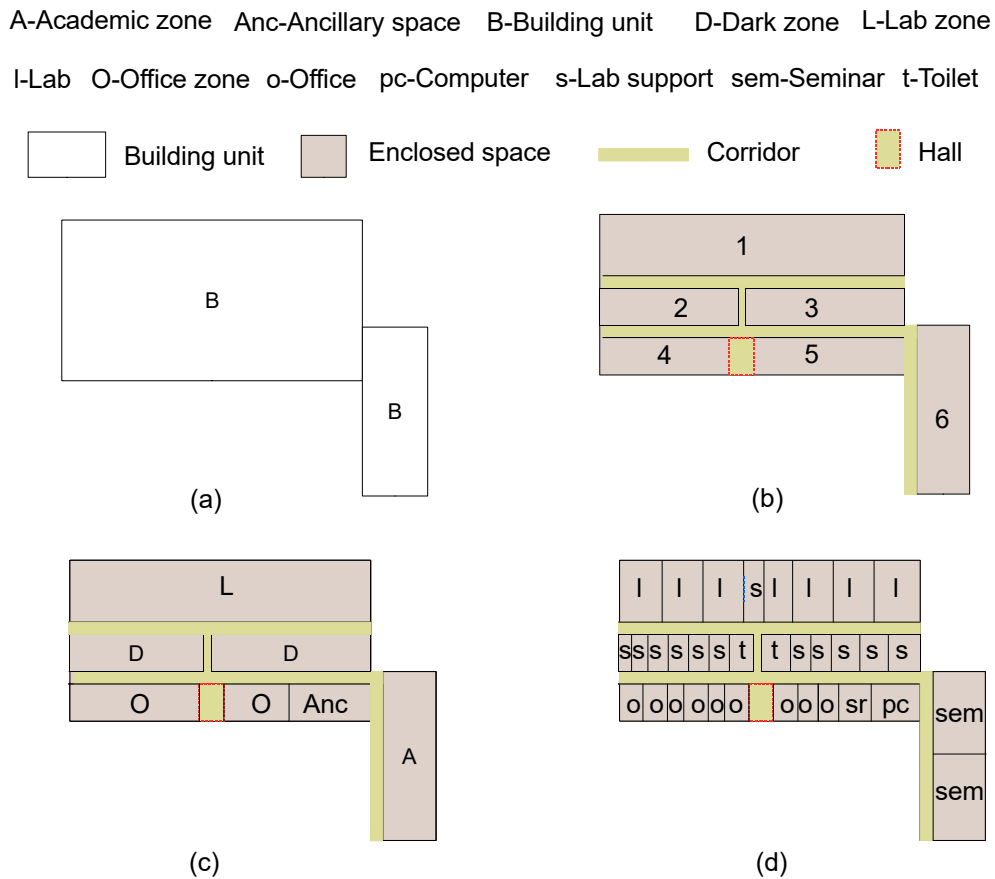


Figure 3. An example of semantic division of research buildings. (a) two building units in a floor. (b) corridors, halls, and enclosed zones in each building unit (c) type of each zone (d) single room types in each zone.

3.3. Constrained Attribute Grammar

Equation (1) formulates a typical rule of constrained attribute grammars [58–60]. p denotes the probability of applying the rule or generating the left-hand object with the right-hand objects. In this work, all the generated left-hand objects are assigned equal probability value of one except the generated objects that corresponds to the 11 room types (such as lab, office, and toilet) whose probability is estimated through the Bayesian inferring method given the geometric properties of primitive rooms. Z represents the parental or superior objects that can be generated by merging the right-hand objects denoted by X_k . x_k and z denote the instance of an object. Constraints define the preconditions that should be satisfied before applying this rule. The attribute part defines the operations that should be conducted on the attribute of the left-hand object.

$$p : Z z \rightarrow X_1 x_1, X_2 x_2 \dots X_k x_k \langle \text{Constraints} \rangle \{ \text{Attribute} \} \tag{1}$$

To simplify the description of rules, we define a collection operation *set*. It is used to represent multiple objects in the same type. For instance, $set(\text{office}, k)$ o . defines a set of (k) office objects, denoted by o .

3.4. Predicates

A condition is a conjunction of predicates applied in rule variables x_i , possibly via attributes. Predicates primarily express geometric requirements but can generally represent any constraints on the underlying objects. In this work, we define several predicates by referring to guidebooks about the design principles of research buildings [23–26].

edgeAdj(a,b): Object a is adjacent to object b via a shared edge without inclusive relationships between a and b .

inclusionAdj(a,b,d): Object a includes object b and they are connected through an internal door d .

withExtDoor(a): Object a has an external door connected to corridors.

onExtWall(a): Object a is at the edge of external walls.

inCenter(a): Most of the rooms in object a (zone) is not located at the external walls of buildings.

conByIntDoor($\{a_1, a_2 \dots a_k\}, \{d_1, d_2 \dots d_m\}$): Multiple objects $\{a_1, a_2 \dots a_k\}$ are connected through internal doors $\{d_1, d_2 \dots d_m\}$.

isTripleLoaded(a): Building a owns a triple-loaded circulation system.

isDoubleLoaded(a): Building a owns a double-loaded circulation system.

formFullArea($\{a_1, a_2 \dots a_k\}$): Multiple objects form a complete area (e.g., a perimeter area or central dark zone), including all the primitive rooms and internal doors.

Figure 4 illustrates the defined predicates.

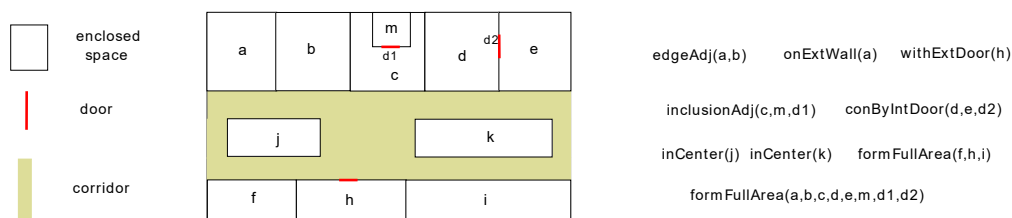


Figure 4. Predicates used in this work.

3.5. Defined Rules

We define 16 rules in total, which can be found in the Appendix A. Note that ‘|’ denotes the OR operation. The objects in the rules correspond to the objects in Figure 2. Specifically, *Ancillary*, *Zone*, *Center*, *CZone*, *BUnit*, and *Building* objects in the rules correspond to the *AncillarySpace*, *PrimaryZone*, *DarkZones*, *PerimeterArea*, *BuildingUnit*, and the *Building* object in Figure 2, respectively. These rules are described as follows:

- A1:** A *room* object can be assigned with one of the eight types. When applying this rule, Bayesian inference methods are used to calculate the initial probability of belonging the room to corresponding type.
- A2:** A *Toilet* object is generated by merging one to three *room* objects when they satisfy the predicate *conByIntDoor* and only one of the *room* objects has an external door. Bayesian inference techniques are used to calculate the mean initial probability of each room to a toilet.
- A3:** Toilet, Copy, Storage, Kitchen, Lounge, Computer, Lecture, and Library objects are interpreted as *Ancillary* objects.
- A4:** A *Library* object is generated by merging a couple of *room* objects when they are connected by internal doors. Bayesian inference methods are used to calculate the mean probability of each room belonging to a library.
- A5:** A couple of *lecture* objects that are adjacent or connected by internal doors can be interpreted as an academic *Zone*.
- A6:** A *Library* object is interpreted as an academic *Zone*.
- A7:** A *Lab* object is generated by merging a single *room* r^l and an optional internal *room* r^w included by r^l when r^l is on external walls. The Bayesian inference method is used to calculate the initial probability of r^l belonging to a lab.
- A8:** A *LGroup* object is generated by merging at least one *Lab* object and optional *Support* objects when they are connected by internal doors.
- A9:** A lab *Zone* is generated by merging multiple adjacent *LGroup* objects.

- A10:** A *room* object r^p with an optional internal *room* r^s contained by r^p can be explained as an *Office* object if r^p has an external door. The Bayesian inference method is used to calculate the initial probability of r^p belonging to an office.
- A11:** An *office Zone* can be generated by merging multiple *Office* objects if they are adjacent or connected through internal doors.
- A12:** A *Center* object can be generated by combining at most three *Ancillary* objects and optional adjacent or connected *Support* objects if the generated object satisfies the predicate *formFullArea*. If no *Support* objects exist, the type of the generated *Center* object is assigned *ancillary* otherwise *support*.
- A13:** A *CZone* object can be generated by combining at most three *Ancillary* objects and a *Zone* object if the generated object satisfies the predicate *formFullArea*.
- A14:** An office-centered or academic-centered building unit can be generated by merging at least one *CZone* object with the type of *office*, at most two *Center* objects with the type of *ancillary*, and at most two *CZone* objects with the type of *academic* if the generated object satisfies the predicate *formFullArea*.
- A15:** A lab-centered building unit can be generated by merging at least one *CZone* object with the type of *lab*, at least one *CZone* object with the type of *office*, and optional *CZone* objects with the type of *academic* if the generated object satisfies the predicate *formFullArea*. Note that if the building unit has a triple-loaded circulation system (with central dark areas), there exists at least one *Center* object with the type of *support*.
- A16:** A *Building* object can be generated by combining all the *BUnit* objects if they are adjacent.

4. Algorithm of Inferring Room Types

4.1. Workflow

The workflow of the proposed method is depicted in Figure 5. The input mainly consists of three parts. The first is the training data, including multiple rooms with their four properties (e.g., area, length, width, and room type). Based on the training data, we can extract the parameters of the Gaussian distribution for each room type. The second is the geometric map of the test scene. The third is the grammar rules, which are first partitioned into layers. Then, they are applied from the lowest layer to the highest layer in primitives to build a parse forest. The primitives are derived from the inputting geometric map, including enclosed rooms and internal doors. The reason that we do not infer corridors, halls, and stairs is that it is easy to identify them with point cloud and trace-based techniques [42,44]. When applying rules to assign rooms with certain types, the initial probability is calculated by using the Bayesian inferring method based on the geometric properties of the rooms (i.e., area, length, and width) that are extracted from the inputting geometric map. Finally, we can calculate the probability of belonging a room to a certain type based on the parse forest. The one with the highest probability is selected as the estimated type of the room.

4.2. Bayesian Inference

Different room types vary in geometric properties, such as length, width, and area. For instance, normally, the area of a seminar room is much larger than an office. We redefine the length and width of a rectangular room that is located at an external wall (denoted by w) as: The width of the room equals the edge that is parallel with w , while the length of the room corresponds to the other edge. For the rooms not located at external walls or located at multiple walls, the width and length follow their original definitions.

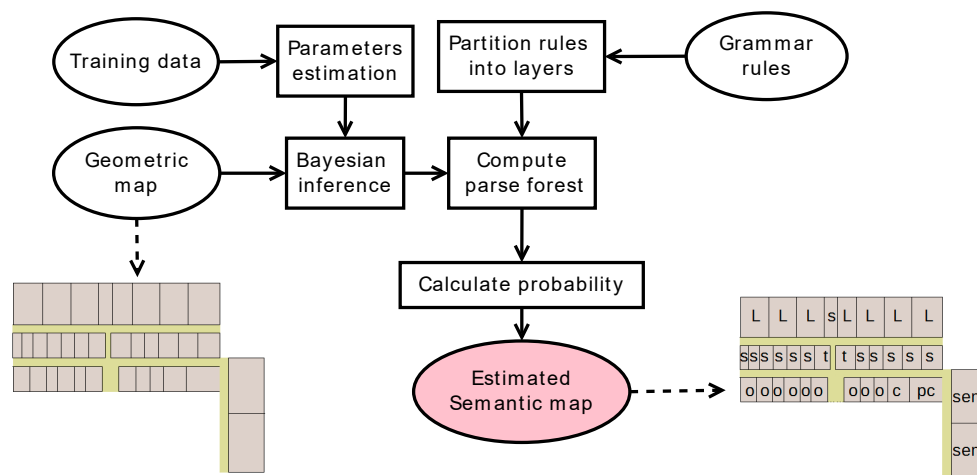


Figure 5. Workflow of proposed algorithm.

Given the geometric properties of a room, we can calculate the initial probability of the room belonging to a certain type by using the Bayesian probability theory, which will be invoked in the bottom-up approach when applying rules 1, 2, 4, 7, and 10. The estimated initial probability represents the probability of generating corresponding superior objects (room type) by applying rules 1, 2, 4, 7, and 10, which is attached to the generated objects. We use vector $x = (w, l, a)$ to denote the geometric properties of a room, where $w, l,$ and a denote the width, length, and area, respectively. We use t to denote the type of rooms. Thus, the probability estimation equation can be written as:

$$p(t|x) = \frac{p(x|t)p(t)}{p(x)}. \tag{2}$$

In the equation, $P(t)$ represents the prior probability, which is approximated as the relative frequency of occurrence of each room type. $P(x)$ is obtained by integrating (or summing) $P(x|t)P(t)$ over all t , and plays the role of an ignorable normalizing constant. $P(x|t)$ refers to the likelihood function. It assesses the probability of the geometric properties of a room arising from the room types. To calculate the likelihood, we assume that the variables $w, l,$ and a follow the normal distribution. The likelihood function is then written in the following notation:

$$p(x|t) = \frac{\exp(-\frac{1}{2}(x - u_t)^T \Sigma_t^{-1}(x - u_t))}{\sqrt{(2\pi)^k |\Sigma_t|}}. \tag{3}$$

In Equation (2), u_t is a 3-vector, denoting the mean value of the geometric properties of rooms in type t . Σ_t is the symmetric covariance matrix of the geometric properties of rooms in type t . Given a room with geometric properties $x = (w, l, a)$, we can first calculate the probabilities of the room belonging to one of the 11 types, denoted by $\hat{p}_i, 1 < i < 11$. Then, the low-ranking candidate types are deleted, and the top T room types are kept. Their probabilities are then normalized. In this work, T is set to 5.

4.3. Compute Parse Forest

A parse tree corresponds to a semantic interpretation of one floor of a building. The proposed approach produces multiple interpretations that are represented by a forest. In this work, we use a bottom-up approach to construct the parse forest. Specifically, we continuously apply rules to merge the inferior objects into the superior objects of the rules if the inferior objects satisfy the preconditions of the rules. This process will terminate until no rules can be applied anymore. The inferior objects

refer to the objects at the right-hand side of a rule and the superior objects refer to the objects at the left-hand side of a rule.

To improve the efficiency of searching proper rules during the merging procedure, we first partition the rules into multiple layers. The rules at the lower layers are applied ahead of the rules at the upper layers.

4.3.1. Partition Grammar Rules into Layers

Certain rules have more than one right-hand object, such as rule $\bar{r}: Zz \rightarrow Xx, Yy, Pp$. These rules can be applied only if all of their right-hand objects have been generated. That is, a rule denoted by \acute{r} with X, Y , or P as the left-hand object should be applied ahead of rule \bar{r} . Then, we define that rule \bar{r} depends on rule \acute{r} . The dependency among the entire rules can be represented with a directed acyclic graph, in which a node denotes a rule and an edge with an arrow denotes the dependency. Based on the dependency graph, we can partition grammar rules into multiple layers. The rules at the lowest layer do not depend on any rules. The process of partitioning rules into multiple layers is described as follows:

- (1) Build dependency graph. Traversal each rule and draw a direct edge from current rule to the rules whose left-hand objects intersect the right-hand objects of this rule. If the right-hand objects of a rule include only primitive objects (e.g., rooms and doors), it is treated as a free rule.
- (2) Delete free rules. Put the free rules at the lowest layer and then delete the free rules and all the edges connecting them from the graph.
- (3) Handle new free rules. Identify new free rules and put them at the next layer. Similarly, delete the free rules and the corresponding edges. Repeat this step until no rules exist in the graph.

4.3.2. Apply Rules

After partitioning rules into layers, we then merge inferior objects into superior objects by applying rules from the lowest layer to the highest layer. During the procedure, if the generated superior objects correspond to a certain room type, the Bayesian inference method that is described in Section 4.2 is used to calculate the initial probability, which is assigned to the generated object. Otherwise, a probability value of one is assigned to the generated object. The process of computing a parse forest is as follows:

- (1) Initialize an object list with the primitives and set the current layer as the first layer.
- (2) Apply all the rules at the current layer to the objects in the list to generate superior objects.
- (3) Fill the child list of the generated object with the inferior objects that form the generated objects.
- (4) Assign a probability value to newly generated objects. When applying rules 1, 2, 4, 7, and 10, the probability is estimated through the Bayesian inference. Otherwise, we assign a probability of one to the generated objects.
- (5) Add the newly generated objects to the object list.
- (6) Move to the next layer and repeat steps (2)–(6).
- (7) Create a *root* node and add all the *Building* objects to its child list.

We take a simplified floor-plan (shown in Figure 6) as an example to illustrate the procedure of creating the parse forest by using the proposed bottom-up method. The floor plan consists of three rooms, one internal and three external doors (denoted by solid red lines), and a corridor (with a yellow background). Based on this floor plan, hundreds of parse trees can be generated that form the parse forest. Here, we only choose three parse trees as examples. To clearly illustrate the bottom-up approach, we divide the procedure of constructing a forest into multiple sub-procedures such that each procedure constructs a single tree. We denote the four primitives including three rooms and an internal door by $r1, r2, r3$, and d , respectively. In Figure 7a, the initial structure of a tree is the four primitives, playing the role of leaf nodes. In the second step, $r1, r2$, and $r3$ are interpreted as three offices denoted by $O1, O2$, and $O3$, respectively by applying rule A10. Next, $O1, O2, O3$, and d are

merged into a *Zone* object, denoted by *Zone1* by applying rule **A11**. Finally, a *Building* object can be created by applying rules **A13**, **A14**, and **A16** successively. Similarly, another two trees can be created as shown in Figure 7b,c, where *Anc*, *T*, and *Sem*, denote *Ancillary*, *Toilet*, and *Lecture* objects described in the rules, respectively. Note that, in the three trees, the nodes with the same name (e.g., the node with the name of *O1* in the first and second trees) refer to the same node in the finally constructed forest. By merging the same nodes in these trees, we can obtain an initial forest (the left forest in Figure 8), where each node points to its children that form the node.

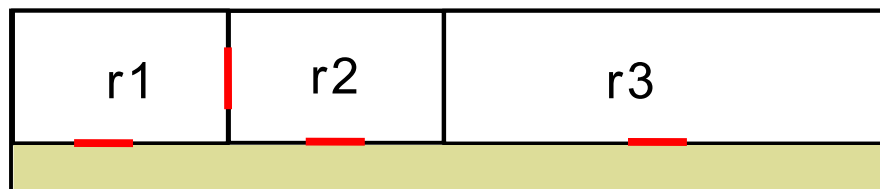


Figure 6. A simplified floor plan with three rooms.

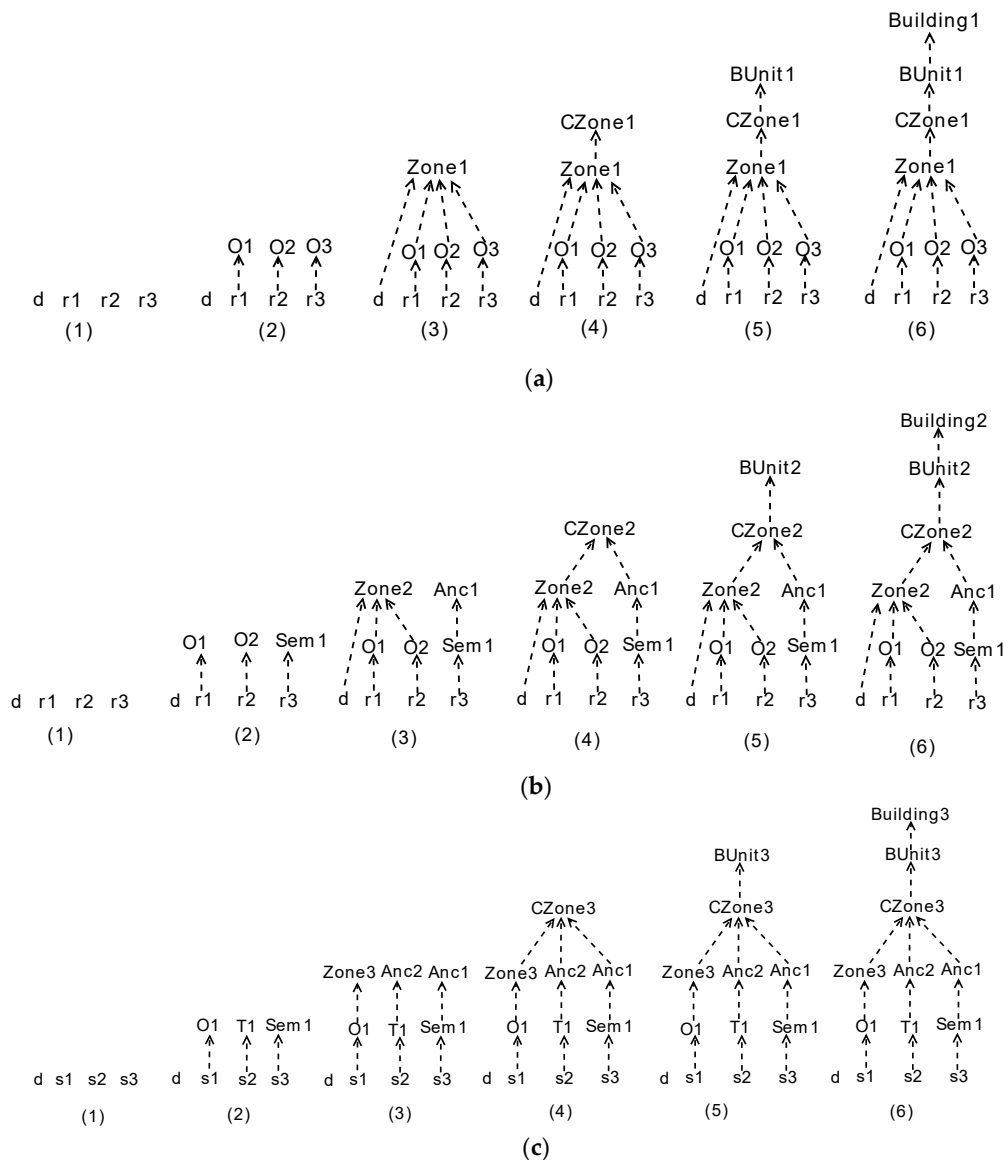


Figure 7. Procedure of creating parse forest by using bottom-up methods. (a) procedure of constructing first parse tree (b) procedure of constructing second parse tree (c) procedure of constructing third parse tree.

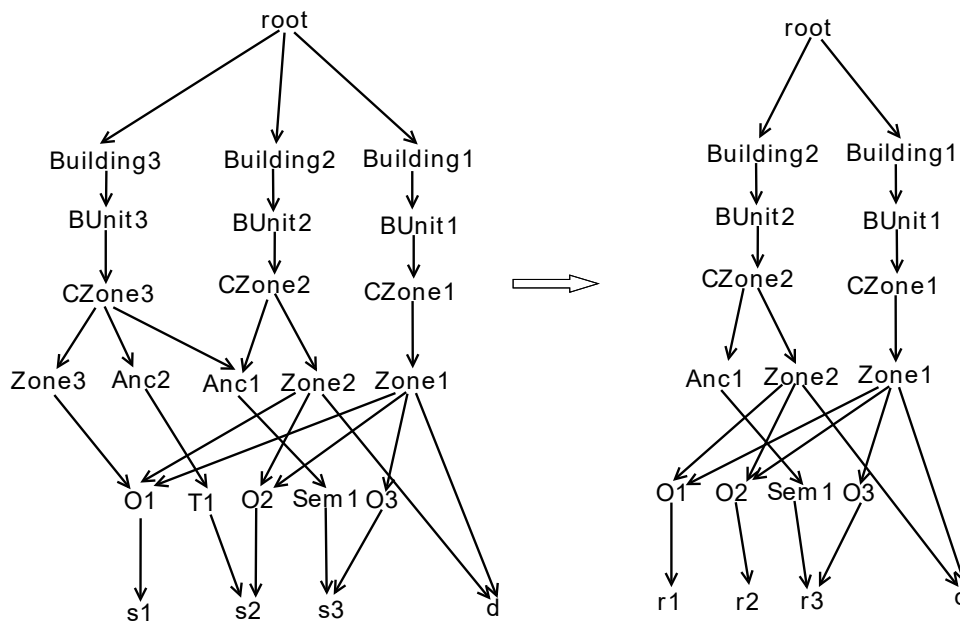


Figure 8. Pruning incomplete trees from parse forest.

The leaf nodes of a parse forest are primitives, including enclosed rooms and internal doors. The root node of the forest links to multiple *Building* nodes. Starting from a *Building* node, we can traversal its child list until the leaf nodes to find a parse tree. During the creation of the parse forest, immature or incomplete trees might be created if the semantic interpretation of these trees violate the defined rules. Thus, they are pruned from the forest. In this way, incorrect semantic interpretation (type) of the rooms can be removed. An incomplete tree refers to the tree whose root node is a *Building* node, but leaf nodes include only partial primitives. For example, the third tree in Figure 7 is an incomplete tree since its leaf nodes miss the internal door *d*. This can be explained by the fact that connecting an office and a toilet with internal doors rarely happens. Thus, this tree is pruned from the forest, as shown in Figure 8. The other two trees are valid parse trees.

The pseudocode of the algorithm that calculate the parse forest is described as follows:

```

Procedure  $F = \text{ComputeParsingForest}((R_i)_{i \leq h}, G);$ 
Input:
 $(R_i)_{i \leq h}$  // partitioned rules.  $h$  denotes the number of layers.
 $G$  // all the primitives: rooms and internal doors
Output:
 $F$  // parse forest
begin
 $O \leftarrow \text{initializeObjects}(G)$  // initialize object list with  $G$ 
for  $i = 1$  to  $h$  do
for each rule  $\bar{r} \in R_i$  do
 $O \leftarrow \text{applyRule}(\bar{r}, O)$ 
end
end
 $F.\text{child\_list} \leftarrow \text{null}$ 
for each object  $\bar{o} \in O$  do
if the type of  $\bar{o}$  is a Building
 $F.\text{child\_list} \leftarrow F.\text{child\_list} \cup \{\bar{o}\}$ 
end
end
end

```

Procedure *initializeObjects*(G) initials the object list with primitive objects (e.g., rooms and internal doors), which are treated as the leaf nodes of a parse forest. Procedure *applyRule*(\bar{r}, O) searches the objects in O that satisfy the preconditions of rule \bar{r} . Then, they are merged to form superior objects that are at the left-hand side of rule \bar{r} . The probability of generating the superior object or applying the rule is estimated through the Bayesian inference method or is set to one.

4.4. Calculating Probability

Given a pruned forest with t parse trees, we can traversal each tree starting from a *Building* node until their leaf nodes (e.g., primitive rooms). For a primitive room r in the parse forest, the probability value attached to its parental object (a certain room type) in a tree is denoted by $\bar{p}_i, 1 \leq i \leq t$ and the probability value attached to its parental object (a certain room type) that matches its true type are denoted by $\tilde{p}_k, 1 \leq k \leq m$, where m denotes the number of trees where room s is correctly assigned the type. The probability of room r belonging to its true type thus equals $\sum_{k=1}^m \tilde{p}_k / \sum_{i=1}^t \bar{p}_i$. Assume that the true type of r_1, r_2 , and r_3 in Figure 6 are office (O), office (O), and lecture (Sem), respectively, and the forest in the right side of Figure 8 is the estimated forest. We denote the assigned probability value to nodes $O1, O2, O3$, and $Sem1$ by $\check{p}_1, \check{p}_2, \check{p}_3$, and \check{p}_4 , respectively, which are estimated through the Bayesian inference method. Thus, the probability of room r_1, r_2 , and r_3 belonging to their true types equal $(\check{p}_1 + \check{p}_1) / (\check{p}_1 + \check{p}_1), (\check{p}_2 + \check{p}_2) / (\check{p}_2 + \check{p}_2)$, and $\check{p}_4 / (\check{p}_3 + \check{p}_4)$, respectively. Similarly, we can calculate the probability of belonging r_1, r_2 , and r_3 to the other room types (apart from the true type). Finally, for a room, the candidate type with the highest probability is selected as the estimated type of the room.

5. Experiments

5.1. Training Data

We collect 2304 rooms from our campuses. A 2304-by-4 matrix D is used to describe the training data. Each row of the matrix corresponds to a room, representing its four properties: Room type, area, length, and width. From the matrix, we can extract the relative frequency of occurrence of each room type, as shown in Figure 9. Further, for each room type, we can calculate the covariance matrix (3-by-3) and mean vector of the area, width, and length.

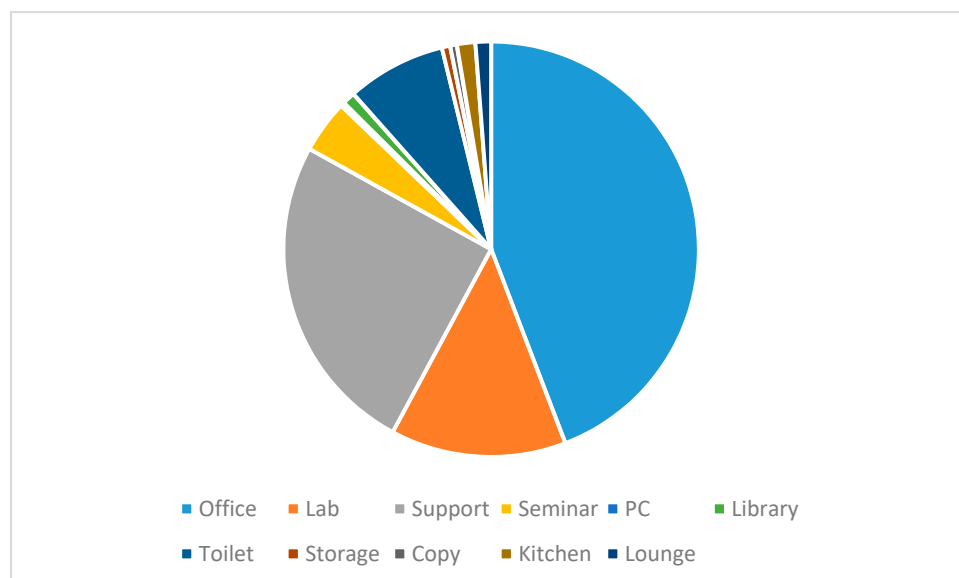


Figure 9. Proportion of different room types in training rooms.

5.2. Testbeds

We choose 15 buildings distributed in two campuses of Heidelberg University as the test bed, as shown in Figure 10. The footprints of these buildings include external passages, foyers, and external vertical passages; therefore, some are non-rectilinear polygons. We manually extract 15 rectilinear floors from these buildings by deleting the external parts, as shown in Figure 11. Table 1 shows the number of lab-centered, office-centered, and academic-centered building units in each floor plan.

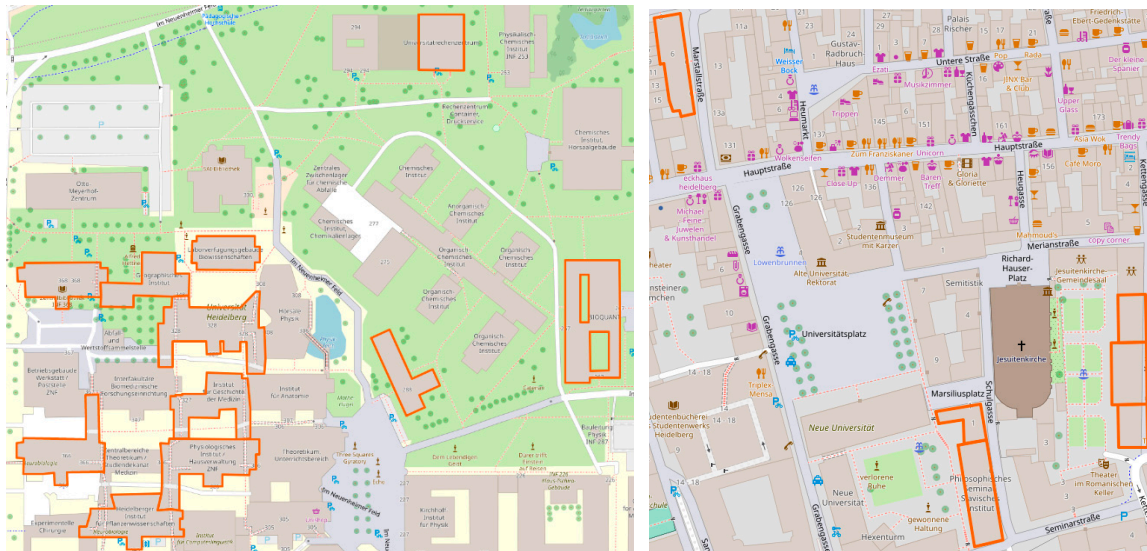


Figure 10. Distribution of test buildings.

Table 1. The number of different types of building units in each floor plan.

Floor Plan	Lab-Centered	Office-Centered	Academic-Centered
(a)	1	0	1
(b)	1	1	0
(c)	0	0	2
(d)	1	0	0
(e)	1	0	0
(f)	1	0	0
(g)	1	0	0
(h)	0	1	0
(i)	0	1	2
(j)	0	1	1
(k)	0	1	0
(l)	0	1	1
(m)	1	0	0
(n)	0	1	0
(o)	0	2	0

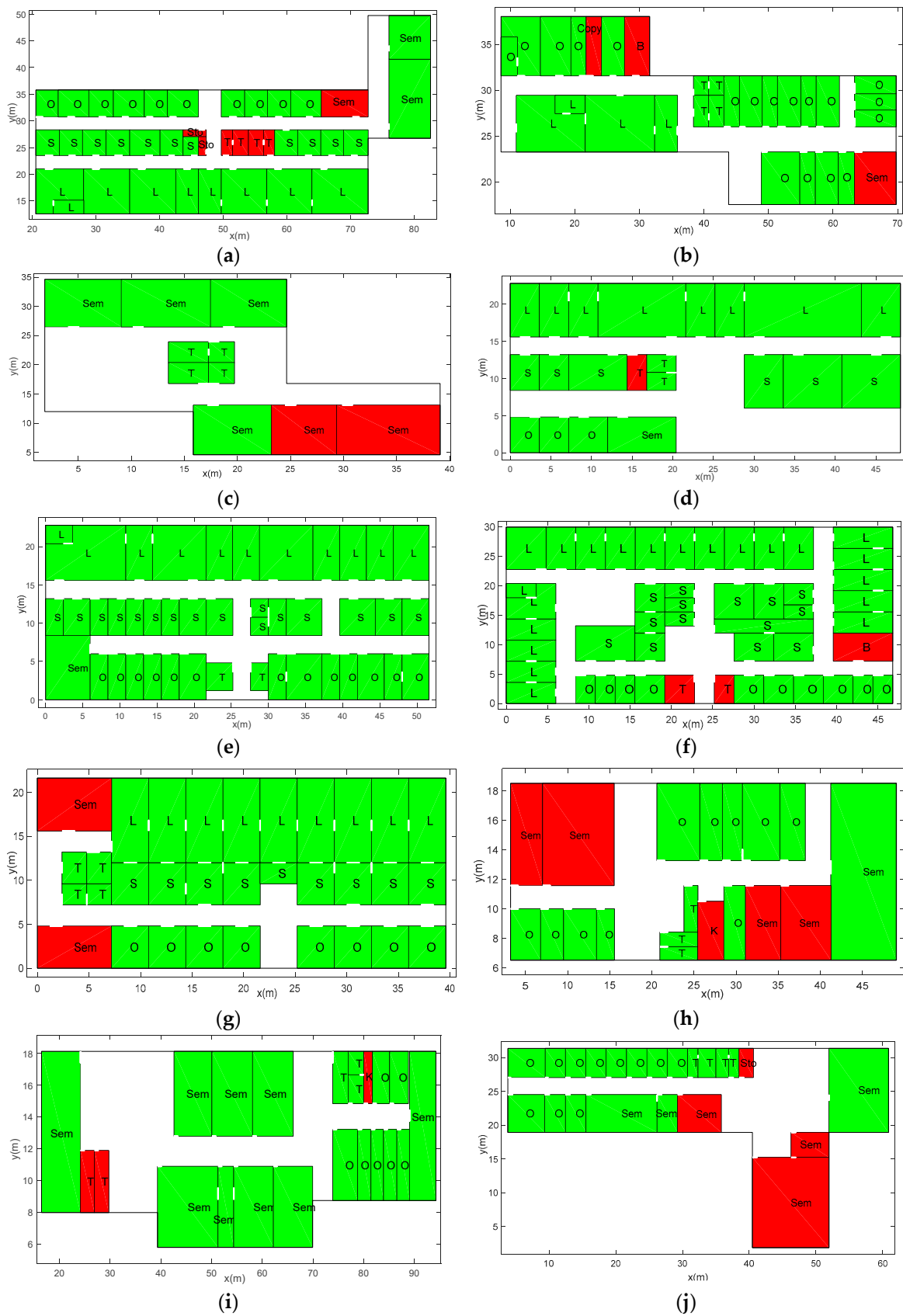


Figure 11. Cont.

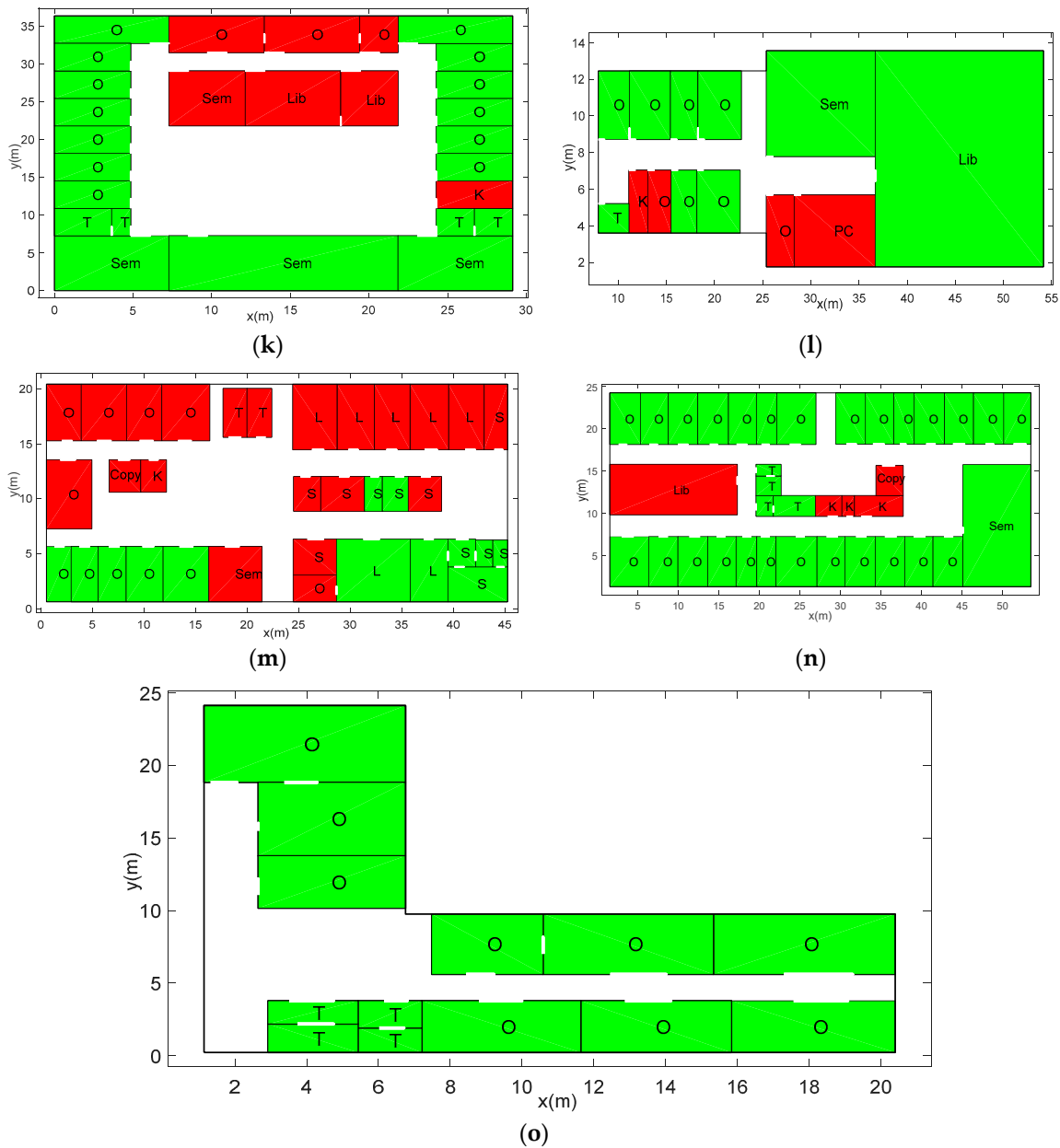


Figure 11. Floor plans (a–o) used for test.

We extract the geometric map from a scanned floor-plan by manually tagging the footprint of the building, the shape of rooms and corridors, and the location of both internal and external doors. All the lines are represented in pixels coordinates. In this procedure, we ignore the furniture in rooms. Then, based on the given area of a room that is tagged on the scanned map, we can convert the pixel coordinate of lines to a local geographic coordinate. Finally, the geometric size of rooms, corridors, and doors, as well as the topology relationship can be obtained. In this work, we assume that these spatial entities are already known since it is easy to detect them by current indoor mapping solutions [42,44,46]. Single rooms and internal doors are treated as the primitives. Internal doors refer to the doors connecting two rooms while external doors refer to the doors connecting a room and corridors. Doors are denoted by blank segments at the edge of rooms. Moreover, we delete a couple of spaces from the testbed since they are insignificant or easily detected by measurement-based approaches, such as electricity room and staircases. There are 408 rooms in total with each having a label, representing its type. Labels O, L, S, Sem, Lib, T, PC, Sto, C, K, and B denote offices, labs, lab support rooms,

seminar rooms, libraries, toilets, computer rooms, storage rooms, kitchens, and lounges (break rooms), respectively. Note that the created grammars cover the room unit that consists of multiple sub-spaces (rooms) connected by internal doors. In this work, each sub-space is assigned to a certain type. In the test floor-plans, there are many room units, such as the one that consists of multiple labs connected by internal doors in floor-plan (a), the one that consists of multiple support spaces connected by internal doors in floor-plan (e), the one that consists of multiple labs and lab support spaces connected by internal doors in floor-plan (g), and the one that consists of multiple seminar rooms in floor-plan (i).

5.3. Experimental Results

As far as we know, currently, no works in indoor mapping (such as image- or Lidar-based approaches) have explicitly detected the room type in research buildings. Therefore, we only demonstrate the room tagging result of our proposed approach without comparing the results with other approaches. For a test floor plan, the identification accuracy denotes the proportion of the rooms whose type are corrected predicted among all the rooms in the floor plan. The average identification accuracy in 15 test floor plans reaches 84% by using our proposed method, as shown in Table 2. We use a green background and a red background to denote the room whose type is correctly and incorrectly identified, respectively, as shown in Figure 11.

Table 2. Identification accuracy of each floor plan.

Floor Plan	Identification Accuracy	Number of Rooms	Time Consumption(s)
Floor plan (a)	0.82	39	8.05
Floor plan (b)	0.90	29	3.93
Floor plan (c)	0.80	10	2.40
Floor plan (d)	0.95	21	3.10
Floor plan (e)	1.00	43	27.02
Floor plan (f)	0.94	48	7.18
Floor plan (g)	0.97	32	459.00
Floor plan (h)	0.74	19	3.68
Floor plan (i)	0.86	22	4.14
Floor plan (j)	0.82	22	4.45
Floor plan (k)	0.74	27	2.62
Floor plan (l)	0.69	13	5.66
Floor plan (m)	0.38	34	13.12
Floor plan (n)	0.86	36	2.33
Floor plan (o)	1.00	13	2.09
Overall	0.84	408	548

The high accuracy is achieved by fusing two kinds of characteristics of room types. The first is the spatial distribution characteristics and topological relationship among different room types, which are represented by grammar rules. We use only the grammars (geometry probability is set to 1) to calculate the probability of assigning rooms to a certain type, achieving an accuracy at around 0.3. The second is the distinguishable frequency and geometries properties (e.g., area, width, and length) of different room types. We use only the Bayesian inference method to estimate the room types of 408 testing rooms based on their geometric properties, achieving an accuracy of 0.38. Meanwhile, we use the random forest algorithm to train a model based on the geometric properties of 2300 rooms. Then, we calculate the probability of assigning each room in the test set (408 rooms) to a certain type given its geometric property, achieving an accuracy of 0.45, which is higher than that of Bayesian inference method. Furthermore, we replace the Bayesian inference method with the random forest method in the proposed solution. The result shows that there is no obvious improvement in the final accuracy.

We must reckon that cases that violate our defined rules still exist. For instance, in floor plan (b), a copy room is located between two office rooms, which is regarded as unreasonable according to our rules. Moreover, during the creation of parse forest, our method first deletes low-ranking candidate types for a room based on the initial probability calculated by the Bayesian inference method. This can greatly speed up the creation of the parse forest but can also rule out the right room type. For instance, in floor plan (k), a kitchen is incorrectly recognized because the estimated initial probability shows this room could not be a kitchen. Floor plan (m) gets a low identification accuracy. This is mainly because labs and offices have similar geometry properties and spatial distribution and topological characteristics. It is difficult to distinguish them.

The time used for calculating the parse forest and predicting the type of rooms based on the parse forest in each floor plan can be seen from the third column of Table 2. For most of the floor plans, it takes about 10 s to build the parse forest and predict the type of room since we have ruled out the low-ranking type for a room based on the geometric probability estimated through Bayesian inference at the beginning of construction of the forest. This avoids an exponential growth of the number of parse trees. However, for floor-plan (g), it takes nearly 8 min. This is because one of the zones contains 23 rooms in total, with 18 connected, which enormously increases the number of possible combinations of room types.

The confusion matrix is shown in Figure 12, where the class labels 1 to 11 denote office, lab, lab support space, seminar, computer room, library, toilet, lounge, storage room, kitchen, and copy, respectively. The accuracy of identifying labs, offices, and lab support spaces is much higher than other types because (1) they are much more common than other types and (2) the defined rules are mainly derived from the guidebooks that focus on exploring the characteristics of these three kinds of rooms and the relationships among them. Moreover, internal doors play a vital role in identifying the type of rooms since only relevant types would be connected through inner doors, such as two offices, a lab and a lab support space, and multiple functional spaces in a toilet. For the ancillary spaces (i.e., lounge, storage room, kitchen, and copy room), the frequency of their occurrence is low, and their dimensional and topological characteristics are inapparent. Thus, the accuracy of identifying these ancillary spaces is much lower than that of other room types.

Figure 13 shows the constructed parse tree with the highest probability on floor-plan (d). *r*, *d*, *S*, *O*, *T*, *Sem*, *Anc*, *L*, and *LG* denote the objects of *room*, *door*, *Support*, *Office*, *Toilet*, *Lecture*, *Ancillary*, *Lab*, and *LGroup* in the rules, respectively. The text in the parentheses denotes the specific type of the object. The final parse forest consists of multiple parse trees, from which we can calculate the probability of assigning each room to a certain type. With the parse forest, we can not only infer the type of rooms, but also the type of zones and building units, as well as understand the whole scene since each parse tree represents a full semantic interpretation of the building. For instance, if we choose the parse tree with the highest probability as the estimated semantic interpretation of the scene, we can describe the scene as follows: Floor plan (d) has one lab-centered building unit, which consists of four enclosed areas or zones with one area mainly for offices, one area mainly for labs, and two areas mainly for lab support spaces located at the center of the building unit. We also infer the type (lab-centered, office-centered, and academic-centered) of building units based on the parse tree with the highest probability in other test floor-plans. Finally, 21 among 23 building units are correctly recognized.

Output Class	1	149 36.5%	5 1.2%	1 0.2%	6 1.5%	0 0.0%	0 0.0%	4 1.0%	0 0.0%	1 0.2%	3 0.7%	1 0.2%	87.6%
	2	6 1.5%	64 15.7%	0 0.0%	3 0.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.2%	86.5%
	3	3 0.7%	0 0.0%	62 15.2%	1 0.2%	0 0.0%	2 0.5%	7 1.7%	2 0.5%	0 0.0%	0 0.0%	0 0.0%	80.5%
	4	0 0.0%	0 0.0%	0 0.0%	26 6.4%	1 0.2%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.9%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN%
	6	0 0.0%	0 0.0%	0 0.0%	4 1.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20.0%
	7	2 0.5%	0 0.0%	4 1.0%	0 0.0%	0 0.0%	0 0.0%	39 9.6%	1 0.2%	2 0.5%	5 1.2%	0 0.0%	73.6%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN%
	9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN%
	10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN%
	11	0 0.0%	0 0.0%	0 0.0%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0%
			93.1% 6.9%	92.8% 7.2%	92.5% 7.5%	63.4% 36.6%	0.0% 100%	25.0% 75.0%	78.0% 22.0%	0.0% 100%	0.0% 100%	0.0% 100%	0.0% 100%
	Target Class	1	2	3	4	5	6	7	8	9	10	11	

Figure 12. Confusion matrix of classification result.

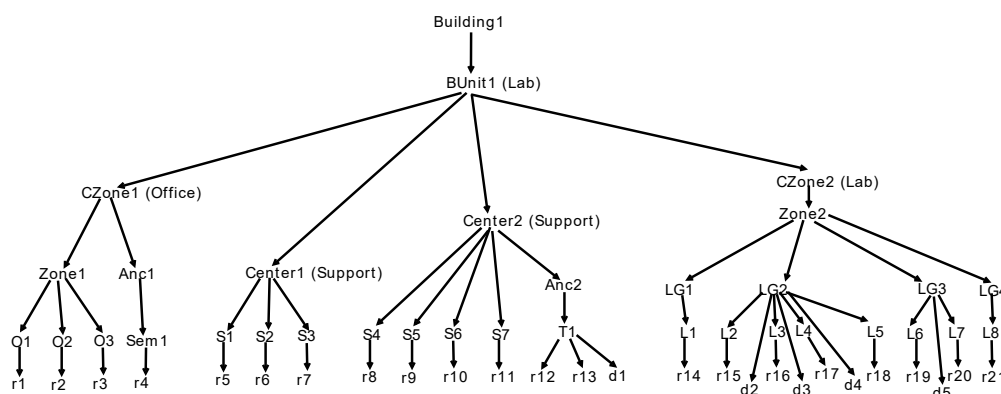


Figure 13. Parse tree with highest probability for floor-plan (d).

6. Discussions

Grammar learning: In this work, grammar rules are defined manually based on guidebooks about research buildings and our prior knowledge. This would produce two problems. One is that manual definition is a time-consuming task and requires a high level of expert knowledge. The other is that the deficiency of some significant rules and the constraints represented in the defined rules both lead to the reduction of the applicability and the accuracy of the proposed method since there exist always the cases that violate our defined rules and constraints. To overcome the two drawbacks, we plan to use grammatical inference techniques [21,22] to automatically learn a probabilistic grammar based on abundant training data in the future. Assigning each rule with a probability can better approximate the ground true since the frequency of occurrence of different rules in the real world varies.

For instance, multiple CZone objects can be merged into a lab-centered building or an academic-centered building. In this work, we assume that the probability of producing a lab-centered building and an academic-centered building is equal. However, the former appears much more frequently than the latter in the real world. Thus, the former should have earned a higher probability. We may argue that learning a reliable grammar for a certain building type is meaningful considering its great advantage in representation, which can benefit many application domains, such as reconstruction, semantic inference, computer-aided building design, and understanding a map by computers.

Deep learning: We may argue that the current advanced technology of deep learning can work for semantic labeling (specifically, room type) as in [61] if abundant images of each type of rooms are collected. However, these deep learning models are restricted in their capacity to reason, for example, to explain why the room should be an office or to further understand the map. Conversely, although grammar-based methods require users' intervention to create rules, they have the advantages of interpreting and representing. Therefore, they have a wide range of applications in GIS and building sectors. First, the grammars we create can not only be used to infer the semantics of rooms but also explain why a room is an office instead of a toilet. Second, the grammars can be used to formally represent a map and help computers to read or understand the map. Last but not least, grammars can benefit computer-aided building design [62].

7. Conclusions

This work investigates the feasibility of using grammars to infer the room type based on geometric maps. We take research buildings as example and create a set of grammar rules to represent the layout of research buildings. Then, we choose 15 floorplans and test the proposed approach. Results show it achieves an accuracy of 84% for 408 rooms. Although the grammar rules we create cannot cover all the research buildings in the world, we still believe the finding of this work is meaningful. It, to a certain extent, proves that grammar can benefit indoor mapping approaches in semantic enrichment. Furthermore, based on the constructed parse trees, we can not only infer the semantics of rooms, but also the type of zones and building units, as well as describe the whole scene.

Several tasks are scheduled for future works. First, we plan to mine useful knowledge from a university's website to enhance the identification of room types, such as the number of offices, the number of people in an office, and the number of conference rooms. This is because the information about researchers' offices and academic reports are accessible to everyone through a university's website. Based on the information, we can further prune parse forests to improve the identification accuracy. Second, a fully automatic solution will be proposed to learn the grammar rules from training data, based on which we can automatically build a more accurate and semantically richer map in a faster way with the help of fewer sensor measurements than the conventional measurement-based reconstruction approaches.

Author Contributions: Conceptualization, H.F.; data curation, X.H.; formal analysis, A.N. and Z.W.; funding acquisition, J.S.; methodology, X.H.; project administration, H.F.; resources, A.Z.; supervision, H.F. and A.Z.; validation, A.N.; writing—original draft, X.H.; writing—review and editing, A.N., Z.W., and J.S.

Funding: This study is supported by the National Key Research and Development Program of China (No.2016YFB0502200), the National Natural Science Foundation of China (Grant No. 41271440), and the China Scholarship Council.

Acknowledgments: We acknowledge financial support by Deutsche Forschungsgemeinschaft within the funding programme Open Access Publishing, by the Baden-Württemberg Ministry of Science, Research and the Arts and by Ruprecht-Karls-Universität Heidelberg.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

$$\text{Copy } c | \text{Storage } s | \text{Kitchen } k | \text{Lounge } l | \text{Lecture } lec | \text{Computer } c | \text{Support } sup \rightarrow \text{room } r \quad (\text{A1})$$

$$\text{Toilet } t \rightarrow \text{set}(\text{room}, k^r) r, \text{set}(\text{door}, k^d) d \left\langle 1 \leq k^r \leq 3; k^d \geq 0; \text{conByIntDoor}(r_i, r_{i+1}, d)_{0 < i < k^r}; \sum_{i=1}^{k^r} \text{withExtDoor}(r_i) == 1 \right\rangle \quad (\text{A2})$$

$$\text{Ancillary } a \rightarrow \text{Toilet } t | \text{Copy } c | \text{Storage } s | \text{Kitchen } k | \text{Lounge } l | \text{Computer } c | \text{Lecture } lec | \text{Library } lib \quad (\text{A3})$$

$$\text{Library } l \rightarrow \text{set}(\text{room}, k^r) r, \text{set}(\text{door}, k^d) d \left\langle k^r \geq 1; k^d \geq 0; \text{conByIntDoor}(r_i, r_{i+1}, d)_{0 < i < k^r} \right\rangle \quad (\text{A4})$$

$$\text{Zone}^{aca} z \rightarrow \text{set}(\text{Lecture}, k^l) l \text{set}(\text{door}, k^d) d \left\langle k^l \geq 1; k^d \geq 0; \text{withExtDoor}(l_i)_{1 \leq i \leq k^l}; \text{edgeAdj}(l_i, l_{i+1})_{0 < i < k^l} | \text{conByIntDoor}(l_i, l_{i+1}, d)_{0 < i < k^l} \right\rangle \{z.type = academic\} \quad (\text{A5})$$

$$\text{Zone}^{aca} z \rightarrow \text{Library } lib \{z.type = academic\} \quad (\text{A6})$$

$$\text{Lab } l \rightarrow \text{room } r^l, (\text{room } r^w, \text{door } d) | \varphi \langle \text{inclusionAdj}(r^l, r^w, d); \text{onExtWall}(r^l) \rangle \quad (\text{A7})$$

$$\text{LGroup } g \rightarrow \text{set}(\text{Lab}, k^l) l, \text{set}(\text{Support}, k^s) s, \text{set}(\text{door}, k^d) d \left\langle k^l \geq 1; k^s \geq 0; \text{conByIntDoor}(l, s, d) \right\rangle \quad (\text{A8})$$

$$\text{Zone}^{lab} z \rightarrow \text{set}(\text{LGroup}, k^l) l \left\langle k^l \geq 1; \text{edgeAdj}(l_i, l_{i+1})_{0 < i < k^l} \right\rangle \{z.type = lab\} \quad (\text{A9})$$

$$\text{Office } o \rightarrow \text{room } r^p, (\text{room } r^s, \text{door } d) | \varphi \langle \text{inclusionAdj}(r^p, r^s, d); \text{onExtWall}(r^p); \text{withExtDoor}(r^p) \rangle \quad (\text{A10})$$

$$\text{Zone}^{office} z \rightarrow \text{set}(\text{Office}, k^o) o, \text{set}(\text{door}, k^d) d \left\langle \text{edgeAdj}(o_i, o_{i+1})_{0 < i < k^o} | \text{conByIntDoor}(o_i, o_{i+1}, d)_{0 < i < k^o} \right\rangle \{z.type = office\} \quad (\text{A11})$$

$$\text{Center } c \rightarrow \text{set}(\text{Ancillary}, k^a) a, \text{set}(\text{Support}, k^s) s$$

$$\left\langle (0 \leq k^a \leq 3; k^s \geq 1) | (k^s == 0; k^a \geq 1); (\text{edgeAdj}(s_i, s_{i+1}) | \text{conByIntDoor}(s_i, s_{i+1}, d))_{0 < i < k^s}; \text{formFullArea}(a, s); \text{inCenter}(c); \right\rangle \quad (\text{A12})$$

$$\{c.type = support | ancillary\}$$

$$CZone\ c \rightarrow set(Ancillary, k)\ a, Zone\ z \langle 0 \leq k \leq 3; formFullArea(a, z) \rangle \{c.type = z.type\} \quad (A13)$$

$$BUnit\ b \rightarrow set(CZone^{office}, k^o)\ z^o, set(Center^{ancillary}, k^c)\ c, set(CZone^{aca}, k^a)\ z^a$$

$$\left\{ \begin{array}{l} k^o \geq 0; 0 \leq k^c \leq 2; 0 \leq k^a \leq 2; k^a + k^o \geq 1; (z^o_i)_{1 \leq i \leq k^o}.type == office; \\ (c_i)_{1 \leq i \leq k^c}.type == ancillary; (z^a_i)_{1 \leq i \leq k^a}.type == academic \end{array} \right\} \{b.type = office|academic\} \quad (A14)$$

$$BUnit^{lab}\ b \rightarrow set(CZone^{lab}, k^l)\ z^l, set(CZone^{office}, k^o)\ z^o, set(CZone^{academic}, k^a)\ z^a, set(Center^{sup}, k^s)\ c^s, set(Center^{anc}, k^{anc})\ c^a$$

$$\left\{ \begin{array}{l} k^l \geq 1; k^o \geq 1; k^a \geq 0; k^{anc} \geq 0; (z^l_i)_{1 \leq i \leq k^l}.type == lab; (z^o_i)_{1 \leq i \leq k^o}.type == office; (z^a_i)_{1 \leq i \leq k^a}.type == academic; \\ (c^s_i)_{1 \leq i \leq k^s}.type == support; (c^a_i)_{1 \leq i \leq k^{anc}}.type == ancillary; isDoubleLoaded(b) | (isTripleLoaded(b); k^s \geq 1) \end{array} \right\} \{b.type = lab\} \quad (A15)$$

$$Building\ f \rightarrow set(BUnit, k^b)\ b \langle k^b \geq 1; edgeAdj(b_i, b_{i+1})_{0 < i < k^b} \rangle \quad (A16)$$

References

1. Zhang, D.; Xia, F.; Yang, Z.; Yao, L.; Zhao, W. Localization technologies for indoor human tracking. In Proceedings of the 2010 5th International Conference on Future Information Technology, Busan, South Korea, 21–23 May 2010; pp. 1–6.
2. Yassin, A.; Nasser, Y.; Awad, M.; Al-Dubai, A.; Liu, R.; Yuen, C.; Raulefs, R.; Aboutanios, E. Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Commun. Surv. Tutor.* **2016**, *19*, 1327–1346. [[CrossRef](#)]
3. Elhamshary, M.; Youssef, M. SemSense: Automatic construction of semantic indoor floorplans. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–11.
4. Youssef, M. Towards truly ubiquitous indoor localization on a worldwide scale. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2015; ACM: New York, NY, USA, 2015.
5. Gao, R.; Zhao, M.; Ye, T.; Ye, F.; Wang, Y.; Bian, K.; Wang, T.; Li, X. Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, Maui, HI, USA, 7–11 September 2014; ACM: New York, NY, USA, 2014; pp. 249–260.
6. Dosch, P.; Tombre, K.; Ah-Soon, C.; Masini, G. A complete system for the analysis of architectural drawings. *Int. J. Doc. Anal. Recognit.* **2000**, *3*, 102–116. [[CrossRef](#)]
7. De las Heras, L.P.; Ahmed, S.; Liwicki, M.; Valveny, E.; S'anchez, G. Statistical segmentation and structural recognition for floor plan interpretation. *Int. J. Doc. Anal. Recognit.* **2014**, *17*, 221–237. [[CrossRef](#)]
8. De las Heras, L.P.; Terrades, O.R.; Robles, S.; S'anchez, G. CVC-FP and SGT: A new database for structural floor plan analysis and its groundtruthing tool. *Int. J. Doc. Anal. Recognit.* **2015**, *18*, 15–30. [[CrossRef](#)]
9. Dodge, S.; Xu, J.; Stenger, B. Parsing floor plan images. In Proceedings of the IEEE IAPR International Conference on Machine Vision Application, Nagoya, Japan, 8–12 May 2017.
10. Xiong, X.; Adan, A.; Akinci, B.; Huber, D. Automatic creation of semantically rich 3D building models from laser scanner data. *Autom. Constr.* **2013**, *31*, 325–337. [[CrossRef](#)]
11. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D semantic parsing of large-scale indoor spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.
12. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
13. Ambruş, R.; Claiaci, S.; Wendt, A. Automatic room segmentation from unstructured 3-d data of indoor environments. *IEEE Robot. Autom. Lett.* **2017**, *2*, 749–756. [[CrossRef](#)]
14. Furukawa, Y.; Curless, B.; Seitz, S.M.; Szeliski, R. Reconstructing building interiors from images. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 80–87.
15. Henry, P.; Krainin, M.; Herbst, E.; Ren, X.; Fox, D. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *Int. J. Robot. Res.* **2012**, *31*, 647–663. [[CrossRef](#)]
16. Ikehata, S.; Yang, H.; Furukawa, Y. Structured indoor modeling. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1323–1331.
17. Alzantot, M.; Youssef, M. Crowdinside: Automatic construction of indoor floorplans. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; ACM: New York, NY, USA, 2012; pp. 99–108.
18. Zhang, J.; Kan, C.; Schwing, A.G.; Urtasun, R. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 1273–1280.
19. Luperto, M.; Riva, A.; Amigoni, F. Semantic classification by reasoning on the whole structure of buildings using statistical relational learning techniques. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2562–2568.
20. Elhamshary, M.; Basalmah, A.; Youssef, M. A fine-grained indoor location-based social network. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1203–1217. [[CrossRef](#)]

21. De la Higuera, C. *Grammatical Inference: Learning Automata and Grammars*; Cambridge University Press: Cambridge, UK, 2010.
22. D'Ulizia, A.; Ferri, F.; Grifoni, P. A survey of grammatical inference methods for natural language learning. *Artif. Intell. Rev.* **2011**, *36*, 1–27. [[CrossRef](#)]
23. Charlotte, K. *New Laboratories: Historical and Critical Perspectives on Contemporary Developments*; Walter de Gruyter GmbH: Berlin, Germany, 2016.
24. Braun, H.; Grömling, D. *Research and Technology Buildings: A Design Manual*; Walter de Gruyter: Berlin, Germany, 2005.
25. Hain, W. *Laboratories: A Briefing and Design Guide*; Taylor & Francis: London, UK, 2003.
26. Watch, D.D. *Building Type Basics for Research Laboratories*; John Wiley & Sons: New York, NY, USA, 2002.
27. Hu, X.; Fan, H.; Zipf, A.; Shang, J.; Gu, F. A conceptual framework for indoor mapping by using grammars. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*; ISPRS Geospatial Week: Wuhan, China, 2017; Volume IV-2/W4, pp. 335–342.
28. Azhar, S. Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. *Leadersh. Manag. Eng.* **2011**, *11*, 241–252. [[CrossRef](#)]
29. Santos, R.; Costa, A.A.; Grilo, A. Bibliometric analysis and review of Building Information Modelling literature published between 2005 and 2015. *Autom. Constr.* **2017**, *80*, 118–136. [[CrossRef](#)]
30. Kolbe, T.H. Representing and exchanging 3D city models with CityGML. In *3D Geo-Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 15–31.
31. Li, K.J.; Kim, T.H.; Ryu, H.G.; Kang, H.K. Comparison of cityGML and indoorGML—A use-case study on indoor spatial information construction at real sites. *Spat. Inf. Res.* **2015**, *23*, 91–101.
32. Kim, J.S.; Yoo, S.J.; Li, K.J. Integrating IndoorGML and CityGML for indoor space. In *International Symposium on Web and Wireless Geographical Information Systems*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 184–196.
33. Kang, H.K.; Li, K.J. A standard indoor spatial data model—OGC IndoorGML and implementation approaches. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 116. [[CrossRef](#)]
34. Macé, S.; Locteau, H.; Valveny, E.; Tabbone, S. A system to detect rooms in architectural floor plan images. In Proceedings of the IEEE IAPR International Workshop on Document Analysis Systems, Boston, MA, USA, 9–11 June 2010.
35. Ahmed, S.; Liwicki, M.; Weber, M.; Dengel, A. Improved automatic analysis of architectural floor plans. In Proceedings of the IEEE International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011.
36. Ahmed, S.; Liwicki, M.; Weber, M.; Dengel, A. Automatic room detection and room labeling from architectural floor plans. In Proceedings of the IEEE IAPR International Workshop on Document Analysis Systems, Gold Coast, QLD, Australia, 27–29 March 2012.
37. Gimenez, L.; Robert, S.; Suard, F.; Zreik, K. Automatic reconstruction of 3D building models from scanned 2D floor plans. *Autom. Constr.* **2016**, *63*, 48–56. [[CrossRef](#)]
38. De las Heras, L.P.; Mas, J.; S'anchez, G.; Valveny, E. Notation-invariant patchbased wall detector in architectural floor plans. In Proceedings of the International Workshop on Graphics Recognition, Seoul, Korea, 15–16 September 2011.
39. Sankar, A.; Seitz, S. Capturing indoor scenes with smartphones. In Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, Cambridge, MA, USA, 7–10 October 2012; ACM: New York, NY, USA, 2012; pp. 403–412.
40. Pintore, G.; Gobbetti, E. Effective mobile mapping of multi-room indoor structures. *Vis. Comput.* **2014**, *30*, 707–716. [[CrossRef](#)]
41. Tsai, G.; Xu, C.; Liu, J.; Kuipers, B. Real-Time Indoor Scene Understanding Using Bayesian Filtering with Motion Cues. In Proceedings of the 2011 International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 121–128.
42. Jiang, Y.; Xiang, Y.; Pan, X.; Li, K.; Lv, Q.; Dick, R.P.; Shang, L.; Hannigan, M. Hallway based automatic indoor floorplan construction using room fingerprints. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013; ACM: New York, NY, USA, 2013; pp. 315–324.

43. Chen, S.; Li, M.; Ren, K.; Qiao, C. Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos. In Proceedings of the IEEE 35th International Conference on Distributed Computing Systems (ICDCS), Columbus, OH, USA, 29 June–2 July 2015; pp. 1–10.
44. Gao, R.; Zhou, B.; Ye, F.; Wang, Y. Knitter: Fast, resilient single-user indoor floor plan construction. In Proceedings of the INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
45. Mura, C.; Mattausch, O.; Villanueva, A.J.; Gobbetti, E.; Pajarola, R. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Comput. Graph.* **2014**, *44*, 20–32. [[CrossRef](#)]
46. Nikoohemat, S.; Peter, M.; Elberink, S.O.; Vosselman, G. Exploiting Indoor Mobile Laser Scanner Trajectories for Semantic Interpretation of Point Clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-2/W4*, 355–362. [[CrossRef](#)]
47. Becker, S.; Peter, M.; Fritsch, D. Grammar-supported 3d Indoor Reconstruction from Point Clouds for “as-built” BIM. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 17. [[CrossRef](#)]
48. Yue, K.; Krishnamurti, R.; Grobler, F. Estimating the interior layout of buildings using a shape grammar to capture building style. *J. Comput. Civ. Eng.* **2011**, *26*, 113–130. [[CrossRef](#)]
49. Philipp, D.; Baier, P.; Dibak, C.; Dürr, F.; Rothermel, K.; Becker, S.; Peter, M.; Fritsch, D. Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data. In Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications, Budapest, Hungary, 24–28 March 2014; pp. 139–147.
50. Rosser, J.F. Data-driven estimation of building interior plans. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1652–1674. [[CrossRef](#)]
51. Luperto, M.; Amigoni, F. Exploiting structural properties of buildings towards general semantic mapping systems. In *Intelligent Autonomous Systems 13*; Springer: Cham, Switzerland, 2016; pp. 375–387.
52. Luperto, M.; Li, A.Q.; Amigoni, F. A system for building semantic maps of indoor environments exploiting the concept of building typology. In *Robot Soccer World Cup*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 504–515.
53. Khoshelham, K.; Díaz-Vilariño, L. 3D modelling of interior spaces: Learning the language of indoor architecture. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *40*, 321. [[CrossRef](#)]
54. Mitchell, W.J. *The Logic of Architecture: Design, Computation, and Cognition*; MIT Press: Cambridge, MA, US, 1990.
55. Dehbi, Y.; Hadiji, F.; Gröger, G.; Kersting, K.; Plümer, L. Statistical relational learning of grammar rules for 3D building reconstruction. *Trans. GIS* **2017**, *21*, 134–150. [[CrossRef](#)]
56. Liu, Z.; von Wichert, G. Extracting semantic indoor maps from occupancy grids. *Robot. Auton. Syst.* **2014**, *62*, 663–674. [[CrossRef](#)]
57. Liu, Z.; von Wichert, G. A generalizable knowledge framework for semantic indoor mapping based on Markov logic networks and data driven MCMC. *Future Gener. Comput. Syst.* **2014**, *36*, 42–56. [[CrossRef](#)]
58. Deransart, P.; Jourdan, M.; Lorho, B. *Attribute Grammars: Definitions, Systems and Bibliography*; Springer Science & Business Media: Heidelberg, Germany, 1988; Volume 323.
59. Deransart, P.; Jourdan, M. Attribute grammars and their applications. In *Lecture Notes in Computer Science*; Springer: Heidelberg, Germany, 1990; Volume 461.
60. Boulch, A.; Houllier, S.; Marlet, R.; Tournaire, O. Semantizing complex 3D scenes using constrained attribute grammars. *Comput. Graph. Forum* **2013**, *32*, 33–42. [[CrossRef](#)]
61. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
62. Müller, P.; Wonka, P.; Haegler, S.; Ulmer, A.; Van Gool, L. Procedural modeling of buildings. *ACM Trans. Graph.* **2006**, *25*, 614–623. [[CrossRef](#)]

