# INAUGURAL-DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich - Mathematischen

Gesamtfakultät

der

Ruprecht–Karls–Universität

Heidelberg

vorgelegt von

Master of Engineering Xuke Hu

aus: Suizhou, China

Tag der mündlichen Prüfung:

Thema

# Building Semantics Reasoning by Using Rules based on Available Geospatial Information

Gutachter: Prof. Dr. Alexander Zipf

Assoc. Prof. Dr. Wenwen Li

# Acknowledgements

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

First, I would like to say a very big thank you to my first supervisor Prof. Dr. Alexander Zipf. I appreciate his acceptance of me being a member of the GIScience Research Group. His wisdom and openness lead to a research group that is passionate, creative and inclusive, and being a member of this group makes me grow up fast.

Second, I want to express my gratitude to my second supervisor Prof. Dr. Hongchao Fan for giving me many insightful comments and supports throughout my PhD no matter at work or in life. Without his constant guidance, feedback, and concern, this PhD would not have been achievable.

Third, sincere thanks go to Bettina Knorr. You are so professional, kind, patient and considerate. Thank you so much for all your assistance and supports in the past five years!

I also want to thank my colleagues in particular, Dr. Yeran Sun, Dr. Ming Li, Dr. Alexey Noskov, Dr. Clemens Jacobs, Sebastian Döring, Dr. Zhiyong Wang, Dr. Wei Huang, Dr. Yingwei Yan, Dr. Jiaoyan Cheng, Dr. Lu Liu, Hao Li, Zhaoyan Wu, Zhendong Yuan, Dr. Tessio Novack, Dr. Michael Schultz, and all the other colleagues, for all the inspirations, encouragements and helps. It is such a pleasure to meet you and work with you in this wonderful group!

My thanks also go out to the support I received from the collaborative work I undertook with Prof. Dr. Jianga Shang and Dr. Fuqiang Gu. Without your support, many of of my research works could not be finished smoothly.

I also want to express gratitude to the reviewers for my papers and dissertation, who helped to significantly enhance my research.

Furthermore, I would like to gratefully acknowledge China Scholarship Council (CSC) for awarding scholarship and giving me an opportunity to pursue my doctoral degree.

Finally, I want to thank my parents and my brother for their endless support and infinite love. In particular, I want to thank Yu from all my heart. Happy or sad, you are always there for me. You give me the courage of making changes. It is dedicated to you!

# Abstract

Sensing and understanding the indoor and outdoor urban space that humans live in is one of the core tasks of GIScience researchers and practitioners. Buildings, as the most important living spaces of humans, attract much attention. Acquisition of accurate, detailed, up-to-date, and full coverage building information is the prerequisite of a variety of key applications, such as city planning, smart city, and location-based services (LBSs). The mainstream indoor and outdoor building reconstruction solutions can be coarsely categorized into two groups: equipment-dependent sensing and volunteered geographic information (VGI). The former detects the building elements by associating the sensor measurements with target building elements. The biggest challenge is that full coverage and refined sensor measurements (e.g., image or LiDAR point cloud) on a large-scale are unavailable. Furthermore, it fails to detect some unobservable building elements, such as the salience of buildings perceived by humans and the main entrance that is normally invisible from the streets and the air. VGI, which leverages the wisdom of humans, is gaining more and more attention. OpenStreetMap (OSM), as one of the most successful VGI projects, provides high-quality and freely editable and accessible geospatial information worldwide. However, due to the characteristics of crowd-sourcing, the missing of building elements (e.g., building type) on OSM occurs quite frequently.

To overcome the challenges faced in the two solutions, this dissertation investigated the potential of inferring distinct indoor and outdoor spatial (specifically building) elements based on existing or available spatial elements on OSM or provided by sensing equipment, leveraging the association relationship between the spatial elements. Furthermore, this dissertation compared and explored the applicability of two kinds of reasoning mechanisms using manually defined explicit rules and learned implicit rules in this context, respectively. Four representative indoor and outdoor building elements (i.e., roof shape, room usage, main entrance, and landmark salience) were taken as examples to explore how and why the four building elements can be inferred by explicit and/or implicit rules. Finally, the results of the four studies were combined to answer the questions related to the research objectives.

The first study investigated how the explicit rules involving in combination and symmetric characteristics of the footprint can be used to reason the possible roof shape combination of complex buildings. Due to the small number of associated elements and numerical elements, manually defining several explicit rules is proper. The results showed that the pure explicit rules perform well in ruling out most of the incorrect options of roof shapes. In the second study, implicit rules derived from imbalanced learning approaches were adopted to predict the location of the main entrance of public buildings based on the footprint and spatial context (e.g., roads) of the building, which are available on OSM. The experimental results revealed that the location of the entrance highly correlates the footprint and spatial context of the building and the derived implicit rules can effectively predict the location of the main entrance. Explicit rules cannot solve the problem faced in entrance prediction since the number of associated elements and numerical elements is large and there exist also internal association relationships. The third study compared the fusion of explicit and implicit rules and pure implicit rules in predicting the room usage of research buildings based on the geometric map. The results reveal that both solutions achieve an acceptable tagging accuracy but the former one is stronger than the later in rule re-usability while the latter is stronger than the former in robustness. It also proves that fusing explicit and implicit rules in room usage tagging is a good option. The fourth study used implicit rules derived from Genetic Programming to model the quantitative relationship between the salience and the visual and semantic attributes of landmarks in shopping malls, which is compared with the model defined by explicit rules. The results show the implicit rules are stronger than the explicit rules in salience prediction accuracy but weaker in interpret-ability.

The results of the conducted studies in this dissertation highlighted the potential of the association of spatial elements in inferring the target spatial (building) elements given existing spatial information on OSM. The results also revealed that both explicit rules defined by experts and implicit rules derived by statistical learning are useful in reasoning the building elements, but they showed distinct performance in prediction accuracy, robustness, rule re-usability, and interpret-ability. Their applicability varies as the complexity of the association relationship (e.g., number of associated elements, numerical elements, and internal association relationships) and the availability of expert knowledge and tagged data. The fusion of both explicit and implicit rules shows great potential.

# Kurzfassung

Das Wahrnehmen und Verstehen des städtischen Innen- und Außenraumes, in dem Menschen leben, ist eine der Kernaufgaben von Forschern und Anwendern in der Geoinformatik. Gebäude als wichtigste Lebensräume des Menschen ziehen viel Aufmerksamkeit auf sich. Die Erfassung exakter, detaillierter, aktueller und vollständiger Gebäudeinformationen ist die Voraussetzung für eine Vielzahl von Schlüsselanwendungen wie Stadtplanung, Smart City und standortbasierte Dienste (LBSs). Die gängigen Lösungen für die Rekonstruktion von Innen- und Außengebäuden lassen sich grob in zwei Gruppen einteilen: geräteabhängige Erfassung und freiwillige geografische Informationen (VGI). Ersteres erkennt die Bauelemente, indem es die Sensormessungen mit Zielbauelementen verknüpft. Die größte Herausforderung besteht dabei darin, dass eine vollständige Abdeckung und präzise Sensormessungen (z. B. Bild oder LIDAR-Punktwolken) nicht in großem Maßstab verfügbar sind. Darüber hinaus werden einige nicht beobachtbare Gebäudeeigenschaften nicht erkannt, wie z. B. Salienz von Gebäuden, die von Menschen wahrgenommen werden, sowie der Haupteingang, der normalerweise von der Straße und der Luft nicht sichtbar ist. VGI, welches das Wissen von Menschen nutzt, gewinnt immer mehr an Aufmerksamkeit. OpenStreetMap (OSM) bietet als eines der erfolgreichsten VGI-Projekte weltweit hochwertige, frei bearbeitbare und zugängliche Geoinformationen. Aufgrund der Eigenschaften von Crowd-Sourcing tritt das Fehlen von Gebäudeelementen (z. B. Gebäudetyp) in OSM jedoch ziemlich häufig auf.

Um die Herausforderungen zu bewältigen, denen die beiden Lösungsansätze gegenüberstehen, untersuchte diese Dissertation das Potenzial, unterschiedliche räumliche (insbesondere Gebäude-) Elemente im Innen- und Außenbereich auf der Grundlage vorhandener oder verfügbarer räumlicher Elemente in OSM oder durch Sensorik bereitzustellen, wobei die Assoziierung zwischen den räumlichen Elementen genutzt wurde. Darüber hinaus verglich und untersuchte diese Dissertation die Anwendbarkeit von zwei Arten von Argumentationsmechanismen unter Verwendung manuell definierter expliziter Regeln und erlernter impliziter Regeln in diesem Zusammenhang. Vier repräsentative Innen- und Außenbauelemente (Dachform, Gebäudenutzung, Haupteingang und Salienz von Landmarken) wurden als Beispiele herangezogen, um zu untersuchen, wie und warum die vier Bauelemente durch explizite und / oder implizite Regeln abgeleitet werden können. Schließlich wurden die Ergebnisse der vier Studien kombiniert, um die Fragen im Zusammenhang mit den Forschungszielen zu beantworten.

In der ersten Studie wurde untersucht, wie die expliziten Regeln für die Kombination und die symmetrischen Eigenschaften des Grundrisses verwendet werden können, um die mögliche Dachformkombination komplexer Gebäude zu begründen. Aufgrund der geringen Anzahl zugeordneter Elemente und numerischer Elemente ist es richtig, mehrere explizite Regeln manuell zu definieren. Die Ergebnisse zeigten, dass mit den expliziten Regeln die meisten falschen Möglichkeiten für Dachformen ausgeschlossen werden können. In der zweiten Studie wurden implizite Regeln angewendet, die aus unbalancierten Lernansätzen abgeleitet wurden, um den Ort des Haupteingangs von öffentlichen Gebäuden basierend auf dem Fußabdruck und dem räumlichen Kontext des Gebäudes (z. B. Straßen), welche auf OSM verfügbar sind, vorherzusagen. Die experimentellen Ergebnisse zeigten, dass die Position des Eingangs in hohem Maße mit der Grundfläche und dem räumlichen Kontext des Gebäudes korreliert und die abgeleiteten impliziten Regeln die Position des Haupteingangs effektiv vorhersagen können. Explizite Regeln können das Problem der Eingangsvorhersage nicht lösen, da die Anzahl der zugeordneten Elemente und numerischen Elemente groß ist und auch interne Assoziierungen bestehen. In der dritten Studie wurde die Verschmelzung von expliziten und impliziten Regeln und rein impliziten Regeln bei der Vorhersage der Raumnutzung von Forschungsgebäuden anhand der geometrischen Karte verglichen. Die Ergebnisse zeigten, dass beide Lösungen eine akzeptable Markierungsgenauigkeit erreichen, die erstere jedoch in der Regel wiederverwendbarer ist als die zweite, während die letztere in ihrer Robustheit stärker ist als die erstere. Es zeigt auch, dass das Zusammenführen expliziter und impliziter Regeln bei der Kennzeichnung der Raumnutzung eine gute Option ist. Die vierte Studie verwendete implizite Regeln, die durch genetischen Programmierung abgeleitet wurden, um die quantitative Beziehung zwischen der Salienz und den visuellen und semantischen Attributen von Orientierungspunkten in Einkaufszentren zu modellieren, die mit dem durch explizite Regeln definierten Modell verglichen wird. Die Ergebnisse zeigen, dass die impliziten Regeln in Bezug auf die Genauigkeit der Salienzvorhersage stärker als die expliziten Regeln sind, jedoch in Bezug auf die Interpretationsfähigkeit schwächer.

Die Ergebnisse der in dieser Dissertation durchgeführten Studien haben das Potenzial der Assoziierung räumlicher Elemente zu OSM bei der Bestimmung der räumlichen Zielelemente (Gebäudeelemente) unter Berücksichtigung vorhandener räumlicher Informationen hervorgehoben. Die Ergebnisse zeigten auch, dass sowohl explizite Regeln, die von Experten definiert wurden, als auch implizite Regeln, die durch statistisches Lernen abgeleitet wurden, nützlich sind, um die Bauelemente zu bestimmen und sie zeigten eine sehr gute Leistung in Bezug auf Vorhersagegenauigkeit, Robustheit, Wiederverwendbarkeit von Regeln und Interpretierbarkeit. Ihre Anwendbarkeit variiert je nach Komplexität der Assoziierungsbeziehung (z. B. Anzahl der zugeordneten Elemente, numerischen Elemente und internen Assoziationsbeziehungen) und der Verfügbarkeit von Expertenwissen und markierten Daten. Die Verschmelzung von expliziten und impliziten Regeln zeigt großes Potenzial.

# Contents

# List of Figures

# List of Tables

# Part I

# Synopsis

# 1. Introduction

Sensing and understanding the urban space that we live in is one of the core tasks of GIScience researchers and practitioners. Buildings, as the most important living spaces of humans, attract much more attention. Acquisition of accurate, detailed, up-to-date, and full coverage (i.e., indoor and outdoor) building information is the prerequisite of a variety of key applications (Haala, Kada 2010; Rottensteiner et al. 2012; Volk et al. 2018; Vosselman et al. 2001), including smart city, city planning, building energy modeling, telecommunication planning, noise simulation, and real-time use such as location-based services (LBSs), gaming, and virtual reality. For instance, through modeling the influence of the roof shape and building height on the wind flow and dispersion of gaseous pollutants from vehicle exhaust within urban canyons, the gas pollution issue in a city can be monitored (Allegrini 2018; Huang et al. 2009). Besides, the detailed 3D information on the building stock will ensure more accurate urban energy analysis that is required for smart city services (Bahu et al. 2014). Specifically, through the geometric calculation based on the 3D building models and other building physical characteristics (e.g., volume, material, height, building type), dynamic heating and cooling demands for buildings can be estimated accurately (Hosseini et al. 2018; Yassin 2011). Furthermore, given the salience of indoor/outdoor landmarks of buildings, a more user-friendly landmark-based navigation system can be implemented, which can reduce pedestrians' cognition burdens and anxiety in way-finding in complex environments (Caduff, Timpf 2008; Millonig, Schechtner 2007; Riehle et al. 2008).

Collecting, managing, and analyzing the building information is necessary for Geography Information System (GIS) no matter how it evolved since its birth (Dore, Murphy 2012; Liu et al. 2017; Maliene et al. 2011). More specifically, from the earliest CGIS (Canadian Geographical Information System), to ESRI (Environmental Systems Research Institute), the pioneer and current leader of the GIS market, to the first open-source GIS, GRASS (Geographic Resources Analysis Support System), till the today' commercial cartography services, such as Google Maps, Bing Maps, and Yahoo Maps, which allow users with little or no technical GIS knowledge to interact with a GIS application and use it through Internet-based services, digitizing the building from outside to inside is the focus as well as one of the most challenging tasks of GIS.

Apart from GIS, the other common building models that represent the details of buildings are CityGML (Gröger, Plümer 2012) and building information model (BIM) (Azhar 2011). CityGML represents not only the graphical appearance and topological aspects of city models, but also the semantic properties, taxonomies, and the aggregation of different features (e.g.

facade and roofs), which are structured in five consecutive levels of detail (LoD). LoD0 defines a coarse regional model and the most detailed LoD4 comprises building interiors, such as rooms, doors, and windows. BIM is capable of restoring both geometric and rich semantic information of building components as well as their relationships. It also enables multi-schema representations of 3D geometry for indoor entities.

Briefly, regardless of today's mature GIS platforms and commercial map providers as well as the widely used building models, collecting complete, accurate, and detailed building information is always significant. Currently, a couple of problems remain regarding this task. First, these platforms and models are incapable of providing geospatial data in many developing countries, such as in Africa (Ayanlade et al. 2010) where a free and widely available geospatial database is highly required. Second, a lot of key building elements are incomplete or ignored, such as the building type, building level, roof shape, entrance, indoor layout, and landmark salience. Therefore, in recent years, many efforts have been taken by the researchers or stakeholders to reconstruct indoor and outdoor, 2D and 3D building elements worldwide in an automatic way (Vosselman et al. 2001) through sensing equipment such as earth observation satellites and LiDAR or by leveraging volunteered geographic information (VGI) (Sui et al. 2012) that utilizes the wisdom of the crowd. Unfortunately, due to the technical limitation of sensing equipment and data quality issue of VGI, some key building elements cannot be automatically detected or are still missing on VGI, but many other associated spatial elements are available on VGI or can be obtained by sensing equipment.

This dissertation, therefore, attempts to uncover the potential of association between spatial elements in inferring indoor and outdoor spatial (specifically building) elements based on available spatial information on VGI platforms or provided by sensing equipment and to compare and investigate the applicability of two kinds of reasoning mechanisms that apply explicit and implicit rules, respectively. The following sections will outline the overall research motivation, research questions, as well as the structure of this dissertation.

## 1.1 Motivation and research context

### 1.1.1 Technological challenge of equipment-based sensing and VGI

Currently, there are two mainstream solutions for reconstructing the outside and inside building elements. The first is traditional manners that leverage sensing equipment, such as the earth observation satellite and LiDAR, which have been intensively investigated by researchers since the early eighties. In 1982, SPOT Image (Day, Muller 1989), the first commercial company to distribute satellite images that cover the entire globe, was created, but only very coarse information can be extracted from the early time satellite images. With the development of new techniques such as LiDAR, which can be used to get elevation data with much more details, new possibilities for large areas such as terrain analysis and 3D model reconstruction have been opened. However, as Habib et al. (2010) point out: *"digital building model generation of complex structures remains to be a challenging issue"*. This can be explained in two aspects. First, fully automatic image understanding is very hard to solve and semi-automatic components are usually required to support the recognition of very complex buildings by a human operator. Besides, LiDAR is limited by its short coverage such that it is not proper for a large-scale building reconstruction. Second, certain physical and semantic elements of buildings cannot be recognized from imagery and LiDAR point clouds, such as the salience of a building perceived by humans and the hidden building elements (e.g., entrance) that cannot be directly observed from the air or streets.

**Table 1.1.** Count of buildings whose roof type, entrance, building type, and building level have been tagged on OSM accordingly.

| City | Total buildings | Entrance | Building type | Building level | Roof shape |
|---|---|---|---|---|---|
| Frankfurt | 40040 | 1228 | 12988 | 1181 | 463 |
| Mannheim | 28853 | 1418 | 5178 | 3538 | 2872 |
| Heidelberg | 13580 | 499 | 2124 | 747 | 122 |
| Karlsruhe | 16695 | 1685 | 4352 | 3268 | 2013 |

The second solution is based on VGI, which was raised by Goodchild in 2007, using the wisdom of the crowd, where the author pictured a network of human sensors with 6 billion components, each an intelligent synthesizer and interpreter of local information. That is, VGI provides us with a new means to sense and understand the urban space. One of the most successful cases of VGI is OpenStreetMap (OSM) (Haklay, Weber 2008), which was founded in 2004, initially focusing on mapping the United Kingdom. Later, it became a worldwide collaborative map freely edited and accessed for anybody. Now, OSM can provide comparable quality (e.g., high coverage and accuracy) geospatial data with its commercial counterparts (e.g., Google Maps) in many regions (e.g., Europe and the US) (Ciepłuch et al. 2010; Hochmair et al. 2013). Every year, millions of OSM volunteers contribute geospatial data worldwide. According to the statistics (the values are derived from our internal OSM database which is updated daily), on November 20, 2019, the number of buildings in OSM was over 80 million. In Germany, there are almost 9 million objects with the key as "building" to the same time point. Furthermore, OSM provides richer building elements than its commercial counterparts. Examples include the material, roof shape, entrance, and indoor structure, which are not widely considered by commercial map providers. However, due to the editable-free characteristics of OSM, volunteers have the total freedom to add certain building elements. This leads to the frequent occurrence of the missing of certain building elements on OSM, such as the entrance of buildings, the roof type, and the internal layout of public buildings. Table 1.1 lists the number of buildings on OSM in six German cities and the number of buildings tagged with building type, building level, roof shape, and entrance, respectively. We can see only a small proportion (below 20%) of buildings have been tagged with these building elements.

### 1.1.2 Association between spatial elements

To overcome the challenges of OSM and equipment-based sensing solutions, this dissertation attempted to infer the missing or target building elements (i.e., roof type, entrance, room usage, and landmark salience) based on available spatial elements on OSM or provided by sensing equipment, leveraging the association between spatial elements. The precondition of this is that there exists strong association between the spatial elements, which is discussed in the following paragraphs.

Buildings are man-made structures that are constructed with plans made by people. The architects would normally follow certain specifications or rules to make sure that the constructed buildings function well. Hence, the buildings have significant socio-economic relevance (Fan et al. 2014). This can be proved by the fact that many guidebooks or design code (Braun, Grömling 2005; Hain 2003; Klonk 2016; Watch 2002) about the buildings (e.g., hospitals, airports, office buildings, and laboratories) have been proposed by the government or the experts in the architecture domain. In the guidebooks or design codes, many general principles or constraints have been proposed, which the architects should pay attention to when designing the building. Some principles and constraints reflect how one spatial element affects or associates the other. For instance, the geometric attributes (e.g., area and length) of a certain type of room (e.g., laboratory) are limited to certain values and the laboratory is normally located at external walls to receive natural lit. That is, the usage of the room correlates with its geometry and spatial context (walls) (Hain 2003; Watch 2002).Kruger, Seville (2012) declared that roofs should be designed to minimize the collection of rainfall, snow, and leaves without creating leaks and narrow spaces, which limits the possible roof shape combination of two adjacent rectangles of the footprint. That is, the roof shape of a building correlates with its footprint. According to the building design specification [1] proposed by OFFICE FOR INFRASTRUCTURE AND LOGISTICS of EUROPEAN COMMISSION, the main entrance of a building should be easily accessed and observed from the main roads. This reveals that the roads affect the location of the main entrance.

The existence of the association relationship also explains why a certain type of buildings

[1] https://ec.europa.eu/oib/pdf/mit-standard-building-specs_en.pdf

worldwide look like the same. Antonio Castro Neto, from the Graphene Research Center of National University of Singapore stated that *"Research labs have been globlised or uniformed in the same way that shopping centers look exactly alike no matter where you go."* (Klonk 2016). Steven L. Bernasek from Depertment of Chemistry, Princeton University also claimed that *"In my experience, there appears to be a universal design for the structure and organization of laboratories. At least in chemistry, to have laboratory benches, hood space and instrumentation space seems to be norm."* (Klonk 2016).

A couple of previous works have utilized the association of spatial elements to reconstruct the buildings elements. For example, Fan et al. (2014) utilized the association between the building type and the shape and size of footprints to estimate the building type using urban morphology analysis. Kang et al. (2010) proposed detecting the entrance from an image by considering the association between the entrance and the windows and walls. Yue et al. (2012) proposed to predict the indoor layout of residential houses in the style of Queen Anne House with the help of a few observations, such as footprints and the location of windows. In these studies, promising prediction results have been achieved by using the association of spatial elements. This also proves the existence of association in spatial elements and unveils the potentials of applying association in building elements reasoning. However, these studies do not answer the questions about which reasoning mechanisms (rule system and statistical learning) should be adopted in different contexts and why. Moreover, these studies normally combine the sensor measurements and the association in reasoning; therefore, it is unclear to what degree the association among spatial elements contributes to the final result.

### 1.1.3 Explicit and implicit rules

As discussed before, there exist association relationships between spatial elements. Given the association relationship, rules can be manually defined or learned to infer the target building element. In the context of building element prediction, a rule can be described as how one or multiple known spatial elements affect or determine the prediction result of the target elements. Spatial elements can be divided into two types: numerical (e.g., area and distance) and categorical (e.g., usage of room).

According to the ways of obtaining the rules, they can be categorized into two types: **explicit and implicit**. Explicit rules are readily understood and defined by humans according to their prior knowledge in the target field. That is, it is clear how one or multiple elements together would affect the result of the target element from humans' perspectives. This statement can be described with formal language that can be understood by computers. For instance, the rule, 'A toilet normally does not locate between two offices' can be described as 'If rooms $a$ and $b$ are offices and room $c$ is between $a$ and $b$, then room c is not likely to be a toilet'.

Implicit rules are those that they do exist but cannot be clearly and exactly understood and defined by humans due to the lack of prior knowledge in the target field or the complex association relationships between elements. Therefore, implicit rules are normally learned from tagged training data. The complexity here refers to the huge number of possible association manners (associating the certain spatial elements with specific values to the result of the target spatial element). For instance, the main entrance is physically close to the main road and the centroid of the building footprint, easily observed from roads, located at convex edge of the footprint, or near the bicycle parking areas. However, how to explicitly defining rules to model the complex association relationship is difficult.

In a building reconstruction task, the target spatial element normally associates with multiple spatial elements. Then an association graph is obtained by representing all the association relationships in a unified manner. It depicts which spatial elements are associated but cannot represent how exactly they are associated. An example of the association graph is shown in Figure 1.1. Circles and squares denote numerical and categorical elements, respectively. Dashed shapes denote unknown elements while solid shape denotes known elements. The shape with a shallow yellow background denotes the target element. Solid lines denote the association relationship. The association between known elements is named internal association.

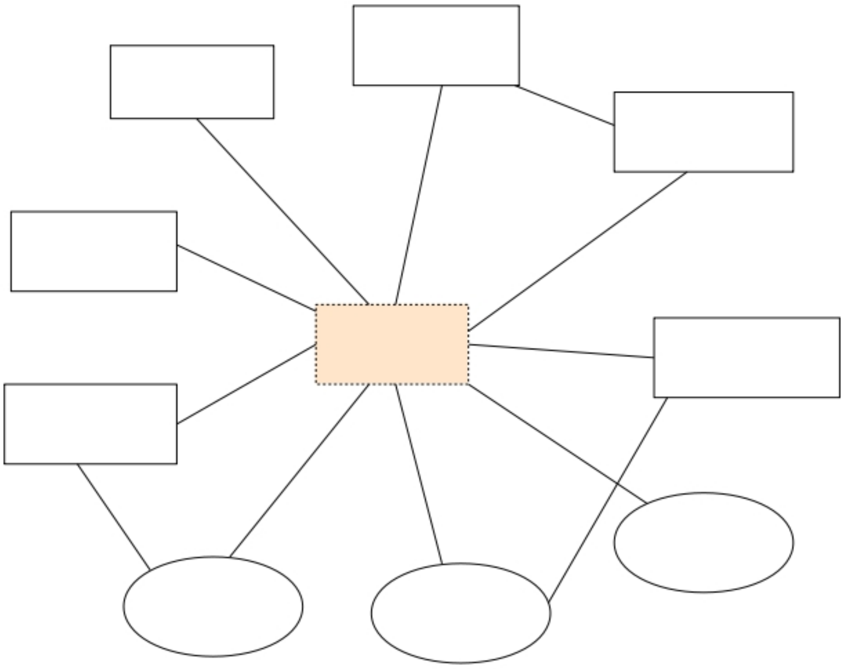


**Figure 1.1.** An example of association graph.

In the graph, the target element is unknown, while the other elements are known. Given an association graph, two common methods can be used to infer the target elements with the known elements in the graph. The first is rule system which uses the explicit rules and the other is statistical learning which utilizes the learned implicit rules. In rule system, experts first manually create rules based on the association graph and prior knowledge to model the association relationship between spatial elements. Then, given the input (known spatial elements), the rule system would make choices consecutively according to the rules and finally output the estimated result of the target element. In statistical learning, features are extracted from the known elements in the association graph in a traditional machine learning manner or automatically extracted from raw data in a deep learning manner without an explicit association graph. Abundant features and tag triples are then gathered to train a model to associate the features with the target element. That is, the trained model can be used to infer the target elements. Apart from the separate application of explicit rules and statistical learning in reasoning spatial elements, the combination of both has also been investigated. In the following sections, the definition of rule system and statistical learning, their applications in building reconstruction as well as the combination of both reasoning techniques will be introduced.

#### 1.1.3.1 Definition of rule system

Rule system is the reasoning method that leverages explicit rules. It is a variant of the expert system, which is a computer system that mimics the decision-making ability of a human expert to solve complex problems by reasoning knowledge. Expert system is the earliest form of artificial intelligence. According to (Tan 2017), the expert system can be divided into four classes: the Rule-Based Expert System, the Frame-Based Expert System, the Fuzzy Logic-Based Expert System and the Expert System Based on Neural Network. This thesis focuses on the first class, the Rule-based Expert System, named rule system for short. Instead of representing knowledge in a declarative, static way as a set of things which are true, rule system represents knowledge in terms of a set of explicit rules that tells what to do or what to conclude in different situations. In this dissertation, the manually created mathematical models by experts, which represent the quantitative relationship between different factors are defined as a special rule system since the execution of the mathematical model can be seen as a reasoning procedure.

A complete rule system contains three parts: explicit rule, database, and control strategy. Any rule consists of two parts: the IF part, called the antecedent (premise or condition) and the THEN part called the consequence (conclusion or action). A rule can have multiple antecedents joined by the keywords AND, OR, or a combination of both. For example, for a rule, its conditions can be ‘if the room area is beyond 200 square meters or if the room is connected to an office through internal doors’, and its consequence could be ’the room is not a toilet’. The database is responsible for storing the conditions and results in the rule statement. When the rule is executed, the corresponding condition is called from the database and the result is put into the database as a condition for other rules. The control strategy is usually executed in a module named ”inference engine”. Its role is to explain how to apply or combine the rules given a certain input. That is to select the appropriate rules in the inference process. The inference process of the inference engine can be divided into forward and reverse.

One important variant of rule system is Grammar (Ginsburg 1966). It is targeted for the objects that comprise multi-hierarchy sub-objects (e.g., language, gene, and building). For instance, for a paragraph, it can be divided into multiple sentences, which can be further divided into multiple words. A grammar consists of a set of rules for rewriting or generating the target objects along with a ”start symbol” from which rewriting starts. That is, a grammar is usually functioning as an object generator. Besides, it can also be used as the basis for a ”recognizer”—a function in computing that determines whether a given object belongs to the target objects or is grammatically incorrect. For instance, to judge if a given indoor layout is correct according to the defined grammar rules. Equation 1.1 denotes the general format of a rule in grammars. $Z$ represents the parental or superior objects that can be split into or replaced by the right-hand objects denoted by $X_k$. $x_k$ and $z$ denote the instance of an object. $p_i$ denotes the parameter that is instantiated when a rule is applied. The *pre* section defines the preconditions that should be satisfied before applying this rule. Grammar was widely used in natural language processing (Doran et al. 1994), bioinformation (Fishman, Porter 2005) , and building reconstruction (Vanegas et al. 2010).

$$Zz(p_1, p_2, ..., p_i)\langle pre\rangle := X_1x_1, X_2x_2, ...X_kx_k \tag{1.1}$$

#### 1.1.3.2 Application of rule system

The first applicable rule system is believed to be the DENDRAL system (Lindsay et al. 1993), which was developed by Stanford University in 1965, according to the requirements of the National Aeronautics and Space Administration. With the pre-entered rule of thumb, the DENDRAL system can automatically generate molecular structures that can interpret spectral data. Since then, rule system has attracted much attention from the researchers and been successfully applied in different domains, such as health diagnosis (Naser, Al-Bayed 2016), lexical analysis to compile or interpret computer programs (Dym 1985), and in natural language processing (Blosseville et al. 1992).

Building reconstruction can be regarded as a special reasoning problem. Hence, rule system has been extensively applied in building reconstruction by introducing prior knowledge about the buildings. This can benefit reconstruction by reducing the demands on the coverage, density, and accuracy of sensor measurements. For instance, Becker, Haala (2009) manually created a couple of grammar rules to represent the common compositional forms of windows and doors on walls. Based on the derived grammar, the accurate façade of buildings can be automatically detected with the help of a few sensor measurements (e.g., LiDAR or images). Likewise, Philipp et al. (2014) manually created a couple of grammar rules to represent the multi-hierarchy layout principles of indoor entities of buildings, such as corridors and rooms. Then, with partial sparse measurements (e.g., point clouds, images, and traces), the complete indoor layout of buildings can be reconstructed. Yang, Tian (2010) created a couple of IF-THEN rules about the shape and appearance characteristics of doors, based on which the doors can be recognized from images.

For simple problems, defining a few explicit rules to aid the reconstruction task is useful and cost-effective. This enhances the intelligence of measurements-based reconstruction solutions. However, when the problem becomes complex, creating thousands of explicit rules to describe the target domain is troublesome. The incorrect definition of a certain rule might even cause the failure of the whole system. For instance, in early times, many researchers manually derived grammar rules for formal language and hoped that this can improve the intelligence of natural language processing tasks (Boguraev, Briscoe 1987; Calder et al. 1988; Wang et al. 2000), such as part-of-speech tagging and syntax parsing. Finally, thousands of explicit rules were created but still could not obtain a practical and stable tagging or parsing system. Today, for such problems, most researchers leverage implicit rules derived by statistical learning (Brants 2000) especially deep learning (Young et al. 2018) when abundant tagged training data becomes possible in the coming big data era.

#### 1.1.3.3 Definition of statistical learning

Statistical learning refers to a set of tools that automatically derive implicit rules to model and understand the complex association relationships. These tools can be classified as supervised or unsupervised. Broadly speaking, supervised statistical learning involves building a statistical model for predicting, or estimating an output based on one or more inputs. Problems of this nature occur in fields as diverse as business, medicine, astrophysics, and public policy. With unsupervised statistical learning, there are inputs but no supervising output; nevertheless, we can learn relationships and structure from such data. This dissertation focuses on supervised learning.

Statistical learning can also be categorized into another two classes according to the feature extraction manners: traditional machine learning and deep learning. In early times, machine learning models were applied to solve many practical problems, and the representative machine learning models include Bayesian-related models, linear regression, logistic regression, genetic programming, support vector machine (SVM), decision tree, and random forest. In these models, each instance in a dataset is described by a set of features or attributes, which are identified by a domain expert to reduce the complexity of the data and make patterns more visible to learning algorithms to work. They normally require further processes such as feature selection and extraction by using Principal component analysis (PCA). However, feature selection and extraction becomes a troublesome task when the number of features is increasing.

Since 2012, when a deep learning model was developed by Google team to recognize humans and cats in YouTube, deep learning is gaining increasing attention due to its supremacy in terms of accuracy when trained with a huge amount of data and its simplicity in automatic learning of high-level features from data in an incremental manner. This eliminates the need for domain expertise and hardcore feature extraction. Most deep learning methods use neural network architectures, which is why deep learning models are often referred as deep neural networks. The term "deep" usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as thousands. At the very beginning, deep learning works on aligned matrix data, namely Euclidean structure, such as image, sound, and sentence, to conduct convolution. Recently, it is extended to the non-Euclidean structure, specifically graph, which is represented by nodes and the relationship between nodes (edges or links). The corresponding deep learning models are based on Convolutional Neural Network (CNN) (Kim 2014) and Graph Convolutional Network (GCN) (Kipf, Welling 2016), respectively.

#### 1.1.3.4 Application of statistical learning

Statistical Learning algorithms have been extensively applied in building reconstruction due to its inherited association characteristics. That is, the core task of building reconstruction is to associate the observations or sensor measurements (e.g., image and point cloud) to certain building elements (e.g., roof shape and building type); therefore, the machine learning algorithms were widely used in building reconstruction in early times (Adan, Huber 2011; Zhang et al. 2013). The image and point cloud are both Euclidean structures, which are perfect for CNN-based deep learning algorithms. Furthermore, the geographical space can be naturally organized as a graph structure, such as with the buildings as nodes and the street as links. Hence, GCN-based deep learning algorithms can also benefit building reconstruction. Next, some representative works that used traditional machine learning algorithms, CNN-based deep learning algorithms, and GCN-based deep learning algorithms in building reconstructions are introduced.

Traditional machine learning approaches were widely used in building reconstruction when the correlated information and dataset is the non-Euclidean structure or the deep-learning approaches are still immature in techniques and limited by the lack of tagged data and low computation power of hardware in the early times. For instance, Mohajeri et al. (2018) used SVM to classify building roofs in relation to their received solar energy and footprint in the city of Geneva in Switzerland. Lu et al. (2014) investigated the classification of building types (i.e., single-family houses, multiple-family houses, and non-residential buildings) from light detection and ranging (LiDAR) remote sensing data by using machine learning approaches (e.g., SVM, Random forest, Decision Tree). Turker, Koc-San (2015) presented an integrated approach for the automatic extraction of rectangular- and circular-shape buildings from high-resolution optical space borne images using the integration of SVM classification, Hough transformation and perceptual grouping.

As the mature of deep learning techniques and the adventure of the big data era, deep learning is becoming mainstream in building reconstruction. For example, many studies (Bischke et al. 2019; Vakalopoulou et al. 2015; Xu et al. 2018) have used CNN-based deep learning approaches to reconstruct buildings with high-resolution remote sensing imagery based on abundant tagged training data. Other studies (Axelsson et al. 2018; Özdemir, Remondino 2019; Wichmann et al. 2018) used CNN-based deep learning algorithms to detect the 3D model of buildings, such as the façade and the roof shape with point cloud data and building footprints. Srivastava et al. (2018) used CNN-based deep learning algorithms to classify the type of building with Google Street View and the tagged training data is automatically extracted from OSM.

CNN-based deep learning approach requires Euclidean structural data (e.g., image, point cloud, and sentence). However, in the real world, many data are not Euclidean structure but graph structure, such as the social network and the layout of geospatial entities. Graph-structured data cannot be well processed by traditional CNN-based algorithms. To meet this need, a graph convolutional network (GCN) was proposed by Kipf, Welling (2016). However, as far as we know, there are no works investigating how to use GCN in building reconstruction. In spite of this, it still has huge potentials. For example, GCN can be used to estimate the type (usage) and landmark-suitability of buildings by representing the buildings as the node of a graph, with the relationships (e.g., topology and contrast) between the surrounding buildings and roads as links. GCN can automatically collect the features from neighboring nodes to classify the node since the relationship, feature, and type of neighboring nodes affect the type of current node.

#### 1.1.3.5 Combination of explicit rules and statistical learning

Both defined explicit rules and implicit rules derived by statistical learning have their own strengths and weaknesses. On one hand, manually defining accurate and complete explicit rules for complex issues by experts is impossible due to the tremendous variety and diversity of the target field, such as the natural language and buildings worldwide. Meanwhile, the lack of tagged training data is always a big challenge for machine learning, especially deep learning algorithms. On the other hand, defining or deriving conceptual knowledge is readily by experts, but much more difficult by statistical learning. Inversely, to obtain accurate numerical rules or knowledge is what statistical learning is good at but difficult for experts. Realizing these facts, some researchers believed an optimal solution might be fusing explicit rules with statistical learning.

Researchers in the building reconstruction domain have conducted some initial attempts in combining statistical learning and explicit rules. The specific idea is to improve the initially defined explicit rules through statistical learning or complement the numerical rules (such as parameters and probability) from training data. For instance, to reconstruct the façade of certain styles of buildings, Dehbi, Plümer (2011) proposed a machine learning approach based on Inductive Logic Programming with the aim of learning grammar rules. The learning consists of inducing rules from a limited number of positive and negative examples together with background knowledge of the considered building part in a logical form. This background knowledge is either provided by a human user as the teacher or automatically extracted from a 3D point cloud. Philipp et al. (2014) used a reconstructed indoor layout with point clouds to learn the parameters of grammar rules in splitting the rooms. The enhanced grammar rules are then used to reconstruct the indoor layouts of other floors. This can reduce the demands on the coverage of sensor measurements. Gadde et al. (2016) presented a novel framework to learn compact grammar from a set of ground-truth images for the purpose of building façade reconstruction. To this end, parse trees of ground-truth annotated images are obtained running existing inference algorithms with a simple, very general grammar. From these parse trees, repeated subtrees are sought and merged together to share derivations and produce a grammar with fewer rules. Furthermore, unsupervised clustering is performed on these rules, so that, rules corresponding to the same complex pattern are grouped together leading to a rich compact grammar. Dehbi et al. (2017) proposed learning weighted attributed context-free grammar rules for 3D building reconstruction. They used SVM 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.

## 1.2 Research objectives and research questions

The main objective of this dissertation is to investigate the potential of applying the association between spatial elements to infer indoor and outdoor building elements given the available spatial information on VGI or provided by the sensing equipment, and to compare and explore the applicability of explicit and implicit rules in different building reconstruction tasks and to assess the feasibility of fusing both to achieve the optimal result. The main objectives are divided into three sub-objectives that are described below:

• Objective 1: To investigate the potential of applying association relationships in inferring building elements given only available spatial information on VGI.

• Objective 2: To explore the applicability of explicit rules and implicit rules derived from statistical learning in inferring distinct building elements.

• Objective 3: To assess the feasibility of fusing explicit rules and learned implicit rules to achieve the optimal prediction result.

In line with the research objectives, this dissertation raises three research questions (RQ). Among them, RQ1, RQ2 and RQ3 are raised with respect to the three objectives. The three research questions are listed as follows:

•RQ 1: What is the potential and limitation of association relationships in inferring the missing spatial elements given available spatial information on VGI?

•RQ 2: How can explicit and learned implicit rules be used to reason indoor and outdoor building elements and what is their strength and weekness?

•RQ 3: If it is possible and how to combine both explicit and learned implicit rules to achieve the best prediction result?

To answer the three listed research questions and draw conclusions from the research objectives, multiple representative building elements across indoors and outdoors should be chosen as examples. Only in this way can the potential of the association and the applicability of explicit and implicit rules in building reconstruction be thoroughly evaluated. Therefore, this dissertation takes four key indoor and outdoor building elements as examples and they are room usage, indoor landmark salience, roof type, and building entrance. Specifically, only the explicit rules and implicit rules work for the complex roof shape prediction and entrance prediction, respectively. Both explicit and implicit rules can be used in room usage and landmark salience prediction, and thus they were compared. The purpose is to investigate how and why the four building elements could be inferred by explicit and / or learned implicit rules based only on existing and available spatial information on VGI. Accordingly, four separate studies have been conducted with each focusing on one of the four building elements that would be inferred by explicit and /or implicit rules. Finally, we combine the results of the four studies to answer the listed three research questions.

### 1.2.1 Feasibility of using explicit rules to infer roof shape of complex buildings

In 3D building reconstruction, roof type is one of the key building elements. On OSM, some simple buildings have been tagged with the roof shape. Unfortunately, it is impossible to use a single tag to indicate the roof shape of a complex building because a complex building might comprise multiple different roof shapes. Furthermore, additional data requirements (e.g. aerial images or 3D point clouds) limit the usage and applicability of automatic roof reconstruction approaches. Intuitively, the footprint that is widely available on OSM correlates the roof shape of complex buildings. Therefore, the first sub-research task is to explore to what extent the association between the footprint and roof shape can be used to estimate the possible roof shape combination of a complex building. Specifically, three sub-research questions are raised:

(1): If the association between the footprint and roof shape exists?

(2): What is the strength and limitation of the explicit rules in reasoning roof shape?

(3): If the combination of explicit and implicit rules derived from statistical learning would perform better in this context?

The main goal of this sub-research task is to provide contributions to the understanding of the feasibility of estimating roof shape for complex buildings by explicit rules given the footprint. This sub-research task is fulfilled with one individual study (Publication 1) that is presented in Chapter 5, and the main findings are also summarized and discussed in Section 2.1.

### 1.2.2 Feasibility of applying learned implicit rules in main entrance tagging

The second studied building element is entrance, which is an important component connecting the internal and external spaces of buildings. For public buildings that are huge and complex, determining their main entrance is necessary for many LBS applications, such as wayfinding. However, only a small proportion of public buildings have been tagged with the main entrance on OSM. Intuitively, the design of the main entrance would follow certain principles, such as close to the centroid of the footprint and easily observed and accessed from main roads. Furthermore, these correlated elements have been mapped on OSM. Therefore, the second sub-research task is to explore to what extent the association between the main entrances and the footprint as well as spatial contexts can be used to predict the location of the main entrance based only learned implicit rules. Specifically, three sub-research questions are raised:

(1): Which spatial elements are associated to the main entrance?

(2): If explicit rules are also suitable for main entrance reasoning?

(3): How would the data quality issue of OSM affect the tagging accuracy?

The main goal of this sub-research task is to provide contributions to the understanding of the feasibility of reasoning the main entrance given footprints and spatial contexts on OSM by using implicit rules derived from statistical learning. This sub-research task is fulfilled with one individual study (Publication 2) that is presented in Chapter 6, and the main findings are also summarized and discussed in Section 2.2.

### 1.2.3 Comparison of explicit rules and learned implicit rules in inferring room usage

The third studied building element is the usage of rooms inside buildings. Currently, the indoor geometric map of many buildings has been tagged on OSM but room usage is normally ignored, which, however, is significant in indoor navigation. Intuitively, the usage of a room correlates its geometry, topology, and spatial distribution attributes that can be derived from a geometric map. This issue might be solved in two different ways. First, the indoor layout of buildings comprises multiple semantic hierarchies, which can be represented by grammars. Second, the geometry, topology, and spatial distribution attributions can be treated as features to predict the room usage. Therefore, the third sub-research task is to explore the feasibility of using explicit rules and statistical learning to infer room usage and to compare the two solutions in this context. Specifically, three sub-research questions are raised:

(1): How to construct the grammar rules that represent the association between room usage and geometry, topology, and spatial distribution attributes?

(2): How could the explicit rules and implicit rules derived by statistical learning be used to infer the room usage?

(3): What is the strength and weakness of explicit rules and implicit rules in room usage reasoning?

The main goal of this sub-research task is to provide contributions to the understanding of the feasibility and limitation of explicit rules and implicit rules derived by statistical learning in room usage inferring. This sub-research task is fulfilled with two studies (Publications 3 and 4) that are presented in Chapter 7 and 8, and the main findings are also summarized and discussed in Section 2.3.

### 1.2.4 Comparison of explicit rules and implicit rules derived from statistical learning in landmark salience prediction

Many indoor landmarks (e.g., shop, staircase, and vending machines) have been tagged on OSM, and knowing the salience of these landmarks is important in landmark-based way finding. The salience of landmarks correlates its visual and semantic attributes. To measure the salience of landmarks, a salience model is required to represent the quantitative relationship between the salience and visual and semantic attributes of landmarks. This sub-research task is to compare the learned implicit rules and explicit rules in predicting the salience. Specifically, three sub-research questions are raised:

(1): How to measure the true salience of landmarks?

(2): How to derive implicit rules by statistical learning to represent the quantitative relationship between the landmark salience and attributes?

(3): What is the strength and weakness of explicit rules and implicit rules derived by statistical learning in calculating the salience?

The main goal of this sub-research task is to provide contributions to the understanding of the strengths and weakness of implicit rules derived from statistical learning and explicit rule defined by experts in predicting the landmark salience. This sub-research task is fulfilled with one individual study (Publication 5) that is presented in Chapter 9, and the main findings are also summarized and discussed in Section 2.4.

## 1.3 Dissertation outline

This section outlines the dissertation with a brief description of the content of each chapter. It further sketches the structure of the dissertation and provides the list of the relevant publications.

### 1.3.1 Dissertation structure

This dissertation comprises two major parts: (I) Synopsis and (II) Publications. The first part (I Synopsis, Chapter 1 through 4) provides an overall description of the research and briefly discusses the individual publications that are presented in the second part (II Publications, Chapter 5 through 9).

The first chapter provides the introduction of the motivation, research context, identification of research objectives and research questions, and the dissertation structure. Afterward, the major results and findings are discussed in Chapter 2. As the research objectives are achieved with several individual publications, each of the first four sections of Chapter 2 discusses one to two publications that are presented in later chapters to answer the research questions with different examples. The major results of these publications are then summarized in Chapter 3 and future work is outlined in Chapter 4.

The second part (II Publications) comprises five peer-reviewed articles. The presented publications constitute new research findings and together contribute to the answer to the proposed research questions. Chapter 5 investigates the feasibility or potential of applying explicit rules to infer the possible roof shape combination for complex buildings on OSM. More specifically, the combination and symmetry characteristics of the footprint are utilized to rule out the incorrect partition of footprints and the impossible roof shape combinations of complex buildings. Chapter 6 demonstrates the performance of implicit rules derived by imbalanced learning algorithms (e.g., weighted random forest) in predicting the location of the main entrance for public buildings by using only available data on OSM. Chapter 7 evaluates the feasibility of using explicit rules to represent the indoor layout templates of research buildings and using statistical learning to derive the geometric rules, with which the room usage can be reasoned by a bottom-up parsing manner. Chapter 8 attempts to utilize random forest and relational graph convolutional network to derive implicit rules to classify the room usage in research buildings. Chapter 9 attempts to use genetic programming to automatically learn a set of implicit rules to represent the quantitative relationship between the landmark salience and the visual and semantic attributes of landmarks. Then, the learned rules were compared with the conventional explicit rules in salience prediction accuracy and interpret-ability.

### 1.3.2 Publications

As explained above, the research questions of this dissertation are explored with several separate studies, and the results of these studies have been submitted and/or accepted in several peer-reviewed journals. Thus, these publications are an integrated part of this dissertation. This section provides a list of these supporting publications.

[1] Hu, X., Fan, H., Noskov, A. (2018): Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints. International Journal of Digital Earth. Vol. 11(10), pp.1039-1063, Doi: 10.1080/17538947.2017.1373867.

[2] Hu, X., Noskov, A., Fan, H., Novack, T., Gu, F., Li, H., Shang, J. (2019). Tagging the Buildings' Main Entrance based on OpenStreetMap and Binary Imbalanced Learning. International Journal of Geographical Information Science. (Major revision)

[3] Hu, X., Fan, H., Noskov, A., Zipf, A., Wang, Z., Shang, J. (2019). Feasibility of Using Grammars to Infer Room Semantics. Remote Sensing. vol. 11(13), p.1535. Doi: 10.3390/rs11131535.

[4] Hu, X., Fan, H., Noskov, A., Wang, Z., Zipf, A., Gu, F., Shang, J. (2019). Room Semantics Inference Using Random Forest and Relational Graph Convolutional Network: A Case Study of Research Build- ing. Transactions in GIS. (Minor revision)

[5] Hu, X., Ding, L., Shang, J., Fan, H., Novack, T., Noskov, A., Zipf, A. (2019). A Data-driven Approach to Learning Saliency Model of Indoor Landmarks by Using Genetic Programming. International Journal of Digital Earth.

Xuke Hu is the first author for all the five publications, and in each publication, Xuke Hu conducted the main work through all stages including conceiving the ideas, designing and implementing the algorithms, performing the experiments as well as the drafting of the articles. The five publications have made the most relevant contributions to this research. Hence, the findings of these five publication will be discussed in separate sections in Chapter 2, and the full content of these publications will also be provided in the second part of this dissertation in Chapter 5-9 in a consistent format. Meanwhile, the coauthors, Prof. Dr. Alexander Zipf, Prof. Dr. Hongchao Fan, Prof. Dr. Jianga Shang, Dr. Alexey Noskov, Dr. Zhiyong Wang, Dr. Tessio Noskov, Dr. Fuqiang Gu, Hao Li, and Lei Ding, have also contributed to these publications by providing constructive comments and suggestions. The authors' contributions to each publication are stated correspondingly in the appending section of ”Declarations of authorship”.

# 2. Results and discussions

This chapter presents the results of the peer-reviewed publications as listed in Section 1.3.2. The findings from each of the five main publications are introduced and discussed in detail with one separate section of this chapter.

## 2.1 Feasibility of using explicit rules to infer roof type of complex buildings

An important task of 3D building reconstruction is to determine the roof shape of buildings, which is still a big challenge specifically for complex buildings. Current approaches for reconstructing the roof shape are based on sensing equipment such as LiDAR, which depends highly on measurements. The study that uses only the available information on OSM to predict the roof shape for complex buildings is still missing, and this forms the first study of this dissertation.

Therefore, the first publication (Chapter 5) attempts to provide insights into the question that to what extent the roof shape of complex buildings can be predicted by explicit rules with only footprints on OSM. In this section, the main findings from this publication will be summarized and discussed.

### 2.1.1 Results

For the complex building with rectilinear polygons, describing its roof shape with a single tag is impossible because it might comprise multiple distinct roof shapes, such as gabled, hipped, and half-hipped. Faced with this challenge, the footprint of complex buildings is first divided into single rectangles with each corresponding to a certain roof shape. A footprint normally has multiple partitions, and the optimal one is chosen based on a couple of explicit rules. These rules involve the number of rectangular fragments, parallel rectangles, and symmetrical sub-clusters in a partition. Given the optimal partition, the roof shape of each rectangle can be then predicted through manually defined rules, considering the symmetric characteristics and the combination manners (e.g., the Linear shape, T-shape, and L-shape) of rectangles in the optimal partition.

These rules are derived mainly from two facts. One is the roof design principle presented in (Kruger, Seville 2012). It declares that roofs should be designed to minimize the collection of rainfall, snow, and leaves without creating leaks and narrow spaces. This principle limits the possible roof shapes of two adjacent rectangles when considering three candidate roof primitives: gabled, hipped, and half-hipped. The other is that the symmetry property in partitions is reflected in roof shapes. These rules are verified based on 30 complex buildings. The experiential results proved that the manually defined rules are effective in ruling out impossible combinations of roof shapes for complex buildings. Figure 2.1 illustrates the corresponding association graph. Solid rectangles and ovals denote known categorical and numerical elements respectively, while the dashed shape denotes unknown elements. The shape with a shallow yellow background denotes the target element that should be predicted. The solid line denotes the association relationship. We can see there is only one numerical element and six associated elements in total in this context. However, there are two unknown elements. The major findings of this study include:

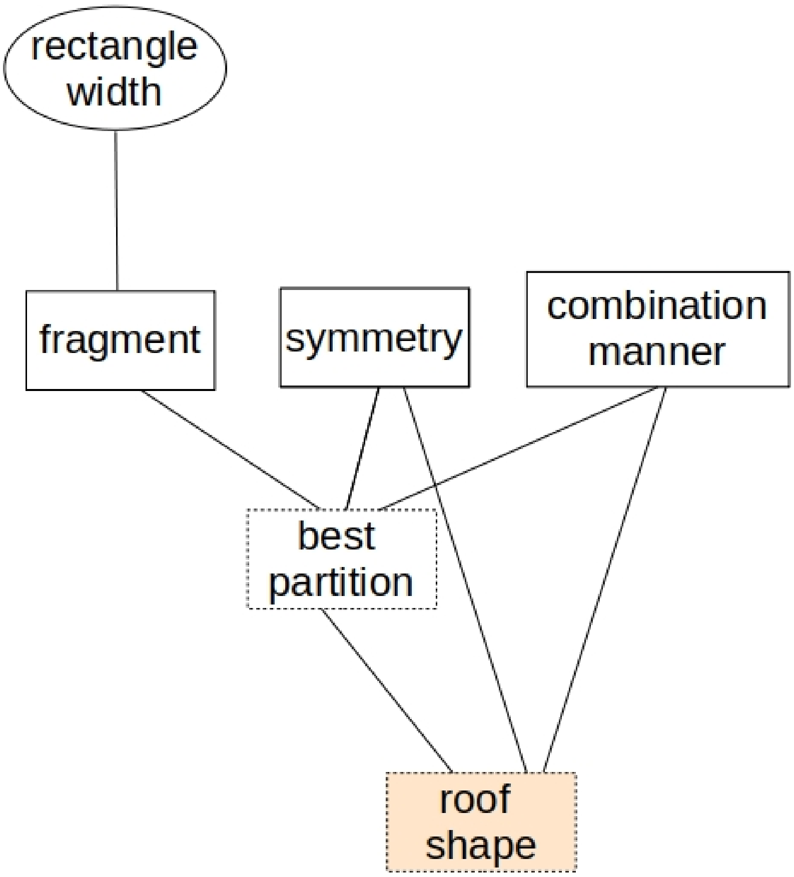


**Figure 2.1.** Association graph for roof shape prediction.

(1). By partitioning the complex buildings into multiple adjacent rectangles, it becomes possible to describe the roof shape of complex buildings.

(2). The roof shape of buildings (especially complex buildings) correlates highly with their footprint. With a couple of simple explicit rules, the true roof shape combination of complex buildings can be limited to a small range based only on the footprint. The top-ranked prediction results can aid the measurements-based solution and also can be recommended to OSM volunteers to facilitate their tagging work.

(3). Implicit rules derived from statistical learning are not suitable for reasoning the roof shape of complex buildings because the reasoning procedure comprises multiple consecutive and interdependent sub tasks (two unknown elements).

### 2.1.2 Discussions

In this study, only the explicit rule is utilized to partition the footprint, select the optimal partition, and determine the possible roof shape combination given the association graph without using implicit rules derived from statistical learning as the previous study did in determining roof shapes. Previous studies normally regarded a building as an inseparable object such that each building is assigned with one certain roof shape and regarded the combination of multiple roof shapes as a special roof shape, such as ‘complex roof shape’. That is, they could not describe the detailed roof shape combination for complex buildings. Therefore, it is quite common to use learned implicit rules to predict the roof shape of a building based on measurements in previous studies. However, the issue in this study is not a simple classification problem but a continuous reasoning problem (pipeline) that comprises multiple consecutive and interdependent sub tasks. This is why statistical learning was not investigated in this study.

In the whole procedure, several explicit rules are defined since the number of total associated elements and numerical element in the association graph are both small. These are simple ”IF THEN” rules as presented in Section 1.1.3. A small number of rules make the developed rule system controllable and also efficient in ruling out the incorrect options of the roof shape combination for complex buildings. From this point of view, the proposed rule system is suitable in this context. Note that, one rule involves in numerical elements, which is also manually defined. For instance, it is defined that the rectangle with the width below 3 meters is regarded as the fragment that should be avoided as much as possible in a partition. The threshold was assigned according to our experience, which might be inaccurate and lead to misjudgment for the buildings outside the test set. A better solution might be deriving this rule using statistical learning given abundant tagged training data.

## 2.2 Potentials of applying learned implicit rules in main entrance tagging

The main entrance is an important component that connects the indoor and outdoor spaces of buildings. Knowing the main entrance of buildings (especially public buildings) can benefit many applications, such as wayfinding, which can dramatically reduce users' efforts in finding the main entrance. Main entrance detection is not a widely investigated topic and only a few studies have proposed detecting the entrance through street-level images, which however is not widely applicable due to the sparse coverage of worldwide street-level images. The study that uses only the available information on OSM to predict the location of the main entrance for public buildings is still missing, and this forms the second study of this dissertation.

Therefore, the second publication (Chapter 6) attempts to provide insights into the question that to what extent the main entrance can be predicted by learned implicit rules using only available OSM data. In this section, the main findings from this publication will be summarized and discussed.

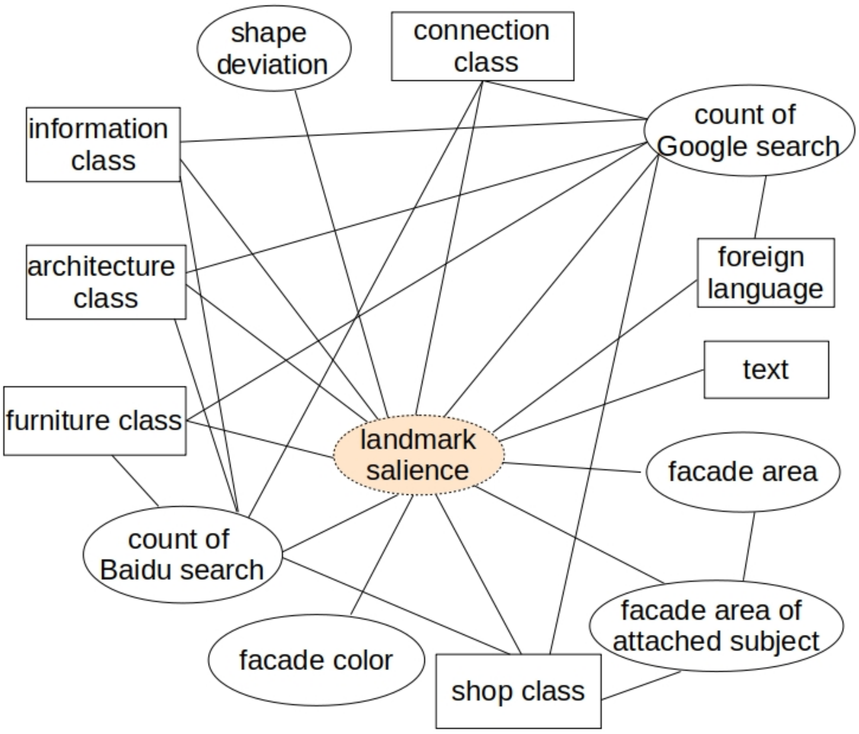


**Figure 2.2.** Association graph for main entrance tagging.

### 2.2.1 Results

Entrance detection is treated as a binary classification problem by discretizing the footprint into single points in an interval of 3 meters. The task is thus to determine which point is most likely the main entrance. Intuitively, the location of the main entrance is correlated with the shape of footprint and spatial contexts (e.g., main road, service way, pedestrian way, and bicycle parking area). 84 features are extracted in total by measuring the spatial distribution characteristics of the point on the footprint (e.g., distance to the centroid) and the relationship (e.g., shortest path distance and visibility) between the candidate point and spatial contexts. The corresponding association graph is shown in Figure 2.2, where only partial associated elements have been depicted since the total number of associated elements is over 80. From the simplified graph, we can still see many elements are numerical variables and there are internal association relationships. These factors make the association graph pretty complex. Manually defining explicit rules to represent the entire association relationship is challenging. Thus, in this study,we adopted three imbalanced learning models (i.e., weighted random forest, balanced random forest, and SmoteBoost) to derive the implicit rules, which are evaluated based on 320 public buildings (collected from seven German cities) with an average perimeter of 350 meters. An acceptable tagging accuracy has been achieved, which can greatly reduce pedestrians' effort in finding the main entrance of buildings.

The major findings of this study include:

(1). The main entrance of public buildings highly correlates the footprint and spatial context, and the tagging accuracy can be further improved by considering more correlated spatial entities on OSM.

(2). Implicit rules derived from statistical learning perform well in entrance detection because this issue can be regarded as a simple binary classification problem with enough training data.

(3). Manually defined explicit rules are unsuitable for entrance detection because the number of correlated spatial elements is quite large (84) and most of them are numeric variables. Moreover, there exists complex internal association among the known elements. It is challenging to explicitly define a set of rules that can clearly and precisely represent the complex association relationships among a large number of elements.

### 2.2.2 Discussion

The large tagging error (over 60 meters) is often caused by inaccurate and incomplete data on OSM. For instance, a green space is wrongly tagged as a freely accessed space, or a fence that surrounds the building is not tagged on OSM. The data quality issue is the biggest challenge of VGI, which restricts its potential in many applications. One of the solutions would be enhancing the VGI with the data from other data sources. Taking the entrance detection as an example, the satellite imagery (e.g., from Bing map) can provide further evidence about the possible locations of the main entrance. For instance, the open space before the main entrance or the inaccessible area (e.g., green space or gardens) can be identified from the satellite imagery. The specific strategy could be fusing the manually defined features extracted from OSM and the features automatically extracted from the satellite imagery with deep learning in an integrated model.

One of the assumptions of the proposed solution is that there is one and only one main entrance in a public building. This is due to two reasons. First, in most cases, this assumption holds. Second, it would be quite challenging to detect a variable number of entrances in a public building if we are uncertain how many main entrances exist. However, when collecting the test buildings, we also found that a public building might be comprised of multiple departments, with each having one house number and one corresponding main entrance. Such buildings are beyond the scope of our work. However, this will be dealt with in future work by considering the house number tagged on OSM since each house number corresponds to one entrance. That is, multiple main entrances can be identified from a building if the tagged location of the house number is known.

## 2.3 Comparison of explicit and implicit rules in inferring room usage

Room usage plays a vital role in indoor location-based search and navigation. However, both the automatic mapping approaches by using sensing equipment and online map providers (such as OSM and MazeMap) focus on the generation of geometric maps, and the room usage is normally ignored. The room usage correlates the room geometric, topological and spatial distribution characteristics, which can be extracted from a geometric map. The study that uses only the geometric maps to predict room usage is still missing, and this forms the third study of this dissertation. In this context, both of the explicit rules and learned implicit rules seem feasible because room usage tagging is a typical association task and the number of total associated spatial elements and numerical elements are both small, which can be seen from Figure 2.3.

Therefore, the third and fourth publications (Chapters 7 and 8) attempt to provide insights into the question that to what extent the room usage can be predicted by explicit rules and implicit rules derived from statistical learning given the geometric map, respectively. In this section, the main findings from the two publications will be summarized and discussed.

### 2.3.1 Results

This study takes the research buildings at universities as examples to explore the two ways to inferring the room usage. For the explicit rule-based approach, the indoor layout of the research buildings is divided into multiple semantic hierarchies, from the footprint at the uppermost level, to the corridors and enclosed spaces at the middle level, and until the single rooms with usage at the lowest lever. The grammar rules are then defined according to the multiple-level semantic division. These rules represent the topological and spatial distributional characteristics

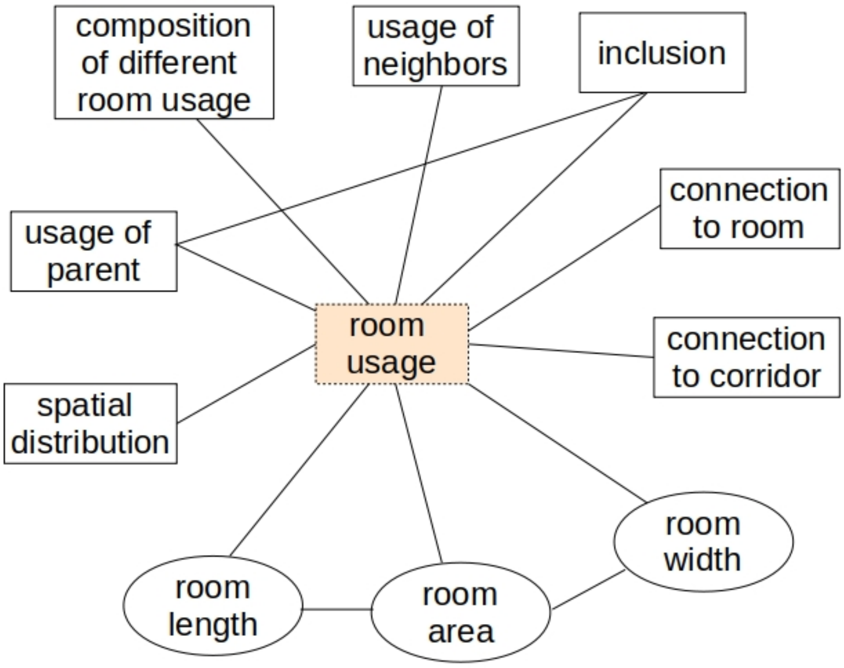


**Figure 2.3.** Association graph for room usage tagging.

of different room types. However, the geometric characteristics (e.g., area) involve numeric constraints. They are thus represented by Bayesian inference. That is, given the geometric characteristics of a room, the probability of assigning it to a certain type is first calculated by Bayesian inference. The grammar rules and Bayesian inference are combined in a bottom-up parsing process that constructs a parse forest for a floor. From the parse forest, the usage of each room is calculated. In total, 20 grammar rules are manually defined. The experimental results on 15 test floor plans show that an acceptable tagging accuracy is achieved using grammars. However, there are still some cases that violate the defined rules.

For the implicit rule-based approach, the geometric, topological, and spatial distribution characteristics of rooms are directly extracted from the geometric map as features. 12 features are extracted in total. Random forest and relational graph convolutional networks (R-GCN) are used to derive implicit rules and conduct prediction. Note that, in R-GCN based approach, the room is modeled as a node and with the adjacency relationship between the rooms as the links. The results show that random forest achieves better results than R-GCN in this context.

The major findings of this study include:

(1) The geometric, spatial distribution, and topological characteristics of rooms highly correlate the room usage such that they can be used to infer the usage of rooms with acceptable

accuracy, at least in research buildings.

(2) Using explicit rules to deal with the categorical elements and using learned implicit rules to deal with the numerical elements is promising in room usage tagging. However, it is still a challenge to cover all the indoor layouts.

(3) Purely using implicit rule derived from statistical learning to predict room usage is a more general way compare to grammars because it can be extended to other building types with slight modification. For the grammar-based solution, a new set of explicit rules should be created for a new building type.

### 2.3.2 Discussion

Compared to explicit rule-based approaches in room usage tagging, implicit rules derived by statistical learning are more robust and extendable (to other types of buildings). For the former one, only a small proportion of test buildings can be accurately predicated because an incorrectly defined rule can cause the failure of tagging the rooms of one floor. Inversely, by using the statistical learning to deriving the rules, an acceptable tagging accuracy of 0.85 is achieved based on the total test floors. Statistical learning is more extendable in this context because the method of deriving implicit rules for research buildings can be reused in other similar building types, such as office buildings and hospitals without much modification. Adding new associated elements or deleting useless associated elements is also quite simple. However, for the explicit rules-based approach, a new set of rules should be carefully defined to meet the requirement of a new building type, which is a troublesome task.

Despite this, the explicit rule-based reasoning approach outperforms statistical learning-based on interpretability and knowledge-re-usability. Explicit rules (specially grammars) represent the underlying knowledge of a certain field, based on which, multiple applications can be promoted. Taking the grammar of indoor layouts as examples, the constructed grammar can be not only used to reason the room usage but also to explain why a room is an office instead of a toilet. Besides, the grammar can be used to formally represent a map and help computers to read or understand the map. Last but not least, it can be used in computer-aided building design. However, the learned implicit rules normally focuses on a specific and single problem in the target field. If a new problem appears in the same field, a new sets of implicit rules should be derived by statistical learning from new training data. For instance, the implicit rules for room usage tagging in this study cannot be used to reconstruct the indoor layout with the help of measurements. They cannot also be used in generating candidate indoor layouts to facilitate the work of building designers.

## 2.4 Comparison of explicit and learned implicit rules in landmark salience calculation

Providing landmark-based wayfinding services can reduce user's anxiety in exploring complex indoor environments, such as shopping malls. The key step is correctly determining the salience of landmarks (e.g., shops, staircases, and vending machines). The traditional solution defines explicit rules to represent the linear relationship between the salience of landmark and its visual and semantic attributes. However, a linear model is believed to be inaccurate due to the existence of internal association, as shown in Figure 2.4. There are 14 associated elements in total. Among them, 6 elements are numerical variables, including the target elements. Besides, there are complex internal association relationships, such as the text and foreign language. Thus, applying defined explicit rules to accurately represent all the association relationships is difficult. The study that proposes deriving rules to represent the non-linear relationship is still missing and this forms the fourth study of this dissertation.

Therefore, the fifth publication (Chapter 9) attempts to provide insights into the question that to what extent the implicit rules derived by genetic programming can represent the accurate non-linear relationship between the salience of landmarks and their visual and semantic attributes. In this section, the main findings from this publication will be summarized and discussed.

### 2.4.1 Results

The study takes shopping malls as examples and collects pictures of 200 scenes in two famous shopping malls in Wuhan City, China. From each scene, 3 to 4 landmarks are marked. To get ground truth data about the salience of a landmark, 200 volunteers (with ages ranging from 18 to 32) are recruited to select the most attractive landmark from the scene. The proportion of volunteers that selected a certain landmark is then used as the salience value (in the range of 0 to 1) of the landmark. This study does not make any assumptions about the mathematical model. Instead, based on tagged training data, genetic programming is used to automatically learn an optimal mathematical model (individual tree) with the basic operators (+,-,*,/, and square) as the non-terminal nodes and the attributes as the terminal nodes.

Next, five-fold cross-validation is used to evaluate the proposed solution and the result is compared with the traditional solutions that manually define rules to represent the form of the salience model and manually or automatically set the parameters. The result shows that in 76% of the cases the learned implicit rules can correctly predict the most attractive landmark from a scene, while an accuracy below 41% is achieved by the traditional solutions.

The major findings of this study include:

(1) The visual and semantic attributes of landmarks highly correlate their salience perceived

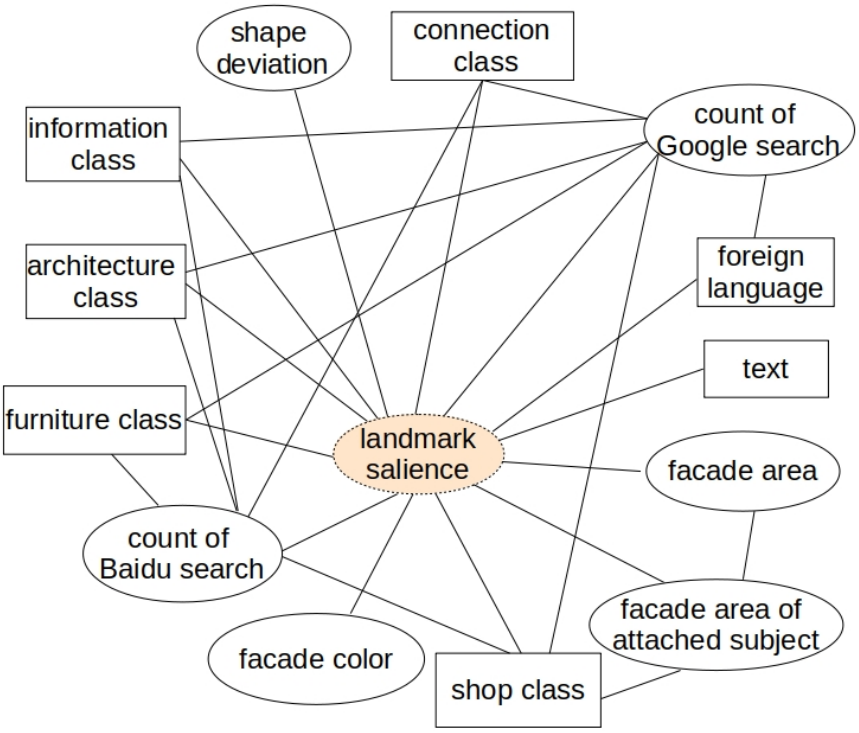


**Figure 2.4.** Association graph for landmark salience calculation.

by humans. The association can be used to accurately predict the salience of a landmark in an indoor environment.

(2) The implicit rules automatically learned from tagged training data can represent a more accurate quantitative relationship between the salience of landmarks and their attributes than explicit rules manually defined by experts.

(3) The manually created explicit rules outperforms the learned implicit rules in interpretability. That is, it is clear that how each attribute of a landmark affects its salience in the manually created rules, which, however, is impossible in the learned rules.

### 2.4.2 Discussions

This study reveals again that implicit rules derived from statistical learning outperform the explicit rules in dealing with the numeric elements. This is concluded in two aspects. First, previous studies have used statistical learning (specifically SVM) to derive the parameters of the explicit rules that are manually defined. The result showed that the semi-automatic manner still outperforms the fully manual manner, in which both the form and parameters are defined by experts. Second, a fully automatic manner (by genetic programming) outperforms a semi-automatic manner.

The explicit rules only suit a specific environment, such as in the street, on campus, or in a shopping mall. That is, the change of environments requires the change of the explicit rules. However, the proposed solution in this study does not require intervention from experts (e.g., manually setting weight values) and can be easily extended to other indoor and outdoor environments without re-proposing a new set of rules. The learned implicit rules is data-dependent and can make accurate predictions given abundant training data but cannot explain the exact influence of distinct attributes on the salience of the landmark as explicit rules do.

# 3. Conclusions

In the first Chapter, three research questions have been raised to achieve the three research objectives. These research questions are explored with five publications that are provided in the second part of this dissertation, meanwhile, the most relevant findings of these publications are summarized and discussed previously in Chapter 2. In this chapter, these results are further summarized to sketch out the main contributions of this dissertation.

The current two mainstream building reconstruction solutions are VGI and equipment-based sensing. The first faces data quality issue (e.g., incompleteness) while the second depends highly on the coverage and accuracy of sensor measurements, which, however, is hardly fulfilled in worldwide scale. Therefore, the main objective of this dissertation is to investigate the potential of inferring the indoor and outdoor building elements given only the existing or available spatial elements on VGI or obtained through sensing equipment. Moreover, this dissertation compared and explored the applicability of explicit rules derived from expert knowledge and implicit rules derived from statistical learning in predicting distinct indoor and outdoor building elements based on the association of spatial elements. To realize the objectives, four representative building elements (i.e., roof shape, room usage, entrance, and landmark salience) are taken as examples to explore how they can be inferred based only on available spatial data at hand by rules.

The first study (Chapter 5) investigated how the explicit rules related to the combination and symmetric characteristics of the footprint on OSM can be used to reason the possible roof shape of complex buildings. The highly ranked candidates can be recommended to OSM contributors. The second study (Chapter 6) attempted to infer the location of the main entrance of a public building based on the footprint and spatial context (e.g., main road) of the building, which are available on OSM. Implicit rules derived by three imbalanced learning approaches are also compared in this context. The third study (Chapter 7 and 8) compared the grammar rules and implicit rules derived by Random Forest and RGCN in predicting the room usage of research buildings based on the geometric map. The drawbacks and strengths of the two solutions are discussed. The fourth study (Chapter 9) used implicit rules derived by statistical learning to predict the salience of landmarks in shopping malls according to the attributes of landmarks. Specifically, genetic programming is used to derive implicit rules which can model the qualitative relationship between the salience and the visual and semantic attributes. Furthermore, the learned implicit rules are compared with the manually defined explicit rules.

The main findings of this dissertation can be summarized as follows:

(1) The association between the spatial elements in the real world is ubiquitous. For instance,

the footprint correlates the roof shape, the usage of rooms correlates the geometric and topological characteristics of rooms, the location of entrances correlates the shape of footprint and the spatial contexts, and the salience of landmarks correlates the visual and semantic attributes of landmarks.

(2) The association between the spatial elements is useful in reasoning the missing building elements, but the performance varies as the degree of association and amount of associated spatial elements. For instance, with only the footprint, the absolute accuracy of predicting the correct roof is low such that the proposed approach cannot be used independently. However, this can rule out most of the incorrect roof shape options of buildings, which can benefit the measurement-based solutions by reducing the requirement on the coverage of the measurements or improving the accuracy. Inversely, given the footprint and spatial context of buildings, the accuracy of estimating the location of the main entrance is pretty high such that the proposed approach can already benefit the real way-finding application without the extra need on the sensor measurements.

(3) Generally, explicit rules work well in a relatively simple problem with a small number of associated elements. The experts can thus manually create a few rules based on their knowledge and experience in the target field. However, if the problem becomes complex or unclear (with a huge number of associated spatial elements and numerical elements and even internal association), the implicit rules derived by statistical learning performs better than explicit rules if abundant training data is available. In this case, manually creating robust rules by experts is impossible. For instance, in the first study (Chapter 5), two main principles are followed, based on which no more than 10 explicit rules are created to infer the possible roof shape given the footprint. However, in the entrance detection issue (Chapter 6), there are no clearly defined codes that describe how the footprint and surrounding contexts correlate the main entrance. Besides, the number of the factors that affect the location of the main entrance is large. In total, 84 associated elements have been extracted and most of them are numerical.

(4) Normally, the learned Implicit rules outperform explicit rules in prediction accuracy and robustness in complex issues when abundant training data is available. This is mainly because the procedure of deriving the implicit rules can be seen as an evolution task and its purpose is to obtain the optimal one that can best predict the result of the training data. However, the explicit rules defined by experts can be regarded as the one without optimization.

(5) Explicit rules outperform the implicit rules derived from statistical learning in dealing with the categorical elements while the implicit rules derived from statistical learning perform well in dealing with the numerical element. For instance, grammar rules are explicitly defined to represent the topological and spatial distribution attributes of different room types by experts but the geometric attributes (e.g., area and length) cannot be readily defined by experts (Chapter 7). However, machine learning can easily obtain the implicit geometric rules of different room types

given abundant training data (Chapter 8). Therefore, combining explicit rules and statistical learning in complementary ways could achieve the best of both worlds.

(6) The domain knowledge represented by explicit rules is normally reusable such that the rules can be easily extended to solve other problems in the same domain. However, the learned implicit rules for a certain problem cannot be reused to solve other problems in the same domain. If a new problem in the same domain appears, new training samples need to be collected and new implicit rules need to be learned. For instance, the explicit grammar rules for indoor layouts cannot only be used to infer the room usage, but also to reconstruct the indoor map, explain a map, and aid the work of architects in building layout generation (Chapter 7). However, the learned rules for room usage tagging cannot be used to solve other problems in the same field (Chapter 8).

(7) The implicit rules derived from machine learning methods, especially deep-learning methods have poor interpret-ability such that no one can explain the cause of the estimation result in the manner that humans can easily understand. However, the explicit rule clearly defines how the reasoning procedure is executed given the input, thus has strong interpret-ability. For instance, in the fifth study (Chapter 9), based on the learned implicit rules, it is unclear how a certain attribute of the landmark would affect the salience of landmarks perceived by humans but the manually defined explicit rules can explain the reason.

(8) When and how to use explicit rules and learned implicit rules in building reconstruction depends on the specific application domain and issue, such as the number of associated elements and internal association relationships, the requirement on the prediction accuracy, and the availability of expert knowledge or tagged data. We cannot conclude that learned the implicit rules outperform the explicit rules, although the former is gaining much more attention than the latter nowadays. As above-concluded, both of them have their suitable application scenes, drawbacks, and strengths.

# 4. Future works

In general, the research limitations and potential enhancements for the proposed methodologies in each study are identified in the corresponding publications in Chapter 5-9. Hence, this chapter is not going to replicate the detailed methodological issues, but to address some open research questions of this dissertation from a broader angle.

A very challenging yet interesting research direction for the future is the deep fusion of explicit rules and statistical learning in building reconstruction. In the third study (Chapter 7), explicit rules have been combined with Bayesian inference, which is used to learn the implicit geometric rules of different room types. However, this is still a shadow fusion of the two kinds of rules. Nowadays, machine learning and deep learning attract a lot of attention from both the academia and industry, in which only the learned implicit rules from tagged data matter without too much attention on the explicit rules. However, on one hand, the decision made by the implicit rules is often stupid, which humans can easily avoid. This is because some easily-understood common sense or rules by humans are hardly learned with the tagged data or sparse tagged data. On the other hand, the knowledge and experience of experts in a certain field are un-utilized, which is a huge waste. To overcome this challenge, one of the possible ways would be using explicit rules to synthesize training samples. This can let the model to 'understand' the explicit rules. Besides, the uncertainty or inaccuracy of the explicit rules can be mitigated during the training process and would not affect the performance of the model. This is because the model would ignore some abnormal points caused by the inaccurate explicit rules to achieve the highest accuracy. However, how many samples should be synthesized and where is the balanced point of the inaccuracy of the explicit rules still need further investigation.

One of the biggest shortage of statistical learning especially deep learning is the lack of transparency and interpret-ability. That is, the trained model consisting of a set of implicit rules is acting as a black box, which is hardly understood by humans. Humans cannot leverage the trained model to explain the cause of the prediction result, which however, in many situations matters. For instance, an transparent model can help the deep learning experts to understand the reason of the failure of a trained model. In this way, they can propose corresponding solutions to avoid the failure. Otherwise, what they can do is just adjusting the architectures or parameters of the model in a random manner and then hoping for good luck. Besides, in service sectors, such as medical diagnosis, the doctor must explain the diagnose result to the patient instead of only telling them the final result. To overcome this challenge, one of the possible solutions would be leveraging available explicit rules to induce other explicit rules from the learned implicit rules. Explicit rules can be regarded as the higher abstract level of the implicit rules. Assume a scene that a couple of explicit rules are known and implicit rules have been learned by deep learning methods. Next, different models can be used to cluster and merge the implicit rules from the lowest level to the highest level (explicit rules). The optimal one would be the one that the generated explicit rules contain the most known explicit rules.

In VGI, such as OSM, the data quality is always an issue. It is true that very small proportion of volunteers would not follow the defined specifications such that they do not correctly tag the data or forgot to tag some key data on OSM. How to find these incorrect data on OSM in a fast and automatic way is still a challenge. To overcome this challenge, one possible solution would be training a model to automatically detect the abnormal points caused by small proportions of volunteers on OSM. This is inspired by the findings of two entity recognition tasks related to OSM. The first is the entrance detection task presented in the second study of this dissertation. It is found that most of the large tagging errors are caused by the missing of data on OSM. The second is place name recognition, training a model based on positive samples from OSM and synthesized negative ones. It is found that many incorrect place names on OSM (such as ‘the road is closed’ and ‘fallen trees’) have been judged as negative by the model. The two examples implied that it is feasible to train a model to detect the abnormal points on OSM.

# References

*Adan Antonio, Huber Daniel.* 3D reconstruction of interior wall surfaces under occlusion and clutter // 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission. 2011. 275–281.

*Allegrini Jonas.* A wind tunnel study on three-dimensional buoyant flows in street canyons with different roof shapes and building lengths // Building and Environment. 2018. 143. 71–88.

*Axelsson Maria, Soderman Ulf, Berg Andreas, Lithen Thomas.* Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds From Aerial Imagery // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2018. 1293–1297.

*Ayanlade A, Babatimehin O, Olawole MO, Orimogunje OOI, others .* Geospatial quality data acquisition problems in Sub-Saharan Africa. // Journal of Sustainable Development in Africa. 2010. 12, 4. 146–152.

*Azhar Salman.* Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry // Leadership and management in engineering. 2011. 11, 3. 241–252.

*Bahu Jean-Marie, Koch Andreas, Kremers Enrique, Murshed Syed Monjur.* Towards a 3D spatial urban energy modelling approach // International Journal of 3-D Information Modeling (IJ3DIM). 2014. 3, 3. 1–16.

*Becker Susanne, Haala Norbert.* Grammar supported facade reconstruction from mobile lidar mapping // ISPRS Workshop, CMRT09-City Models, Roads and Traffic. 38. 2009. 13.

*Bischke Benjamin, Helber Patrick, Folz Joachim, Borth Damian, Dengel Andreas.* Multi-task learning for segmentation of building footprints with deep neural networks // 2019 IEEE International Conference on Image Processing (ICIP). 2019. 1480–1484.

*Blosseville Marie-Joëlle, Hebrail Georges, Monteil Marie-Gaëlle, Penot Nadine.* Automatic document classification: natural language processing, statistical analysis, and expert system techniques used together // Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval. 1992. 51–58.

*Boguraev Bran, Briscoe Ted.* Large lexicons for natural language processing: utilising the grammar coding system of LDOCE // Computational Linguistics. 1987. 13, 3-4. 203–218.

*Brants Thorsten.* TnT: a statistical part-of-speech tagger // Proceedings of the sixth conference on Applied natural language processing. 2000. 224–231.

*Braun Hardo, Grömling Dieter.* Research and technology buildings: A design manual. 2005.

*Caduff David, Timpf Sabine.* On the assessment of landmark salience for human navigation // Cognitive processing. 2008. 9, 4. 249–267.

*Calder Jonathan, Klein Ewan, Zeevat Henk.* Unification categorial grammar: A concise, extendable grammar for natural language processing // Proceedings of the 12th conference on Computational linguistics-Volume 1. 1988. 83–86.

*Ciepłuch Błażej, Jacob Ricky, Mooney Peter, Winstanley Adam C.* Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps // Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resuorces and Enviromental Sciences 20-23rd July 2010. 2010. 337.

*Day Tim, Muller Jan-Peter.* Digital elevation model production by stereo-matching spot image-pairs: a comparison of algorithms // Image and Vision Computing. 1989. 7, 2. 95–101.

*Dehbi Youness, Hadiji Fabian, Gröger Gerhard, Kersting Kristian, Plümer Lutz.* Statistical relational learning of grammar rules for 3D building reconstruction // Transactions in GIS. 2017. 21, 1. 134–150.

*Dehbi Youness, Plümer Lutz.* Learning grammar rules of building parts from precise models and noisy observations

// ISPRS journal of photogrammetry and remote sensing. 2011. 66, 2. 166–176.

*Doran Christy, Egedi Dania, Hockey Beth Ann, Srinivas Bangalore, Zaidel Martin.* XTAG system-a wide coverage grammar for English // arXiv preprint cmp-lg/9410010. 1994.

*Dore Conor, Murphy Maurice.* Integration of Historic Building Information Modeling (HBIM) and 3D GIS for recording and managing cultural heritage sites // 2012 18th International Conference on Virtual Systems and Multimedia. 2012. 369–376.

*Dym Clive L.* Expert systems: New approaches to computer-aided engineering // Engineering with computers. 1985. 1, 1. 9–25.

*Fan Hongchao, Zipf Alexander, Fu Qing.* Estimation of building types on OpenStreetMap based on urban morphology analysis // Connecting a Digital Europe Through Location and Place. 2014. 19–35.

*Fishman Mark C, Porter Jeffery A.* A new grammar for drug discovery // Nature. 2005. 437, 7058. 491–493.

*Gadde Raghudeep, Marlet Renaud, Paragios Nikos.* Learning grammars for architecture-specific facade parsing // International Journal of Computer Vision. 2016. 117, 3. 290–316.

*Ginsburg Seymour.* The Mathematical Theory of Context Free Languages.[Mit Fig.]. 1966.

*Gröger Gerhard, Plümer Lutz.* CityGML–Interoperable semantic 3D city models // ISPRS Journal of Photogrammetry and Remote Sensing. 2012. 71. 12–33.

*Haala Norbert, Kada Martin.* An update on automatic 3D building reconstruction // ISPRS Journal of Photogrammetry and Remote Sensing. 2010. 65, 6. 570–580.

*Habib Ayman F, Zhai Ruifang, Kim Changjae.* Generation of complex polyhedral building models by integrating stereo-aerial imagery and lidar data // Photogrammetric engineering & remote sensing. 2010. 76, 5. 609–623.

*Hain Walter.* Laboratories: A Briefing and Design Guide. 2003.

*Haklay Mordechai, Weber Patrick.* Openstreetmap: User-generated street maps // IEEE Pervasive Computing. 2008. 7, 4. 12–18.

*Hochmair Hartwig H, Zielstra Dennis, Neis Pascal, Hartwig PN, Hochmair H, Zielstra D.* Assessing the completeness of bicycle trails and designated lane features in OpenStreetMap for the United States and Europe // Transportation Research Board Annual Meeting. 2013.

*Hosseini Mirata, Tardy François, Lee Bruno.* Cooling and heating energy performance of a building with a variety of roof designs; the effects of future weather data in a cold climate // Journal of Building Engineering. 2018. 17. 107–114.

*Huang Yuandong, Hu Xiaonan, Zeng Ningbin.* Impact of wedge-shaped roofs on airflow and pollutant dispersion inside urban street canyons // Building and Environment. 2009. 44, 12. 2335–2347.

*Kang Suk-Ju, Trinh Hoang-Hon, Kim Dae-Nyeon, Jo Kang-Hyun.* Entrance detection of buildings using multiple cues // Asian Conference on Intelligent Information and Database Systems. 2010. 251–260.

*Kim Yoon.* Convolutional neural networks for sentence classification // arXiv preprint arXiv:1408.5882. 2014.

*Kipf Thomas N, Welling Max.* Semi-supervised classification with graph convolutional networks // arXiv preprint arXiv:1609.02907. 2016.

*Klonk Charlotte.* New laboratories: Historical and critical perspectives on contemporary developments. 2016.

*Kruger Abe, Seville Carl.* Green building: principles and practices in residential construction. 2012.

*Lindsay Robert K, Buchanan Bruce G, Feigenbaum Edward A, Lederberg Joshua.* DENDRAL: a case study of the first expert system for scientific hypothesis formation // Artificial intelligence. 1993. 61, 2. 209–261.

*Liu Xin, Wang Xiangyu, Wright Graeme, Cheng Jack CP, Li Xiao, Liu Rui.* A state-of-the-art review on the integration of Building Information Modeling (BIM) and Geographic Information System (GIS) // ISPRS International Journal of Geo-Information. 2017. 6, 2. 53.

*Lu Zhenyu, Im Jungho, Rhee Jinyoung, Hodgson Michael.* Building type classification using spatial and landscape attributes derived from LiDAR remote sensing data // Landscape and Urban Planning. 2014. 130. 134–148.

*Maliene Vida, Grigonis Vytautas, Palevičius Vytautas, Griffiths Sam.* Geographic information system: Old principles with new capabilities // Urban Design International. 2011. 16, 1. 1–6.

*Millonig Alexandra, Schechtner Katja.* Developing landmark-based pedestrian-navigation systems // IEEE Transactions on intelligent transportation systems. 2007. 8, 1. 43–49.

*Mohajeri Nahid, Assouline Dan, Guiboud Berenice, Bill Andreas, Gudmundsson Agust, Scartezzini Jean-Louis.* A city-scale roof shape classification using machine learning for solar energy applications // Renewable energy. 2018. 121. 81–93.

Detecting Health Problems Related to Addiction of Video Game Playing Using an Expert System. // . 2016.

*Özdemir E, Remondino F.* CLASSIFICATION OF AERIAL POINT CLOUDS WITH DEEP LEARNING. // International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences. 2019.

*Philipp Damian, Baier Patrick, Dibak Christoph, Dürr Frank, Rothermel Kurt, Becker Susanne, Peter Michael, Fritsch Dieter.* Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data // 2014 IEEE International Conference on Pervasive Computing and Communications (PerCom). 2014. 139–147.

*Riehle Timothy H, Lichter P, Giudice Nicholas A.* An indoor navigation system to support the visually impaired // 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2008. 4435–4438.

*Rottensteiner Franz, Sohn Gunho, Jung Jaewook, Gerke Markus, Baillard Caroline, Benitez Sebastien, Breitkopf Uwe.* The ISPRS benchmark on urban object classification and 3D building reconstruction // ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1. 2012. 1, 1. 293–298.

*Srivastava Shivangi, Vargas Muñoz John E, Lobry Sylvain, Tuia Devis.* Fine-grained landuse characterization using ground-based pictures: a deep learning solution based on globally available data // International Journal of Geographical Information Science. 2018. 1–20.

*Sui Daniel, Elwood Sarah, Goodchild Michael.* Crowdsourcing geographic knowledge: volunteered geographic information (VGI) in theory and practice. 2012.

*Tan Haocheng.* A brief history and technical review of the expert system research // IOP Conference Series: Materials Science and Engineering. 242, 1. 2017. 012111.

*Turker Mustafa, Koc-San Dilek.* Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, Hough transformation and perceptual grouping // International Journal of Applied Earth Observation and Geoinformation. 2015. 34. 58–69.

*Vakalopoulou Maria, Karantzalos Konstantinos, Komodakis Nikos, Paragios Nikos.* Building detection in very high resolution multispectral data with deep learning features // 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2015. 1873–1876.

*Vanegas Carlos A, Aliaga Daniel G, Beneš Bedřich.* Building reconstruction using manhattan-world grammars // 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010. 358–365.

*Volk Rebekka, Luu Thu Huong, Mueller-Roemer Johannes Sebastian, Sevilmis Neyir, Schultmann Frank.* Deconstruction project planning of existing buildings based on automated acquisition and reconstruction of building information // Automation in construction. 2018. 91. 226–245.

*Vosselman George, Dijkman Sander, others .* 3D building model reconstruction from point clouds and ground plans // International archives of photogrammetry remote sensing and spatial information sciences. 2001. 34, 3/W4. 37–44.

*Wang Ye-Yi, Mahajan Milind, Huang Xuedong.* A unified context-free grammar and n-gram model for spoken language processing // 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100). 3. 2000. 1639–1642.

*Watch Daniel D.* Building type basics for research laboratories. 5. 2002.

*Wichmann Andreas, Agoub Amgad, Kada Martin.* ROOFN3D: DEEP LEARNING TRAINING DATA FOR 3D

BUILDING RECONSTRUCTION. // International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences. 2018. 42, 2.

*Xu Yongyang, Wu Liang, Xie Zhong, Chen Zhanlong*. Building extraction in very high resolution remote sensing imagery using deep learning and guided filters // Remote Sensing. 2018. 10, 1. 144.

*Yang Xiaodong, Tian Yingli*. Robust door detection in unfamiliar environments by combining edge and corner features // 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops. 2010. 57–64.

*Yassin Mohamed F*. Impact of height and shape of building roof on air quality in urban street canyons // Atmospheric Environment. 2011. 45, 29. 5220–5229.

*Young Tom, Hazarika Devamanyu, Poria Soujanya, Cambria Erik*. Recent trends in deep learning based natural language processing // ieee Computational intelligenCe magazine. 2018. 13, 3. 55–75.

*Yue Kui, Krishnamurti Ramesh, Grobler Francois*. Estimating the interior layout of buildings using a shape grammar to capture building style // Journal of computing in civil engineering. 2012. 26, 1. 113–130.

*Zhang Jixian, Lin Xiangguo, Ning Xiaogang*. SVM-based classification of segmented airborne LiDAR point clouds in urban areas // Remote Sensing. 2013. 5, 8. 3749–3775.

# Part II

# Publications

# 5. Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints

## Article information

| | |
|---|---|
| Authors | Xuke Hu, Hongchao Fan, Alexey Noskov |
| Journal | International Journal of Digital Earth |
| History | Received: 21 April 2017<br>Revised: 13 July 2017<br>Accepted: 28 August 2017 |
| Reference | Hu, X., Fan, H. and Noskov, A.(2017). Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints. International Journal of Digital Earth, 11(10), pp.1039-1063. |

## Abstract

The building footprints on OpenStreetMap are used for 3D city reconstructions in more and more applications. Unfortunately, very few roof shape information for complex buildings is available on OpenStreetMap. Additional data requirements (e.g., aerial images or 3D point clouds) limit the usage and applicability of many roof reconstruction approaches. To mitigate this issue, we propose an approach to roof shape recommendations for complex buildings by exploring the inherited characteristics of building footprints: the disclosure of rectangles combinations in a partition of footprints and the symmetrical features of footprints. First, it decomposes a complex footprint into rectangles by using an advanced minimal non-overlapping cover algorithm, which can list all the partitions with the fewest number of rectangles. Second, a graph-based symmetry detection algorithm is proposed to identify all the symmetrical subclusters in partitions. Then, a set of selection rules are defined to rank partitions, and the best ones are chosen for roof shape recommendation. Finally, a set of combination rules and a symmetry rule are defined. It enables to evaluate the probability of a footprint being a certain combination of roof shapes and of each rectangle and L-unit being a certain roof shape. Experimental results show the growth of the probability of correctly recommending roof shapes for single rectangles and buildings from a prior probability of 17% to 45% and from a prior probability of 0.29% to 14.3%, removing 60% and 93% of the incorrect roof shape options, respectively.

**Keywords:** Roof recommendation;footprint decomposition;symmetry detection;symmetry rule; combination rules; intrinsic roof reconstruction; OpenStreetMap;

## 5.1 Introduction

OpenStreetMap (OSM) is one of the most successful and popular Volunteered Geographic Information projects (Haklay, Weber 2008). In its current state, nearly three million registered members contribute to the rapid growth of OSM. The OSM community has not only captured roads and paths, but also more and more POIs, urban facilities, land use areas, and buildings. The latter can be extracted and extruded into 3D (Goetz and Zipf 2012). Several projects generate and visualize 3D buildings from OSM: OSM-3D [1], OSM Buildings[2], and Glosm [3], etc. The main limitation of these projects is that the majority of buildings are modelled at lower levels of detail (Kolbe et al. 2005) comprising box models with flat roofs because of the lack of roof information.

3D roof reconstruction (Vosselman et al. 2001; Elberink and Vosselman 2009; Henn et al. 2013) is normally achieved by using LiDAR data and aerial image data. With enough sensor data covering a certain area, the roof shapes of buildings in this area can be accurately reconstructed. However, the main drawback of these approaches is their high computational load and high dependency on sensor measurements. Deficiency of measurements can lead to inaccurate and incomplete roof reconstruction. Moreover, a quite simple description about the roof shape is usually provided for OSM buildings. OSM allows the user to specify roof shapes of individual polygons as the flat, the gabled, the half-hipped, the hipped, and the pyramidal. Currently, it is almost impossible to tag roof information for buildings with complex roof. To solve this problem, we propose an approach to partitioning a complex footprint into rectangles and predicting the roof shape of rectangles. The approach is intrinsic since it discloses rectangles combinations and detects symmetrical features of footprint partitions without the requirement of additional data (e.g., sensor measurements). In most countries, footprints are readily available from cadastral data and OSM.

[1] www.osm-3d.org

[2] https://osmbuildings.org

[3] http://wiki.openstreetmap.org/wiki/Glosm

Our approach is motivated by two facts. One is the roof design principle presented in (Kruger, Seville 2012). It declares that roofs should be designed to minimize the collection of rainfall, snow, and leaves without creating leaks and narrow spaces. This principle can constrain the roof shapes of two adjacent rectangles when considering three roof primitives: gable, hipped or half-hipped. The other is that symmetry is a fundamental element in buildings, which is mainly attributed to culture, economical and aesthetic reasons. Figure 5.1 shows the examples of building roofs and the partitions of their associated footprints. The symmetry property in partitions is reflected in roof shapes. Thus, footprint symmetry information is a strong prior towards a robust interpretation of roof shapes.

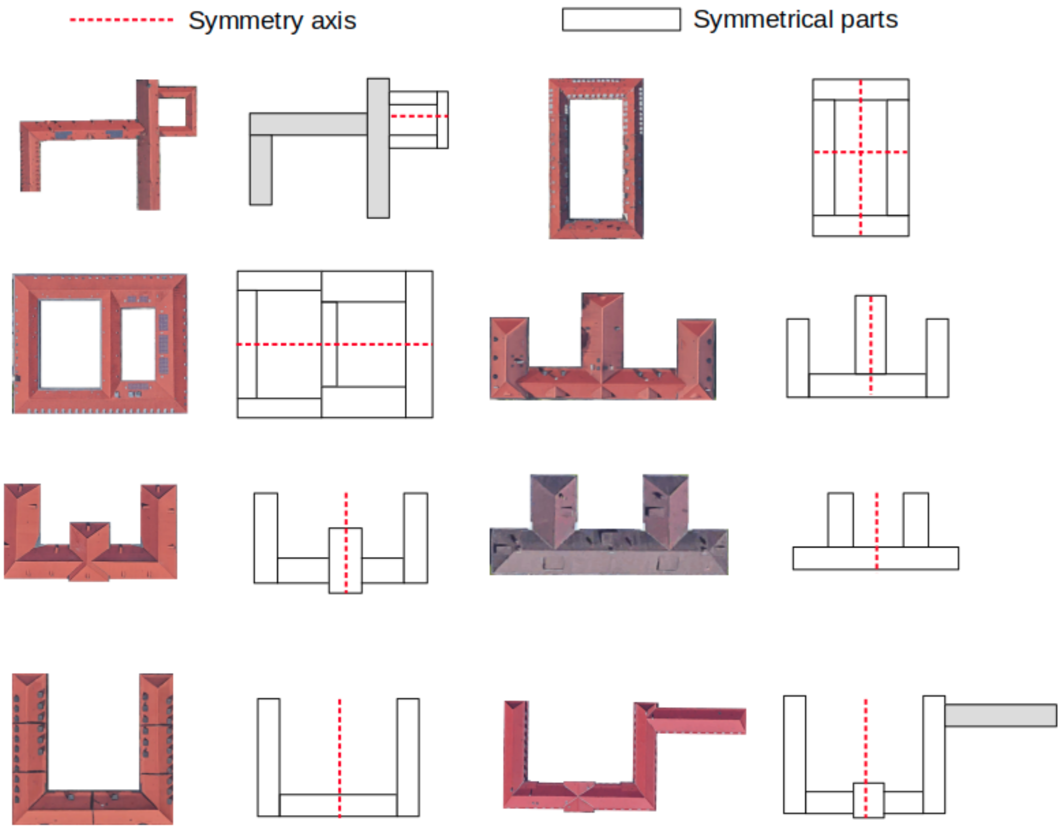


**Figure 5.1.** Symmetry feature in roof shapes and associated footprints.

In the proposed approach, a complex building footprint is first decomposed into several rectangles by using an enhanced minimal non-overlapping cover (MNC) algorithm (Ohtsuki 1982). It uses a recursive function to list all the partitions with the fewest number of rectangles. Second, a graph-based symmetry detection algorithm is introduced to the symmetrical rectangles recognition in partitions. It represents rectangles as one-dimensional line segments, enabling us to process a partition as an undirected graph. Our task is thus converted to find all the connected sub-graphs being symmetrical to a certain axis. Next, a set of selection rules is defined to evaluate the partitions and the one with the highest score is chosen for roof shape recommendation. Then, we define a set of combination rules and a symmetry rule to eliminate incorrect roof options. Finally, we calculate the probability of the single rectangle or L-unit and the footprint being certain roof shapes, which is followed by ranking recommended roof shapes.

In summary, our main contributions are as follows:

(1) We propose an algorithm of recommending roof shapes for a complex building by using a symmetry rule and a set of combination rules in footprints. It improves the traditional roof reconstruction approaches by ruling out most of the incorrect options. The approach does not require additional data. It allows improving the performance of the approaches relying on sensor measurements data and increasing their accuracy by applying as a preliminary step. In addition, it can enrich the OSM roof information by recommending volunteers several optimal options of roof shapes. To the best of our knowledge, this is the first work of predicting roof shapes of complex buildings by using only footprints without relying on any sensor measurements.

(2) We propose a novel graph-based symmetry detection algorithm, considering a partition as a graph and distinguishing all the axial-symmetry sub-graphs in complex footprints.

(3) We can achieve the best decomposition of complex footprints with regard to roof modelling by using an advanced MNC algorithm and a set of selection rules.

The rest of this paper is structured as follows: Section 5.2 describes the related works. Section 5.3 presents the concept of the proposed recommendation algorithm and then describes the details of its core components: footprint decomposition, symmetry detection, selection rules, combination rules, symmetry rule, and probability calculation. Section 5.4 presents experimental results. Section 5.5 discusses some issues involved in this paper. Section 5.6 concludes the paper and introduces the future works of this paper.

## 5.2 Related works

**3D roof reconstruction**: Roof reconstruction approaches can be divided into two main categories: data-driven (Kim and Shan 2011; Tarsha-Kurdi et al. 2007) and model-driven (Henn et al. 2013; Kada and McKinley 2009; Suveg and Vosselman 2004; Xiong et al. 2014). The former reconstructs 3D geometries directly from point clouds or features extracted from single or multi-source data. This process does not require complex building decomposition into primitives. Many data-driven algorithms (e.g., region growing, 3D Hough-transform, and Random sample consensus) are proposed to detect the roof planes. The model-driven algorithms require a model library of roof primitives. The model library allows users to establish correspondences between predefined 3D primitives and subsets of 3D point clouds. For complex roof models, building footprints are often decomposed into rectangles. Then, parametric building models in a predefined library are used as hypotheses and are subsequently verified by using information derived from images or point clouds. Both approaches rely highly on sensor measurements.

The dearth of measurements can result in inaccurate and incomplete roof reconstruction. Furthermore, although the model-driven approach uses the footprint of buildings, the purpose is to partition the footprints into rectangles that are associated with roof primitives. In contrast to the mentioned algorithms, our work introduces a novel intrinsic approach based on the analysis of inherent characteristics of footprints, such as rectangles combination way and symmetry features, to predict roof shapes.

In the work (Pita et al. 2008), the proportion of different roof shapes in several countries has been studied. For instance, in the considered countries, the approximate rate of gabled and hipped roofs is 72% and 20%, respectively. Obviously, this can be a reference for roof shapes prediction, which can achieve a relatively high accuracy if the test area has been surveyed in advance. The problem is the unavailability of the prior knowledge of the probability distribution of roof shapes in most areas.

**Symmetry detection**: Symmetry detection is an active topic in building modelling (Zhang et al. 2013). For instance, the work in (Musialski et al. 2009) tried to remove the unwanted content from facade images and replace the content with regular structures by using the symmetry property in facades. In this paper, we focus on the detection of the symmetric features of building footprints, which are normally represented as a two-dimensional polygon. The commonly used symmetry detection algorithm for polygons is based on string match approaches, which encode the polygon as a string, for instance, as a sequence of angles and edge lengths (Lladós et al. 1997; Wolter et al. 1985). Haunert (2012) presented a symmetry detection algorithm based on the string match approach for identifying axial symmetries and representative structures in footprints for map generation tasks. Instead of using the string match approach, the work in (Dehbi et al. 2016) utilized Support Vector Machines and a formal grammar to identify and model the symmetries and the hierarchical structures in building footprints, which can benefit facade and roof reconstruction. However, the problem of symmetry detection we are faced in this work is quite different. It requires detecting the symmetrical sub-clusters consisting of connected rectangles in the partition of footprints. Thus, none of the existing symmetry detection algorithms can be adopted for tackling the issue.

**Footprint decomposition**: Many works have been conducted in partitioning complex polygons into smaller parts for roof reconstruction. Suveg and Vosselman (2004) suggested extending the footprint lines that intersect in concave corners. This can generate multiple partitions because the line can be extended in both directions: vertical or horizontal. If two rectangles share a common edge, they will be merged. The Minimum Description Length (MDL) principle is used to rank partitioning results, giving higher priority to the partitioning schemes with a smaller number of rectangles. A disadvantage of this approach is that, in particular cases, several possible partitions with the least number of rectangles might be produced, which also include unreasonable partitions. Vallet et al. (2009) proposed decomposing a polygonal footprint into a set of non-overlapping polygonal sub footprints in order to address the inconsistency issue in the footprint. It mainly uses a splitting and merging operation, which is controlled by an energy function that incorporates the horizontal and vertical gradient of the digital elevation model. The works in (Vosselman et al. 2001; Kada and McKinley 2009) divided footprints into small cells that are mostly quadrangular sections. Then, the partitioning results are refined by merging or splitting the cells with LIDAR point clouds. The cell decomposition is an initial stage of a complete decomposition. To achieve a complete decomposition, the approach must leverage on sensor measurements. Gooding et al. (2015) tried to find a maximal area rectangle in a footprint. However, it could not achieve the best partition in many cases due to the production of many rectangle fragments, which are too small to be associated with a roof primitive.

## 5.3 Roof shape recommendation

As shown in Figure 5.2, the proposed method consists of six steps. In the first step, an advanced MNC algorithm is used to decompose a footprint into rectangles, which can list all the partitions with the least rectangles. The second step is detecting the symmetrical structures in the partitions by using a graph based symmetry detection algorithm, which represents the partition as a connected graph. Next, a set of rules are defined to evaluate the partitions, and the best ones are chosen for roof shape recommendation. Then, we define a set of combination rules and a symmetry rule. The combination rules state that each combination of rectangles can only correspond to a couple of combinations of roof primitives, while the symmetry rule defines that two symmetrical rectangles must have equal roof options. In this way, we can rule out many incorrect roof options that violate these rules. Now, we can calculate the probability of each rectangle being a certain roof shape and the probability of the footprint being a certain roof shape combination. Results with the highest probabilities can be recommended to OSM volunteers. In Figure 5.2, a workflow of the proposed approach is presented. The workflow (on the right) is illustrated by concrete examples (on the left). As shown in the figure, the footprint can be partitioned in two ways (partition 1 and partition 2) based on the advanced MNC algorithm. In the next step, a symmetry axis is identified, which is represented as a red dash line. The sub partition consisting of A and the left half of B and the other sub partition consisting of C and the right half of B are symmetrical. Because partition 2 contains two small fragments, it earns a lower score than partition 1, which is regarded as the best one. Then, the probability of each rectangle being a certain roof shape and the probability of the footprint being a certain combination of roof shapes can be calculated using the combination and symmetry rules. Finally, only two candidate combinations of roof shapes remain with each having a probability of 1/2.

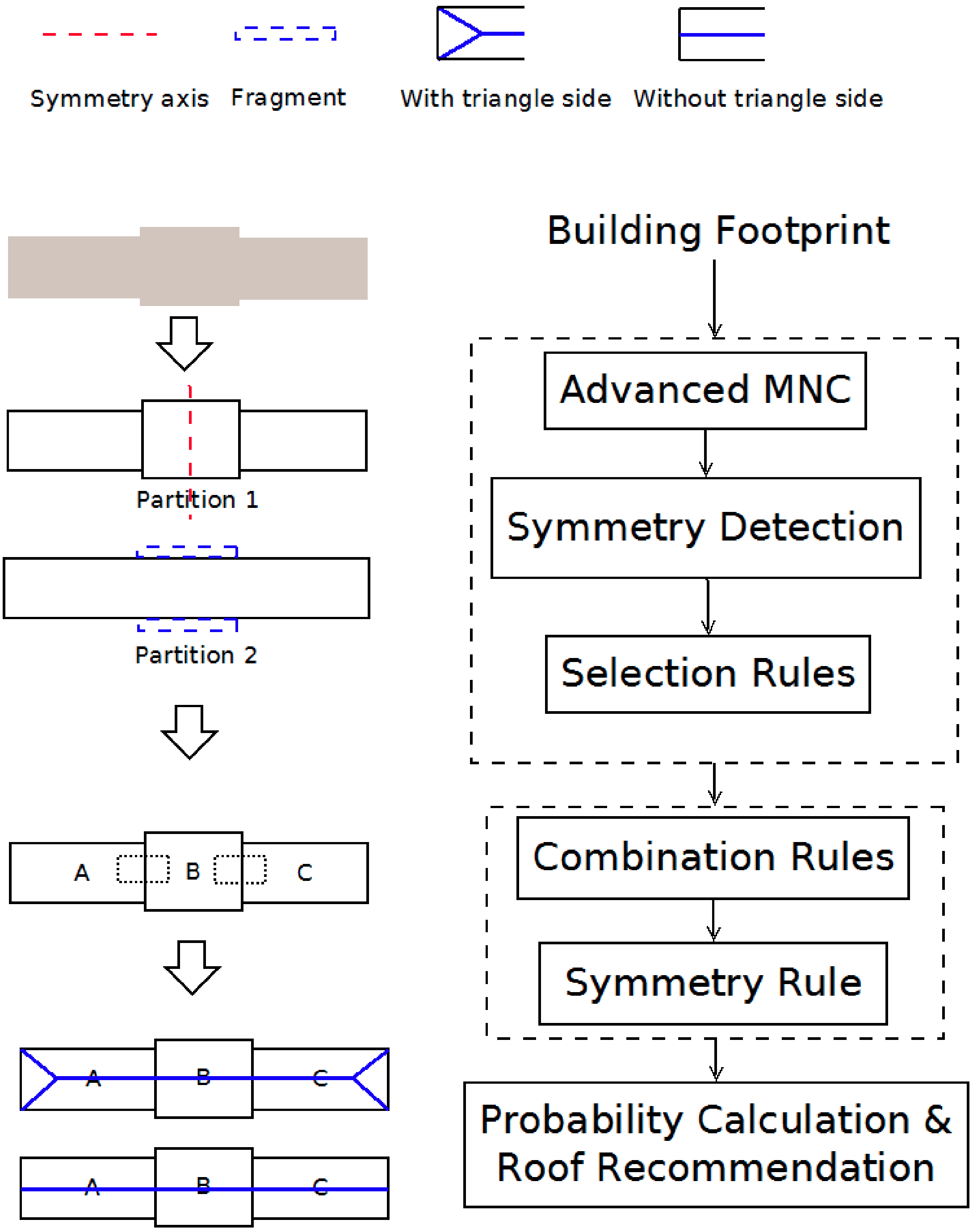


**Figure 5.2.** Workflow of roof shapes recommendation algorithm.

### 5.3.1 Footprint decomposition

We assume that building footprints consisting of rectangle structures are quite common in real environments. In order to achieve reasonable partitions from footprints, an advanced MNC partitioning algorithm is used. A reasonable partition means each rectangle in a partition correctly corresponds to a roof primitive of a true building roof. We assume the horizontal and vertical edges of rectilinear polygons are parallel to x-axis and y-axis, respectively. The conventional MNC algorithm (Wu and Sahni 1994) consists of the following six steps:

(1) Identify the concave vertexes of a rectilinear polygon.

(2) Generate vertical and horizontal chords by connecting two concave vertexes that have an equal $x$ and $y$ coordinate, respectively. Due to the noise in footprints, we define that two $x$ or $y$ coordinates are equal when their difference is below a threshold of 0.3 m. The concave vertex whose x or y coordinate is unequal to the $x$ or $y$ coordinate of any other concave vertexes is called free concave vertex.

(3) Obtain a maximum matching (MM) by using the Hungarian algorithm (Kuhn 1956). We treat vertical and horizontal chords as the left and right parts of a bipartite graph, respectively, where each chord is denoted by an endpoint. Then, we add an edge to connect two chords if they are intersected. Therefore, a bipartite graph can be denoted by $G = (H \cup V, U)$ , where H, V, and E represent horizontal chords, vertical chords, and the edges among them. A matching in a bipartite graph is a set of edges chosen in such a way that no two edges share an endpoint. A maximum matching is a matching of maximum number of edges.

(4) Find a maximum independent set (MIS) from the bipartite graph (Wu and Sahni 1994). An independent set in a graph is a set of endpoints chosen in such a way that no two endpoints are connected by an edge. A maximum independent set is an independent set of maximum number of endpoints.

(5) Draw the chords in MIS to partition the polygon.

(6) Draw a horizontal or vertical line segment from the free concave vertexes to the nearest edge.

The original MNC algorithm can only generate one partition. To overcome this issue, we slightly modify the conventional MNC algorithm (Wu and Sahni 1994) for listing all the partitions with the fewest number of rectangles through the following two steps:

(1) Produce all the maximum independent sets in step 4.

(2) Draw a line segment from the free concave vertexes in both vertical and horizontal directions in step 6.

For the first step, we modified the original MaxInd algorithm in (Wu and Sahni 1994). In the algorithm, only one of the two endpoints of an edge of the MM is added to an MIS, and based on the endpoint rest legal endpoints are found from the MM and added to the MIS. To list the entire MISs, we need to traversal both the branches of the two endpoints of an edge of the MM by using a recursive process. The pseudocode of the modified MaxInd algorithm is as follows:

**Algorithm 1** Advanced MaxInd

```
Input:
    $G = (H \cup V, E)$ // bipartite graph;
    $M$ // the maximum matching of the graph;
    $F$ // set of free endpoints relative to $M$;
    $s^{'}$ // current MIS;
    $s$ // current MIS array
Output
    $S$ // all the MISs $S = \{s^{'}\}$; such that $s^{'} \in H \cup V$;
procedure AllMaxInd
    while $(F \neq null)$ or $(M \neq null)$ do
        if $F \neq null$ then
            let $u \in F$; $F \leftarrow F - \{u\}$; $s^{'} \leftarrow s^{'} \cup \{u\}$
            $[G, M, F] = Process\_endpoint(u, G, M, F)$
        else
            let $(u, v) \in M$
            $M \leftarrow M - \{(u, v)\}$ ; $G \leftarrow G - \{(u, v)\}$
            $\bar{u} \leftarrow v$; $\bar{s} \leftarrow s^{'} \cup \{\bar{u}\}$ ; $s^{'} \leftarrow s^{'} \cup \{u\}$
            $\bar{M} \leftarrow M$; $\bar{G} \leftarrow G$; $\bar{F} \leftarrow F$
            $[G, M, F] = Process\_endpoint(u, G, M, F)$
            $[\bar{G}, \bar{M}, \bar{F}] = Process\_endpoint(\bar{u}, \bar{G}, \bar{M}, \bar{F})$
            $S =$ AllMaxInd $(\bar{G}, \bar{M}, \bar{F}, \bar{s}, S)$
    $S \leftarrow S \cup \{s^{'}\}$
    return $S$
```

An example in Figure 5.3 is taken to explain the advanced MNC algorithm. First, 15 concave vertexes were found (small blue circles). Vertexes $f, p, q, r$, and $s$ are five free concave vertexes. Lines are drawn from these vertexes, in either horizontal or vertical direction, producing new edges depicted by red dashed lines. Next, 13 chords that are denoted by black dotted lines are generated by connecting two concave vertexes with an equal $x$ or $y$ coordinate. Then, a bipartite graph is formed, consisting of six vertical chords and seven horizontal chords, as shown in Figure 5.4. The next step is to find a Maximum Matching from the bipartite graph. One of the MMs is $\{(a, i), (h, i)\}$,$\{(b, h), (a, b)\}$ , $\{(c, d), (c, o)\}$, $\{(d, n), (n, o)\}$, $\{(e, g), (e, m)\}$, and $\{(g, j), (j, m)\}$, which are denoted by red dot lines. The modified MaxInd algorithm is then

**Algorithm 2** Process Endpoint

```
procedure PROCESS_ENDPOINT(u,G,M,F)
    for all $(h, u) \in G$ do
        $G \leftarrow G - \{(h, u)\}$;
        if there is a $v$ such that $(h, v) \in M$ then
            $M \leftarrow M - (h, v); F \leftarrow F \cup \{v\}$
    return $G, M, F$
```

invoked, generating five MISs. The following step is drawing the chords in each MIS to form rectangles. Finally, 160 (5·25) partitions are generated in total. The first '5' represents the number of MISs and the second represents the number of free concave vertexes.

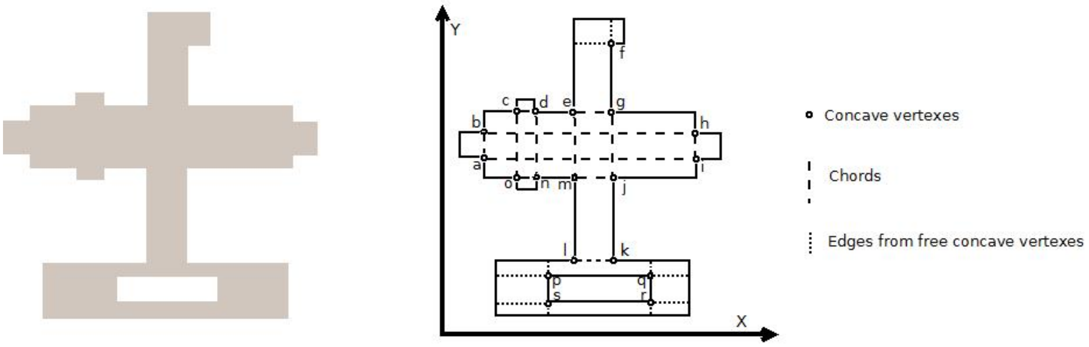


**Figure 5.3.** An example of MNC algorithm.

## 5.3.2 Symmetry detection in partitions

Symmetry is the fundamental feature in buildings and preferred by architectures. The symmetry feature in the partition of footprints is reflected in roofs, thus detecting the symmetry feature can benefit roof reconstruction. For simplification, we represent two-dimensional rectangles in partitions by one-dimensional line segments with length and width properties. The line segment is the central line of rectangles along their long side. The abstraction might lead to two disconnected segments although their corresponding rectangles are adjacent. To solve this problem, we extend original segments or add an extra segment for connecting two segments, as illustrated in Figure 5.5. Two adjacent and collinear segments with equal width value are treated as a single segment. In this way, we can consider a partition as an undirected graph, in which nodes represent line segments and links represent the connectivity among them.

Two sub-clusters are symmetrical only if they satisfy the following four conditions: 1) the two sub-clusters are both connected subgraphs; 2) they are connected; 3) the number of the elements in both sub-clusters exceeds two; 4) the segments in two sub-clusters are symmetrical to an axis. A symmetry axis is actually the perpendicular bisector of a segment since it is

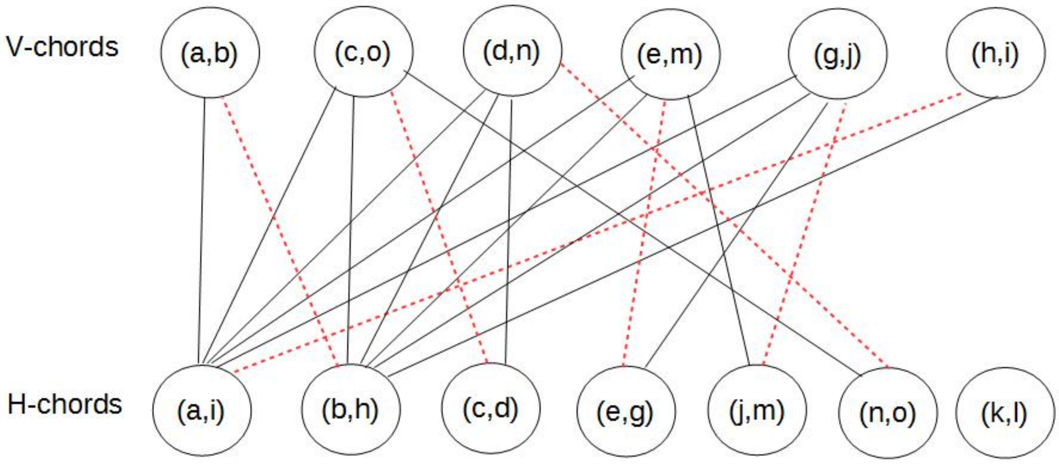


**Figure 5.4.** Bipartite graph consisting of horizontal and vertical chords.

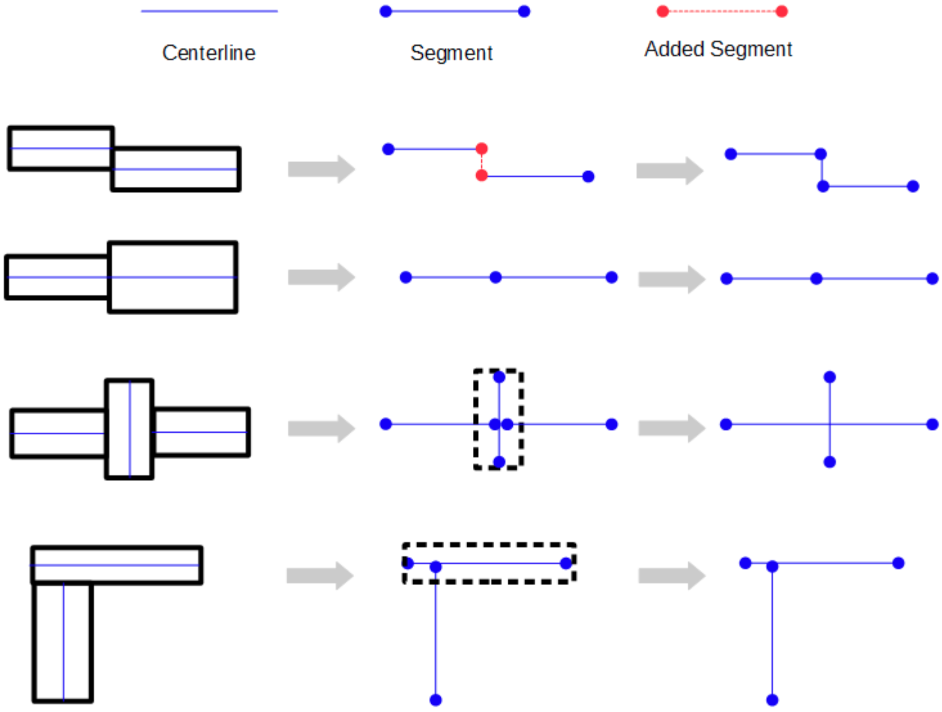


**Figure 5.5.** Representation of 2D rectangles by 1D line segments.

perpendicular with the segment and divides the segment into two equal sub-segments belonging to two symmetrical sub-clusters, respectively. We define that the upper half part and the lower half part of a segment $S$ as the first half part and the second half part, respectively, which are denoted by $s^1$ and $s^2$ when the segment is vertical. Similarly, when a segment is horizontal, the right half part and the left half part of the segment are treated as the first half part and second half part, respectively. The symmetry detection algorithm is described as follows.

Traversal each line segment $s$ in graph $G$ and treat the perpendicular bisector of $s$ as the symmetrical axis, which is denoted by $a$. $s$ is divided into two equal sub-segments, $s^1$ and $s^2$ representing the first segment in two symmetrical sub-clusters, respectively. Then, the segments that are connected to are checked to determine if they are connected to $s^1$ or $s^2$. The next step is to traverse all the sub-graphs that start from $s^1$ and simultaneously to find a symmetrical subgraph that starts from $s^2$ by invocating a recursion procedure FindMatch. The pseudocode of the symmetry detection algorithm is as follows:

**Algorithm 3** Detect Symmetry

```
    Input:
        $G$ // Connection graph
    Output
        $O$ // Contain all the symmetrical subgraphs in $G$
procedure SymmetryDetection(G)
    $V_s \leftarrow null$
    $O \leftarrow null$
    for any $s \in G$ do
        if $s \notin V_s$ then
            Treat the perpendicular bisector $a$ of $s$ as the symmetrical axis
            $M \leftarrow null$
            $N \leftarrow null$
            $M \leftarrow M \cup \{s^1\}$
            $N \leftarrow N \cup \{s^2\}$
            Assume $L$ is the array that contains segments connected to $s^2$
            for all $l \in L$ do
                $[M, N] = FindMatch(G, M, N, s^1, s^2, l, a, V_s)$
                if $M$ and $N$ include at least two elements then
                    $O \leftarrow O \cup \{M, N\}$
    return $O$
```

In the pseudocode, a variable $V_s$ is used to reduce the loop count of the first layer loop. It records the line segments that have been checked and divided into two symmetrical sub-segments. Segments divided into two symmetrical sub-segments and added into two symmetrical sub-graphs in previous loops do not require a repeated evaluation in the first layer of the loop. Figure 5.6 illustrates an example of the symmetry detection algorithm.

Because of the noise in footprints, for segment comparison we use a threshold of 0.3 m for width difference and a threshold of 0.5 m for length difference. Two segments with their width and length difference below 0.3 m and 0.5 m respectively are considered as equal in size.

**Algorithm 4** Find Matched Part

```
procedure FindMatch($G, M, N, p\_m, p\_n, l, a, V_s$)
    if $l$ can be divided into two parts being symmetrical to axis $a$ then
        Assume $p\_m$ connects $l^1$
        if there exist s a segment $s$ such that $s$ connects $p\_n$ and $l^2$ then
            $M = M \cup \{l^1\}; N = N \cup \{l^2\}; V_s = V_s \cup \{l\}$
            Assume $P$ is array that contains the segments connected to $l^1$
            for all $p \in P$ do
                $[M, N] = FindMatch(G, M, N, l^1, l^2, p, a, V_s)$
    else
        Assume $C$ is the array that contains the segments connected to $p\_n$
        for all $c \in C$ and $c \notin M$ and $c \notin N$ do
            if $c$ and $l$ are symmetrical to axis $a$ and $c.w = l.w$ then
                $M = M \cup \{l\}; N = N \cup \{c\}$
                Assume $P$ is array containing the segments connected to $l$;
                for all $p \in P$ do
                    $[M, N] = FindMatch(G, M, N, l, c, p, a, V_s)$
                break
    return $M, N$
```

### 5.3.3 Selection rules

After obtaining all the partitions, the next step is to select the most reasonable ones, in which each rectangle correctly corresponds to a roof part. We assume that each partition has an initial score of zero, and a set of rules are defined for changing the score. The partition with the highest score is treated as the best one. The defined rules are as follows.

(1) Each rectangle fragment causes the score to decrease by one. We define the rectangle with the length and width over three meters is valid for corresponding to a roof primitive. Otherwise, the rectangle is treated as a fragment that should be avoided as many as possible in partitions. Although fragments might exist in the best partitions, playing the role of balconies or entrance awnings, they are ignored in the subsequent processes. As shown in Figure 5.7, partitions 2 and 3 earn a lower score than partition 1 because of the existence of rectangle fragments in partitions 2 and 3. Thus, partition 1 is the best one correctly corresponding to the roof primitives of the true building roofs.

(2) Each combination of a pair of parallel rectangles causes the score to decrease by two. Two parallel rectangles mean their long sides are adjacent and collinear. This rule is derived from the roof design principle (Kruger, Seville 2012) that roofs should be designed to avoid

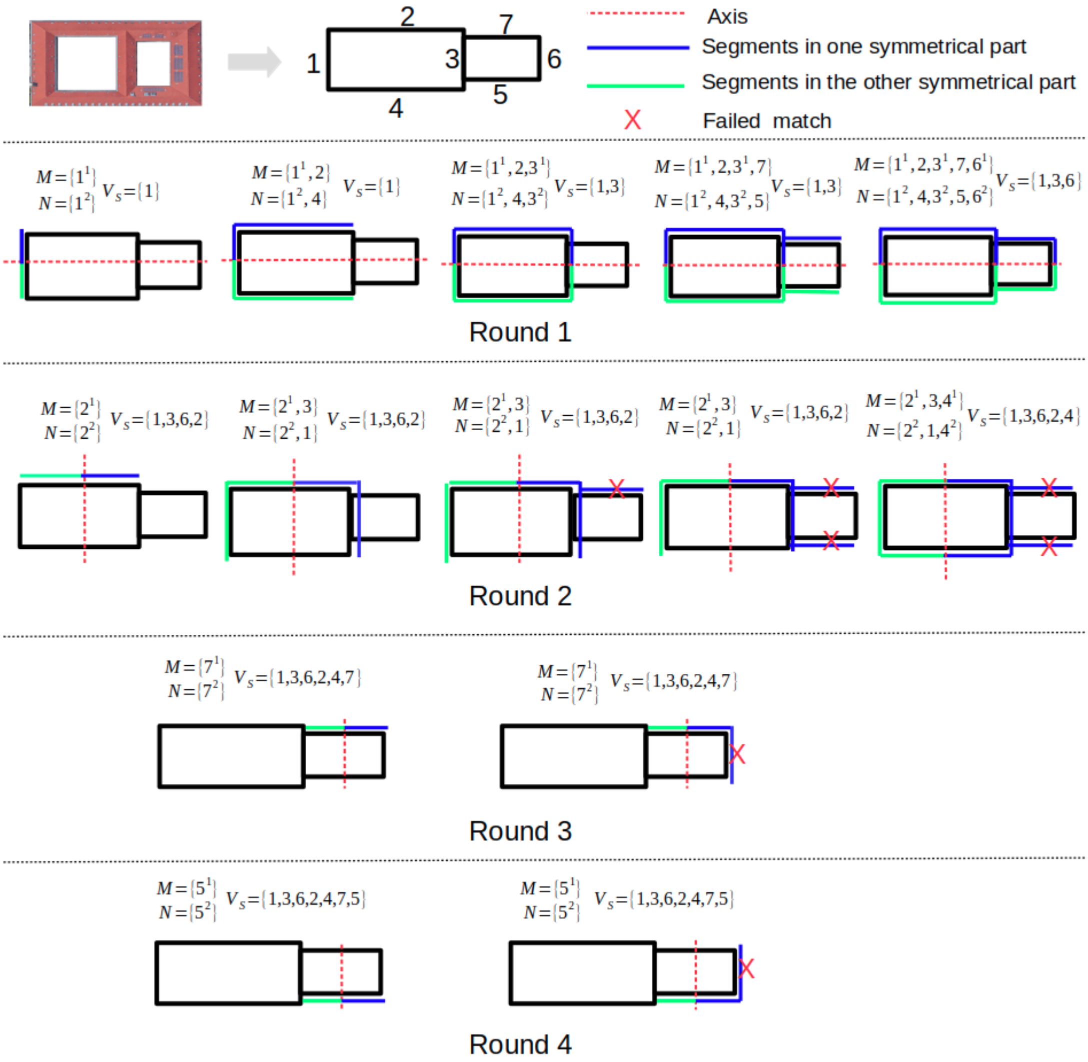

2
7
1
3
6
4
5
Axis
Segments in one symmetrical part
Segments in the other symmetrical part
X Failed match
$M=\{1^1\}$ $N=\{1^2\}$ $V_s=\{1\}$
$M=\{1^1,2\}$ $N=\{1^2,4\}$ $V_s=\{1\}$
$M=\{1^1,2,3^1\}$ $N=\{1^2,4,3^2\}$ $V_s=\{1,3\}$
$M=\{1^1,2,3^1,7\}$ $N=\{1^2,4,3^2,5\}$ $V_s=\{1,3\}$
$M=\{1^1,2,3^1,7,6^1\}$ $N=\{1^2,4,3^2,5,6^2\}$ $V_s=\{1,3,6\}$
Round 1
$M=\{2^1\}$ $N=\{2^2\}$ $V_s=\{1,3,6,2\}$
$M=\{2^1,3\}$ $N=\{2^2,1\}$ $V_s=\{1,3,6,2\}$
$M=\{2^1,3\}$ $N=\{2^2,1\}$ $V_s=\{1,3,6,2\}$
$M=\{2^1,3\}$ $N=\{2^2,1\}$ $V_s=\{1,3,6,2\}$
$M=\{2^1,3,4^1\}$ $N=\{2^2,1,4^2\}$ $V_s=\{1,3,6,2,4\}$
Round 2
$M=\{7^1\}$ $N=\{7^2\}$ $V_s=\{1,3,6,2,4,7\}$
$M=\{7^1\}$ $N=\{7^2\}$ $V_s=\{1,3,6,2,4,7\}$
Round 3
$M=\{5^1\}$ $N=\{5^2\}$ $V_s=\{1,3,6,2,4,7,5\}$
$M=\{5^1\}$ $N=\{5^2\}$ $V_s=\{1,3,6,2,4,7,5\}$
Round 4

**Figure 5.6.** An example of detecting symmetrical structures in a complex footprint.

narrow spaces that can carry snow and leaves. Two parallel gable, hipped or half-hipped roofs would definitely result in a narrow space. We believe the partition with a rectangle segment that plays the role of a balcony or an entrance awning is more common than the partition with a pair of parallel rectangles. Figure 5.8 shows a building footprint and its two partitions. Partition 1 consists of a pair of parallel rectangles, while partition 2 includes a fragment. In this case, we prefer partition 2 since it is more common in the real world than partition 1. Thus, we assign a lower score (e.g., -2) to partition 1 than partition 2 that earns a score of -1 according to selection rule 1.

(3) Each pair of symmetrical sub-clusters increases the score by 1. As we know, symmetry is the fundamental element in buildings and preferred by architectures. Therefore, a partition with symmetrical sub-clusters is more reasonable than the one without symmetrical sub-clusters. In Figure 5.9, a footprint corresponds to four partitions. Partitions 1 and 2 earn a higher score than

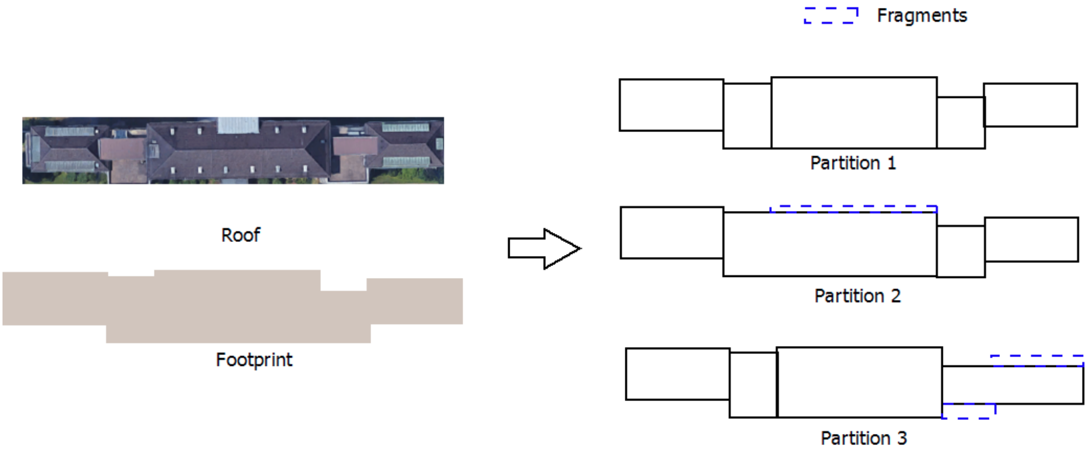


**Figure 5.7.** An example of selection rule based on rectangle fragments.

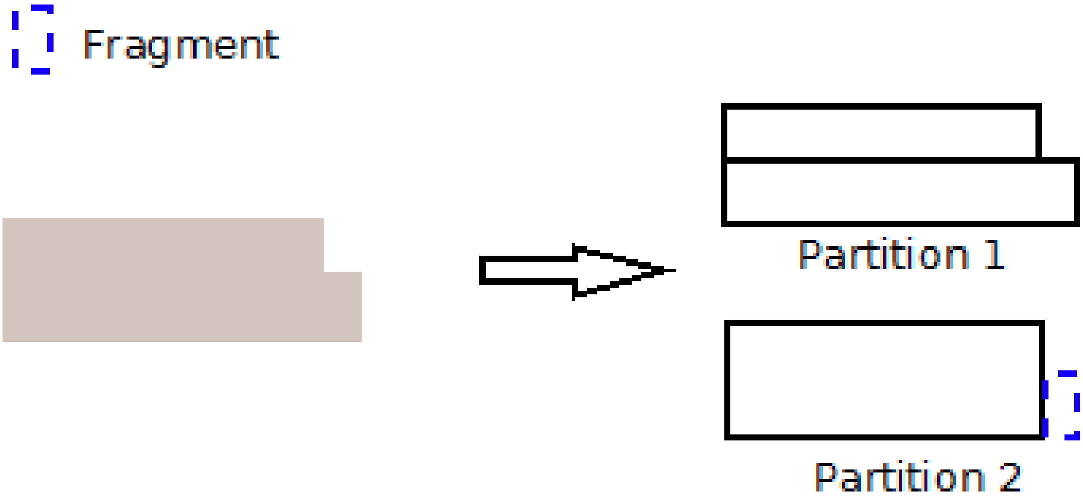


**Figure 5.8.** An example of selection rule based on parallel rectangles.

partitions 3 and 4 since the formers can be divided into two symmetrical sub-clusters.

Take the footprint in Figure 5.3 as an example to explain the selection rules. Figure 5.10 shows four partitions of the footprint. According to the selection rules, partitions 1, 2, 3, and 4 are assigned a score of -3, -13, -10, and -5, respectively. Thus, partition 1 is treated as the best.

### 5.3.4 Combination rules

Given the best partitions, we can recommend roof shapes for the rectangles in the partitions by identifying three combination ways of rectangles: one-line, L-shape, and T-shape. Each combination way of rectangles corresponds to several possible combinations of roof shapes. The gabled, the hipped, the half hipped, and the flat roofs are the four most common roof types. We consider the former three roof primitives in this paper since the flat roof is very easy to be identified with Lidar data or aerial images. The difference amongst these three primitives lies in the existence or nonexistence of the triangular side at the two short sides or ends of a

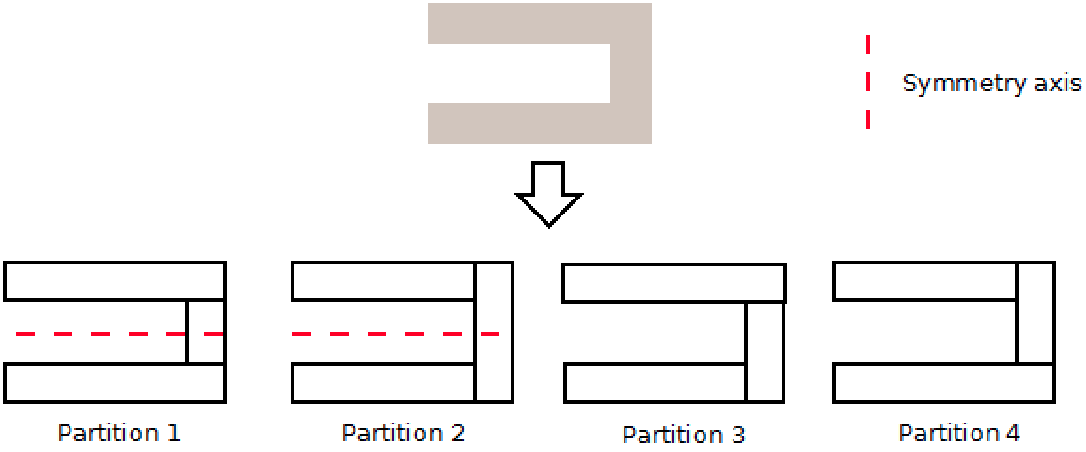


**Figure 5.9.** An example of selection rule based on symmetry feature.

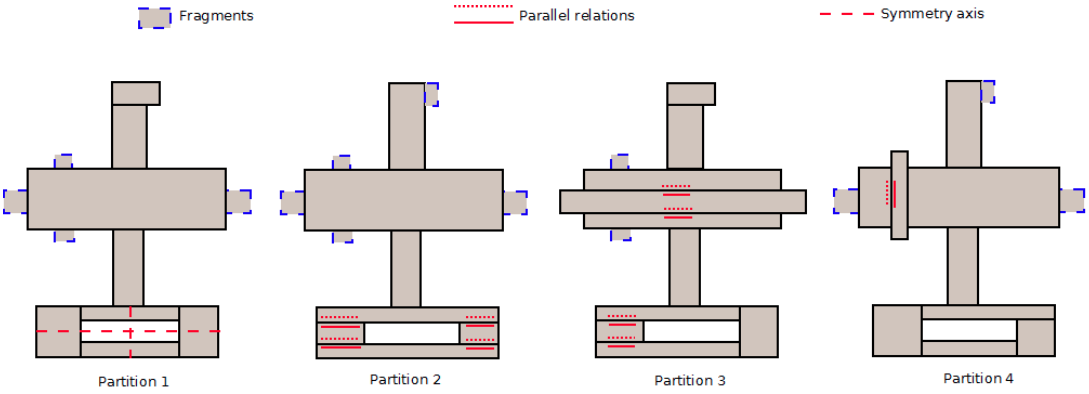


**Figure 5.10.** An example of scoring partitions with selection rules.

rectangle. We use two Boolean variables to represent the option of the triangle side at the two ends of a rectangle, denoted by $R_1$ and $R_2$, respectively. $R_1$ denotes the option of the triangle side of the left and lower end of a rectangle when the long side of the rectangle is horizontal and vertical, respectively, as shown in Figure 5.11. $R_2$ denotes the option of the triangle side of the right and upper end of a rectangle when the long side of the rectangle is horizontal and vertical, respectively. $R = 1$ means the end has two options: with and without a triangle side. $R = 0$ means the end has only one option: without a triangle side. When two rectangles are connected in one of the combination ways, the $R$ value of the two rectangles will change. The combination rules are described as follows.

Combination 1 (one-line): One-line combination means the short sides of two rectangles are adjacent and collinear. $R_{a,1}$ and $R_{a,2}$ denote the two R values of rectangle A, while $R_{b,1}$ and $R_{b,2}$ denote the two R values of rectangle B. The $R$ value of the two adjacent ends of two rectangles A and B in this combination is set to zero. As shown in Figure 5.12, the blue line denotes the nonexistence of the triangular side. This rule is derived from the roof design principle (Kruger, Seville 2012)] that roofs should be designed to avoid narrow spaces. If the right end of A or the

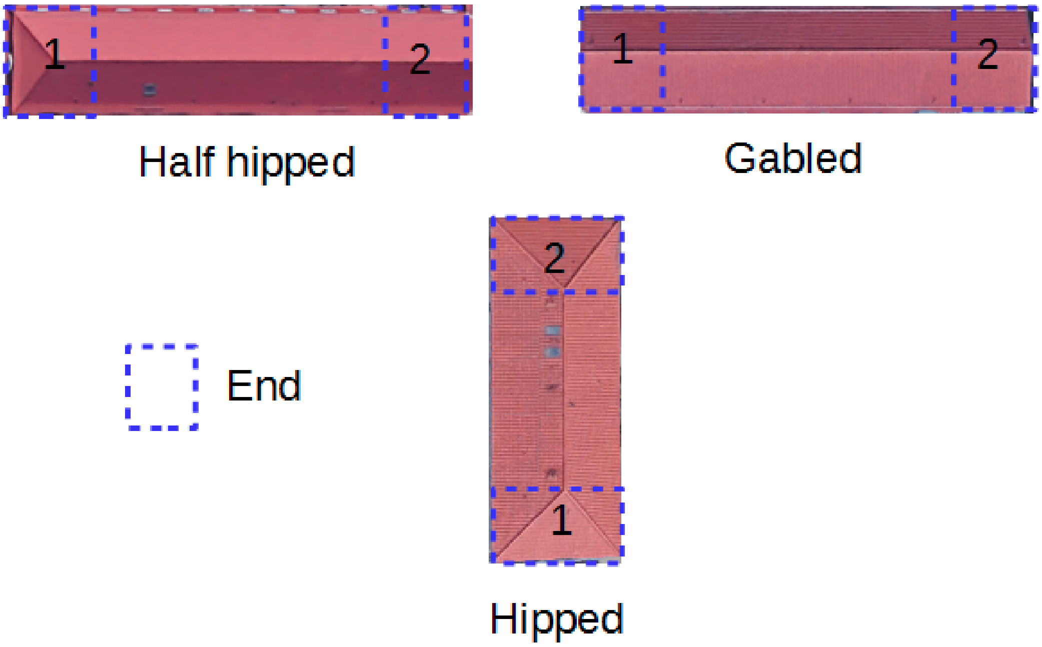


**Figure 5.11.** Difference and numbering of two ends in three roof primitives.

left end of B has a triangle side, a narrow space is produced, as shown in Figure 5.13.

$$R_{a,2} \leftarrow 0, R_{b,1} \leftarrow 0 \tag{5.1}$$

Combination 2 (T-shape): The value of the upper end in B is set to zero, as shown in Figure 5.14. The principle behind this rule is same as that of I-shape combination rule. If the upper end of B has a triangular side, a narrow space is produced.

$$R_{b,2} \leftarrow 0 \tag{5.2}$$

Combination 3 (L-shape): Each L-shape can be divided into three sections, including two separated sections A and B, and a public section C that is called L-junction end, as shown in Figure 5.15. The L-shape combination only affects the roof shape of the L-junction end. Typically, an L-shape has two best partitions with the public section belonging to either separated section. The rectangle occupying the public section is tagged with +, such as $A^+$ and $B^+$. In each best partition, the L-junction end has two possible combinations of roof shapes: the shared combination, which is denoted by $\{A^{+1/2}, B^{+1/2}\}$, and T combination, as shown in Figure 5.16. Thus, in an L-shape with two best partitions, the probabilities of the L-junction end being the T combination in $\{A^+, B\}$, being the T combination in $\{A, B^+\}$, and being the shared combination are 1/4, 1/4, and 1/2, respectively.

However, there might exist only one best partition for an L-shape. For example, in Figure 5.17, the partition $\{A, B^+\}$ earns a score at -1 because A is a small fragment. Partition

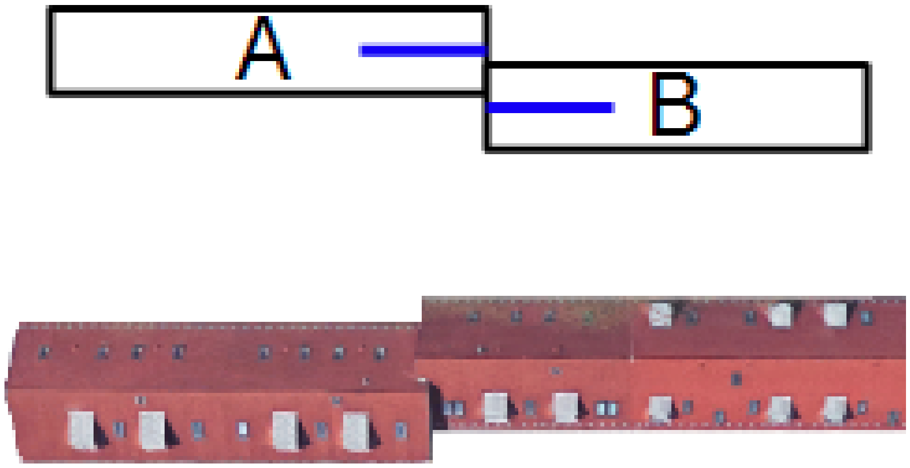


**Figure 5.12.** Change of R value in one-line combination.

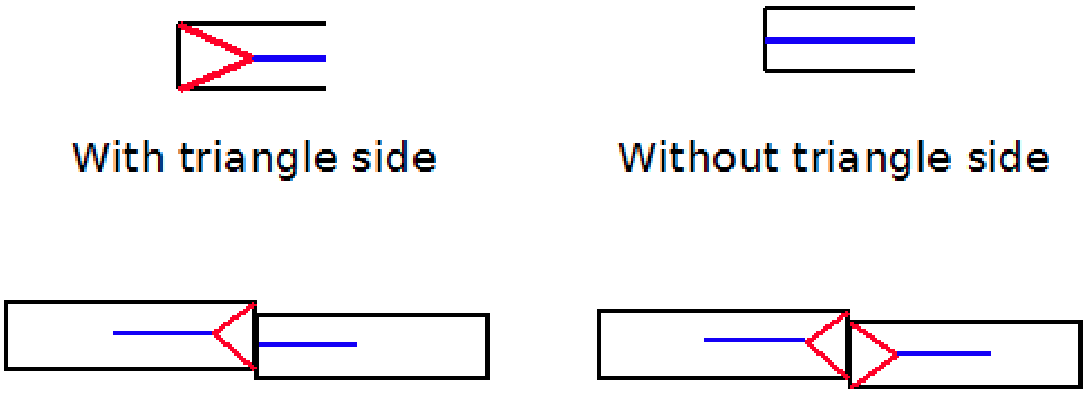


**Figure 5.13.** Unreasonable roof shapes in one-line combination.

$\{A^+, B\}$ earns a score of zero and thus is treated as the best. In this case, the L-junction end has only two options in the roof shape combination: 50% $\{A^+, B\}$ in T combination and 50% $\{A^{+1/2}, B^{+1/2}\}$ in shared combination.

If more than one combination ways of rectangles are identified on the same end of a rectangle, the roof shape combination satisfying the constraints of all the combination ways is used. As shown in Figure 5.18, the combination $\{A^+, B\}$ has two roof shape combinations: the shared combination and the T combination, while no triangle side exists in the two adjacent ends of the one-line combination $\{A^+, C\}$. To satisfy the constraints from both $\{A^+, B\}$ and $\{A^+, C\}$, only the T combination is assigned for the combination $\{A^+, B\}$.

The last step is analysing the existence or nonexistence of triangle sides at each end of a rectangle and calculate the probability of the rectangle being certain roof primitives. For instance, for a rectangle with $R_1 = 0$ and $R_2 = 1$, the probability of it being the gabled roof and the half hipped roof are both 1/2. This is because the first end of the rectangle has no triangle

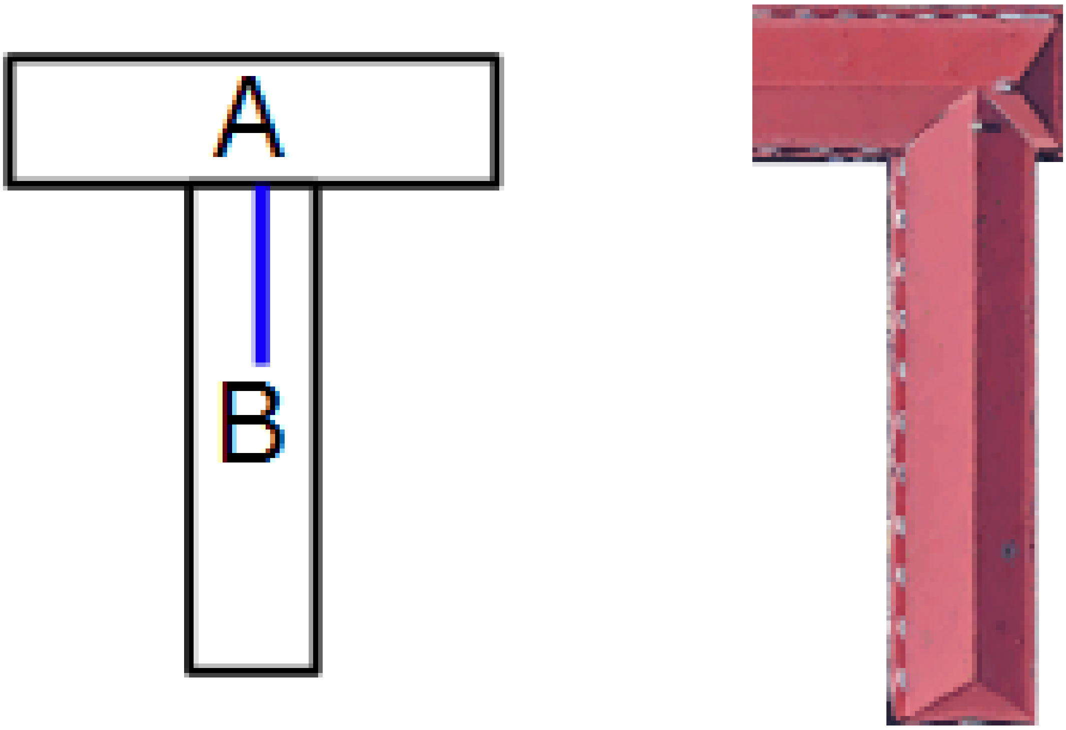


**Figure 5.14.** Change of R value in T-shape combination rule.

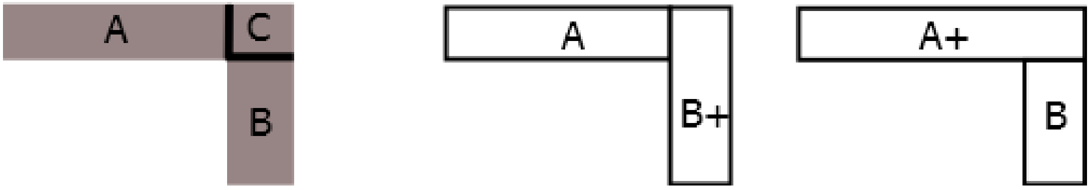


**Figure 5.15.** Two best partitions in an L-shape footprint.

side, while the other end has two options: with or without a triangle side.

### 5.3.5 Symmetry rules

Symmetrical features in the partition of footprints also reflect on roofs. Therefore, we rule that if two rectangles or two ends in a partition are symmetrical, they should have equal roof shapes. After identifying symmetrical parts in a partition, symmetrical ends can be found. Moreover, the symmetry relation can be transited amongst ends. For instance, if ends A and B are symmetrical, while ends A and C are symmetrical, then B and C are symmetrical. The symmetry rule is defined as follows: two symmetrical ends with equal $R$ value that is inferred through combination rules have equal roof shape options. For instance, if two symmetrical ends have equal R values, $R_{a,1} = 1$ and $R_{b,1} = 1$, the two ends have equal options of triangular

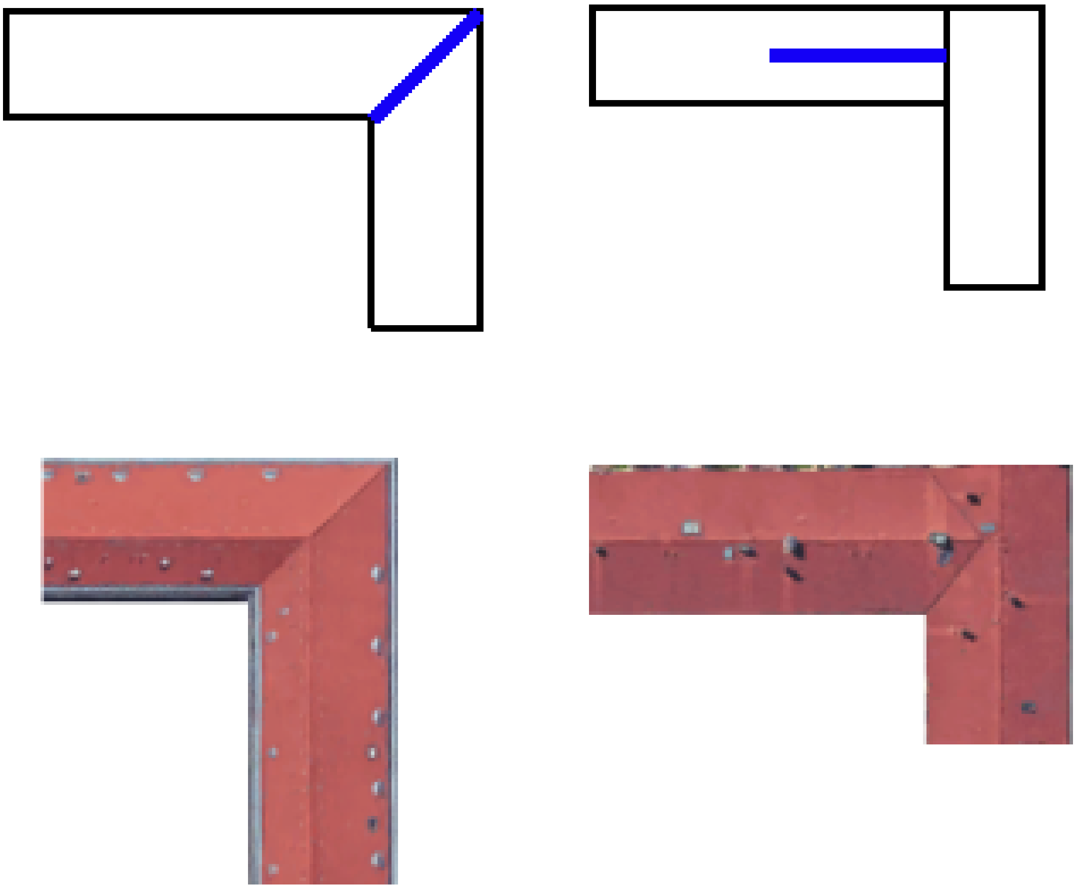


Figure 5.16. Two possible combination manners of roof shapes in L-shape.

sides: both with a triangle side or both without a triangular side. The rule also works for two symmetrical L-junction ends.

### 5.3.6 Probability calculation

With the combination rules and symmetry rule, many incorrect roof options can be ruled out. Then, we can calculate the probability of selecting the right one from the remaining roof options. We define the event of a rectangle end with or without a triangle side, and an L-junction end being a certain roof shape as an atomic event. From subsection 5.3.4 and 5.3.5, we can obtain the probability of atomic events. We define that an L-unit comprises adjacent L-junction ends and the rectangle ends that are adjacent to these L-junction ends. For instance, rectangles D, E, and F in Figure 5.19 forms an L-unit. Then, the event of each rectangle or L-unit being a certain roof primitive is defined as a single event, while the event of each footprint being a certain combination of roof primitives is defined as a joint event. Thus, the probability of a single event and of a joint event can be calculated by multiplying the probability of multiple atomic events because a rectangle, an L-unit, and a partition of footprints consist of rectangle ends and L-junction ends.

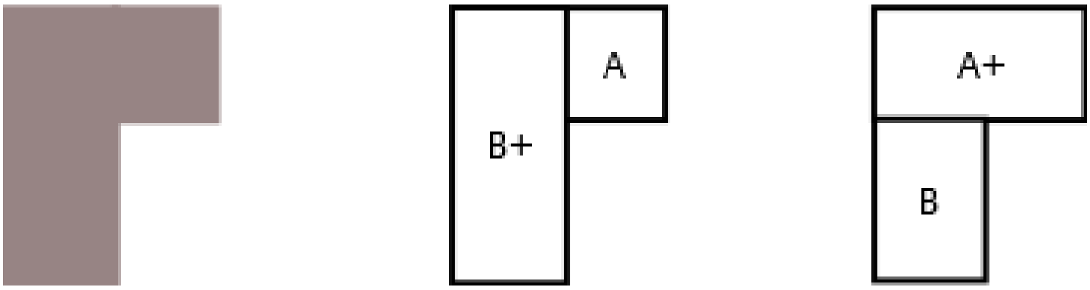


**Figure 5.17.** L-shape with only one best partition.

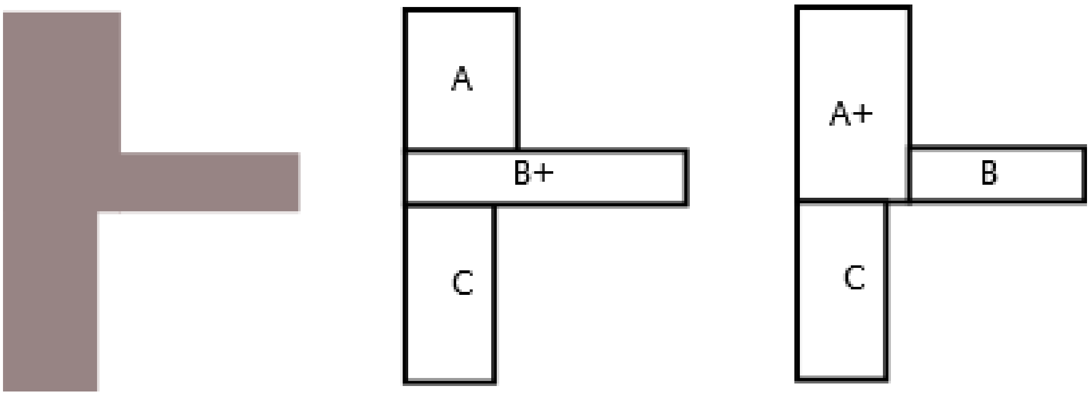


**Figure 5.18.** An end is imposed two combination ways of rectangles.

A building is taken as an example in Figure 5.19 to demonstrate the process of calculating the probability. The partition comprises a rectangle C, and two L-units consisting of rectangles $A$ and $B$, and of rectangles D, E, and F, respectively. The initial R value of all the rectangle ends in the partition equals 1, except for the three L-junction ends. Six combination ways are identified and the R values of all ends are updated as follows.

Finally, the lower end of rectangle $A$ has two options since it has a R value of $Ra, 1 = 1$. Thus, the probability of this end without a triangle side equals 1/2. From the result of step 1, we can obtain that the probability of L-junction end L-AB being shared roof $\left\{A^{+1/2}, B^{+1/2}\right\}$ equals 1/2. The right end of rectangle B has only one option since it has a R value of $R_{b,2} = 0$. Thus, the probability of this end without a triangle side equals 1. The upper and lower ends of rectangle C both have two options with $R_{c,1} = 1$ and $R_{c,2} = 1$. Thus, the probability of the two ends without the triangle side equals 1/2. The left end of rectangle F has only one option since it has a R value of $R_{f,1} = 0$. Thus, the probability of this end without a triangle side equals one. Similarly, the probability of the left end of rectangle D without a triangle side equals 1 since this end has a R value of $R_{d,1} = 0$. From the result of step 4, we can obtain the probability of L-junction end L-EF being shared roof $\left\{E^{+1/2}, F^{+1/2}\right\}$, equalling 1/2. Similarly, the probability of L-junction end L-DE being shared roof $\left\{D^{+1/2}, E^{+1/2}\right\}$ equals 1/2.

Note that the left end of rectangle $F$ and the left end of rectangle $D$ are symmetrical and have equal R values. In addition, two L-junction ends L-EF and L-DE are symmetrical and

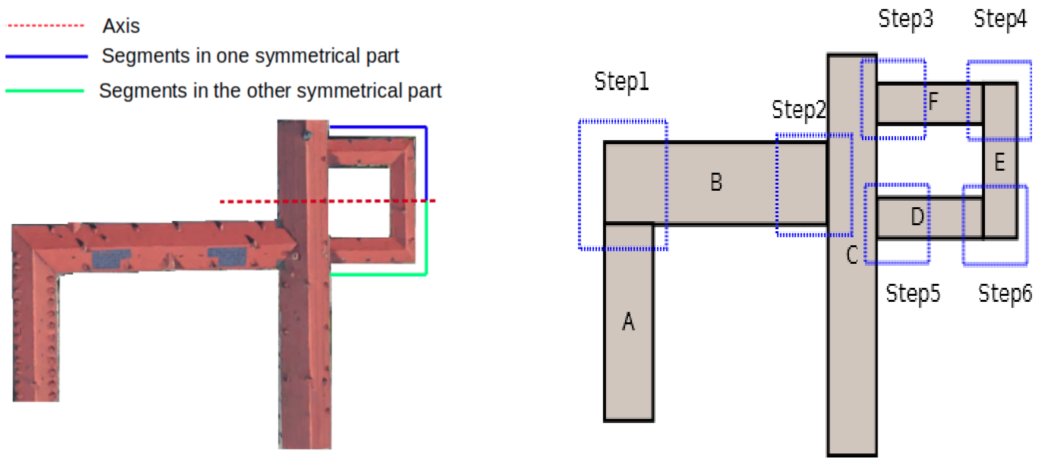

Axis
Segments in one symmetrical part
Segments in the other symmetrical part
Step1
Step2
Step3
Step4
Step5
Step6
A
B
C
D
E
F

**Figure 5.19.** A building with symmetrical parts and one of the best partitions.

have equal R values. According to the symmetry rule, the left end of rectangle D and F have equal roof option, and L-junction ends L-EF and L-DE have equal roof option. Thus, the joint probability of L-EF being $\left\{E^{+1/2}, F^{+1/2}\right\}$ and L-DE being $\left\{D^{+1/2}, E^{+1/2}\right\}$ equals 1/2 instead of 1/4. The probabilities of correctly recommending roof shapes for rectangle C, for the L-unit consisting of rectangle A and B, and for the L-unit consisting of rectangle $D$, $E$, and $F$ are 1/4, 1/4, and 1/2, respectively. Finally, the product of these three probabilities is the probability of correctly recommending the roof shape combination for the building, equalling 1/32.

## 5.4 Experiments

The proposed approach is implemented and tested using 30 complex building footprints on OSM. Results are evaluated by comparing the roof shapes on Google images, as shown in Figure 5.20. In order to demonstrate the benefits of the proposed recommendation algorithm, we compare the probability of rectangles or L-units being the right roof shapes and the probability of the footprint being the right roof shape combination before (prior probability) and after (estimated probability) using the recommendation algorithm. Prior probability refers to the probability of selecting the right one by using only the prior knowledge of roof primitives. It equals the reciprocal of the number of roof options. For example, a rectangle consists of two ends with each having two options: with or without a triangle side. A rectangle has four roof options in total, and the prior probability of correctly recommending a roof shape for the rectangle is thus 1/4. Initially, an L-junction end has five and nine roof options if its corresponding L shape has one and two best partitions, respectively. Thus, the prior probability of correctly selecting the roof for an L-junction end is 1/9 or 1/5. Two kinds of probabilities are evaluated: single event probability and joint probability. Single event probabilities refer to the probability of selecting right roof shapes for a rectangle or an L-unit. Joint probabilities refer to the probability of selecting the right combination of roof shapes for a footprint.

Note that buildings 13, 14, 15, and 16 consist of several single or complex buildings, while other buildings are an independent complex building. Our proposed rules are still applicable for those consisting of single and complex buildings. Their footprints are decomposed into multiple single and/or complex footprints on OSM with each corresponding to an independent building. In this situation, we just apply the MNC algorithm in these partitioned footprints.

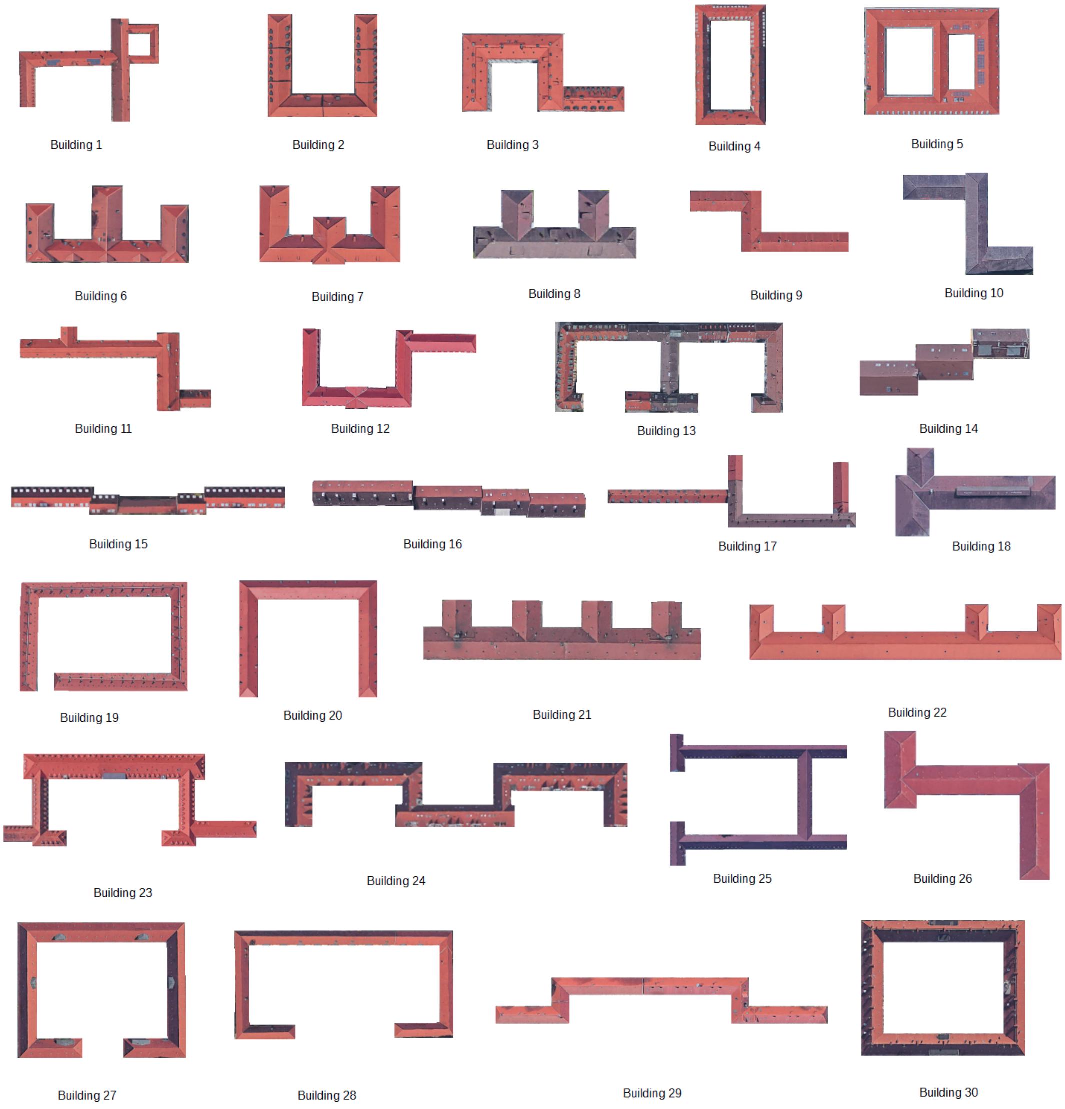


**Figure 5.20.** Test buildings from google images.

### 5.4.1 Comparison of joint probability

Estimated joint probabilities can be calculated according to our proposed probability calculation approach. For prior joint probabilities, we first record the number of possible roof shape options for each rectangle end and L-junction end, denoted by $m_i (i \in [1, n])$. In this equation, $n$ denotes the number of ends in a partition. Then, the total number of roof options of one partition can be calculated by using Equation 5.3. As we mentioned before, a footprint may has multiple best partitions, and the options of roof shape combinations derived from different partitions need to be integrated by counting repeated options only once. We denote integrated options by $O^{'}$, and the prior joint probability of correctly selecting roof shape combinations for a footprint is thus $1/O^{'}$.

$$O = \prod_{i=1}^{n} m_i \tag{5.3}$$

We use building 1 in Figure 5.20 as an example to explain the process of calculating prior joint probabilities. The L-unit consisting of rectangles D, E, and F has two best partitions according to the symmetry selection rule, corresponding to partition 1 and partition 2 in Figure 5.9. In each partition, the two L-junction ends have 25 (5·5) roof options in total. The number of integrated option is 49 (25+25-1) since the option of two L-junction ends being the shared combination appears totally twice in two partitions. The number of the roof options of the lower end of rectangle A, L-junction end L-AB , the right end of rectangle B, the lower and upper ends of rectangle C, the left end of rectangle D and E, and the two L-junctions ends L-DE and L-EF are 2, 9, 2, 2, 2, 2, 2, and 49, respectively. Then, the total number of roof shape combinations is 28224, and the prior joint probability is thus 1/28224.

The prior and estimated probability of joint events for other test buildings can be calculated in a similar way. Table 5.1 shows the results of prior and estimated probabilities. From the table, we can see a considerable improvement in the probability of correctly selecting the combination of roof shapes for footprints from 0.29% to 14.3% after using our proposed algorithm. The roof shape combinations that satisfy all the constraints of combination rules and symmetry rules are called candidate options. The roof shape combinations that are ruled out by the recommendation algorithm are called removed options. In Figure 5.21, each bar refers to a building. The blue section represents the proportion of removed options, while the sum of the red section and the green section represents the proportion of candidate options. The red section represents the proportion of the candidate options that have a lower estimated probability than that of the true combination of roof shapes. The green section represents the ranking of the true combination of roof shapes. From the table, we can see the amount of removed options is much greater than the amount of candidate options, occupying nearly 93 percent of the whole options. This proves the high efficiency of the proposed recommendation algorithm in ruling out incorrect roof shape combinations. Table 5.2 shows the ranking result of true roofs. A roof with a ranking result of 1% means the roof is ranked among the top 1%. The table demonstrates that the true roofs of the entire buildings are recommended and highly ranked among the entire roof options. Furthermore, the truth roofs of several buildings are still highly ranked among candidates. Although, in some cases, the truth roofs with a value of 100% are lowest ranked among candidates, most of the incorrect options have been removed, which can be observed from the last column of the table. Achieved results can be used to improve roof reconstruction since most of the incorrect options have been ruled out, and only very few measurements are needed to choose the true one from candidate options.

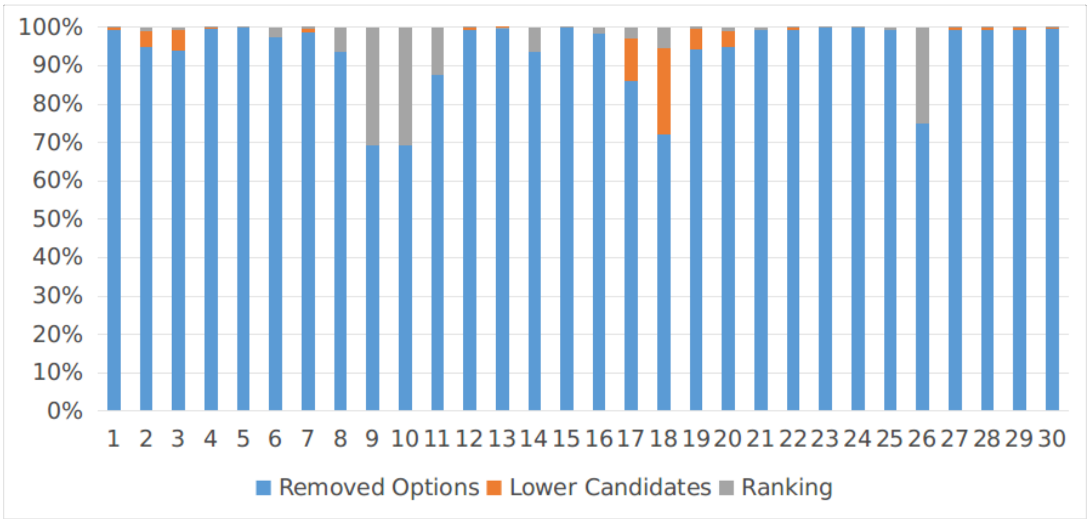


**Figure 5.21.** Comparison of removed and candidate options and ranking of true options for buildings.

Buildings 4, 8 and 15 are taken as examples to show the recommended roof shape combinations that are ranked highest among candidate options, as illustrated in Figure 5.22. For building 4, there exists one roof shape combination that is ranked within top 1 and four roof shape combinations that are ranked within top 5. A roof shape combination is ranked within top 5 means five roof shape combinations have an equal or higher estimated probability than this combination. The recommended roof shape combination ranked within top 1 is the true roofs of building 4. The true roof shape combinations of buildings 8 and 15 are ranked within top 4 and top 2, respectively.

### 5.4.2 Comparison of single event probability

In this subsection, we evaluate the probability of recommending right roof shapes for a single rectangle and an L-unit. We can calculate the prior and estimated probability of a single event in a similar way as a joint event. A single event consists of multiple atomic events and thus can be also seen as a ‘joint’ event. After obtaining the probability of recommending the right roof shape for each rectangle and L-unit, the mean probability is calculated according to Equation 5.4. In the equation, $p_i$ denotes probabilities, and $q$ denotes the number of rectangles and L-units in a building.

The mean probability of all buildings is shown in Table 5.3. Furthermore, a total mean probability can be calculated using a weighted average method according to Equation 5.5, where

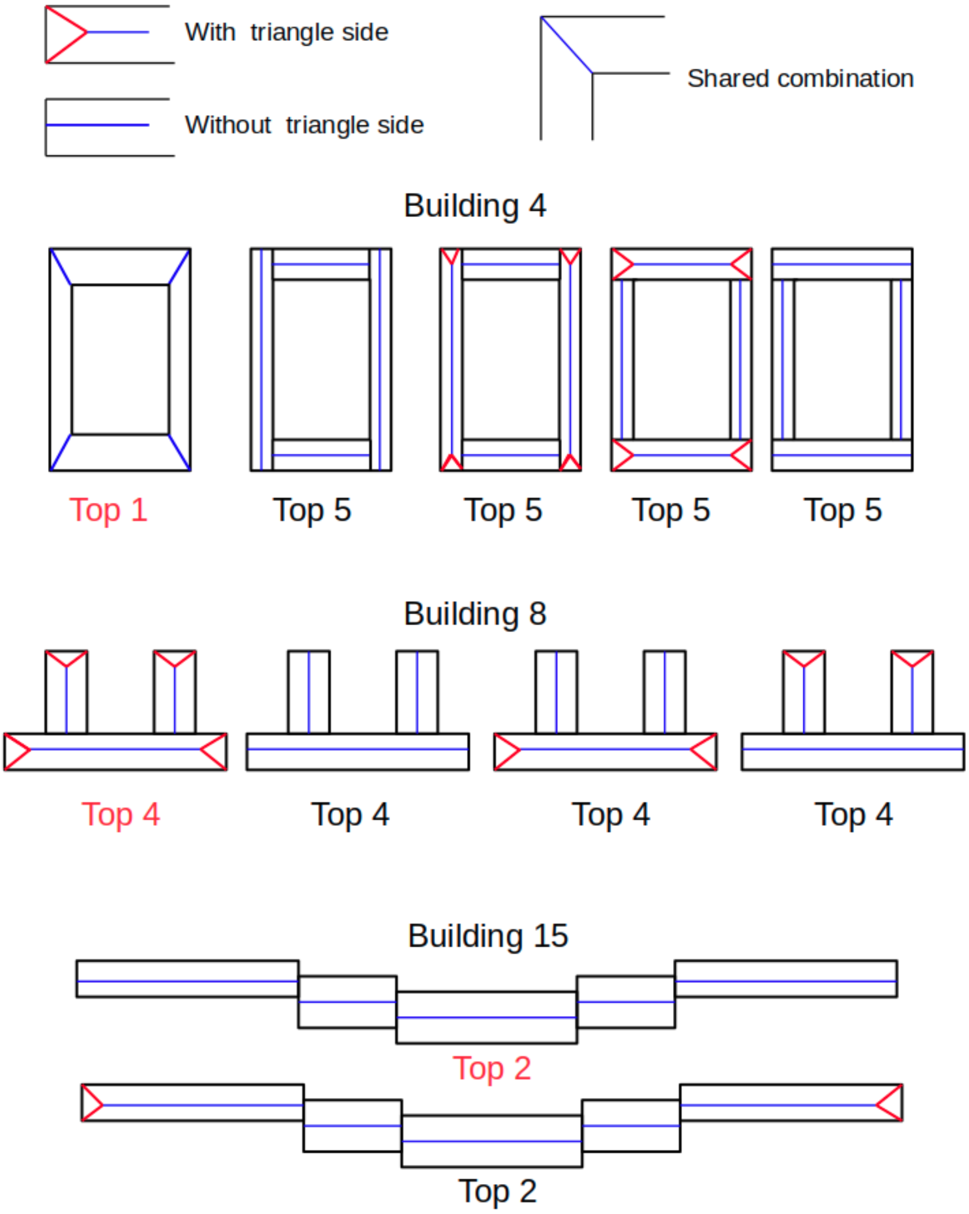


**Figure 5.22.** Examples of highly ranked roof shape combinations of buildings 4, 8, and 15.

$s$ denotes the number of buildings. The prior mean probability equals 17%, while the estimated mean probability reaches 45% after using the recommendation algorithm. The result shows a great improvement in the probability of choosing right roof shapes for single rectangles and L-units.

$$p^{'} = \frac{\sum_{i=1}^{q} p_i}{q} \tag{5.4}$$

$$\bar{p} = \frac{\sum_{j=1}^{s} p_j{}^{'} q_j}{\sum_{j=1}^{s} q_j} \tag{5.5}$$

Figure 5.23 shows the proportion of removed options and candidate options, as well as the ranking of the true roof shapes for a single rectangle or an L-unit. Each bar corresponds to a rectangle or an L-unit, and the blue, red, and green sections have the same meaning with that of Figure 5.22. From the figure, we can see removed options are much more than candidate options for most of the rectangles or L-units, occupying nearly 60 percent of the whole options. The results also show it is 21% and 77% the cases that the true roof shape of a rectangle or an L-unit is ranked within top 1 and top 2, respectively.

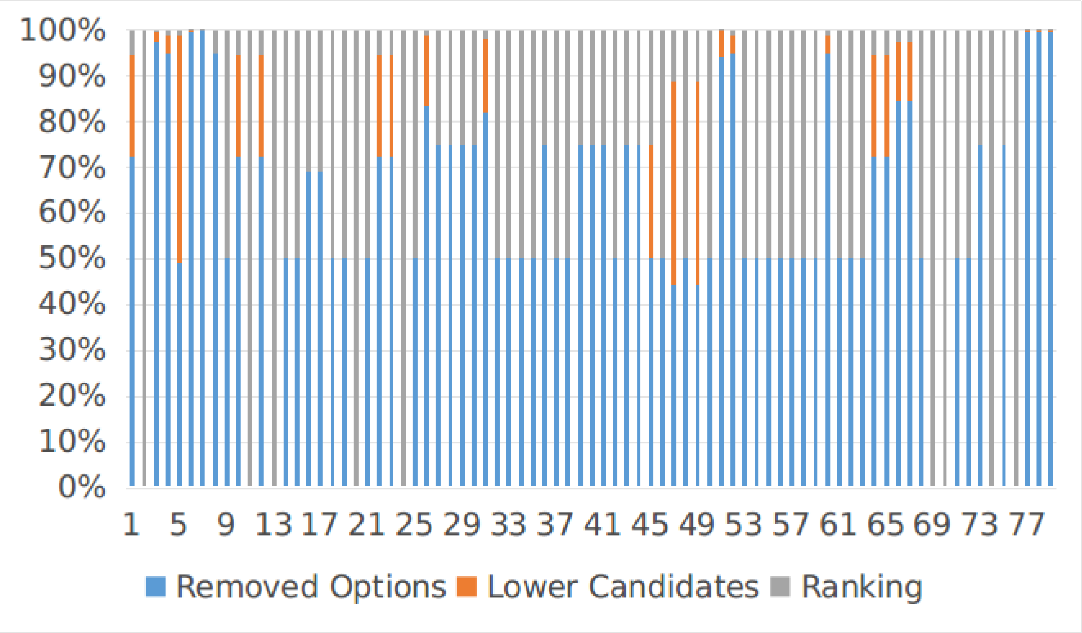


**Figure 5.23.** Comparison of removed and candidate options and ranking of true options for rectangles and L-units.

**Table 5.1.** Prior and estimated probability of joint events.

| | Prior probability | Estimated probability |
|---|---|---|
| Building 1 | 1/28224 | 1/32 |
| Building 2 | 1/196 | 1/4 |
| Building 3 | 1/980 | 1/32 |
| Building 4 | 1/1249 | 1/2 |
| Building 5 | 1/581042 | 1/8 |
| Building 6 | 1/784 | 1/32 |
| Building 7 | 1/3136 | 1/16 |
| Building 8 | 1/64 | 1/4 |
| Building 9 | 1/324 | 1/256 |
| Building 10 | 1/324 | 1/256 |
| Building 11 | 1/256 | 1/32 |
| Building 12 | 1/12544 | 1/32 |
| Building 13 | 1/73728000 | 1/256 |
| Building 14 | 1/64 | 1/4 |
| Building 15 | 1/1024 | 1/2 |
| Building 16 | 1/256 | 1/4 |
| Building 17 | 1/576 | 1/32 |
| Building 18 | 1/144 | 1/16 |
| Building 19 | 1/1764 | 1/100 |
| Building 20 | 1/196 | 1/4 |
| Building 21 | 1/1024 | 1/8 |
| Building 22 | 1/3136 | 1/20 |
| Building 23 | 1/50176 | 1/32 |
| Building 24 | 1/153664 | 1/16 |
| Building 25 | 1/1024 | 1/8 |
| Building 26 | 1/64 | 1/16 |
| Building 27 | 1/9604 | 1/8 |
| Building 28 | 1/9604 | 1/8 |
| Building 29 | 1/9604 | 1/8 |
| Building 30 | 1/1249 | 1/2 |

**Table 5.2.** Ranking of truth among entire options and candidates.

| | If true roofs are recommended | Ranking of truth among entire options | Ranking of truth among candidates | Proportion of removed options among entire options |
|---|---|---|---|---|
| Building 1 | Yes | 0,03% | 4,00% | 99,29% |
| Building 2 | Yes | 1,02% | 20,00% | 94,90% |
| Building 3 | Yes | 0,82% | 13,33% | 93,88% |
| Building 4 | Yes | 0,08% | 20,00% | 99,60% |
| Building 5 | Yes | 0,00% | 0,80% | 99,98% |
| Building 6 | Yes | 2,55% | 100,00% | 97,45% |
| Building 7 | Yes | 0,26% | 20,00% | 98,72% |
| Building 8 | Yes | 6,25% | 100,00% | 93,75% |
| Building 9 | Yes | 30,86% | 100,00% | 69,14% |
| Building 10 | Yes | 30,86% | 100,00% | 69,14% |
| Building 11 | Yes | 12,50% | 100,00% | 87,50% |
| Building 12 | Yes | 0,13% | 20,00% | 99,36% |
| Building 13 | Yes | 0,00% | 0,74% | 100,00% |
| Building 14 | Yes | 6,25% | 100,00% | 93,75% |
| Building 15 | Yes | 0,20% | 100,00% | 99,80% |
| Building 16 | Yes | 1,56% | 100,00% | 98,44% |
| Building 17 | Yes | 2,78% | 20,00% | 86,11% |
| Building 18 | Yes | 5,56% | 20,00% | 72,22% |
| Building 19 | Yes | 0,23% | 4,00% | 94,33% |
| Building 20 | Yes | 1,02% | 20,00% | 94,90% |
| Building 21 | Yes | 0,78% | 100,00% | 99,22% |
| Building 22 | Yes | 0,13% | 20,00% | 99,36% |
| Building 23 | Yes | 0,03% | 20,00% | 99,84% |
| Building 24 | Yes | 0,00% | 4,00% | 99,93% |
| Building 25 | Yes | 0,78% | 100,00% | 99,22% |
| Building 26 | Yes | 25,00% | 100,00% | 75,00% |
| Building 27 | Yes | 0,02% | 4,00% | 99,48% |
| Building 28 | Yes | 0,02% | 4,00% | 99,48% |
| Building 29 | Yes | 0,02% | 4,00% | 99,48% |
| Building 30 | Yes | 0,08% | 20,00% | 99,60% |

**Table 5.3.** Prior and estimated probability of single events.

| | Prior probability | Estimated probability | The number of rectangles and L-units |
|---|---|---|---|
| Building 1 | 499/5292 | 1/3 | 3 |
| Building 2 | 1/196 | 1/4 | 1 |
| Building 3 | 1/980 | 1/32 | 1 |
| Building 4 | 1/1249 | 1/2 | 1 |
| Building 5 | 1/581042 | 1/8 | 1 |
| Building 6 | 25/196 | 9/32 | 2 |
| Building 7 | 11/108 | 1/4 | 3 |
| Building 8 | 1/4 | 5/12 | 3 |
| Building 9 | 1/324 | 1/256 | 1 |
| Building 10 | 1/324 | 1/256 | 1 |
| Building 11 | 1/4 | 7/16 | 4 |
| Building 12 | 5/36 | 5/16 | 4 |
| Building 13 | 341/1800 | 21/32 | 8 |
| Building 14 | 1/4 | 2/3 | 3 |
| Building 15 | 1/4 | 4/5 | 5 |
| Building 16 | 1/4 | 3/4 | 4 |
| Building 17 | 19/108 | 3/8 | 3 |
| Building 18 | 5/36 | 5/16 | 2 |
| Building 19 | 1/1764 | 1/100 | 1 |
| Building 20 | 1/196 | 1/4 | 1 |
| Building 21 | 1/4 | 1/2 | 5 |
| Building 22 | 33/196 | 5/12 | 3 |
| Building 23 | 29/180 | 2/5 | 5 |
| Building 24 | 83/972 | 1/4 | 3 |
| Building 25 | 1/4 | 1/2 | 5 |
| Building 26 | 1/4 | 1/2 | 3 |
| Building 27 | 1/9604 | 1/8 | 1 |
| Building 28 | 1/9604 | 1/8 | 1 |
| Building 29 | 1/9604 | 1/8 | 1 |
| Building 30 | 1/1249 | 1/2 | 1 |

## 5.5 Discussions

**Theoretical limitations**: We make two main assumptions. One is rectilinear footprints. In this paper, we only choose the buildings with rectilinear footprints as test data. For the buildings whose footprints have non-right angles, an additional generalization of the footprint can be introduced in our future work as in (Vosselman et al. 2001; Kada 2007; Noskov and Doytsher 2013). The other is that buildings follow the roof design principle that a narrow space between roofs is avoided in buildings. Although some buildings might violate this principle, we believe the cases are rare. In our future work, we can resolve this issue by modelling it as a small probability event, which is integrated into our probability model. In this way, more roof shape options are produced, but the roof shapes that violate the design principle would earn a quite low probability.

**Empirical thresholds**: We use two groups of empirical thresholds in this work. The first is related to noisy footprints. More specifically, we use a distance threshold of 0.3 m to determine the equality of two $x$ or $y$ coordinates in the algorithm of decomposing footprints. In addition, we use a threshold of 0.3 m as width difference and a threshold of 0.5 m as length difference to determine the equality of two rectangles in the algorithm of symmetry detection. To make these thresholds applicable in a wider range of test data, they can be learned from annotated data. The second is related to the definition of fragments. We define that the rectangle with a width or length below three meters is treated as a fragment. Fragments might be the derivative of an incorrect partition or play the role of a balcony, vertical passage, and entrance awning. The value of this threshold is derived from two facts. One is the size of these three building parts. They normally have a width and length less than three meters. The other is the size of a garage, which has a minimum width at about three meters [4]. According to our prior knowledge about roofs, a garage is the smallest building unit that has a roof. Similarly, we can select a better threshold value by learning from annotated data.

**Applications of proposed approach**: Achieved results can be mainly applied in two aspects. First, they can benefit roof reconstruction that is implemented by using LiDAR data or aerial images. For example, the right one can be identified from candidate roof shapes by using very few LiDAR data or aerial images. In this way, computation loads and needed sensor data are dramatically reduced. This is especially well suited for the situation where sensor data is deficient. Second, it can facilitate the contribution activities of OMS volunteers when they are contributing the roof information of a complex building. For example, top ranked roof shapes for a single rectangle and for the whole building are a significant reference to volunteers when they are unclear with true roof shapes.

[4] http://moud.gov.in/upload/uploadfiles/files/Chap-4.pdf

## 5.6 Conclusions

In this paper, a rule-based approach to roof shape recommendation for complex buildings is proposed. First, complex footprints are decomposed into rectangles by using an advanced MNC algorithm. Next, we capture symmetrical rectangles in partitions. A set of selection rules are then defined to rank partitions and the one with the highest score is selected for roof recommendation. Finally, the probability of a rectangle or the whole building being a certain roof shape or roof shape combination is calculated by analysing the combination ways of rectangles and by using the symmetry rule. The evaluation of single and joint event probabilities shows an obvious improvement in the probability of correctly choosing roof shapes. This is because that most of the erroneous roof shapes have been ruled out by the recommendation algorithm. This can bring many benefits in both data-driven and model-driven roof reconstruction methods. More specifically, only a few measurements such as LiDAR point clouds or images are needed to identify the right one from remaining roof shapes. In addition, top-ranked roof shapes can be recommended to OSM volunteers as an important reference when they are unclear with the true roof shapes of a complex building.

Several tasks are scheduled for future works. First, we plan to take into account more roof primitives by leveraging on the size and length-to-width ratio information of rectangles. For instance, the rectangle with a small length-to-width ratio and a large area normally corresponds to a flat roof, while the footprint of a pyramidal roof has equal length and width. Second, in order to improve the accuracy of correctly selecting roof shapes, we plan to combine the estimated probability that is calculated by the recommendation algorithm and the probability distribution of roof primitives in a certain area, which can be learned from annotated roof data of neighbouring areas.

# Reference

Dehbi, Y., Gröger, G., and Plümer, L. (2016). Identification and modelling of translational and axial symmetries and their hierarchical structures in building footprints by formal grammars. *Transactions in GIS*, 20(5):645–663.

Elberink, S. O. and Vosselman, G. (2009). Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations. *Sensors*, 9(8):6101–6118.

Goetz, M. and Zipf, A. (2012). Openstreetmap in 3d–detailed insights on the current situation in germany. In *Proceedings of the AGILE 2012 International Conference on Geographic Information Science, Avignon, France*, volume 2427, page 2427.

Gooding, J., Crook, R., and Tomlin, A. S. (2015). Modelling of roof geometries from low-resolution lidar data for city-scale solar energy applications using a neighbouring buildings method. *Applied Energy*, 148:93–104.

Haklay, M. and Weber, P. (2008). Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18.

Haunert, J.-H. (2012). A symmetry detector for map generalization and urban-space analysis. *ISPRS journal of photogrammetry and remote sensing*, 74:66–77.

Henn, A., Gröger, G., Stroh, V., and Plümer, L. (2013). Model driven reconstruction of roofs from sparse lidar point clouds. *ISPRS Journal of photogrammetry and remote sensing*, 76:17–29.

Kada, M. (2007). Scale-dependent simplification of 3d building models based on cell decomposition and primitive instancing. In *International Conference on Spatial Information Theory*, pages 222–237. Springer.

Kada, M. and McKinley, L. (2009). 3d building reconstruction from lidar based on a cell decomposition approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3):W4.

Kim, K. and Shan, J. (2011). Building roof modeling from airborne laser scanning data based on level set approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4):484–497.

Kolbe, T. H., Gröger, G., and Plümer, L. (2005). Citygml: Interoperable access to 3d city models. In *Geo-information for disaster management*, pages 883–899. Springer.

Kruger, A. and Seville, C. (2012). *Green building: principles and practices in residential construction*. Cengage Learning.

Kuhn, H. W. (1956). Variants of the hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3(4):253–258.

Lladós, J., Bunke, H., and Martı, E. (1997). Finding rotational symmetries by cyclic string matching. *Pattern Recognition Letters*, 18(14):1435–1442.

Musialski, P., Wonka, P., Recheis, M., Maierhofer, S., and Purgathofer, W. (2009). Symmetry-based façade repair. In *VMV*, pages 3–10. Citeseer.

Noskov, A. and Doytsher, Y. (2013). Hierarchical quarters model approach toward 3d raster based generalization of urban environments. *International Journal on Advances in Software*, 6(3&4):343–353.

Ohtsuki, T. (1982). Minimum dissection of rectilinear regions. In *Proc. IEEE International Symposium on Circuits and Systems, Rome*, pages 1210–1213.

Pita, G., Pinelli, J., Subramanian, C., Gurley, K., and Hamid, S. (2008). Hurricane vulnerability of multi-story residential buildings in florida. *Proceedings ESREL 2008*.

Suveg, I. and Vosselman, G. (2004). Reconstruction of 3d building models from aerial images and maps. *ISPRS Journal of Photogrammetry and remote sensing*, 58(3-4):202–224.

Tarsha-Kurdi, F., Landes, T., and Grussenmeyer, P. (2007). Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from lidar data.

Vallet, B., Pierrot-Deseilligny, M., and Boldo, D. (2009). Building footprint database improvement for 3d reconstruction: a direction aware split and merge approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(Part 3):W4.

Vosselman, G., Dijkman, S., et al. (2001). 3d building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W4):37–44.

Wolter, J. D., Woo, T. C., and Volz, R. A. (1985). Optimal algorithms for symmetry detection in two and three dimensions. *The Visual Computer*, 1(1):37–48.

Wu, S.-Y. and Sahni, S. (1994). Fast algorithms to partition simple rectilinear polygons. *VLSI Design*, 1(3):193–215.

Xiong, B., Elberink, S. O., and Vosselman, G. (2014). A graph edit dictionary for correcting errors in roof topology graphs reconstructed from point clouds. *ISPRS Journal of photogrammetry and remote sensing*, 93:227–242.

Zhang, H., Xu, K., Jiang, W., Lin, J., Cohen-Or, D., and Chen, B. (2013). Layered analysis of irregular facades via symmetry maximization. *ACM Trans. Graph.*, 32(4):121–1.

# 6. Tagging the buildings’ main entrance based on OpenStreetMap and binary imbalanced learning

## Article information

| | |
|---|---|
| Authors | Xuke Hu, Alexey Noskov, Tessio Noskov, Hongchao Fan, Fuqiang Gu, Hao Li, Jianga Shang |
| Journal | International Journal of Geographical Information Science |
| History | Received: 25 November 2019 |
| Reference | Hu, X., Noskov, A., Novack, T.,Fan, H., Gu, F., Li, H., Shang, J. (2020). Tagging the buildings’ main entrance based on OpenStreetMap and binary imbalanced learning. International Journal of Geographical Information Science. (Submitted) |

## Abstract

The entrance of buildings is an important feature that connects their internal and external environments. Most frequently, automatic approaches for detecting building entrances are based on street-level images, which, however, are not widely available worldwide. To address this issue, we propose a more general approach for inferring the location of the main entrance of public buildings based only on OpenStreetMap data. In particular, we adopt three binary classification models: Weighted Random Forest, Balanced Random Forest, and SmoteBoost. The features considered in the classification are of two types: (1) intrinsic features derived from the footprint, such as the distance to the centroid of the footprint, and (2) extrinsic features derived from spatial contexts, such as the shortest path distance to the main roads. Extensive experiments have been conducted on 320 public buildings with an average perimeter of 350 meters. The experimental results showed that a mean linear distance error of 21 meters and a mean path distance error of 22 meters were achieved by using the Weighted Random Forest and Balanced Random Forest models, ruling out 90% of the incorrect locations of the main entrance at buildings. Our work finds relevance, for example, in saving pedestrians' way-finding efforts.

**Keywords:** Main entrance tagging; Random forest; OpenStreetMap;

## 6.1 Introduction

The entrance of public buildings plays a vital role in connecting outdoor and indoor spaces. Determining the location of the main entrance is essential in many location-based service (LBS) applications, such as way-finding since it is normally the end destination of outdoor way-finding (Zeng and Weber 2015). However, the entrance information is missing on current mainstream map providers, such as Bing Maps and Google Maps. This can lead to several issues (e.g., inaccurate navigation and misleading) when using these map services. For example, when following the planned route by map providers to a certain building, users are often guided to the wrong location, which is far away from the main entrance. Consequently, they need to spend even more efforts to find the main entrance by themselves. Times way-finding efforts can be saved and shorter and simpler routes can be derived if the main entrance of buildings is a mapped feature. This is an unpleasant experience especially for the people with mobility constraints because public buildings are normally complex and of large proportions. Figure 6.1 shows two real examples when using Google Maps to plan a route to a certain building. Realizing the importance of mapping the building entrance, the OpenStreetMap (OSM) contributors have created a tag to represent the main entrance as a node with the OSM key 'entrance' and value 'main' (Goetz and Zipf 2011).

However, to the present date, only a small proportion of buildings on OSM have an entrance tag feature. For instance, in the London area, there are only about 60 buildings that are tagged with the main entrance. This is because it is difficult for volunteers to contribute with the entrance of the building in comparison to other features, such as the buildings' footprints and the venues' names, which can be obtained from personal experience, public information, and Bing satellite imagery. Only the volunteers who are familiar with the building would mark the main entrance on OSM. To overcome this challenge, some automatic solutions have been proposed to identify the entrance of buildings from street-level images (Kang et al. 2010; Liu et al. 2014; 2017) and remarkable tagging results have been achieved. However, the data they leverage on limit the applicability of their approach, as street-level images that cover a wide range of areas are not guaranteed to be available even from Google Street View, which is the largest provider of street view images to date, specially outside developed countries. Furthermore, the entrance of many buildings can not be directly observed from streets due to the existence of obstacles or the entrance does not face any street.

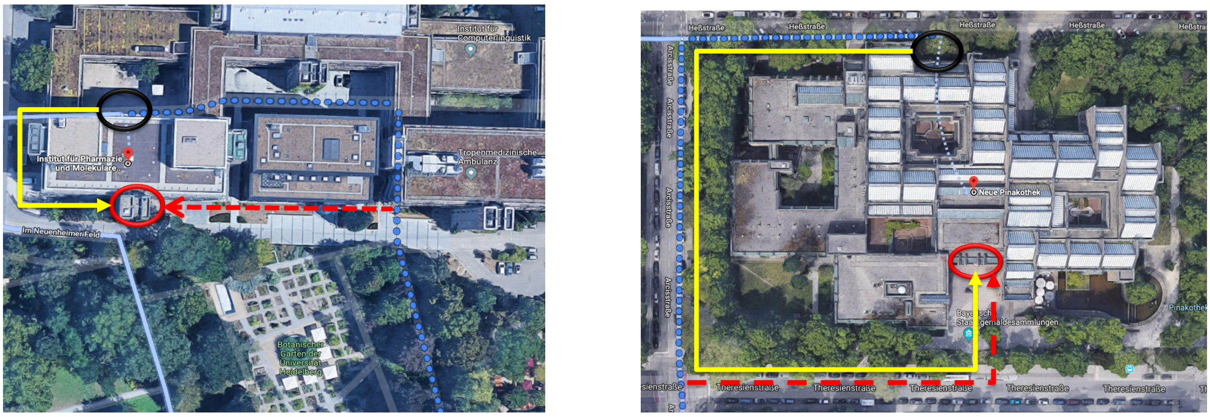
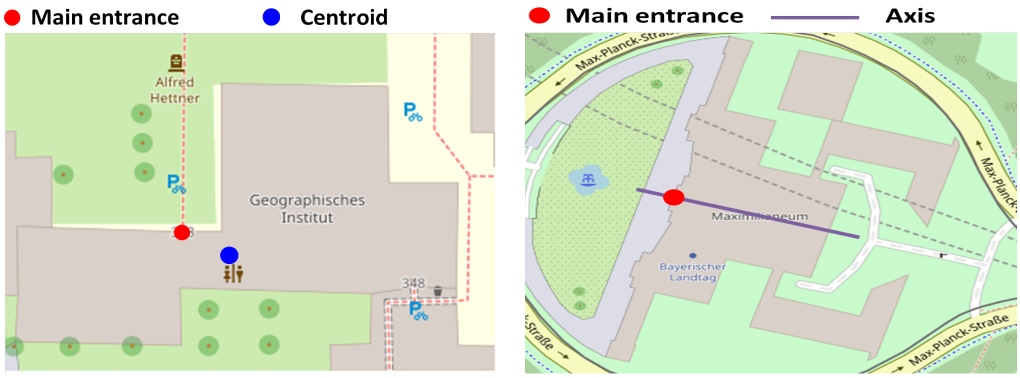

Institut für Computerlinguistik
Institut für Pharmazie und Molekulare...
Tropenmedizinische Ambulanz
Im Neuenheimer Feld
Botanischer Garten der Universität Heidelberg
Heßstraße
Arcisstraße
Neue Pinakothek
Bayerische Staatsgemäldesammlungen
Pinakothek
Theresienstraße

**Figure 6.1.** Inaccurate and misleading navigation by Google Maps due to missing of entrance information. Location tagged by black and red circle are planned target point by Google Maps and true main entrance, respectively. The blue dotted line represents the planned path by Google Map. The yellow line shows the extra path taken to find the true entrance to the planned target location. The red dashed line denotes the shortcut that is not found by Google Maps.

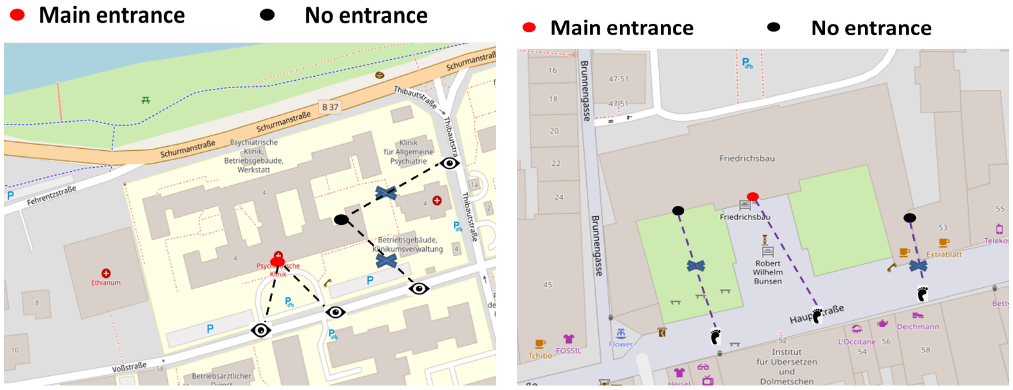
To mitigate this gap, a more general and applicable main entrance tagging approach for public buildings (e.g., hospital, office building, and museum) is proposed by leveraging OSM, which provides high-quality geography information in many regions, such as Europe and the United States (Hochmair et al. 2013) and is freely accessible. The reason that we focus only on public buildings rather than private buildings such as residential house, is that their shape is complex and large-scaled and they are the most frequent route destinations. Therefore, to guide users to find the entrance of public buildings is of larger public interest. Besides, this work focuses on the detection of the main entrance, ignoring the possible secondary or ancillary entrances since in many cases the public is not allowed to use secondary entrances, commonly used mostly for special purposes, such as emergency evacuations. Therefore, from the perspective of navigation, the main entrance is more important than the secondary entrance.

The idea of this work is inspired by two intuitions: (1) The location of the main entrance of a public building is correlated with the shape of its footprint. For instance, the main entrance is located normally near the centroid of the footprint, as shown in Figure 6.2a. If the footprint is reflection symmetry, the main entrance is very likely located close to the symmetry axis to maintain the symmetric characteristics of the building, as shown in Figure 6.2b. Some previous works have applied the symmetric characteristics of the footprint in reconstructing building elements, such as roof type (Hu et al. 2018). Another example is that the main entrance sometimes is located at the convex and concave edge of the footprint, which corresponds to the rain-shed, independent vertical passage or entrance foyer. (2) The main entrance of a building is correlated with its surrounding spatial contexts, such as the streets. Generally, the main entrance should be easily accessed and observed from the streets, which often has shorter path distance to the streets and more observable points from the street than the other locations at the footprint, as shown in Figure 6.3.



**(a)** Entrance is close to centroid of footprint **(b)** Entrance is close to axis of symmetric footprint

**Figure 6.2.** Location of entrance is correlated with shape of footprint.



**(a)** Entrance is easily observed from roads **(b)** Entrance is easily accessed from roads

**Figure 6.3.** Location of entrance is correlated with spatial contexts of buildings.

In this work, we consider the main entrance tagging issue as a binary classification problem. By splitting the footprint into discrete equidistant points (also named samples), the task is thus converted to identify which one is the most likely location of the main entrance (positive). For each point, the corresponding intrinsic and extrinsic features are extracted by measuring the relationship between the samples and the footprint as well as its spatial context, respectively. The proposed approach consists of two stages, namely, model training and entrance tagging. During the training stage, three different classification models are fitted. These are the Weighted Random Forest (WRF)(Effendy et al. 2014), Balanced Random Forest (BRF) (Khalilia et al. 2011), and the SmoteBoost (Chawla et al. 2002) algorithms. The reason for testing and comparing these models is that they are robust to class imbalance situations. During the tagging stage, the fitted model is used to calculate the probability of assigning each sample in a test building as positive, and the one with the highest probability is chosen as the estimated location of the main entrance.

The main contributions of this work are twofold:

(1). To the best of our knowledge, this work is the first to propose an automatic approach to estimating the physical location of the main entrance of buildings based only OSM data.

(2). Our proposed approach is broadly applicable, as it relies only on OSM data, which is freely accessible and covers a wide range of areas in the world.

The remainder of this paper is structured as follows: In Section 6.2, we introduce the relevant works. In Section 6.3, we present the workflow of the proposed approach and give the details of each step. We evaluate the proposed approach through experiments on 320 public buildings in Section 6.4, and discuss relevant issues in Section 6.5. Conclusions are drawn in Section 6.6.

## 6.2 Related works

We categorize the entrance detection approaches into two groups: door detection (indoor) and entrance detection (outdoor) since to some degree an entrance is also a door but the detection approach is different.

**Door detection:** Door detection approaches are widely investigated due to two reasons. First, robots need to recognize the location of doors for autonomous navigation. Second, indoor reconstruction solutions also need to detect the location of doors to build a complete indoor navigation network for pedestrians. For instance, Murillo et al. (2008) presented a technique for detecting doors using only visual information for robot navigation. The probability distribution is learned in a parametric form from a few reference images in a supervised setting. A model-based approach is used, where the door model is described by a small set of parameters characterizing the shape and the appearance of the object. The geometry of the door is specified by a small number of parameters and the appearance is learned from the reference data. The constraints of man-made environments were used to generate multiple hypotheses of the model and the learned probability distribution was used to evaluate their likelihood. Zhao et al. (2015) proposed a light-weight and broadly applicable door detection approach based on the magnetometer embedded on a smartphone. It analyzes readings from the built-in magnetic sensors since the anomalies or sharp fluctuations of magnetic signals normally happened at doors. Nikoohemat et al. (2017) 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 results showed that using structured learning methods for semantic classification is promising. Recently, Quintana et al.

(2018) presented an approach that detects open, semi-open and closed doors in 3D laser scanned data of indoor environments. It integrates the information regarding of both the geometry and colour provided by a calibrated set of 3D laser scanner and a colour camera. The integration of geometry and colour makes it robust to occlusion and variations in colours resulting from varying lighting conditions at each scanning location and different scanning locations.

**Entrance detection:** Apart from door detection, a couple of automatic methods have also been proposed to detect the entrance of buildings, which is the focus of this work. The traditional ways to detect the entrance is through images analysis. That is, the detection of entrance is treated as the issue of semantic tagging from images. For instance, Liu et al. (2014) proposed a three-stage system that starts with a high-recall entrance candidate extractor, which is followed by classifying candidates based on local image features. The final stage fuses results from multiple views by using Markov chain Monte Carlo to solve a Bayesian inference problem, and to select the best set of entrances that explain the image of a facade. The system achieves a recall of 70% on a challenging data set of urban scene images. Kang et al. (2010) proposed an approach to detecting the entrance of building for robot navigation based on the images that can be collected in real-time by mobile robots during navigation. They adopted a probabilistic model for entrance detection by defining the likelihood of various features for entrance hypotheses. The basic idea is to exclude non-entrance regions in the surface of a building, such as walls and windows, which are extracted from the image of the surface. The reminding region is considered as the candidate of entrance, which is then evaluated by their proposed probabilistic model. Recently, Talebi et al. (2018) presented a vision-based method for detecting building entrances with outdoor images. They first converted the RGB image into gray-scale image, from which the vertical and horizontal line segments can be detected by using Line Segment Detector (LSD) algorithm. Then, the regions between the vertical lines were specified and the features including height, width, location, color, texture and the number of lines inside the regions are obtained. Finally, they used some additional knowledge such as door existence at the bottom of the image and a reasonable height and width of a door to decide if a door is detected or not. Different from the aforementioned works that use manually defined features to detect entrance, Liu et al. (2017) proposed using random forest classifier to perform automatic feature selection and entrance classification. The process of the algorithm is as follows: first, the scene geometry was exploited and the multi-dimensional problem is reduced down to a one-dimensional (1D) problem. Then, a rich set of discriminative image features for entrances was explored according to constructed designs, specifically focusing on properties such as symmetry and color consistency. Lastly, a joint model was formulated in three dimensions (3D) for entrances on a given facade, which enables the exploitation of physical constraints between different entrances on the same facade in a systematic manner to prune false positives, and thereby selected an optimum set of entrances on a given facade. The drawback of these works is that they rely on the stree-level image, which can be obtained from some map providers, such as Google Street View and through the cameras equipped on the robot during navigation. The street-level image does not always contain the entrance of all the buildings since the entrance might not face any street. Meanwhile, Google Street View covers only partial large cities in the world. The robot-based solution is not applicable for pedestrian way-finding, which needs to know the location of the main entrance in advance.

## 6.3 Approach

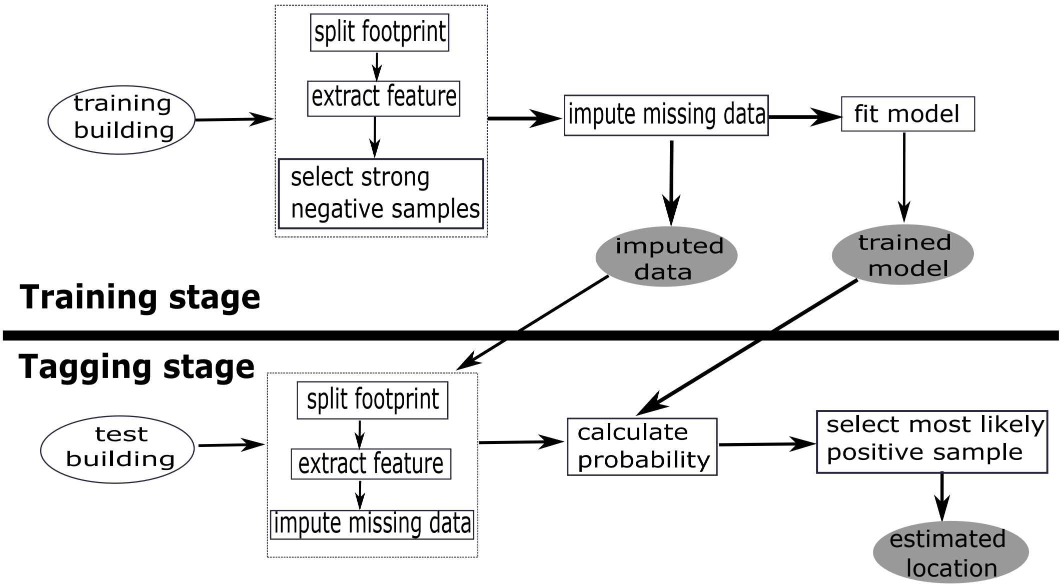


**Figure 6.4.** Workflow of proposed approach.

As illustrated in Figure 6.4, the proposed approach consists of two stages: training and tagging. In the training stage, the edges of each building are first split into single points, also named samples. They are then tagged as positive (true main entrance) and negative accordingly. The next step is to extract features for each sample by measuring the relationship between the sample and the footprint (intrinsic features) and the surrounding spatial entities (extrinsic features), such as the distance to the centroid of the footprint and the shortest path distance to the main roads. However, some negative samples are neighbors of the positive sample, which may cause the mis-classification of the positive sample. To solve this issue, only the ‘strong’ negative samples are used in the training data. The ‘strong’ negative samples are those whose physical or feature distance is far away from the positive sample. After collecting the samples from all of the training buildings, the missing data of the training samples would be filled out, which is caused by the lacking of some spatial entities around a certain building. For instance, not all the buildings have main roads around. Finally, a classification model that can deal with the unbalanced class issue is fitted based on the training samples.

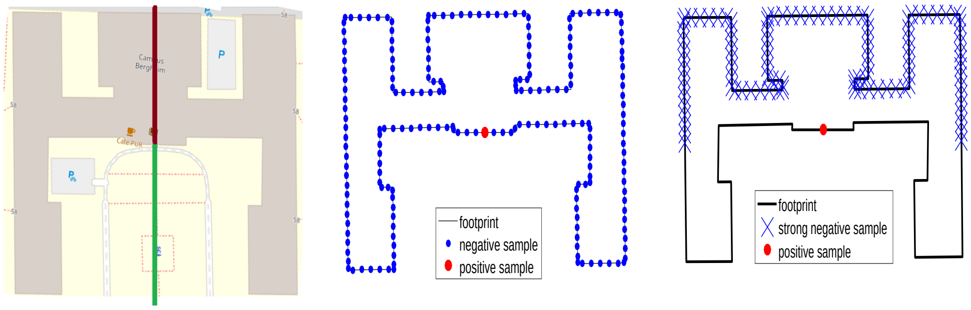
In the tagging stage, the footprint of a test building is split into single points and the corresponding features are extracted in the same way as in the training stage. The next step is to impute the missing data by using the strawman strategory (Tang and Ishwaran 2017). Specifically, the missing value of a numerical feature is filled out with the median value of the non-missing values of this feature in the training samples. Likewise, the missing value of a categorical feature is filled out with the most frequent value of the non-missing values of this feature in the training samples. Then, the trained model is used to calculate the probability of assigning each sample to positive or negative. Finally, the one with the highest positive probability among all the samples in a building is chosen as the estimated location of the main entrance. In the following sections, we will elaborate on the key steps of the training stage.

### 6.3.1 Data pre-processing

The input of the training stage is buildings. For each one, its external edges are first split into smaller segments with an interval of three meters. For the segment whose length is below three meters, they are directly treated as a complete segment. Then, the midpoint of the segment is chosen as a sample (the candidate location of the main entrance). The one whose parental segment contains the true main entrance is tagged as positive, and the others are tagged as negative. We define the edge that contains a sample as the master edge of the sample. The features of each sample can be extracted by measuring the relationship between the sample and the footprint (intrinsic features) and the surrounding spatial entities (extrinsic features), which will be elaborated in the following section. Figure 6.5a shows the footprint of a building. Figure 6.5b shows the discretized result, from which we can see: (1) the number of the negative sample is much larger than that of the positive sample (only one); (2) the positive sample is physically surrounded by some negative samples. If we fit a normal classification model with these samples, all the test samples would most likely to be categorized as negative to achieve the highest classification accuracy. However, what we expect is to correctly pick out the positive samples from the negative ones.

To handle the inbalanced data issue, this work adopts three classification models namely, SmoteBoost, Balanced Random Forest, and Weighted Random Forest. To address the second issue, the negative samples that are close to the positive sample in either physical or feature distance are ruled out from the training samples in order to reduce the interference of the negative samples on the positive sample. That is, only the 'strong' negative samples are preserved. The physical and feature distance thresholds are denoted by $P_T$ and $F_T$, respectively. The physical distance between two samples is defined as the shortest linear distance along the footprint, as shown in Figure 6.10. The feature distance is defined as the Euclidean distance of the feature vector of two samples. Note that before calculating the feature distance, each variable in the vector is first normalized by using the Min-Max Normalization method, as shown in Formula 6.1, limiting the value of all features to the range of zero and one. Figure 6.5c shows the selected 'strong' negative samples.

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{6.1}$$



**(a)** footprint of a building on OSM **(b)** discretization of footprint **(c)** resampling of nagative samples

**Figure 6.5.** Process of footprint split and sample extraction.

After obtaining positive and 'strong' negative samples for all the training buildings, the straw-man imputation strategy is adopted to deal with the missing data issue in the training samples. Specifically, we fill out the missing value of a numerical feature with the median value of the non-missing values of this feature in the training samples. Likewise, we fill out the missing value of a categorical feature with the most frequent value of the non-missing values of this feature in the training samples. More approaches that impute the missing data will be investigated in our future work, such as KNN, missing forest, and multiple imputation by chained equations (MICE) (Deng et al. 2016; Tang and Ishwaran 2017).

## 6.3.2 Feature extraction

This section introduces the procedure of extracting features for each sample in a building. Given a building, its footprint and surrounding spatial contexts or entities are obtained from OSM, on which the intrinsic and extrinsic features can be derived, respectively. In total, we define 84 features. The detailed definition of these features can be found in the shared files online.

### 6.3.2.1 Extrinsic feature

The spatial contexts include *address street*, *main road*, *pedestrian way*, *service way*, *railway*, *bicycle parking area*, *landmark*, and *postbox*. Partial buildings have been tagged with the address street. The key is 'addr_street' in OSM and the corresponding value is renamed as 'addr_street_value' in this work, based on which the address street of the building are retrieved. The key and value of these contexts in OSM are given in Table 6.1. The relationship between

**Table 6.1.** OSM key and value of spatial entities used to extract external features.

| | key | value |
|---|---|---|
| address street | name | addr_street_value |
| main road | highway | primary / secondary/ tertiary / unclassified/ residential |
| pedestrian way | highway | pedestrian |
| service way | highway | service |
| railway | railway | rail |
| bicycle parking area | amenity | bicycle_parking |
| landmark | artwork_type | sculpture |
| | tourism | artwork |
| | historic | memorial |
| | amenity | fountain |
| | man_made | water_well |
| | man_made | flagpole |
| postbox | amenity | post_box |

the sample and the spatial contexts can be measured in multiple ways, as shown in the first column of Table 6.2. Note that, we do not choose the pathways connected to the building as the spatial context since it is too strong features that indicate the location of the entrance, as shown in Figure 6.6.

Before introducing the specific features, we first define the outer perpendicular line (OPL) and inner perpendicular line (IPL) of a sample, which are needed in defining some features. OPL of a sample is the line with the sample as the start point, extending along the line that is perpendicular to the master edge of the sample and deviating from the building. Conversely, IPL is the line with the sample as the start point, extending along the line that is perpendicular to the master edge of the sample and toward the footprint. For example, the OPL and IPL of a sample in Figure 6.2 are denoted by the green and brown lines, respectively. The following measures are used to define the external feature:

**Shortest path distance**: It refers to the shortest path distance from a sample to multiple spatial contexts of the same type, such as multiple service ways. Normally, the true sample (main entrance) can be easily accessed (with a shorter path distance than other samples) from address streets and main roads. To calculate the path distance, the path obstacle is first extracted, including building, barrier, grass, water, railway, and garden. Next, the spatial contexts in the form of line segments or polygons are split into points at a certain interval (5 meters in this

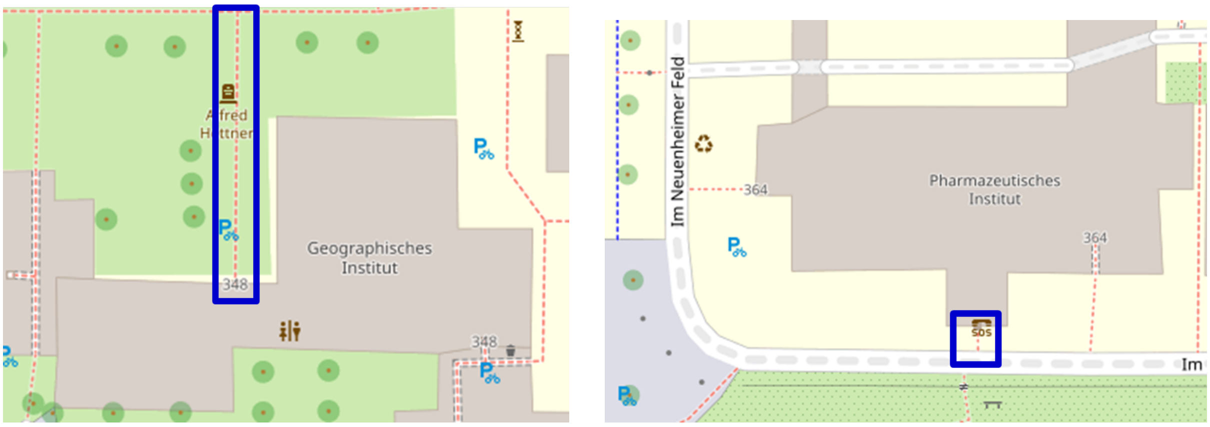


**Figure 6.6.** Pathway surrounded by blue rectangles is connected to buildings on OSM with the connection point as the location of main entrance.

work). The path distance from the sample to these context points is then calculated with the A-star algorithm (Hart et al. 1968), and among them the shortest path distance is obtained.

**Turning degree**: It refers to the turning degree of the shortest path from a sample to a certain spatial context. It is calculated by dividing the shortest path distance by the euclidean distance from the sample to the target location on the shortest path. The larger the value, the more turnings on the shortest path.

**Accessible**: It measures if a sample is accessible from a certain spatial context. It can be obtained from the result of the shortest path distance.

**Degree of visibility**: It measures how easily a sample (candidate entrance) can be observed from certain spatial contexts. Generally, the main entrance is easily observed from main roads. The obstacles that hinder visibility are buildings and barriers. Specifically, the key of visual obstacles on OSM is ‘barrier’. To calculate the degree of visibility of a sample, the spatial contexts (e.g., main roads) are first discretized into points at a certain interval (5 meters in this work) and the number of the points from which a sample can be directly observed without obstruction is used as the degree of visibility.

**Visible**: It measures if a sample is visible from a certain type of spatial contexts. It can be derived from the result of the degree of visibility.

**Euclidean distance**: It measures the Euclidean distance between a sample and the spatial contexts.

The other important extrinsic features are:

**Open area (*)**: It measures the size of an open area before a sample. To calcualte the feature, the OPL is first obtained, which is followed by searching all the intersection point of the OPL and the obstacles. The open area then equals the shortest Euclidean distance between the intersection points and the sample. The obstacles here are the building, grass, main road, barrier, water, and railway.

**Table 6.2.** Extrinsic feature extraction by measuring the relationship between samples and spatial contexts.

| | address street | main road | pedestrian way | service way | railway | bicycle parking area | landmark | postbox |
|---|---|---|---|---|---|---|---|---|
| Shortest path distance (*) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Accessible | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Turning degree | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Degree of visibility (*) | ✓ | ✓ | | ✓ | ✓ | | | |
| Visible | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Euclidean distance (*) | | | | | | | ✓ | ✓ |

**Distance to buildings(*)**: It measures the Euclidean distance from the sample to the nearest building. It is calculated in the same way as **Open area**. The only difference is that the obstacle here contains only buildings.

#### 6.3.2.2 Intrinsic feature

Intrinsic features refer to the features extracted from the OSM footprint. Some important intrinsic features are as follows:

**Distance to centroid (*)**: It represents the Euclidean distance from a sample to the centroid of the footprint.

**Proportion (*)**: It measures how close a sample is to the midpoint of its master edge. It is calculated by dividing the distance between the sample and the midpoint of its master edge by the length of its master edge. The value ranges from 0 to 0.5.

**Existence of axis**: It indicates if the reflection symmetry axis exists since not every building is symmetric. For instance, the footprint in Figure 6.5b is reflection-symmetric and the axis is the perpendicular bisector of the master edge of the positive sample.

**Distance to reflection symmetry axis (*)**: It represents the perpendicular distance from a sample to the reflection symmetry axis of the footprint if the axis exists.

**At intersected edge of axis (*)**: It indicates if a sample is located at the edge that intersects the axis of the building if the axis exists.

**Length of master edge (*)**: It represents the length of the edge that contains the sample.

**Face inner(*)**: It indicates if the OPL of a sample intersects the other edges of the building (except the master edge). It has been observed that the OPL of the entrance sample does not normally intersect the edges of the footprint, such as the positive sample in Figure 6.5b.

**Concavity and convexity**: It indicates if the master edge of a sample is concave (0), convex(1), or neither (-1). An edge is defined as convex only when the inner angles of the two endpoints of this edge approximate 90 degrees, while the neighboring two angles approximate 270 degrees. In contrast, an edge is defined as concave only when the two angles of this edge approximate 270 degrees, while the neighboring two angles approximate 90 degrees. For instance, the master edge of the positive sample in Figure 6.5b is concave.

**Opposite shape**: It indicates if the opposite edge of the master edge of the sample is concave (0), convex(1), or neither (-1). The opposite edge of an edge is defined as the closest exterior edge of a building, which intersects the perpendicular bisector of the edge.

Note that, for both intrinsic and extrinsic features, the one with the star symbol (*) means that apart from the absolute measurements, the sorting result of measurement of a sample among the total samples in the same building is also treated as features. It measures if one sample is closer to some spatial contexts or easier to be observed from some places than the other samples in the same building. Intuitively, the positive sample (entrance) is closer to the centroid of a building than most of the negative samples. The sorting result of each sample in a building, denoted by $S = \{s_1, s_2, ...s_n\}$ is normalized, denoted by $NS = \{s_i/n\}_{i \in [1,n]}$. $s_i$ denotes the sorting result of $i$-th sample, ranging from 1 to $n$, while $n$ denotes the number of samples in a building. In this way, the value of the sorting feature is limited in the range of 0 and 1, making it globally comparable.

### 6.3.3 Classification models for imbalanced data

As we mentioned before, the positive sample is far less than the negative ones, which cause imbalanced data issues (Sun et al. 2009). The common ways to deal with this issue include oversampling the minority class, under-sampling the majority class, and giving more weight to the minority class. This work adopts three different classification models: SmoteBoost, Balanced Random Forest, and Weighted Random Forest, which are the representative methods of the three strategies.

**SmoteBoost**: It was first proposed by Chawla et al. (2003) for countering imbalance in a dataset, which combines the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al. 2002) and Adaptive Boost (AdaBoost) (Schapire 2013). Specifically, before each boosting step, a SMOTE resampling calculates new synthetic examples for the minority class. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples from the k minority class nearest neighbors. AdaBoost works to improve the performance of weak learners (poor predictive models, but better than random guessing). It iteratively builds an ensemble of weak learners by assigning a higher weight to samples that the current weak learner misclassified during each iteration. This weight determines the probability that the sample will appear in the training of the next weak learner. For this reason, boosting algorithms like AdaBoost are particularly useful for class imbalance problems because higher weight is given to the minority class at each successive iteration as data from this class is often misclassified. More details of the AdaBoost can be found in (Chawla et al. 2003).

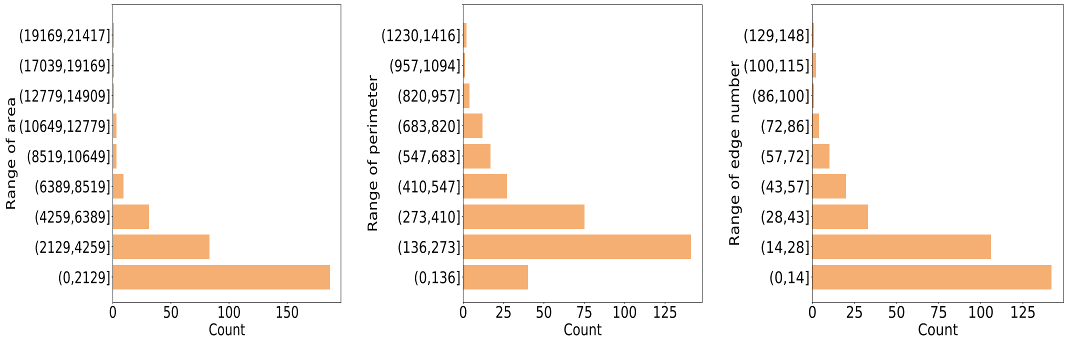
**Balanced Random Forest**: Balanced Random Forest (BRF) is a variant of the random forest by under-sampling the majority class in building each decision tree. BRF algorithm consists of three steps. (1) For each iteration (building a tree) in random forest, draw a bootstrap sample from the minority class and randomly draw the same number of samples, with replacement, from the majority class. (2) Induce a classification tree from the data to maximum size, without pruning. (3) Repeat the two steps above for the number of trees desired. During the tagging stage, the predictions of all the trees in the forest are aggregated to make the final prediction. More details of the Balanced Random Forest can be found in (Khalilia et al. 2011).

**Weighted Random Forest**: Weighted Random Forest (WRF) is another variant of random forest, which follows the idea of cost-sensitive learning. That is, a heavier penalty would be placed on the misclassification of the minority class, by assigning the minority class a larger weight (i.e., higher misclassification cost). The class weights are used in two places of the RF algorithm. In the tree induction procedure, class weights are used to weight the Gini criterion for finding splits. In the terminal nodes of each tree, class weights are again taken into consideration. The class prediction of each terminal node is determined by “weighted majority vote”; i.e., the weighted vote of a class is the weight for that class times the number of cases for that class at the terminal node. The final class prediction for RF is then determined by aggregating the weighted vote from each tree, where the weights are average weights in the terminal nodes. More details of Weighted Random Forest can be found in (Effendy et al. 2014).

## 6.4 Experiments

### 6.4.1 Experimental setting

We have collected 320 public buildings from seven German cities: Frankfurt (60), Mannheim (28), Heidelberg (46), Karlsruhe (40), München (44), Stuttgart(38), Berlin (40), and Köln (24). The digital in the parentheses denotes the number of buildings collected in the corresponding city. We use IGIS.TK and its spatial data model to export the OSM data of the seven cities into the Spatialite database, from which the corresponding OSM entities around a building are retrieved (Noskov and Zipf 2018). Specifically, the OSM elements (i.e., node, way, and relation) that locate in or intersect with the buffer of the building are retrieved from the database. The buffer takes the centroid of the building as the center and 150 meters as the radius. The corresponding SQL script is as *'SELECT elements.id, AsGeoJson(Transform(geom,32630)),keys.txt, vals.txt FROM elements JOIN tags ON elements.id=tags.id JOIN keys ON tags.key = keys.rowid join vals on tags.val=vals.rowid WHERE MbrIntersects(Transform(Buffer(Transform(MakePoint (8.3728, 49.0159, 4326), 32630), 150), 4326),elements.geom)'*. (8.372814, 49.015944) represents the latitude and longitude coordinates of the centroid of a building that should be modified as buildings when querying the corresponding buffer. Based on the retrieved result, the required OSM entities and spatial contexts of a building can be then extracted.



**(a)** distribution of area **(b)** distribution of perimeter **(c)** distribution of edge number

**Figure 6.7.** Distribution of area, perimeter, and edge number of test buildings.

Furthermore, we analyze the distribution of the perimeter, area, and number of edges of the total buildings, which is shown in Figure 6.7. We can observe that the shape of the buildings varies greatly. Then, the spatial contexts that are located around the buildings and the symmetric buildings (with axis) are analyzed. The occurrence frequency of different spatial contexts and the symmetric building is shown in Figure 6.8. We can see the missing data issue is quite serious with the frequency of only the main road, service way, and address street over 0.7, making the classification task much challenging. To know which feature is important in recognizing the main entrance, we measured the importance of each feature (84 in total) by calculating how much the accuracy decreases when the feature is excluded in the random forest. From which, the top 20 most significant features are picked out, and their normalized weights are shown in Figure 6.9.

The top 20 features are '*sort of distance to centroid*', '*proportion*', '*distance to centroid*', '*sort of shortest path distance to main road*', '*distance to nearest building*', '*sort of shortest path distance to service ways*', '*turning degree of shortest path distance to service ways*', '*sort* *of open area*', '*open area*', '*sort of shortest path distance to address street*', '*sort of distance to nearest building*', '*turning degree of shortest path distance to main road*', ' *sort of visibility degree from main road*', '*visibility degree from main road*', '*visibility degree from service way*', '*sort of shortest path distance to service way*', '*shortest path distance to main road*', '*sort of visibility degree from service way*', '*sort of turning degree of shortest path distance to address street*', and '*opposite shape*'. They play the most significant role in identifying the location of the main entrance. Axis related features are not ranked among the top 20 as we expected, which is because only a small proportion of buildings are symmetric.

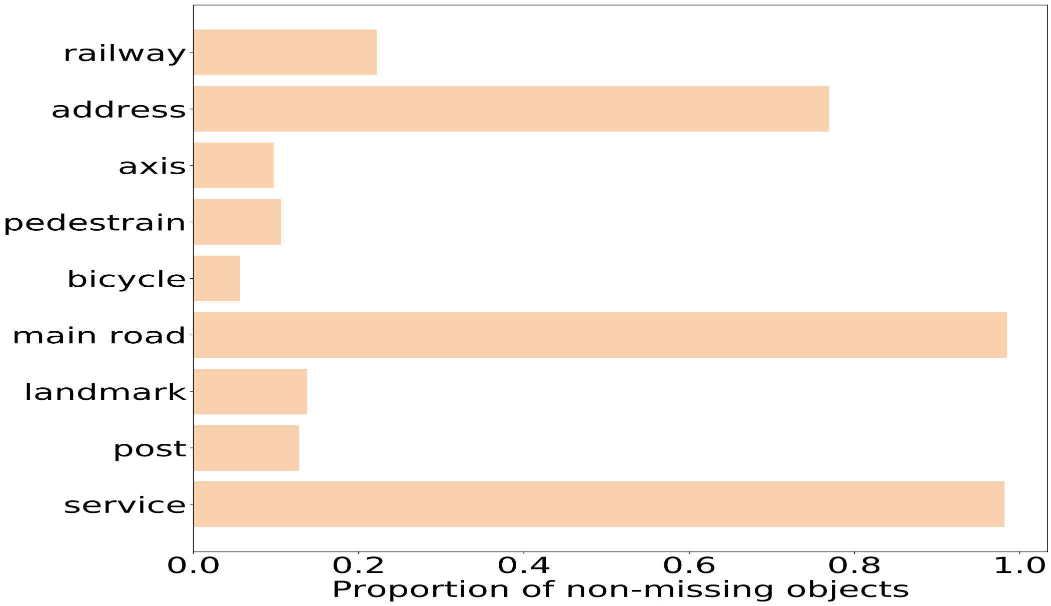


**Figure 6.8.** Occurrence frequency of spatial contexts and symmetric buildings in test buildings.

## 6.4.2 Tagging accuracy

In this experiment, we compare the three classification models for the imbalanced class issue and the general random forest model. A couple of important parameters need to be set for our proposed solutions and the classification models, to achieve the optimal performance. Specifically, the physical distance threshold ($P_T$) and the feature distance threshold ($F_T$) that are used to select the 'strong' negative samples are set to 24 (m) and 0.04, respectively. For the WRF approach, the important parameters include the number of trees, the maximum depth of the tree, and the weight of the minority class compared to the majority class, which are set to 80 and 12, and 160:1, respectively. For the BRF approach, the key parameters include the number of trees and the maximum depth of the tree, which are set to 140 and 14, respectively. For the Smote-Boost approach, the key parameters include the number of new synthetic samples per boosting step, the maximum number of estimators at which boosting is terminated, and the number of

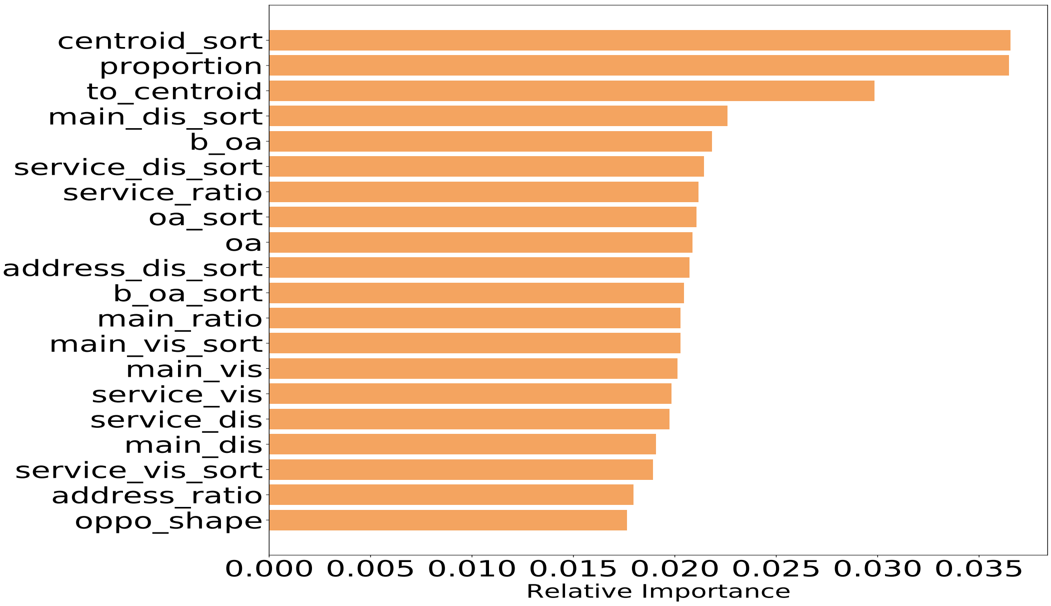


**Figure 6.9.** Importance of top 20 features.

the nearest neighbors that are used to generate new samples for a minority class sample, which are set to 130, 4, and 90, respectively. Note that, the SmoteBoost model is not stable in our tests such that different prediction results are achieved with the same parameters. For the general RF approach, the important parameters include the number of trees and the maximum depth of the tree, which are set to 110 and 14, respectively. These approaches are implemented based on scikit and the imbalanced-learn package of Python.

The five-fold cross-validation is used to evaluate the approaches based on 320 public buildings. That is, the 320 buildings are divided into five test groups with each containing 64 buildings. In each test group, the 64 buildings are treated as the testing set, and the remaining 258 buildings are treated as the training set, in which the location of the main entrance is known. We measure the deviation between the true entrance and the estimated entrance in two ways. The first is the shortest linear distance between them along the footprint. The second is the distance of the shortest path that the users need to take to walk from the estimated entrance to the true entrance. Note that, due to the existence of obstacles such as barriers and buildings, the path distance between two locations might be much larger than their linear distance, as shown in Figure 6.10.

In Appendix A, we present the tagging results of partial testing buildings by using the four models. In the figures, the red square denotes the position of the true entrance, while the brown upper-pointing triangle, the yellow star, the light blue diamond, and the blue right-pointing triangle denote the estimated position of the entrance by WRF, BRF, RF, and SmoteBoost, respectively. The complete data set, python code, and tagging results have been uploaded online. Figure 6.11 shows the cumulative linear distance error of the total five test groups. We can see

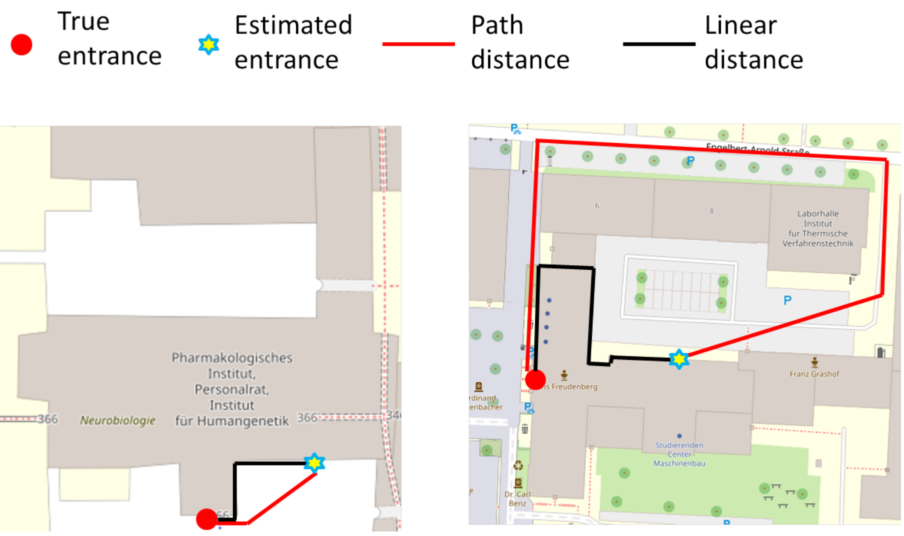

● True entrance
Estimated entrance
Path distance
Linear distance
Im Neuenheimer Feld
Pharmakologisches Institut, Personalrat, Institut für Humangenetik
Neurobiologie
366
Engelbert-Arnold-Straße
Laborhalle Institut für Thermische Verfahrenstechnik
Franz Grashof
Studierenden Center Maschinenbau
P

**Figure 6.10.** Two kinds of distance errors. In the left figure, the path distance is smaller than the linear distance. In the right figure, the path distance is much larger than linear distance due to the obstruction of buildings.

WRF and BRF achieve the best tagging result with an average error of around 21 meters. 30% of the buildings are correctly tagged with the linear distance error at 0 meters, and in 80% of the cases, the distance error is below 30 meters. For SmoteBoost and the general random forest approaches, the mean error is at around 35 meters. BRF and WRF can better deal with the imbalanced class issue than the SmoteBoost approach in this context.

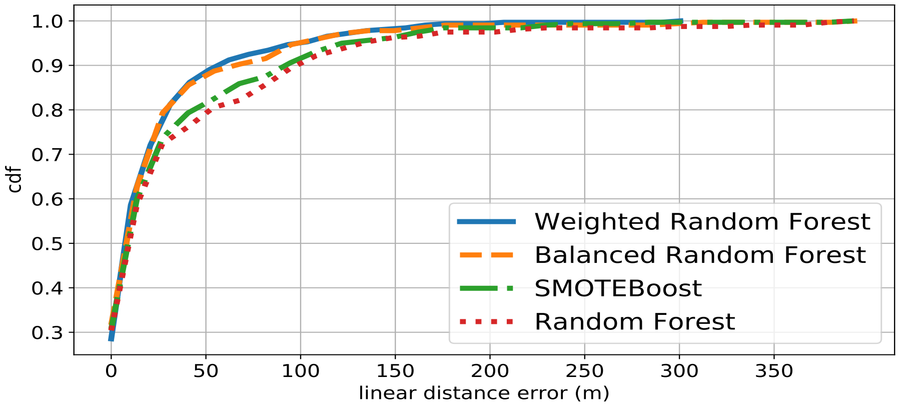


**Figure 6.11.** CDF of linear distance error of four classification models.

However, the liner distance error between the estimated and the true entrance does not reflect the actual walking distance that users need to take from the estimated to the true entrance due to the existence of obstacles, including buildings and barriers (e.g., fence) in this context, as shown in Figure 6.10. Therefore, we further calculate the shortest path between the estimated and true entrance for the five test groups. If the true entrance is unreachable from the estimated

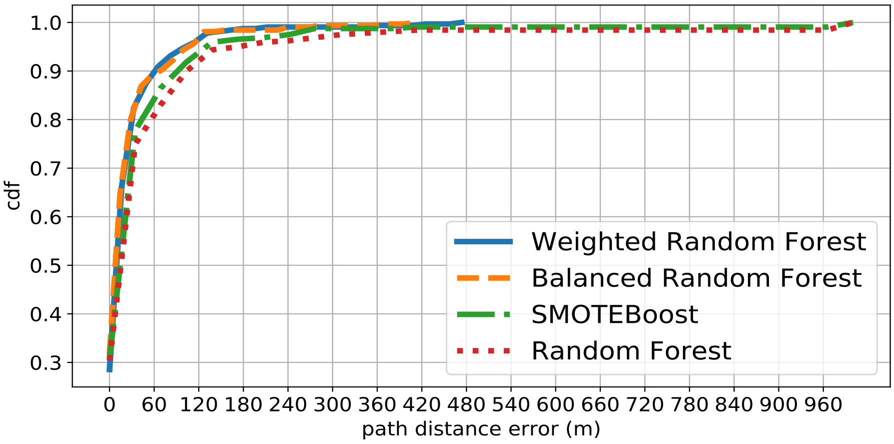


**Figure 6.12.** CDF of path distance error of four classification models.

entrance, the shortest path distance is set to 1000 meters. Figure 6.12 shows the CDF of the path distance error of the four approaches. We can see, BRF and WRF still achieve a promising result, with a mean error at 22 meters, and in 80% of the cases, the path distance error is below 30 meters. However, for SmoteBoost and the general RF approaches, the path distance error becomes larger at 38 and 46 meters, respectively, compared to their liner distance errors. We believe that a distance deviation at around 30 meters would not cause the failure of finding the true entrance because humans have powerful spatial cognition capability (Foo et al. 2005). For instance, pedestrians can easily notice the entrance when they are following the route to the estimated entrance if the estimated and the true entrance is not far away.

Furthermore, we analyzed the test buildings whose linear distance error is over 60 meters. We found that three reasons mainly cause the large tagging error. The first is inaccurate or incomplete OSM data. For instance, the building in Figures (k) and (u) of Appendix A have a tagging error over 60 meters. A fence in front of the estimated entrance location of Figure (k) by WRF is missing on OSM, leading to the estimated entrance easily accessed from roads, which is not true. Likewise, a deep hole in front of the estimated entrance location of Figure (u) is missing on OSM. The second is that the estimated location is the ancillary entrance but not the main entrance, as shown in Figures (s) and (ad) of Appendix A, where the estimated location by WRF is very close to the ancillary entrance. The third reason is that there are always numbers of exceptional buildings that do not follow the general layout principles of the main entrance.

Finally, we analyze the ranking of positive probability assigned to the true entrance sample among the total samples in a building. The ranking result has two types. The first is the absolute ranking result among all the samples with the value ranging from 1 to N, where N represents the number of samples in a building. Figure 6.13 shows the CDF of the absolute ranking result achieved by the four approaches. Still, BRF performs the best. In 55% of the cases, the true positive sample is ranked among Top 4. In 75% of the cases, it is ranked among Top 10. The second is the relative ranking result, which considers the varying number of samples in test buildings. It is calculated by dividing the absolute ranking result by the total number of samples in the corresponding building, limiting the value in the range of zero to one. Figure 6.14 shows the CDF of the relative ranking result achieved by the four approaches. Likewise, BRF performs the best. In 50% of the cases, the true entrance sample is ranked among the top 2%. In 74% of the cases, it is ranked among the top 10%. The ranking result looks promising, which confirms that the trained models (i.e., BRF and WRF) is close to the ground truth.

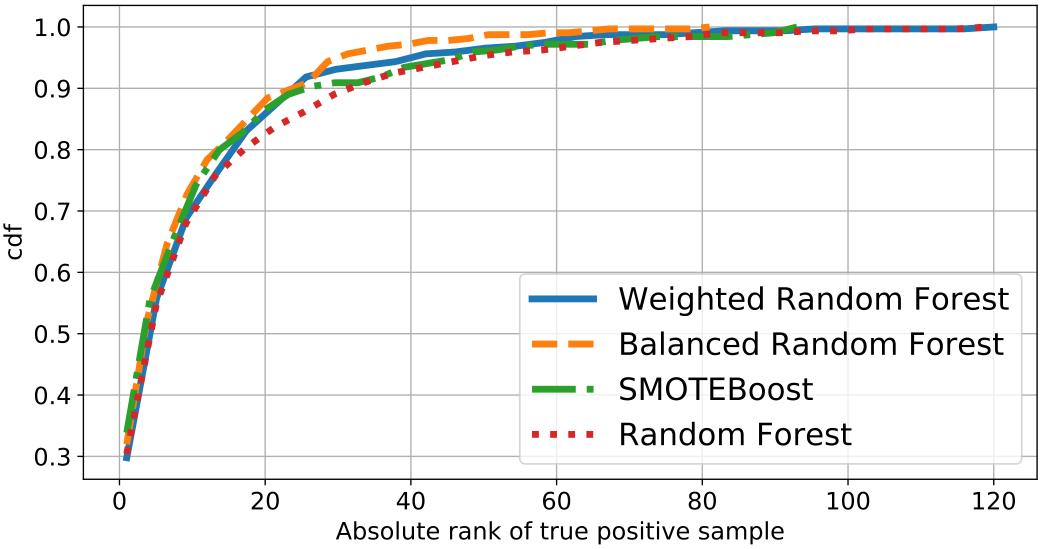


**Figure 6.13.** CDF of absolute ranking result of estimated positive probability of true entrance by four classification models.

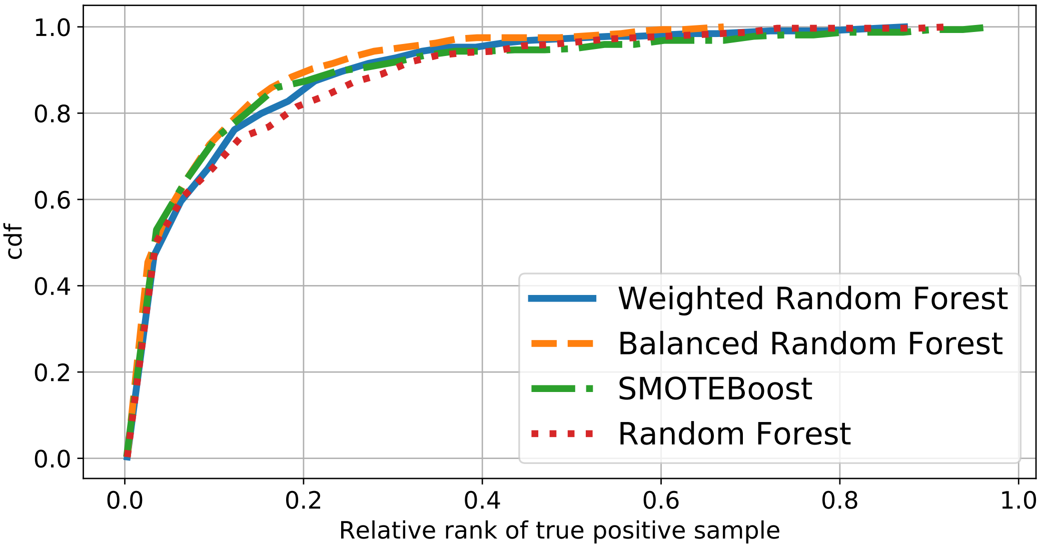


**Figure 6.14.** CDF of relative ranking result of estimated positive probability of true entrance by four classification models.

## 6.5 Discussions

**Main entrance assumption:** One of the assumptions of the proposed solution is that there is one and only one main entrance in a public building. This is due to two reasons. First, in most of the cases, this assumption holds. Second, it would be quite challenging to detect a variable number of main entrances in a public building if we are uncertain how many main entrances exist. However, when collecting the test buildings, we also found that a public building might be comprised of multiple departments, each having one house number and one corresponding main entrance. Such cases of buildings are beyond the scope of our work. However, this will be dealt with in future work by considering the house number tagged on OSM since each house number corresponds to a main entrance. That is, multiple main entrances can be identified from a building if the tagged location of the house number are known.

**Fusion of OSM and satellite imagery**: As we have mentioned in the experimental section, the tagging error is often caused by missing or incomplete data in OSM. This greatly reduces the applicability and robustness of the proposed solution. To mitigate the issue, in the future, we plan to use the satellite imagery (e.g., from Bing map) to provide more cues about the possible locations of the main entrance. For instance, in Figure 6.15, an open space is in front of the main entrance, which can be identified from the satellite imagery. However, by using only the data from OSM, a big tagging error is produced, as shown in Figure (v) of Appendix A. One of the cues of the impossible entrance position is the green space, as shown in Figure 6.16, which can be observed from the satellite imagery. However, with only the OSM data, the estimated entrance by BRF is located at the green space, as shown in Figure (ab) of Appendix A. The possible solution is to combine the manually defined features extracted from OSM, and the features automatically extracted from the satellite imagery with deep learning in an integrated model.

## 6.6 Conclusion

To mitigate the misleading and inaccurate navigation issues caused by the missing main entrances of public buildings on current map providers (e.g., Google Maps and OSM), we proposed a broadly applicable main entrance tagging approach based only on extrinsic and intrinsic features extracted from OSM. Three classification models have been applied to deal with the imbalanced data issue, namely WRF, BRF, and SmoteBoost. Experimental results show that WRF and BRF have a low tagging error in both linear distance and shortest path distance errors, which we believe can greatly save pedestrians' effort in finding the main entrance. We also found the most frequent tagging error is normally caused by inaccurate and incomplete OSM data. Realizing this interesting finding, we will investigate the possibility of automati-

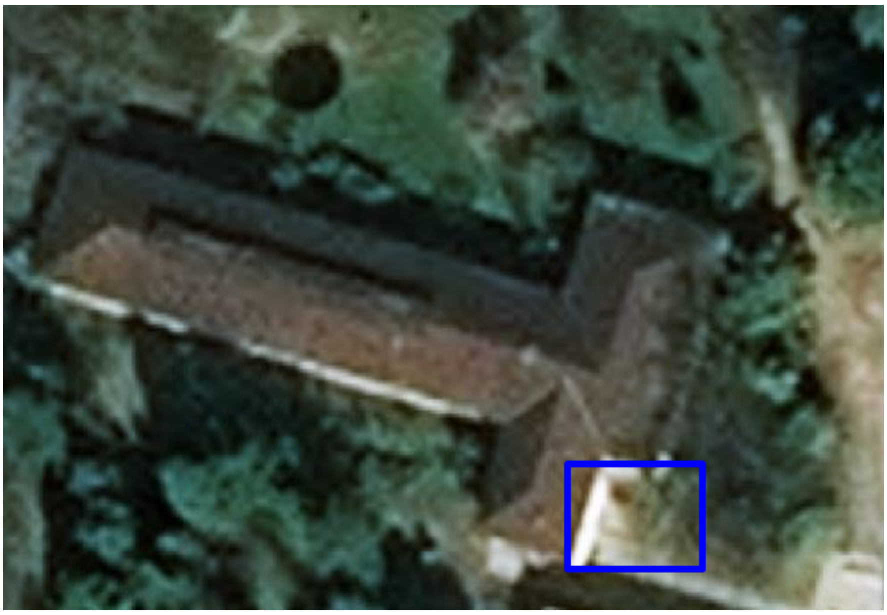


**Figure 6.15.** Blue square indicates an open space where the main entrance is located.

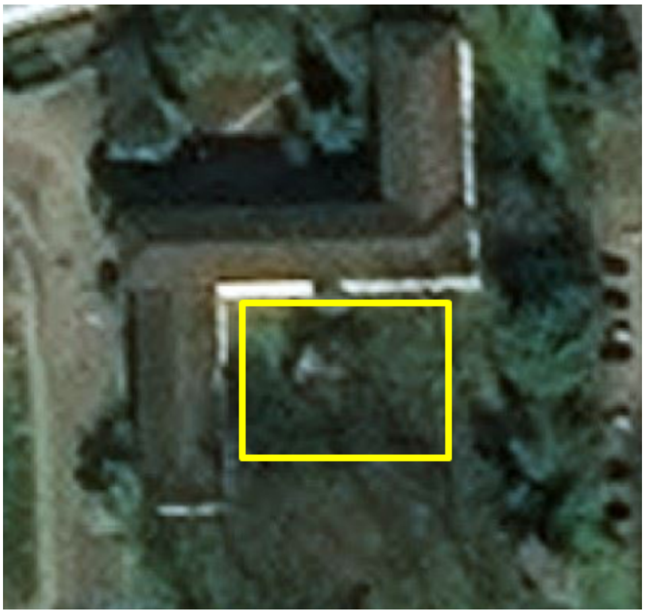


**Figure 6.16.** Yellow square denotes a green space where the main entrance is not likely to be located.

cally reporting erroneous data on OSM based on the tagged entrance since the big tagging error might be related to erroneous OSM data. Apart from this, in the future, we plan to combine the satellite imagery to provide further evidences about the possible location of the main entrance to mitigate the large tagging error.

## 6.7 Data and codes availability statement

The data and codes that support the findings of this study are available in entrance_tagging with the identifier at the private link https://figshare.com/s/00612ebbc369a980bd7b

# Reference

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer.

Deng, Y., Chang, C., Ido, M. S., and Long, Q. (2016). Multiple imputation for general missing data patterns in the presence of high-dimensional data. *Scientific reports*, 6:21689.

Effendy, V., Baizal, Z. A., et al. (2014). Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest. In *2014 2nd International Conference on Information and Communication Technology (ICoICT)*, pages 325–330. IEEE.

Foo, P., Warren, W. H., Duchon, A., and Tarr, M. J. (2005). Do humans integrate routes into a cognitive map? map-versus landmark-based navigation of novel shortcuts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(2):195.

Goetz, M. and Zipf, A. (2011). Extending openstreetmap to indoor environments: bringing volunteered geographic information to the next level. *Urban and regional data management: UDMS annual*, 2011:47–58.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107.

Hochmair, H. H., Zielstra, D., Neis, P., Hartwig, P., Hochmair, H., and Zielstra, D. (2013). Assessing the completeness of bicycle trails and designated lane features in openstreetmap for the united states and europe. In *Transportation Research Board Annual Meeting*.

Hu, X., Fan, H., and Noskov, A. (2018). Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints. *International Journal of Digital Earth*, 11(10):1039–1063.

Kang, S.-J., Trinh, H.-H., Kim, D.-N., and Jo, K.-H. (2010). Entrance detection of buildings using multiple cues. In *Asian Conference on Intelligent Information and Database Systems*, pages 251–260. Springer.

Khalilia, M., Chakraborty, S., and Popescu, M. (2011). Predicting disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11(1):51.

Liu, J., Korah, T., Hedau, V., Parameswaran, V., Grzeszczuk, R., and Liu, Y. (2014). Entrance detection from street-view images. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshop (CVPR), Columbus*.

Liu, J., Parameswaran, V., Korah, T., Hedau, V., Grzeszczuk, R., and Liu, Y. (2017). Entrance detection from street-level imagery. US Patent 9,798,931.

Murillo, A. C., Košecká, J., Guerrero, J. J., and Sagüés, C. (2008). Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, 56(6):512–521.

Nikoohemat, S., Peter, M., Elberink, S. O., and Vosselman, G. (2017). Exploiting indoor mobile laser scanner trajectories for semantic interpretation of point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Noskov, A. and Zipf, A. (2018). Open-data-driven embeddable quality management services for map-based web applications. *Big Earth Data*, 2(4):395–422.

Quintana, B., Prieto, S. A., Adán, A., and Bosché, F. (2018). Door detection in 3d coloured point clouds of indoor environments. *Automation in Construction*, 85:146–166.

Schapire, R. E. (2013). Explaining adaboost. In *Empirical inference*, pages 37–52. Springer.

Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719.

Talebi, M., Vafaei, A., and Monadjemi, A. (2018). Vision-based entrance detection in outdoor scenes. *Multimedia Tools and Applications*, 77(20):26219–26238.

Tang, F. and Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(6):363–377.

Zeng, L. and Weber, G. (2015). A pilot study of collaborative accessibility: How blind people find an entrance. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 347–356. ACM.

Zhao, Y., Qian, C., Gong, L., Li, Z., and Liu, Y. (2015). Lmdd: Light-weight magnetic-based door detection with your smartphone. In *2015 44th International Conference on Parallel Processing*, pages 919–928. IEEE.

# 7. Feasibility of using grammars to infer room semantics

## Article information

| | |
|---|---|
| Authors | Xuke Hu, Hongchao Fan, Alexey Noskov, Alexander Zipf, Zhiyong Wang, Jianga Shang |
| Journal | Remote Sensing |
| History | Received: 13 May 2019<br>Revised: 14 June 2019<br>Accepted: 26 June 2019 |
| Reference | Hu, X., Fan, H., Noskov, A., Zipf, A., Wang, Z., Shang, J. (2019). Feasibility of Using Grammars to Infer Room Semantics. Remote Sensing. vol. 11(13), p.1535. Doi: 10.3390/rs11131535 |

## Abstract

Current indoor mapping approaches can detect accurate geometric information but are incapable of detecting the room type or dismiss this issue. This study 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;

## 7.1 Introduction

People spend most of their time indoors, such as offices, houses, and shopping malls (Zhang et al. 2010). 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 (Yassin et al. 2016). 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 (Youssef 2015; Elhamshary and Youssef 2015). 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 (Gao et al. 2014), 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 (de las Heras et al. 2015; 2014; Dodge et al. 2017; Dosch et al. 2000). 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 (Ambruş et al. 2017; Armeni et al. 2016; Qi et al. 2017; Xiong et al. 2013), image (Henry et al. 2012; Furukawa et al. 2009; Ikehata et al. 2015), and volunteers' trace (Alzantot and Youssef 2012; Gao et al. 2014). 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) (Armeni et al. 2016; Zhang et al. 2013) 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, (Luperto et al. 2017) proposed using a statistical relational learning approach to reason the type of rooms as well as buildings at schools and office buildings. (Elhamshary et al. 2016; Elhamshary and

Youssef 2015) 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 (De la Higuera 2010; D'Ulizia et al. 2011) 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 (Braun, Grömling 2005; Hain 2003; Klonk 2016; Watch 2002) 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 (Hu et al. 2017) 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 sematic 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 to infer the type 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 7.2, 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 7.3. In Section 7.4, we present the workflow and the details of each step of the proposed approach. In Section 7.5, we evaluate our approach using 15 floor-plans and discuss some issues in Section 7.6. We conclude the paper in Section 7.7.

## 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 (Azhar 2011). 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 (Santos et al. 2017). Different from IFC, CityGML (Kolbe 2009) 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 (Li et al. 2015). 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 (Kim et al. 2014). 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 (Kang and Li 2017). 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 (Ahmed et al. 2012; 2011; Dosch et al. 2000; Gimenez et al. 2016; Macé et al. 2010). 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. (2015; 2014; 2011) 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. (2017) 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 (2012) 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 (2014) proposed generating metric scale floor plans based on individual room measurements by using commodity smartphones equipped with accelerometers, magnetometers, and cameras. Furukawa et al. (2009) 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. (2011) 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. (2015) 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. (2013) 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, Youssef (2015); Gao et al. (2014); Jiang et al. (2013) 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. (2015) 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. (2017) 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 et al. (2014) 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. (2013) 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. (2017) 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. (2017) 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. (2016) 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. (2017) 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 (Philipp et al. 2014; Yue et al. 2012; Becker et al. 2015; Hu et al. 2017) or machine learning techniques (Luperto and Amigoni 2016; Luperto et al. 2013; 2017; Rosser et al. 2017). Yue et al. (2012) 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 (Philipp et al. 2014) 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 (2014) used a shape grammar (Mitchell 1990) 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. (2017) 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. (2017) 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. (2013) 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 (e.g., small room, medium room, big room, corridor, and hall) identified from data acquired by laser range scanners. Furthermore,Luperto et al. (2017) 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 and von Wichert (2014a) 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 and von Wichert (2014b) 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.

## 7.3 Formal representation of layout principles of research buildings

### 7.3.1 Definition of research buildings

Research buildings are the core buildings at universities, including laboratories (Watch 2002) 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. According to (Watch 2002), 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 (Braun, Grömling 2005; Hain 2003) 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 (Hain 2003).

This work focuses on the typical research building, which refers to those research buildings whose layouts are corridor based. Figure 7.1 shows the three types of layouts of typical research buildings based on the layout of corridors: Single-loaded, double-loaded, and triple-loaded (Braun, Grömling 2005). Most research buildings are the variations of them.

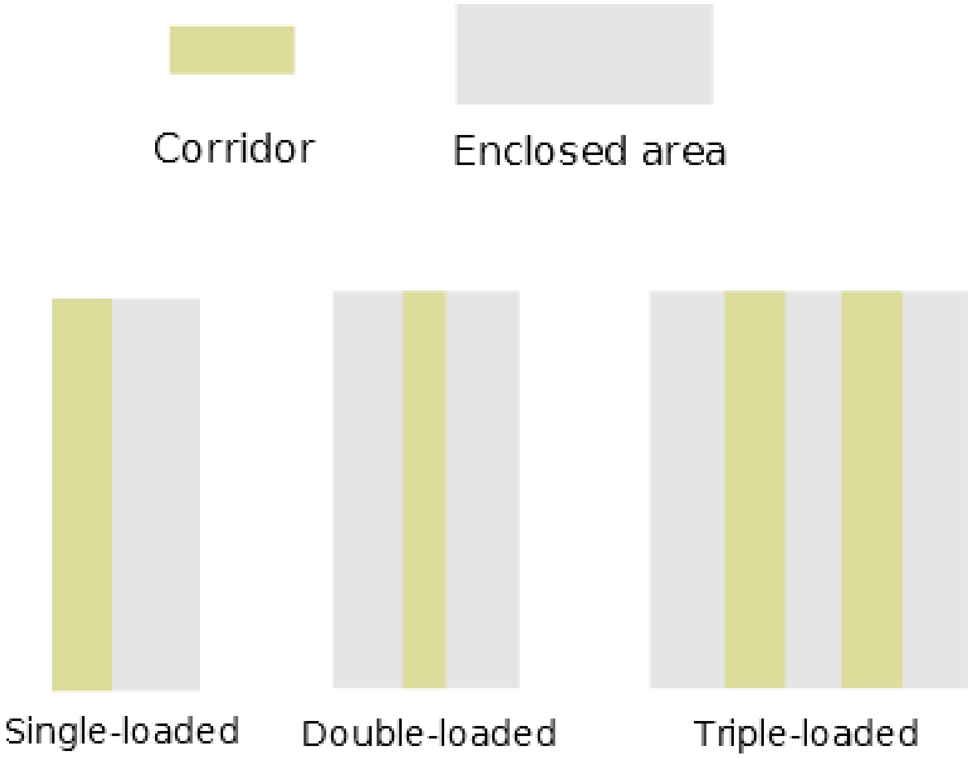


**Figure 7.1.** Three typical plans of research buildings.

### 7.3.2 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 7.2. 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 (Braun, Grömling 2005). 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.

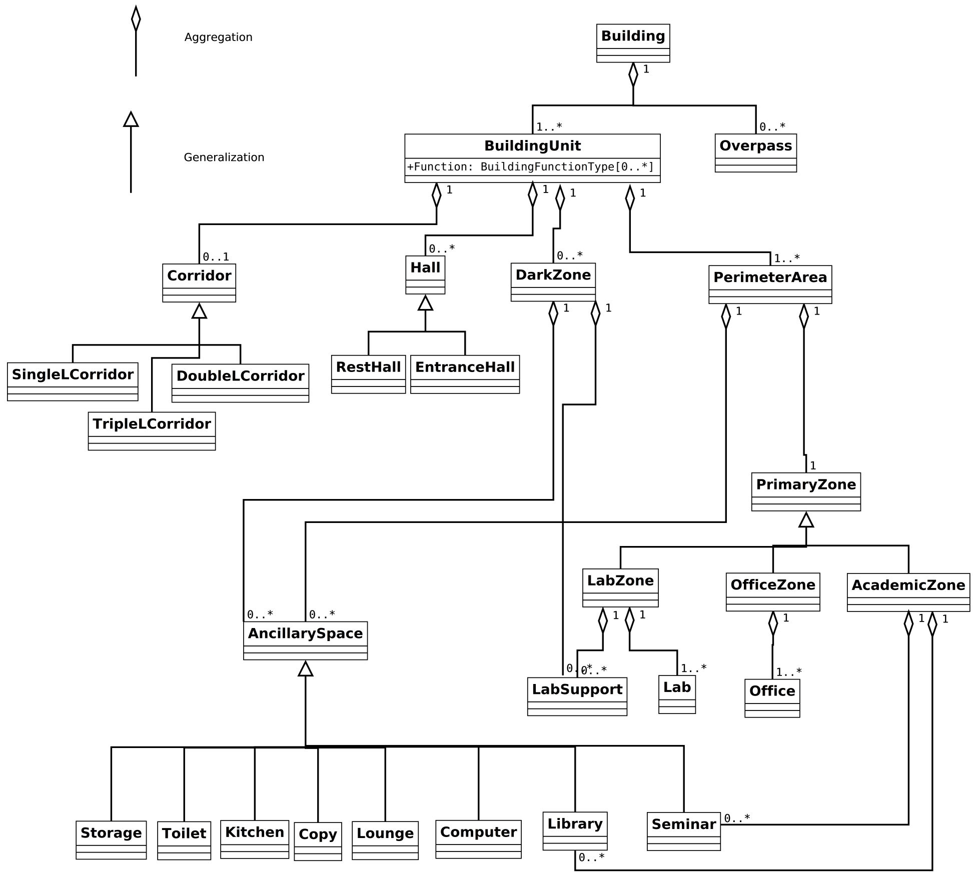


**Figure 7.2.** Semantic division of research buildings.

We take an example in Figure 7.3 to explain the semantic division of research buildings. Figure 7.3 (1) shows a building consists of two adjacent building units. In Figure 7.3 (2), 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 7.3 (3), 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 7.3 (4), 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.

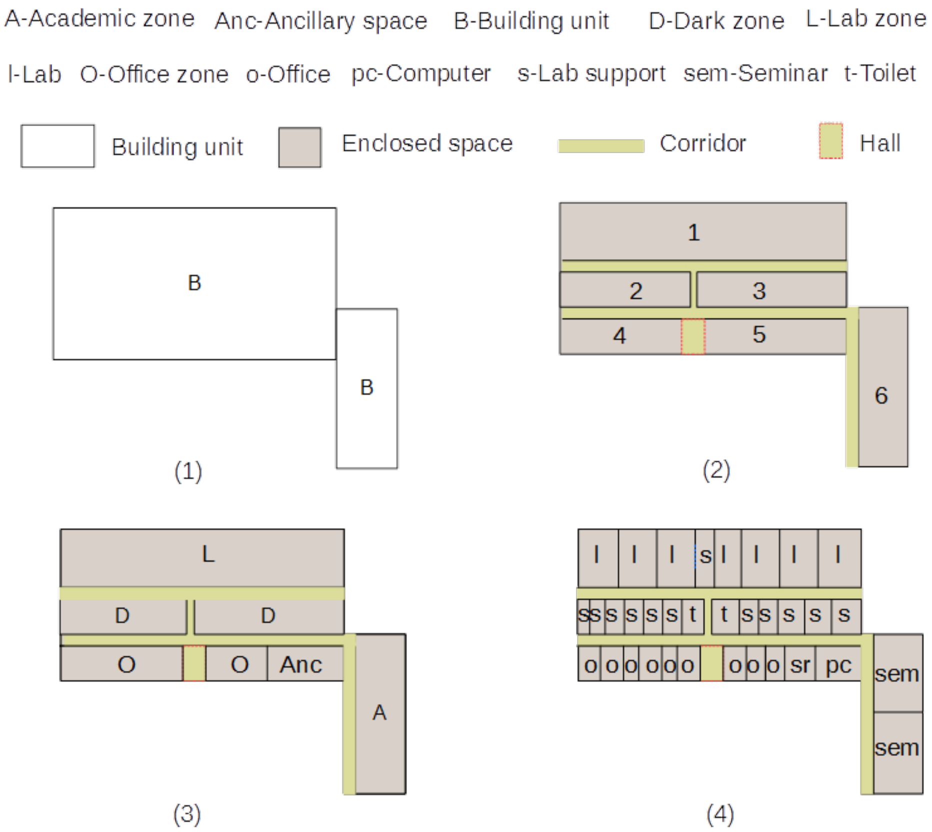


**Figure 7.3.** An example of semantic division of research buildings. (1) two building units in a floor; (2) corridors, halls, and enclosed zones in each building unit; (3) type of each zone; (4) single room types in each zone.

## 7.3.3 Constrained attribute grammar

Equation 7.1 formulates a typical rule of constrained attribute grammars (Boulch et al. 2013; Deransart and Jourdan 1990; Deransart et al. 1988). $p$ 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.

$$p : Zz \rightarrow X_1x_1, X_2x_2, ...X_kx_k\langle Constraints\rangle\langle Attribute\rangle \tag{7.1}$$

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(office,k) o* defines a set of ($k$) office objects, denoted by $o$.

### 7.3.4 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 (Braun, Grömling 2005; Hain 2003; Klonk 2016; Watch 2002).

$edgeAdj(a, b)$: 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.

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

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

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

$formFullArea(\{a_1, a_2, a_k\})$: Multiple objects form a complete area (e.g., a perimeter area or central dark zone), including all the primitive rooms and internal doors. Figure 7.4 illustrates the defined predicates.

### 7.3.5 Defined rules

We define 16 rules in total, which can be found in the Appendix B. Note that '|' denotes the OR operation. The objects in the rules correspond to the objects in Figure 7.2. 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 7.2, respectively. These rules are described as follows:

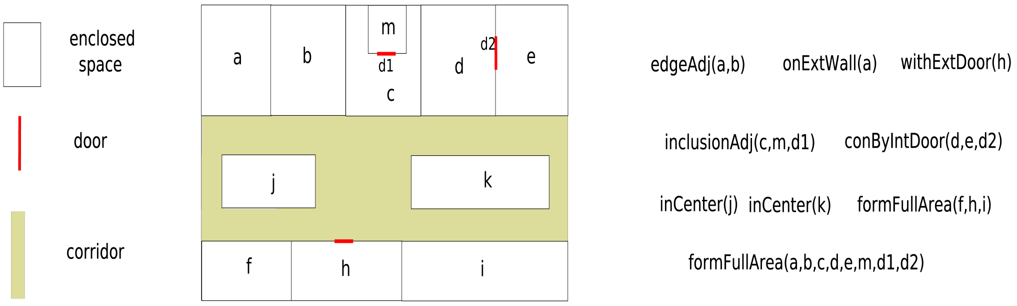


**Figure 7.4.** Predicates used in this work.

**B1**: 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.

**B2**: 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.

**B3**: Toilet, Copy, Storage, Kitchen, Lounge, Computer, Lecture, and Library objects are interpreted as Ancillary objects.

**B4**: 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.

**B5**: A couple of lecture objects that are adjacent or connected by internal doors can be interpreted as an academic Zone.

**B6**: A Library object is interpreted as an academic Zone.

**B7**: 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.

**B8**: A LGroup object is generated by merging at least one Lab object and optional Support objects when they are connected by internal doors.

**B9**: A lab Zone is generated by merging multiple adjacent LGroup objects.

**B10**: A room object $r^p$with an optional internal room rs 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.

**B11**: An office Zone can be generated by merging multiple Office objects if they are adjacent or connected through internal doors.

**B12**: 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.

**B13**: 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*.

**B14**: 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*.

**B15**: 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.

**B16**: A *Building* object can be generated by combining all the *BUnit* objects if they are adjacent.

## 7.4 Algorithm of inferring room types

### 7.4.1 Workflow

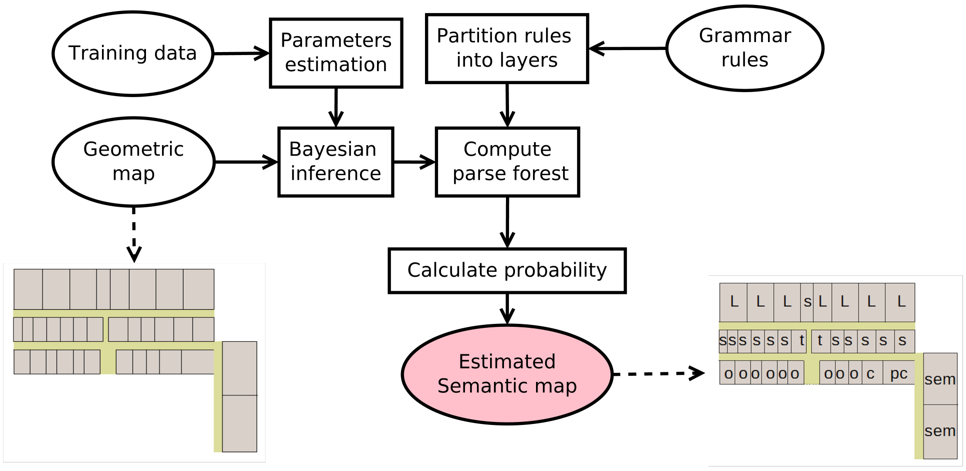


**Figure 7.5.** Workflow of proposed algorithm.

The workflow of the proposed method is depicted in Figure 7.5. The input consist of mainly 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 (Mura et al. 2014; Becker et al. 2015). 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.

### 7.4.2 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 are 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 the other edge. For the rooms not located at external walls or located at multiple walls, the width and length follow their original definitions.

Given the geometric properties of a room, we can calculate the initial probability of belonging the room 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 $w$, $x$, and $a$ denote the width, length, and area, respectively. We use $t$ to denote the type of rooms. Thus, the probability can be estimated by Equation 7.2.

$$p(t \mid x) = \frac{p(x \mid t)p(t)}{p(x)} \tag{7.2}$$

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) $p(x \mid t)p(t)$ over all $t$, and plays the role of an ignorable normalizing constant. 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:

$$p(x \mid t) = \frac{exp(-\frac{1}{2}(x - u_t)^T \sum_t^{-1}(x - u_t))}{\sqrt{(2\pi)^k}} \tag{7.3}$$

In the equation, $u_t$ is a 3-vector, denoting the mean value of the geometric properties of rooms in type $t$. $\sum_t$ 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 belonging the room 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.

### 7.4.3 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.

#### 7.4.3.1 Partition Grammar Rules into Layers

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.

Certain rules have more than one right-hand object, such as rule $\bar{r} : Zz \to Xx, Yy$. 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 $X$ and $Y$ as the left-hand object should be applied ahead of rule $\bar{r}$. Then, we define that rule $\bar{r}$ dependents 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 dependent 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.

#### 7.4.3.2 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 7.4.2 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 7.6) 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 $r_1$,$r_2$,$r_3$ , and $d$, respectively. In Figure 7.7 (a), the initial structure of a tree is the four primitives, playing the role of leaf nodes. In the second step, $r_1$,$r_2$, and $r_3$ are interpreted as three offices denoted by $O_1$,$O_2$, and $O_3$, respectively by applying rule **B10**. Next, $O_1$,$O_2$,$O_3$, and $d$ are merged into a Zone object, denoted by *Zone1* by applying rule **B11**. Finally, a Building object can be created by applying rules **B13**, **B14**, and **B16** successively. Similarly, another two trees can be created as shown in Figure 7.7 (b),(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 $O_1$ 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 7.8), where each node points to its children that form the node.

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,

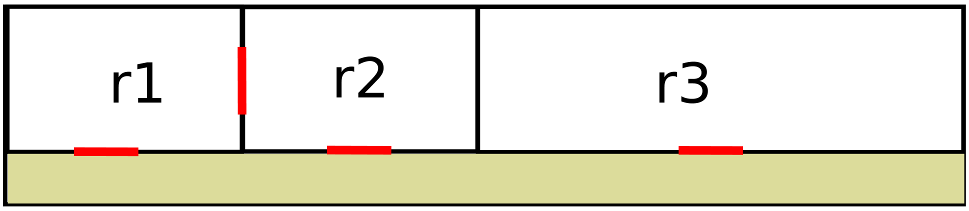


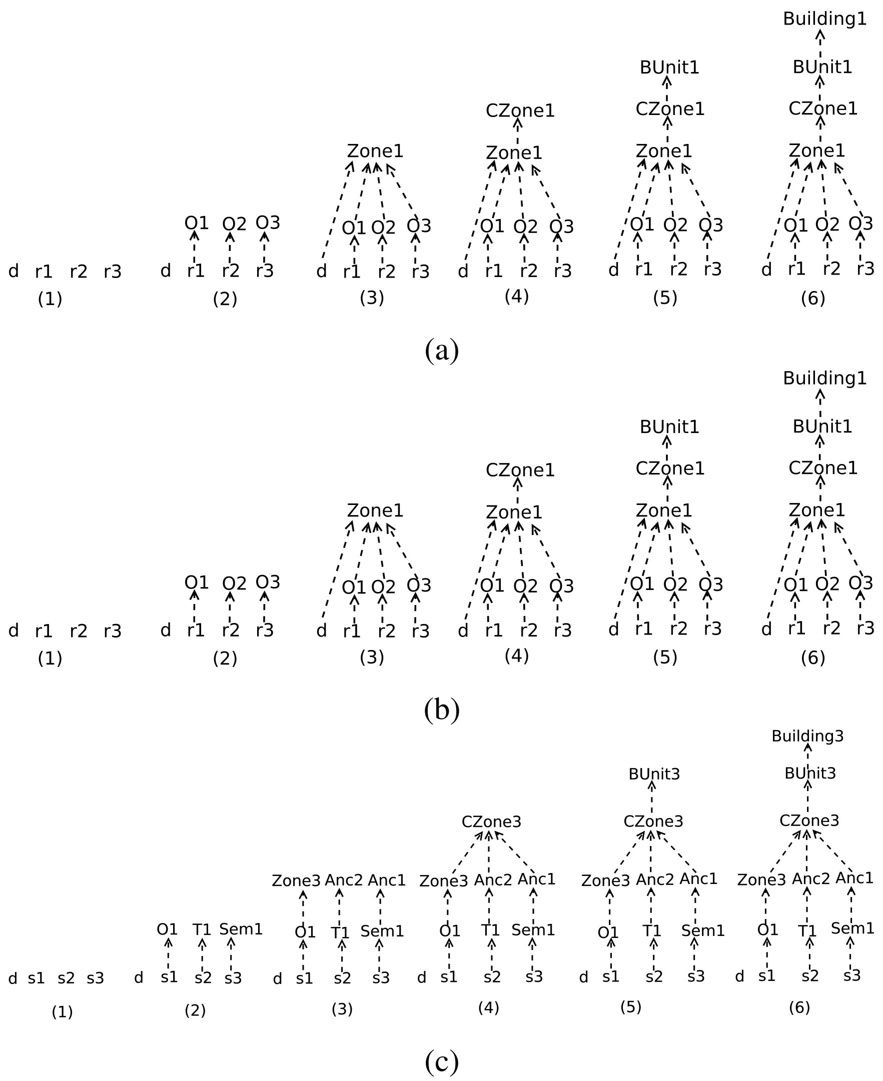
**Figure 7.6.** A simplified floor plan with three rooms.

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 7.7 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 7.8. The other two trees are valid parse trees.

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

**Algorithm 5** Parse Forest Computation

```
procedure COMPUTEPARSINGFOREST($(R_i)_{1 \le i \le h}, G$) ▷ $R$ denotes the partitioned rules with $h$ layers; $G$ denotes all the primitives
    $O \leftarrow initializeObjects(G)$                ▷ initialize object list with $G$
    for $i = 1 : h$ do
        for each rule $\bar{r} \in R_i$ do
            $O \leftarrow applyRule(\bar{r}, O)$
    $F.Child\_list \leftarrow null$
    for each object $\bar{o} \in O$ do
        if the type of $\bar{o}$ is a Building then
            $F.Child\_list \leftarrow F.Child\_list \cup \{\bar{o}\}$
    return $F$
```



(a)

(b)

(c)

**Figure 7.7.** 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.

### 7.4.4 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≤$i$≤$t$ 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≤$k$≤$m$, 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 $\sum_{k=1}^{m} \tilde{p}_k / \sum_{i=1}^{t} \bar{p}_k$. Assume that the true type of $r_1$,$r_2$, and $r_3$ in Figure 7.6 are office ($O$), office ($O$), and lecture ($Sem$), respectively, and the forest in the right side of Figure 7.8 is the estimated forest. We denote the assigned probability value to nodes *O1*,*O2*,*O3*, and *Sem1* by $\ddot{p}_1$,

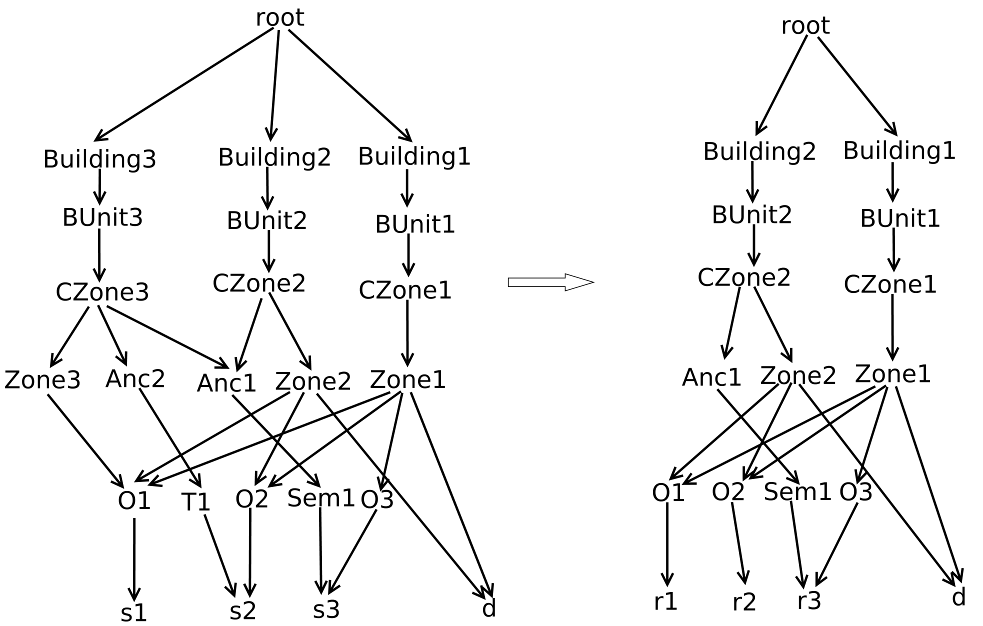


**Figure 7.8.** Pruning incomplete trees from parse forest.

$\ddot{p}_2$, $\ddot{p}_3$, and $\ddot{p}_4$, 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 $(\ddot{p}_1 + \ddot{p}_1)/(\ddot{p}_1 + \ddot{p}_1)$, $(\ddot{p}_2 + \ddot{p}_2)/(\ddot{p}_2 + \ddot{p}_2)$, and $\ddot{p}_4/(\ddot{p}_4 + \ddot{p}_4)$, 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.

## 7.5 Experiments

### 7.5.1 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 7.9. Further, for each room type, we can calculate the covariance matrix (3-by-3) and mean vector of the area, width, and length.

### 7.5.2 Testbeds

We choose 15 buildings distributed in two campuses of Heidelberg University as the test bed, as shown in Figure 7.10. 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 7.11.

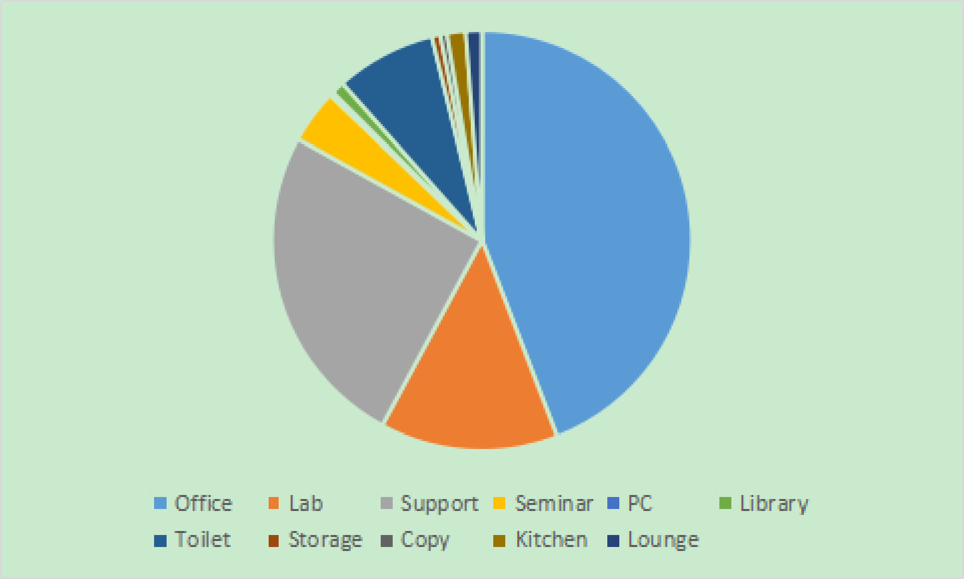


**Figure 7.9.** Proportion of different room types in training rooms.

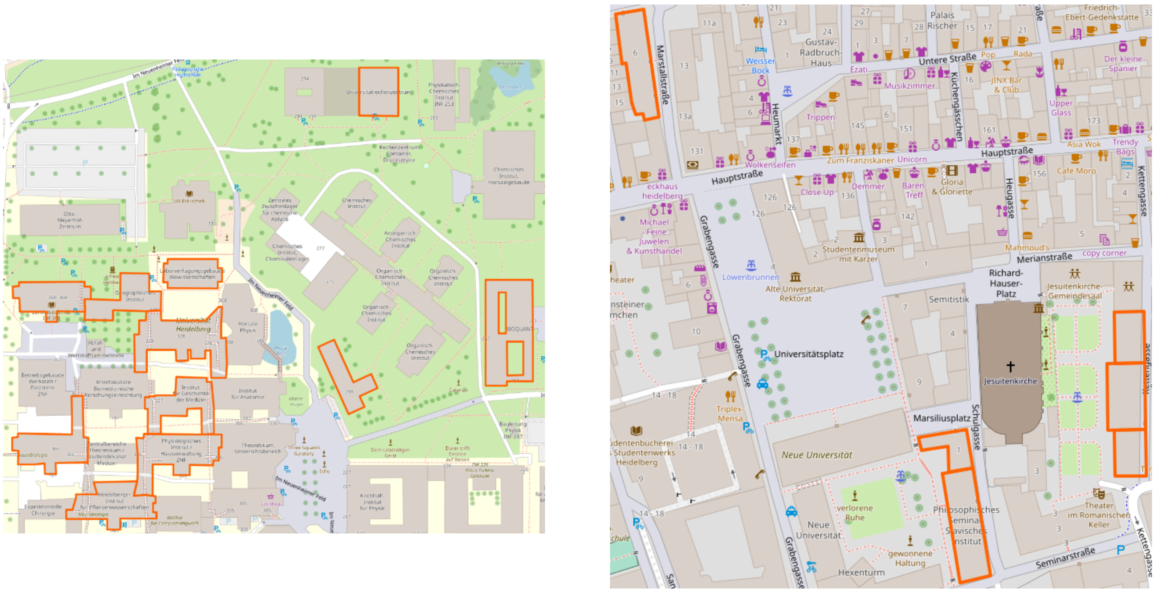


**Figure 7.10.** Distribution of test buildings.

Table 7.1 shows the number of lab-centered, office-centered, and academic-centered building units in each floor plan.

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

**Table 7.1.** 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 |

to detect them by current indoor mapping solutions (Nikoohemat et al. 2017; Gao et al. 2017; Jiang et al. 2013). 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).

### 7.5.3 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 7.2. 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 7.11.

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 7.2. For most of the floor plans, it takes about 10s 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

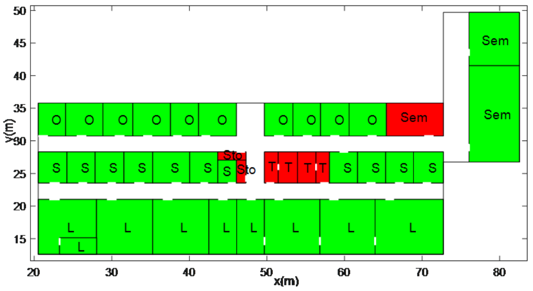
Bayesian inference at the beginning of construction of the forest. This avoids an exponential growth of the number of parse trees. However, for floor-plan (g), it takes nearly 8 minutes. This is because one of the zones contains 23 rooms in total, with 18 connected, which enormously increases the number of possible combinations of room types.

**Table 7.2.** Identification accuracy of each floor plan.

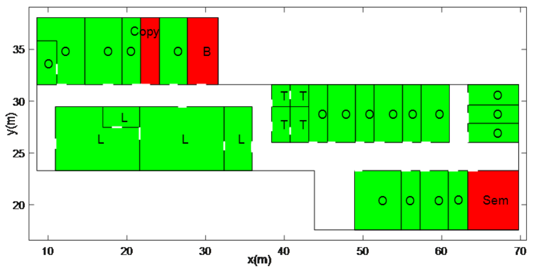
| Floor Plan | Identification Accuracy | Number of Rooms | Time Consumption(s) |
|---|---|---|---|
| Floor plan (a) | 0.82 | 39 | 8.05 |
| Floor plan (b) | 0.90 | 29 | 3.93 |
| Floor plan (c) | 0.80 | 10 | 2.40 |
| Floor plan (d) | 0.95 | 21 | 3.10 |
| Floor plan (e) | 1.00 | 43 | 27.02 |
| Floor plan (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 confusion matrix is shown in Figure 7.12, 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.

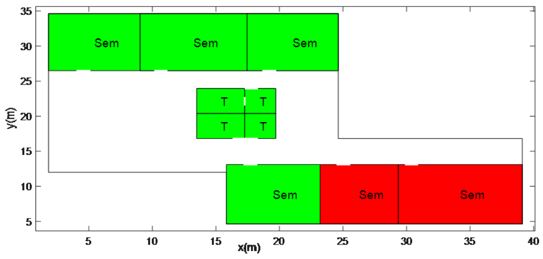
Figure 7.13 shows the constructed parse tree with the highest probability on floor-plan (d). *r*, *d*, *S*, *O*, *T*, *Sem*, *Anc*, *L*, and *LG* denote the objects of room, door, Support, Office, Toilet, Lecture, Ancillary, Lab, and LGroup in the rules, respectively. The text in the parentheses denotes 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.


y(m)
x(m)
Sem
Sem
O O O O O O O O O O Sem
S S S S S S Sto S Sto T T T T S S S S
L L L L L L L L L L
20 30 40 50 60 70 80
15 20 25 30 35 40 45 50

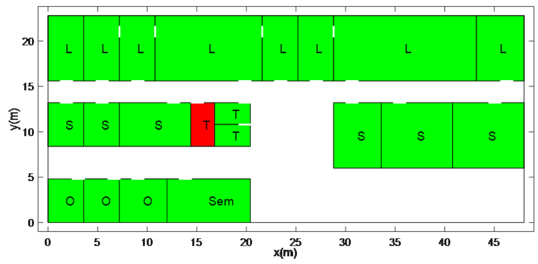
(a)


y(m)
x(m)
O O O O Copy O B
T T T T O O O O O O O O O
L L L L
O O O O Sem
10 20 30 40 50 60 70
20 25 30 35

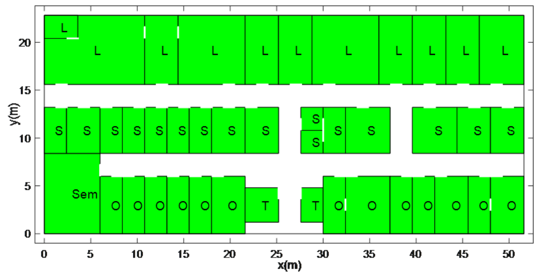
(b)


y(m)
x(m)
Sem Sem Sem
T T T T
Sem Sem Sem
5 10 15 20 25 30 35 40
5 10 15 20 25 30 35

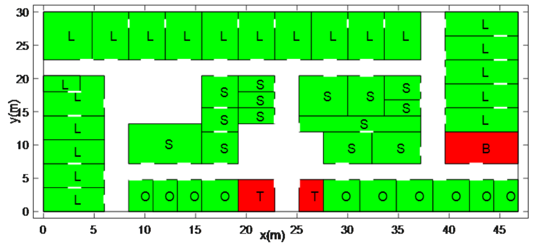
(c)


y(m)
x(m)
L L L L L L L L
S S S T T T S S S
O O O Sem
0 5 10 15 20 25 30 35 40 45
0 5 10 15 20

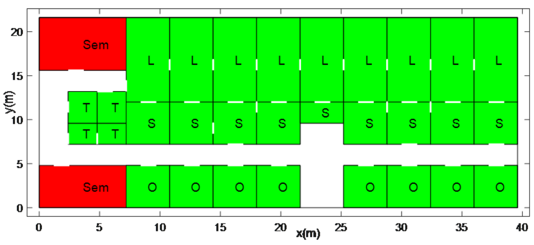
(d)


y(m)
x(m)
L L L L L L L L L L L L L
S S S S S S S S S S S S S S S S
Sem O O O O O O T T O O O O O O O O
0 5 10 15 20 25 30 35 40 45 50
0 5 10 15 20

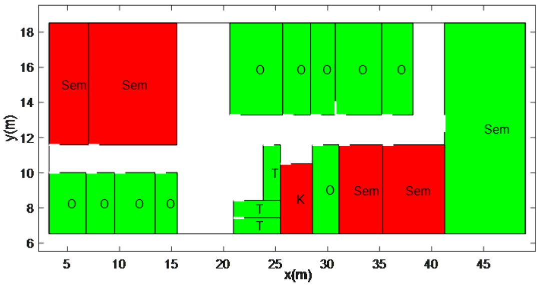
(e)


y(m)
x(m)
L L L L L L L L L L L
L L L L L L L L L L
S S S S S S S S S S S S S
B
O O O O T T O O O O O O
0 5 10 15 20 25 30 35 40 45
0 5 10 15 20 25 30

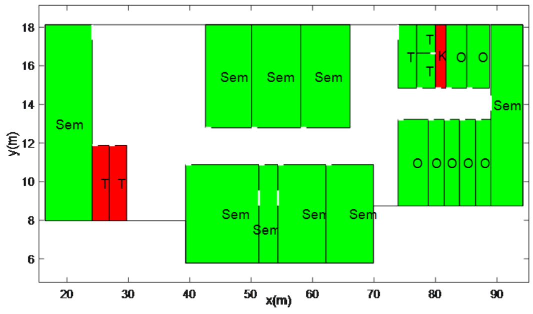
(f)


y(m)
x(m)
Sem
L L L L L L L L L
T T T T
S S S S S S S S S
Sem O O O O O O O O
0 5 10 15 20 25 30 35 40
0 5 10 15 20

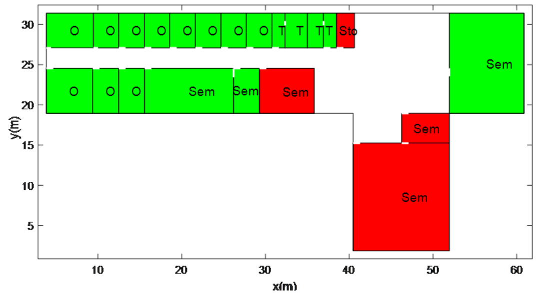
(g)


y(m)
x(m)
Sem Sem
O O O O O
Sem
T K O Sem Sem
O O O O T T
5 10 15 20 25 30 35 40 45
6 8 10 12 14 16 18

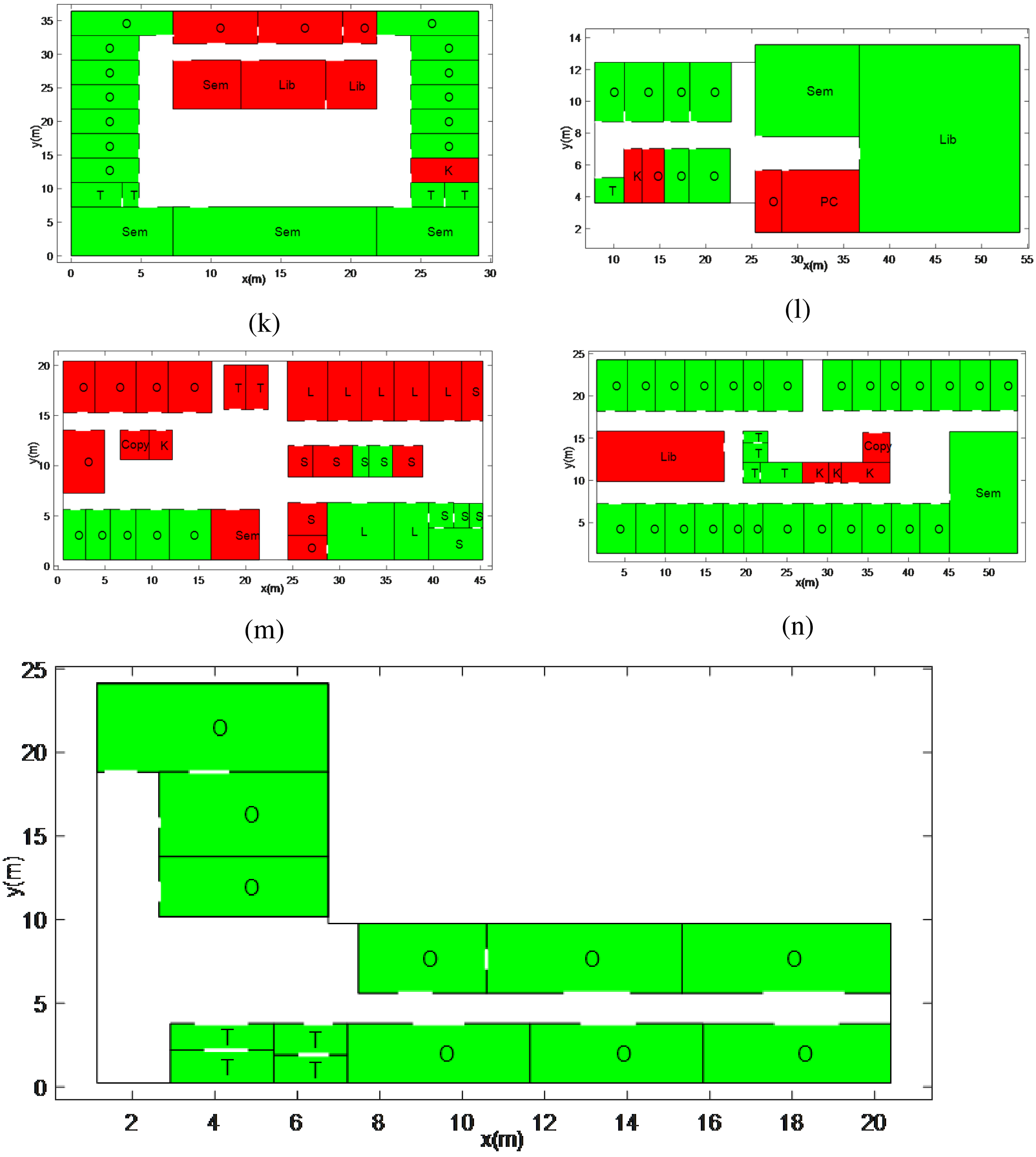
(h)


y(m)
x(m)
Sem Sem Sem
T T K O O T
Sem
Sem
T T
O O O O O
Sem Sen Sem Sem
20 30 40 50 60 70 80 90
6 8 10 12 14 16 18

(i)


y(m)
x(m)
O O O O O O O O T T TT Sto
Sem
O O O Sem Sem Sem
Sem
Sem
10 20 30 40 50 60
5 10 15 20 25 30

(j)



(k)

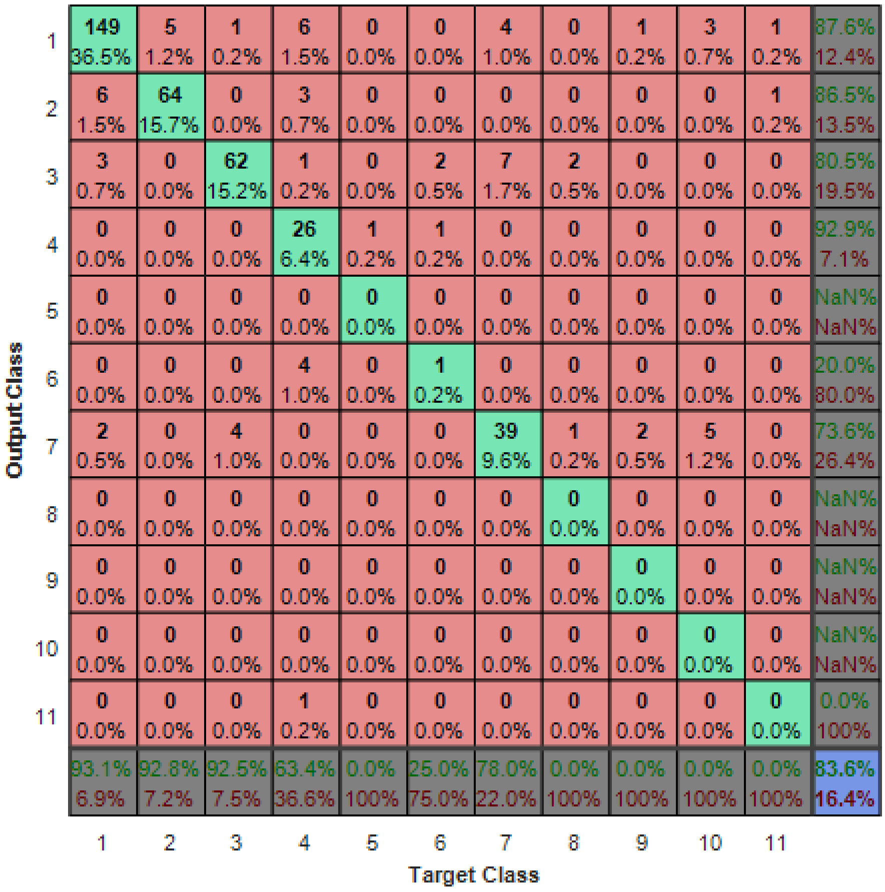
(l)

(m)

(n)

(o)

**Figure 7.11.** Floor plans (a–o) used for test.


Output Class
Target Class

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 149<br>36.5% | 5<br>1.2% | 1<br>0.2% | 6<br>1.5% | 0<br>0.0% | 0<br>0.0% | 4<br>1.0% | 0<br>0.0% | 1<br>0.2% | 3<br>0.7% | 1<br>0.2% | 87.6%<br>12.4% |
| 2 | 6<br>1.5% | 64<br>15.7% | 0<br>0.0% | 3<br>0.7% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.2% | 86.5%<br>13.5% |
| 3 | 3<br>0.7% | 0<br>0.0% | 62<br>15.2% | 1<br>0.2% | 0<br>0.0% | 2<br>0.5% | 7<br>1.7% | 2<br>0.5% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 80.5%<br>19.5% |
| 4 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 26<br>6.4% | 1<br>0.2% | 1<br>0.2% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 92.9%<br>7.1% |
| 5 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | NaN%<br>NaN% |
| 6 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 4<br>1.0% | 0<br>0.0% | 1<br>0.2% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 20.0%<br>80.0% |
| 7 | 2<br>0.5% | 0<br>0.0% | 4<br>1.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 39<br>9.6% | 1<br>0.2% | 2<br>0.5% | 5<br>1.2% | 0<br>0.0% | 73.6%<br>26.4% |
| 8 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | NaN%<br>NaN% |
| 9 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | NaN%<br>NaN% |
| 10 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | NaN%<br>NaN% |
| 11 | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.2% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0.0%<br>100% |
| | 93.1%<br>6.9% | 92.8%<br>7.2% | 92.5%<br>7.5% | 63.4%<br>36.6% | 0.0%<br>100% | 25.0%<br>75.0% | 78.0%<br>22.0% | 0.0%<br>100% | 0.0%<br>100% | 0.0%<br>100% | 0.0%<br>100% | 83.6%<br>16.4% |

**Figure 7.12.** Confusion matrix of classification result.

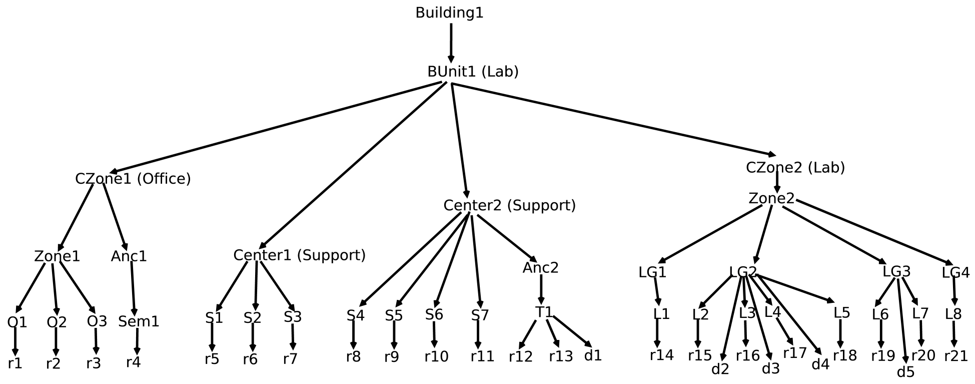


**Figure 7.13.** Parse tree with highest probability for floor-plan.

## 7.6 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 (De la Higuera 2010; D'Ulizia et al. 2011) 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 (Russakovsky et al. 2015) 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 (Müller et al. 2006).

## 7.7 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.

# Reference

Ahmed, S., Liwicki, M., Weber, M., and Dengel, A. (2011). Improved automatic analysis of architectural floor plans. In *2011 International Conference on Document Analysis and Recognition*, pages 864–869. IEEE.

Ahmed, S., Liwicki, M., Weber, M., and Dengel, A. (2012). Automatic room detection and room labeling from architectural floor plans. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 339–343. IEEE.

Alzantot, M. and Youssef, M. (2012). Crowdinside: automatic construction of indoor floorplans. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 99–108.

Ambruş, R., Claici, S., and Wendt, A. (2017). Automatic room segmentation from unstructured 3-d data of indoor environments. *IEEE Robotics and Automation Letters*, 2(2):749–756.

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S. (2016). 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543.

Azhar, S. (2011). Building information modeling (bim): Trends, benefits, risks, and challenges for the aec industry. *Leadership and management in engineering*, 11(3):241–252.

Becker, S., Peter, M., and Fritsch, D. (2015). Grammar-supported 3d indoor reconstruction from point clouds for" as-built" bim. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2.

Boulch, A., Houllier, S., Marlet, R., and Tournaire, O. (2013). Semantizing complex 3d scenes using constrained attribute grammars. In *Computer Graphics Forum*, volume 32, pages 33–42. Wiley Online Library.

Braun, H. and Grömling, D. (2005). *Research and technology buildings: A design manual*. Walter de Gruyter.

Chen, S., Li, M., Ren, K., and Qiao, C. (2015). Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos. In *2015 IEEE 35th International conference on distributed computing systems*, pages 1–10. IEEE.

De la Higuera, C. (2010). *Grammatical inference: learning automata and grammars*. Cambridge University Press.

de las Heras, L.-P., Ahmed, S., Liwicki, M., Valveny, E., and Sánchez, G. (2014). Statistical segmentation and structural recognition for floor plan interpretation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):221–237.

de las Heras, L.-P., Mas, J., Sánchez, G., and Valveny, E. (2011). Notation-invariant patch-based wall detector in architectural floor plans. In *International Workshop on Graphics Recognition*, pages 79–88. Springer.

de las Heras, L.-P., Terrades, O. R., Robles, S., and Sánchez, G. (2015). Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(1):15–30.

Dehbi, Y., Hadiji, F., Gröger, G., Kersting, K., and Plümer, L. (2017). Statistical relational learning of grammar rules for 3d building reconstruction. *Transactions in GIS*, 21(1):134–150.

Deransart, P. and Jourdan, M. (1990). Attribute grammars and their applications. *Lecture Notes in Computer Science*, 461.

Deransart, P., Jourdan, M., and Lorho, B. (1988). *Attribute grammars: definitions, systems and bibliography*, volume 323. Springer Science & Business Media.

Dodge, S., Xu, J., and Stenger, B. (2017). Parsing floor plan images. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 358–361. IEEE.

Dosch, P., Tombre, K., Ah-Soon, C., and Masini, G. (2000). A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 3(2):102–116.

D'Ulizia, A., Ferri, F., and Grifoni, P. (2011). A survey of grammatical inference methods for natural language learning. *Artificial Intelligence Review*, 36(1):1–27.

Elhamshary, M., Basalmah, A., and Youssef, M. (2016). A fine-grained indoor location-based social network. *IEEE Transactions on Mobile Computing*, 16(5):1203–1217.

Elhamshary, M. and Youssef, M. (2015). Semsense: Automatic construction of semantic indoor floorplans. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–11. IEEE.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2009). Reconstructing building interiors from images. In *2009 IEEE 12th International Conference on Computer Vision*, pages 80–87. IEEE.

Gao, R., Zhao, M., Ye, T., Ye, F., Wang, Y., Bian, K., Wang, T., and Li, X. (2014). Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 249–260.

Gao, R., Zhou, B., Ye, F., and Wang, Y. (2017). Knitter: Fast, resilient single-user indoor floor plan construction. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE.

Gimenez, L., Robert, S., Suard, F., and Zreik, K. (2016). Automatic reconstruction of 3d building models from scanned 2d floor plans. *Automation in Construction*, 63:48–56.

Hain, W. (2003). *Laboratories: A Briefing and Design Guide*. Taylor & Francis.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. (2012). Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663.

Hu, X., Fan, H., Zipf, A., Shang, J., and Gu, F. (2017). A conceptual framework for indoor mapping by using grammars. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:335.

Ikehata, S., Yang, H., and Furukawa, Y. (2015). Structured indoor modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1323–1331.

Jiang, Y., Xiang, Y., Pan, X., Li, K., Lv, Q., Dick, R. P., Shang, L., and Hannigan, M. (2013). Hallway based automatic indoor floorplan construction using room fingerprints. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 315–324.

Kang, H.-K. and Li, K.-J. (2017). A standard indoor spatial data model—ogc indoorgml and implementation approaches. *ISPRS International Journal of Geo-Information*, 6(4):116.

Khoshelham, K. and Díaz-Vilariño, L. (2014). 3d modelling of interior spaces: Learning the language of indoor architecture. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):321.

Kim, J.-S., Yoo, S.-J., and Li, K.-J. (2014). Integrating indoorgml and citygml for indoor space. In *International Symposium on Web and Wireless Geographical Information Systems*, pages 184–196. Springer.

Klonk, C. (2016). *New laboratories: Historical and critical perspectives on contemporary developments*. Walter de Gruyter GmbH & Co KG.

Kolbe, T. H. (2009). Representing and exchanging 3d city models with citygml. In *3D geo-information sciences*, pages 15–31. Springer.

Li, K.-J., Kim, T.-H., Ryu, H.-G., and Kang, H.-K. (2015). Comparison of citygml and indoorgml-a use-case study on indoor spatial information construction at real sites. *Spatial Information Research*, 23(4):91–101.

Liu, Z. and von Wichert, G. (2014a). Extracting semantic indoor maps from occupancy grids. *Robotics and Autonomous Systems*, 62(5):663–674.

Liu, Z. and von Wichert, G. (2014b). A generalizable knowledge framework for semantic indoor mapping based on markov logic networks and data driven mcmc. *Future Generation Computer Systems*, 36:42–56.

Luperto, M. and Amigoni, F. (2016). Exploiting structural properties of buildings towards general semantic mapping systems. In *Intelligent Autonomous Systems 13*, pages 375–387. Springer.

Luperto, M., Li, A. Q., and Amigoni, F. (2013). A system for building semantic maps of indoor environments exploiting the concept of building typology. In *Robot Soccer World Cup*, pages 504–515. Springer.

Luperto, M., Riva, A., and Amigoni, F. (2017). Semantic classification by reasoning on the whole structure of buildings using statistical relational learning techniques. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2562–2568. IEEE.

Macé, S., Locteau, H., Valveny, E., and Tabbone, S. (2010). A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 167–174.

Mitchell, W. J. (1990). *The logic of architecture: Design, computation, and cognition*. MIT press.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., and Van Gool, L. (2006). Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, pages 614–623.

Mura, C., Mattausch, O., Villanueva, A. J., Gobbetti, E., and Pajarola, R. (2014). Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics*, 44:20–32.

Nikoohemat, S., Peter, M., Elberink, S. O., and Vosselman, G. (2017). Exploiting indoor mobile laser scanner trajectories for semantic interpretation of point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Philipp, D., Baier, P., Dibak, C., Dürr, F., Rothermel, K., Becker, S., Peter, M., and Fritsch, D. (2014). Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 139–147. IEEE.

Pintore, G. and Gobbetti, E. (2014). Effective mobile mapping of multi-room indoor structures. *The visual computer*, 30(6-8):707–716.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.

Rosser, J. F., Smith, G., and Morley, J. G. (2017). Data-driven estimation of building interior plans. *International Journal of Geographical Information Science*, 31(8):1652–1674.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Sankar, A. and Seitz, S. (2012). Capturing indoor scenes with smartphones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 403–412.

Santos, R., Costa, A. A., and Grilo, A. (2017). Bibliometric analysis and review of building information modelling literature published between 2005 and 2015. *Automation in Construction*, 80:118–136.

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

Watch, D. D. (2002). *Building type basics for research laboratories*, volume 5. John Wiley & Sons.

Xiong, X., Adan, A., Akinci, B., and Huber, D. (2013). Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in construction*, 31:325–337.

Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., and Aboutanios, E. (2016). Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys & Tutorials*, 19(2):1327–1346.

Youssef, M. (2015). Towards truly ubiquitous indoor localization on a worldwide scale. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–4.

Yue, K., Krishnamurti, R., and Grobler, F. (2012). Estimating the interior layout of buildings using a shape grammar to capture building style. *Journal of computing in civil engineering*, 26(1):113–130.

Zhang, D., Xia, F., Yang, Z., Yao, L., and Zhao, W. (2010). Localization technologies for indoor human tracking.

In *2010 5th international conference on future information technology*, pages 1–6. IEEE.

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

# 8. Room semantics inference using random forest and relational graph convolutional network: A case study of research building

## Article information

| | |
|---|---|
| Authors | Xuke Hu, Hongchao Fan, Alexey Noskov, Zhiyong Wang, Zipf Alexander, Fuqiang Gu, Jianga Shang |
| Journal | Transactions in GIS |
| History | Received: 28 October 2019<br>Revised: 7 March 2020 |
| Reference | Hu, X., Fan, H., Noskov, A., Wang, Z., Zipf, A., Gu, F., Shang, J. (2020). Room Semantics Inference Using Random Forest and Relational Graph Convolutional Network: A Case Study of Research Building. Transactions in GIS. (Submitted) |

## Abstract

Semantically-rich maps are the foundation of indoor location-based services. Many map providers such as OpenStreetMap and automatic mapping solutions focus on the representation and detection of geometric information (e.g., shape of room) and a few semantics (e.g., stairs and furniture) but neglect room usage. To mitigate the issue, this work proposes a general room tagging method for public buildings, which can benefit both existing map providers and automatic mapping solutions by inferring the missing room usage based on indoor geometric maps. Two kinds of statistical learning-based room tagging methods are adopted: traditional machine learning (e.g., random forest) and deep learning, specifically Relational Graph Convolutional Network (R-GCN), based on the geometric properties (e.g., area), topological relationships (e.g., adjacent and inclusion), and spatial distribution characteristics of rooms. In the machine learning-based approach, a bi-directional beam search strategy is proposed to deal with the issue that the tag of a room depends on the tag of its neighbors in an undirected room sequence. In R-GCN-based approach, useful properties of neighboring nodes (rooms) in the graph is automatically gathered to classify the nodes. Research buildings are taken as examples to evaluate the proposed approaches based on 130 floor plans with 3330 rooms by using five-fold cross validation. The conducted experiments show that the random forest-based approach achieves a higher tagging accuracy (0.85) than R-GCN (0.79).

**Keywords:** Indoor mapping; Room usage tagging; Semantic inference; Random forest; Relational Graph Convolutional Network;

## 8.1 Introduction

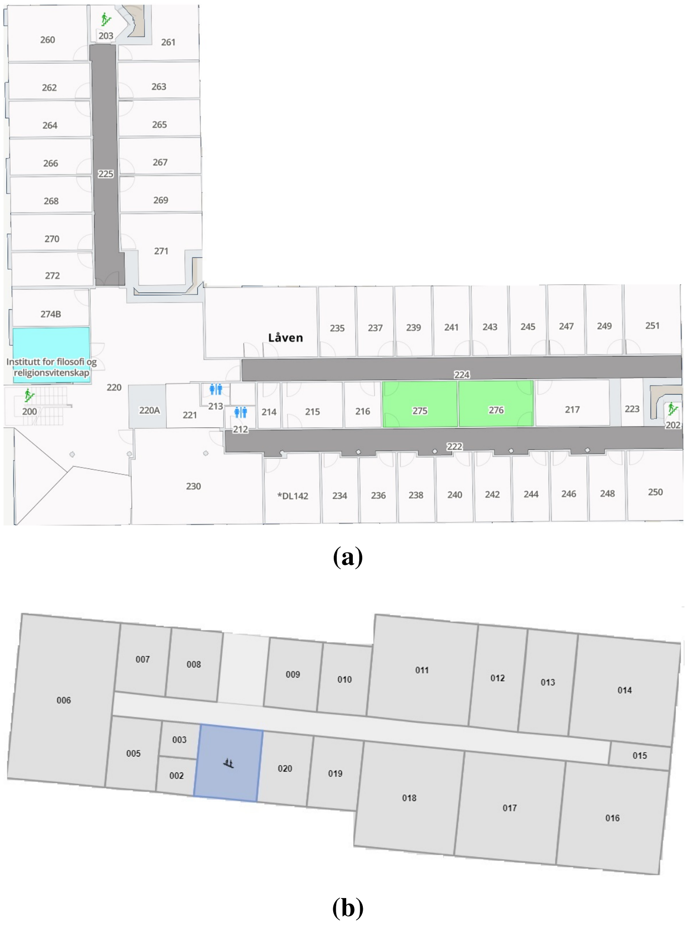
Nowadays, indoor mobile applications are becoming popular, such as way-finding, location-based recommendation, and ambient assisted living and health applications (Huang and Gartner 2009; Kattenbeck 2015; Yassin et al. 2016). People use them because they spend most of their time indoors, such as offices, universities, and shopping malls. Semantically-rich indoor maps that contain the usage of rooms (e.g., office, restaurant, or book shop) are an indispensable part of indoor location-based services (Elhamshary and Youssef 2015). Existing indoor models such as building information modeling (BIM), industry foundation classes (IFC), and computer-aided design and drafting (CAD), and GIS systems (e.g., ArcGIS and Google Maps) provide rich semantic information, including doors, walls, corridors, staircases, and the usage of rooms. However, currently, only a small fraction of millions of indoor environments are mapped (Gao et al. 2014); let alone the usage of rooms.

There are two main solutions to collecting indoor maps. The first is manually surveying and uploading the maps by companies or volunteers, named manual mapping, such as OSM and MazeMap. Thousands of indoor maps of public buildings (e.g., hospitals and research buildings at universities) in Europe have been published on MazeMap. On OSM, tons of indoor spatial entities in public buildings have been tagged by volunteers, such as shopping malls, office buildings, and airports. However, the room usage (type) is normally missing on these maps. Figure 8.1 shows the indoor map of two public buildings without usage of rooms on MazeMap and OSM, respectively. The second is automatic mapping, i.e., reconstructing the indoor map from sensor measurements, such as LIDAR point cloud (Armeni et al. 2016; Qi et al. 2017; Xiong et al. 2013), image (Ambruş et al. 2017; de las Heras et al. 2015; Furukawa et al. 2009), and volunteers' trace (Alzantot and Youssef 2012; Gao et al. 2014), and by digitalizing scanned maps (de las Heras et al. 2015; Dodge et al. 2017; Dosch et al. 2000). These automatic solutions mainly focus on the detection of geometries with limited semantic information (e.g., doors, corridors, stairs, and furniture). They are incapable of detecting the room usage (e.g., for digitization-based and traces-based solutions) or ignore this issue (e.g., for image or point cloud-based solutions). Although point cloud-based and image-based solutions are able to identify the room usage based on deep-learning techniques given enough annotated data, they fail to overcome the challenge faced in the manual solution, providing room usage information based on the published geometric maps. To solve the issues. Elhamshary and Youssef (2015)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, the check-in information is only available in popular indoor venues. Most of the indoor venues do not have check-in information. Hu et al. (2019) proposed inferring the room usage of research buildings at universities by using grammars and Bayesian inference based on geometric maps. However, the defined grammar rules can only cover partial research buildings, which cannot be applied in other styles of research buildings.

To resolve the described gap, we propose a more general solution, using traditional machine learning (e.g., random forest (Breiman 2001)) and deep learning (i.e., Relational Graph Convolutional Network (Schlichtkrull et al. 2018)) algorithms to infer the usage of rooms in public buildings based on the geometric maps, which can benefit both the manual and automatic mapping solutions. The input of the proposed method (specifically geometric maps) can be obtained from existing map providers such as OSM or from automatic mapping solutions. In machine learning-based approaches, a bi-directional beam search strategy is proposed to deal with the issue that the tag of a room depends on the tag of its neighbors in an undirected room sequence. In R-GCN-based approach, useful properties of neighboring nodes (rooms) in the graph is automatically gathered to classify the nodes. The idea of the work is inspired by the following intuitions: (1) there exists a close correlation between the geometry and semantics of indoor spaces. Specifically, different space types (semantics) vary in geometric properties. For instance, in office buildings, a seminar room is normally much larger than a private office; (2) there exists a close correlation between the topology and semantics. That is, certain topological relations normally exist among certain space types. For instance, in research buildings, a lab is often adjacent to another lab, connected by internal doors. However, it rarely happens that a lab is adjacent to a toilet, connected by internal doors; (3) there exists a close correlation between the spatial distribution features and semantics. The spatial distribution of certain spaces follows certain principles. For instance, the checkout area is located in the front of supermarkets, while the storage room is generally located in the back.

The main contribution of the work consists of two parts:

(1) Proposing two general statistical-learning based room usage tagging approaches, which can be applied in multiple types of public buildings, such as hospitals, office buildings, and research buildings.



(a)

(b)

**Figure 8.1.** Indoor maps of public buildings published online without room usage.(a)indoor map of a building at universities on MazeMap. (b)indoor map of a building at universities on OSM.

(2) Proving the existence of the correlation among the three spatial elements: geometry, topology, and semantics; the proposed approach can be further extended to enrich other spatial elements, such as geometry and topology given the semantics of spatial elements.

The remainder of this paper is structured as follows: In Section 8.2, we present a relevant literature review. In Section 8.3, we present the workflow of the proposed methods and give the details of each step. In Section 8.4, we evaluate our approaches using 130 floor plans and discuss some issues in Section 8.5. We conclude the paper in Section 8.6.

## 8.2 Related work

**Digitalization-based indoor mapping.** The classical approach of parsing the scanned map or the image of floor plans consists of two stages: primitive detection and semantics recognition (Dosch et al. 2000; Gimenez et al. 2016; Macé et al. 2010). Dosch et al. (2000) proposed an approach to tiling high-resolution images and to segmenting the pixels of thin and thick lines by morphological filtering after separating graphics and texts in the images. Then, the segmented pixels are vectorized into segments by skeletonization. Ahmed et al. (2011) proposed a complete system for automated floor plan analysis. The approach consists of three main stages, i.e., information segmentation, structural analysis, and semantics analysis. Its novelty lies in the preprocessing methods, e.g., the differentiation between thick, medium, and thin lines and the removal of components outside the convex hull of the outer walls, which can increase the performance of the final system. In recent years, machine-learning techniques have been applied to detect the semantic classes (e.g., room, doors, and walls). de las Heras et al. (2014; 2011; 2015) 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. As 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. (2017) 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 scanned 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.

**Measurement-based indoor mapping.** According to the type of used measurements, this group of techniques can be further divided into image-based, trace-based, and point cloud-based. Image-based techniques are cost-effective solutions, requiring mainly cameras. Sankar and Seitz (2012) proposed modeling indoor scenes including offices and houses by using cameras and inertial sensors from smartphones. That allows users to create an accurate 2D and 3D model based on simple interactive photogrammetric modeling. However, it is still a semi-automatic mapping solution and provides only simple semantics, such as rooms. Ikehata et al. (2015) 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. 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. (2013) proposed an approach to estimating 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. Point cloud-based approach can achieves the highest geometric accuracy. Xiong et al. (2013) proposed a method to automatically converting the raw 3D point data into a semantically rich information model. It mainly models the structural components of an indoor environment, such as walls, floors, ceilings, windows, and doorways. Armeni et al. (2016) 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. (2017) proposed a Multilayer Perceptron (MLP) architecture named PointNet on point clouds for 3D classification and segmentation. 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 high-accurate semantic scene understanding. Traces-based solutions assume that users' traces reflect accessible spaces, including unoccupied internal spaces, corridors, and halls. With enough traces, they can infer the outline of rooms, corridors, and halls. For instance, Alzantot and Youssef (2012); Gao et al. (2014); Jiang et al. (2013) use volunteers' motion traces and the location of landmarks derived from inertial sensor data or Wi-Fi to determine the geometry of rooms and corridors. The disadvantage of these methods is that the furniture or other obstacles often block the edge of a room. Thus, users' traces could not cover these places, leading to inaccurate detection of room shapes. To resolve this problem, Chen et al. (2015) proposed a CrowdMap system that combines crowd-sourced sensory and images to track volunteers. Based on images and estimated motion traces, it can then create an accurate floor plan. Gao et al. (2017) proposed a Knitter system that can fast construct the indoor map 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. In a nutshell, measurements-based approaches focus mainly on the reconstruction of geometric maps with semantics, such as doors, rooms, windows, walls, ceilings, chairs, and tables, but not the usage of rooms.

**Rule or machine learning-based indoor mapping**. This group of approaches uses the structural rules or features in a certain building type to assist the reconstruction of maps. Such rules or features can be gained through manual definitions (Becker et al. 2015; Hu et al. 2017; Yue et al. 2011) or machine learning techniques (Rosser et al. 2017). Yue et al. (2011) proposed using a shape grammar (Mitchell 1990) 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. Peter et al. (2013) reconstruct a coarse building model from an evacuation plan and refine it using inertial measurement unit data and a grammar to constrain particular representations. Philipp et al. (2014) 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. Similarly, Khoshelham and Díaz-Vilariño (2014) used a shape grammar to reconstruct indoor maps that contain walls, doors, and windows. The collected point clouds can be used to learn the parameters of rules. Rosser et al. (2017) 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. The above-mentioned approaches normally ignore the reconstruction of room usage. To mitigate this, Aydemir et al. (2012) fused heterogeneous and uncertain information such as object observations, shape, size, appearance of rooms and human input for semantic mapping. Specifically, a probabilistic graphical model was used to represent the conceptual information and perform spatial reasoning, such as room categories and the structure of unexplored space. Pronobis and Jensfelt (2012) used a graph to represent the indoor environment, where rooms are regarded as the nodes of the graph. Graph-based reasoning approach was then adopted to infer the semantics of rooms according to their context information. Luperto et al. (2013) 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 Luperto et al. (2017) 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. Furthermore, Luperto and Amigoni (2019) adopted a generative model to represent the topological structures and the semantic labeling schemes of buildings and to generate plausible hypotheses for unvisited portions of these environments. Specifically, the buildings are represented as undirected graphs, whose nodes are rooms and edges are physical connections between them. Dehbi et al. (2018) addressed the automatic learning of a classifier which predicts the functional use of housing rooms based on features which are widely available such as room areas and orientation. It achieved a promising result but the layout and functional use of residential house are different with public buildings, such as research buildings and hospitals. Hu et al. (2019) proposed inferring the room usage of research buildings at universities by using grammars and Bayesian inference based on geometric maps. The approach was evaluated based on 15 maps with 408 rooms and a promising tagging accuracy at 0.84 is achieved. However, the defined grammar rules can only cover partial research buildings, which cannot be applied in other styles of research buildings. Rule or machine learning-based indoor mapping solutions take the most advantages of the intrinsic rules or features of certain building types, which can reduce the dependence on measurements or complement measurements-based approaches. However, there is still a lack of general solutions for room usage tagging.

## 8.3 Methods

### 8.3.1 Machine learning-based room type tagging

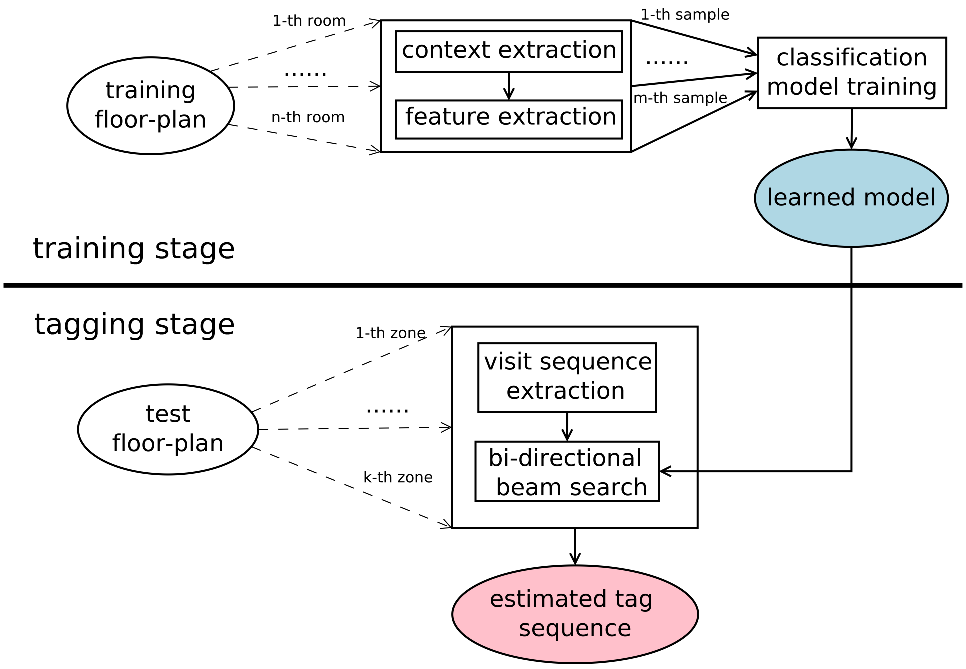


**Figure 8.2.** Workflow of machine learning-based method.

The workflow of the machine learning-based method is shown in Figure 8.2. It consists of two stages: training and tagging. During the training stage, the context containing the neighbors of a room is first determined, based on which its feature representation can be generated. A training sample is then generated by combining the feature representation and the usage of the room. Note that, the number of samples is normally larger than the number of rooms since most of the rooms have more than one contexts. The last step is to train a classification model (e.g., random forest) based on all the samples.

During the tagging stage, the tagging unit is a zone, which is formed by clustering the rooms with corridors and the building footprint as the boundary of the clusters. For a zone in a test floor plan, the longest linear sequence of rooms is first extracted based on the adjacency relationships among the rooms in the zone. The next step is to extract the visit sequence of rooms based on the longest linear sequence and the dependency relationship among rooms. Last, a bi-directional beam search strategy is applied in the visit sequence to find the tag sequence with the highest probability.

#### 8.3.1.1 Training stage

During the training stage, we first extract the context of each room. The reason for extracting context is that the type of a room is correlated with its context, specifically, the type of its neighbouring rooms. For instance, a lab is always surrounded by labs and lab support spaces. Thus, for a lab, the probability of neighbouring labs and lab support spaces is much higher than other room types. For a toilet, it is highly likely that its neighbouring room is still a toilet.

The context available when predicting the type of a room $r_i$ in a zone $R = \{r_1, r_2, ..., r_n\}$ with types $T = \{t_1, t_2, ..., t_n\}$ is $h_i = \left\{r_i, r^{i1}, \ldots, r^{iL}, r^p, t^{i1}, \ldots, t^{iL}, t^p\right\}$. $r^{ij}$ denotes the $j$-hop neighbor of room $r_i(r^{i0})$ with $1 \leq j \leq L$. $t^{i1}, \ldots, t^{iL}$ denotes the corresponding room types. $L$ denotes the maximum hop count of neighbors. Note that the neighbors here are defined according to the connection and adjacency relationship. Thus, the sub-room that is included by other rooms is not added into the neighbor list. Instead, the inclusion relationship is represented in $r^p$, which denotes the room that includes $r_i$. $t^p$ denotes the type of $r^p$. That is, if room $a$ is included by room $b$, room $a$ is not in the context of room $b$. Inversely, room $b$ is in the context of room $a$.

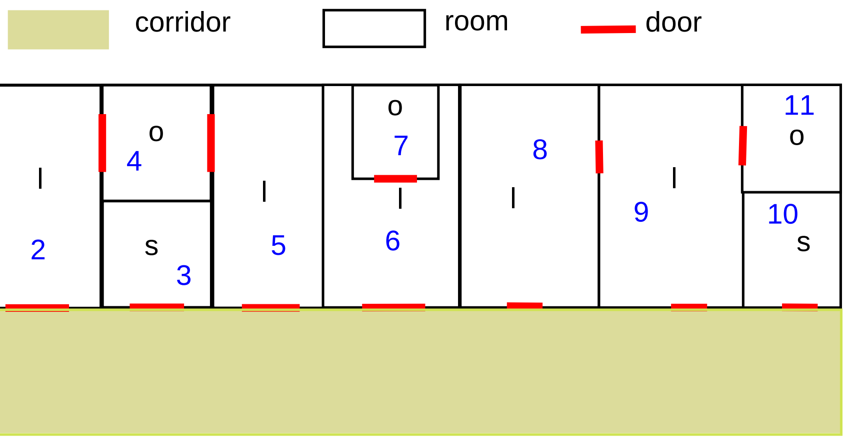


**Figure 8.3.** An example of a zone with annotated room usage with $s, l$, and $o$ denoting lab support space, lab, and office, respectively.

Figure 8.3 shows a zone with 10 rooms. Integers in blue denote the room number. Letters $s, l$, and $o$ represent lab support space, lab, and office, respectively. Red lines denote doors. The yellow space represents corridors. When L equals 3, the contexts for room 1 are denoted by $h_1 = \left\{ \begin{array}{l} h_1^1 : (r_1, r_2, r_3, r_5, l, s, l) \\ h_1^2 : (r_1, r_2, r_4, r_5, l, o, l) \end{array} \right\}$,which contains two contexts. The context for room 3 are denoted by $h_3 = \left\{ \begin{array}{r} h_3^1 : (r_3, r_5, r_6, r_8, l, l, l) \\ h_3^2 : (r_3, r_2, r_1, l, s) \\ h_3^3 : (r_3, r_4, o) \end{array} \right\}$, which contains three sub contexts. Room 6 is the parental room of room 7. Thus, the context of room 7 is $h_7 = \{r_7, r_6, l\}$, which contains only one sub context. From each sub context, a sample can be produced. Thus, for room 3, three training samples are generated. For room $r_i$ with one of the contexts as $h_i = \{r_i, r^{i1}, ..., r^{iL}, r^p, t^{i1}, ..., t^{iL}, t^p\}$, the used features are listed in Table 8.1. We name the features relying only on the room itself as intrinsic features and the features relying on other rooms as extrinsic features. Thus, the last four features in Table 8.1 are extrinsic features and the remaining features are intrinsic features.

**Table 8.1.** Features used in machine learning approach.

| **Features** | **Type** |
|---|---|
| area of room | float |
| width of room | float |
| length or room | float |
| area of building | float |
| length of building | float |
| width of building | float |
| withExtDoor($r_i$) | category |
| inCenter($r_i$) | category |
| extWallNum($r_i$) | category |
| parentExist($r_i$) | category |
| $t^p$ | category |
| connection($r_i, r^{i1}$); $t^{i1}$ | category |
| connection($r^{i1}, r^{i2}$); $t^{i2}$ | category |

A couple of functions are used to define the features in Table 8.1.

**withExtDoor**($a$): If room $a$ is connected to corridors through doors. In Figure 8.3, room 4 is not connected to corridors, while room 3 is connected to corridors.

**extWallNum** ($a$): the number of walls that room $a$ is located at. In Figure 8.3, rooms 1, 2, and 3 are located at two, one, and zero walls, respectively.

**parentExist**($a$): If room $a$ is included by another room. For instance, in Figure 8.3, room 6 contains room 7.

**inCenter** ($a$): If the zone that room $a$ belongs to is at the center of a building. A zone is in the center of a building only when the number of its rooms that are located at external walls is smaller than the number of its rooms that are not located at external walls.

**connection**($a, b$): rooms $a$ and $b$ are connected through at least one door. In Figure 8.3, room 2 is connected to room 4 but disconnected to room 3.

The categorical features are encoded as one-hot numeric arrays. The length of the array equals the number of possible values of the categorical feature. For instance, $t^p$ refers to the type of the parental room with eleven possible values. One of the values can be encoded as '00010000000'. Note that, the type of neighbouring rooms and the connection to neighbouring rooms are combined as one feature. Thus, there are at most $2 * m$ possible options, where $m$ denotes the number of candidate room types. For missing categorical features, such as the missing parental room and the neighboring rooms, all the bits are encoded as zeros.

Two assumptions are made in defining the features. One is that the rooms are assumed to be rectangles. The other is that building footprints are rectilinear polygons. The length and width of a rectilinear building is the larger and smaller one in the sum of horizontal and vertical edges, respectively. In our future work, automatic solutions, such as the generalization method will be adopted to deal with the non-rectangular and non-rectilinear issues.

#### 8.3.1.2 Online tagging: bi-directional beam search

The feature representation of a room is correlated with its context, specifically, the type of neighbouring rooms, which however is unknown. To solve this problem, the beam search algorithm is adopted, which predicts the tag of current room based on previously estimated tags of neighboring rooms, to find an approximately optimal type sequence for a cluster of rooms within an acceptable time bound. The probability model is defined as $\hat{p}(t|h)$, where $t$ is the type of room and $h$ is the context of the room. The probability of a type sequence $\{t_1, ..., t_n\}$ given a zone with room sequence $\{r_1, ..., r_n\}$ can be calculated by Equation 1:

$$\hat{p}\left(\{t_1, ..., t_n\} \mid \{r_1, ..., r_n\}\right) = \prod_{i=1}^{n} \hat{p}\left(t_i | h_i\right) \tag{8.1}$$

The first step of the online-tagging algorithm is to extract two visit sequences from a graph-structural or tree-structural zone, which is used as the input of the beam search algorithm. The visit sequence should keep the dependency or topological relationship in the graph-structural zone as much as possible. This step consists of six sub-steps.

(1) **Extract the longest linear sequence**. The longest linear (sub) sequence is extracted from the zone to make sure that each room is only connected or adjacent to its direct preceding and succeeding rooms. The sub-rooms that are included by other rooms are ignored at this step.

(2) **Initialize one visit sequence.** Select one of the two end points (rooms) of the longest linear sequence and initialize one visit sequence with the selected end point.

(3) **Add neighboring rooms to the visit sequence.** From the zone, select the rooms that are directly adjacent or connected to the rooms in the visit sequence, which are then added to

the visit sequence. Repeat this step in a breadth-first strategy until all the rooms (except the sub-rooms) in the zone have been added into the visit sequence.

(4) **Insert the sub-rooms into the visit sequence**. The sub-rooms are inserted into the visit sequence, directly following their parental rooms.

(5) **Get the second visit sequence.** In a similar way, the second visit sequence can be generated by initializing it with the second end point of the longest linear sequence and then executing steps (3) and (4).

For example, one of the longest sequences of the zone in Figure 8.3 is $\{1, 2, 4, 5, 6, 8, 9, 10\}$, in which each room is only connected or adjacent to its directly preceding and succeeding rooms. Then, based on it, the two visit sequences {1,2,4,3,5,6,7,8,9,10,11} and {10,11,9,8,6,7,5,4,3,2,1} can be obtained. Based on the two visit sequences, the beam search algorithm is execute twice. Thus, two room type sequences for a zone can be obtained, and the one with the highest probability is chosen as the tagging result. We name this strategy as bi-directional beam search. The reason of using the bi-directional strategy is that the rooms in a zone can be searched from either direction (assumed linear), which however, can produce distinct tagging results.

Let $V = \{r_1, ..., r_n\}$ be one of the visit sequences of a test zone, and let $s_{i,j}$ be the $j$-th highest probability tag sequence up to and including room $r_i$. $j$ ranges from 1 to $N$. $N$ represents the number of the best tagging sequences that are kept as candidates. The procedure of beam search is to recursively calculate $s_{i,j}$ when $i$ increases from $1$ to $n$. The detail of the beam search algorithm are given in (Ratnaparkhi 1996). When calculating the probability of belong $r_i$ to a certain type, the context of $r_i$ is extracted from previously tagged room sequence, $\{r_1, ..., r_{i-1}\}$. If multiple sub contexts for a room are extracted, the estimated probability values based on all the contexts are multiplied and then raised to the power of the reciprocal of the number of contexts as Equation 8.2. $k$ denotes the number of sub contexts and $\hat{p}_j$ denotes the estimated probability based on the $j$-th sub context.

$$\hat{p} = \left( \prod_{j=1}^{k} \hat{p}_j \right)^{\frac{1}{k}} \tag{8.2}$$

We choose the floor plan in Figure 8.4 as an example to explain the process of online tagging. Enclosed rectangles denote rooms. The other spaces are corridors. In each room, the integer denotes the room number. The solid blue lines denote footprints, while solid red lines denote internal doors. The rooms surrounded by solid green lines have no external doors.

The zones are automatically generated by clustering the rooms that are adjacent to or included by at least one of the rooms in the zone without depending on the shape of corridors. Five zones are extracted from the floor plan of Figure 8.4, comprising rooms with the number ranging from 1 to 7, 8 to 15, 16 to 30, 31 to 42, and 43 to 57, respectively. We assume that the

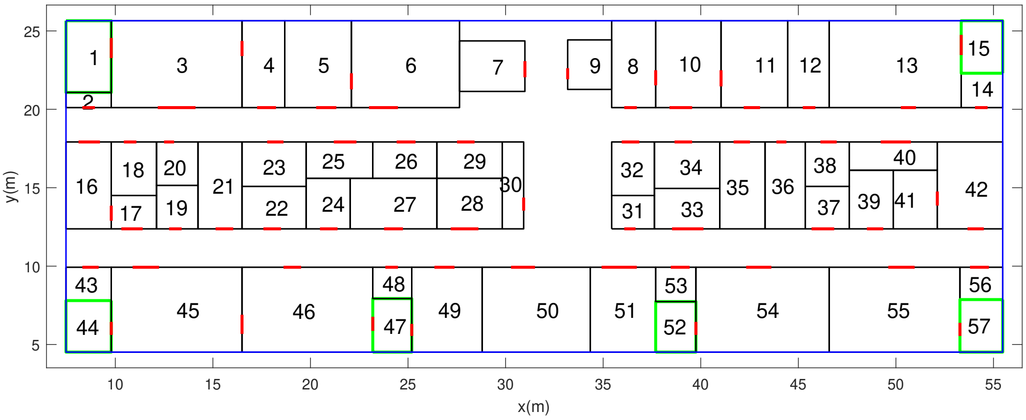


**Figure 8.4.** A floor plan with 57 unmarked rooms.

classification model has been trained. In the online-tagging stage, the tagging unit is a zone. We take the first zone comprising rooms with the number ranging from 1 to 7 is as an example to show how the rooms in a zone are tagged. This procedure is illustrated in Figure 8.5. First, two visit sequences are extracted from the zone based on the aforementioned algorithm. They are $\{2, 1, 3, 4, 5, 6, 7\}$ and $\{7, 6, 5, 4, 3, 1, 2\}$. The element in the sequence denotes the room number. Next, the beam search algorithm is conducted on each visit sequence to find the tag sequence with the highest probability. In the first visit sequence, the start room is room 2, whose context contains only itself without neighbors since the usage of its neighbors is currently unknown. In this case, the features related to the usage of neighbours are assigned default values: all the bits are assigned 0. Then, the complete feature representation of room 2 is obtained, based on which, the probability of belonging it to each candidate type is calculated by using the trained classification model. Let $s_i$ be the top $N$ (set to 2 in this example) highest probability tag sequence for the visit sequence. Let $p_i$ be the probability estimation of each tag sequence in $s_i$. We assume that the value of $s_1$ and $p_1$ are $s_1 = \{s, t\}$ and $p_1 = \{0.4, 0.1\}$,where $s$ and $t$ denote the support space and toilet, respectively. Next, room 1 is visited. Now, the usage of room 2 is known. Thus, the context of room 1 contains itself and one neighbour: room 2. Based on one of the tag sequences in $s_1$, the feature representation of room 1 is generated, which contains the usage of its neighbor:room 2. Then, the probability of belonging room 1 to each type is estimated, which is multiplied with the probability estimation of the chosen tag sequence in $s_1$. Thus, we can obtain totally m*N tag sequences for room sequence $\{2, 1\}$, where m denotes the number of room types. Similarly, Top $N$ sequences is preserved in $s_2$. The possible value of $s_2$ could be $\{so, tt\}$ with $p_2 = \{0.2, 0.05\}$ . And so on, $s_7$ can be calculated. Top 1 tag sequence in $s_7$ is preserved. Then, the beam search is conducted on the second visit sequence to find the second Top 1 tag sequence. The one among the two Top 1 tag sequences with the highest probability is chosen as the output tag sequence. The other zones can be tagged in a similar manner.

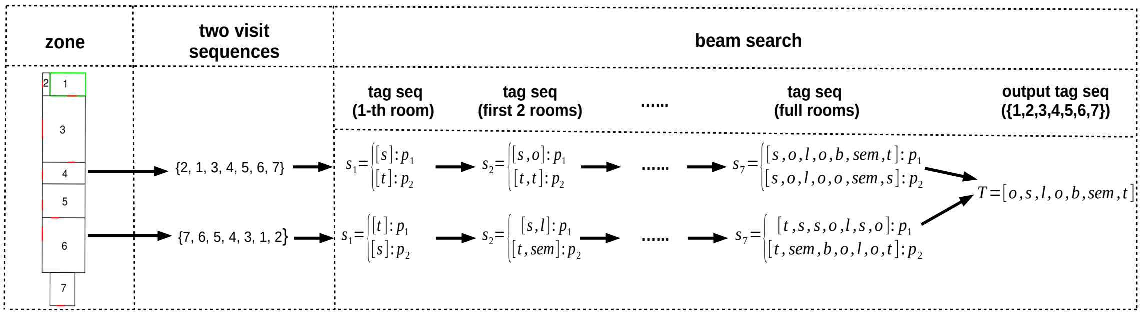

zone
two visit sequences
beam search
2
1
3
4
5
6
7
{2, 1, 3, 4, 5, 6, 7}
{7, 6, 5, 4, 3, 1, 2}
tag seq (1-th room)
tag seq (first 2 rooms)
......
tag seq (full rooms)
output tag seq ({1,2,3,4,5,6,7})
$s_1 = [s]: p_1$ ; $[t]: p_2$
$s_2 = [s,o]: p_1$ ; $[t,t]: p_2$
$s_7 = [s,o,l,o,b,sem,t]: p_1$ ; $[s,o,l,o,o,sem,s]: p_2$
$s_1 = [t]: p_1$ ; $[s]: p_2$
$s_2 = [s,l]: p_1$ ; $[t,sem]: p_2$
$s_7 = [t,s,s,o,l,s,o]: p_1$ ; $[t,sem,b,o,l,o,t]: p_2$
$T = [o,s,l,o,b,sem,t]$

**Figure 8.5.** Workflow of online tagging stage.

### 8.3.2 Deep learning-based room type tagging

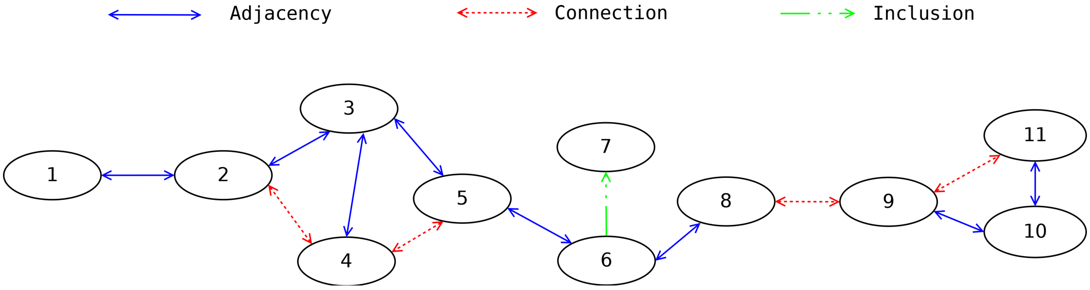


**Figure 8.6.** Directed multi-relational graph of the zone in Figure 8.3.

In machine learning-based approaches, several extra data processing procedures are required, such as the visit sequence extraction and bi-directional beam search, in order to deal with the issues that (1) the zone is a graph structure rather than a linear sequence (2) the tag of a room depends on the tag of its neighbors, which however, is unknown. These procedures increase the computational-complexity of the machine learning-based approach and might lead to the incomplete representation of the topological features of a room in a graph structure. To address these issues, we adopt a graph-based deep learning method, named Relational Graph Convolutional Networks (R-GCNs) to classifying the nodes (rooms) by automatically collecting useful information from neighbour nodes. It models each zone as a directed graph with multiple types of links (relationships) between nodes (rooms). Figure 8.6 shows the directed multi-relational graph of the zone in Figure 8.3. For simplicity, we rename the two relationships: adjacency without connection and adjacency with connection as adjacency and connection, respectively. Adjacency and connection are defined as bi-directional relationships, while inclusion is defined as a unidirectional relationship.

R-GCN generalizes GCN (Kipf, Welling 2016) to handle different relations between nodes. In GCN-based models, the most important part is how to effectively accumulate and encode features from local, structured neighborhoods. R-GCN uses the following propagation model for calculating the forward-pass update of a node denoted by $v_i$ in a relational (directed and labeled) multi-graph:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right) \tag{8.3}$$

where $h_i^{(l+1)} \in R^{d^{(i)}}$ is the hidden state of node $v_i$ in the l-th layer of the neural network, with $d^{(i)}$ being the dimensionality of this layer's representations. $h_i^{(0)}$ equals the input feature vector of node $i$. $N_i^r$ denotes the set of neighbor indices of node $i$ under relation $r \in R$. $c_{i,r}$ is a normalization constant that is set as $|N_i^r|$ in this work. $W$ denotes the model parameters. $W_0^{(l)}h_i^{(l)}$ is a single self-connection of a special relation type to each node in the data. $\sigma(\cdot)$ is the activation function. In this work, we use ReLU(·)= max(0,·) as the activation function. Intuitively, (2) accumulates the feature vectors of current nodes $i$ and the transformed feature vectors of neighboring nodes through a normalized sum based on the direction and type of edges.

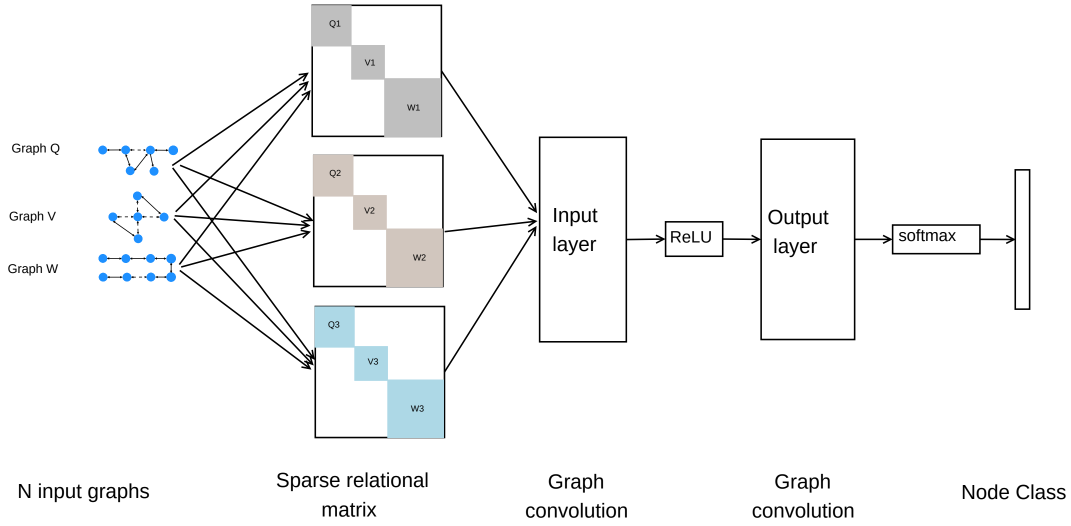


**Figure 8.7.** Architecture of R-GCN for room tagging.

In this work, we use a 2-layer architecture: Input layer and output layer, as shown in Figure 8.7. $G$, $V$, and $W$ are the input data, which are bunches of small and separated graphs with each representing a zone of the total floor plans. All the graphs are combined as a virtually full graph. Then, the full graph can be represented by three sparse relational matrices with each corresponding to one of the three relationships: adjacency, connection, and inclusion. Q1, V1, and W1 denote the relation matrix of the three input graphs in the first relationship, respectively, which are combined to generate the sparse matrix of the full graph in the first relationship. Similarly, Q2, V2, W2, and Q3, V3, W3 denote the relation matrix in the second and third relationship, respectively. A softmax (·) activation (per node) is used on the output of the last layer. The following cross-entropy loss on all labeled nodes (while ignoring unlabelled nodes) is minimized:

$$L = -\sum_{i \in Y} \sum_{k=1}^{K} t_{ik} \ln h_{ik}^{l} \tag{8.4}$$

where $Y$ is the set of node indices that have labels (training nodes) and $h_{ik}^{l}$ is the $k$-th entry of the network output for the $k$-th labeled node. $t_{ik}$ denotes its respective ground truth label. More details about the node classification by using R-GCN can be found in (Schlichtkrull et al. 2018)

The input data are the total floor plans, including the training floor plans and the test floor plans, which are combined to generate the full graph. The rooms (nodes) in the training floor plans have been labelled, while the rooms (nodes) in the test floor plans are unlabelled. During the training stage, the labelled nodes are used together to update the parameters shared by both labelled and unlabelled nodes. That is, RGCN is applied globally on the whole graph. When the training process ends, the softmax function is used to classify the unlabelled nodes based on their feature representation of the output layers. In Equation 8.4, $h_i^{(0)}$ is the input feature vector of node $i$, which denotes the intrinsic features of room (node) $i$. During the convolutional process, both the intrinsic and extrinsic or structural features of a node are propagated to its neighbors. The intrinsic features of room (node) $i$ used in RGCN are listed in Table 8.2, which is a subset of the features in Table 8.1. The geometric properties of the buildings are removed since we found the addition of these features do not improve the accuracy. The categorical features are encoded as one-hot numeric arrays. The numerical features (area, length and width) are encoded as their binary form. Then, the representation of these intrinsic features for node $i$ are concatenate as a one-dimensional vector, which is used as the input feature vector of node $i$.

**Table 8.2.** Node features used in R-GCN approach.

| **Features** | **Type** |
|---|---|
| area of room | float |
| width of room | float |
| length or room | float |
| withExtDoor($r_i$) | category |
| inCenter($r_i$) | category |
| extWallNum($r_i$) | category |

The graph in Figure 8.6 is taken as an example to explain the training procedure. Assume that the graph in Figure 8.6 is the training graph and all the nodes are thus the training nodes. A

2-layer architecture is adopted and three relationships are defined. Thus, there are in total eight parameter matrices needed to be learned from the training nodes, denoted by $W_r^{(l)}$, following the definition in Equation 8.4 with $r \in [0, 1, 2, 3]$, corresponding self-connection, adjacency, connection, and inclusion relationship, respectively, and $l \in [0, 1]$, corresponding the two layers. Assume that the area, width, and length of node 5 are 50, 5, and 10, respectively. Next, the three attributes are encoded as their binary forms with 6 bytes, which are then connect to one vector. The result vector is thus [**110010**000101**001010**], which is treated as the input vector of node 5, denoted by $h_5^{(0)}$. Similarly, the input vector of the other nodes can be obtained. The hidden state of each node in the first layer is then estimated according to Equation 8.4. Node 5 is taken as an example, which has three neighbors (nodes 3,4,and 6) in two relationships (adjacency and connection) apart from the self-connection relationship. The hidden state (feature vector) of node 5 in the first layer is calculated by Equation 8.5 that instantiates Equation 8.4.

$$h_5^{(1)} = \sigma\left(\frac{1}{2}W_1^{(0)}h_3^{(0)} + \frac{1}{2}W_1^{(0)}h_6^{(0)} + W_2^{(0)}h_4^{(0)} + W_0^{(0)}h_5^{(0)}\right) \tag{8.5}$$

In this way, apart from the intrinsic features ($h_5^{(0)}$) of node 5 itself, the intrinsic features of neighboring nodes ($h_3^{(0)}, h_4^{(0)}$, and $h_6^{(0)}$) are also propagated to node 5. Moreover, the features of neighboring nodes are multiplied with two different matrix with each corresponding to one relationship. Thus, the structural features have also been integrated to the hidden state of node 5. Similarly, the hidden state of other nodes at the first layer can be calculated. likewise, the hidden state of node 5 at the second layer can be calculated from $h_3^{(1)}, h_4^{(1)}$, and $h_6^{(1)}$. By doing so, the features of farther neighbors (e.g., node 7) of node 5 is also propagated to node 5 because they have been integrated into the hidden state of the direct neighbor (e.g., node 6) of node 5 at the first layer. The softmax function is then applied in the hidden state of each node at the second layer to output the classification result of the nodes. The classification loss is calculated based on the training nodes and used to optimize the parameter matrices. When the training procedure is done, the test nodes can be classified based on the optimized parameter matrices.

## 8.4 Experiments

We choose research buildings at universities as testbeds to evaluate the proposed approaches. 130 floor plans of research buildings with 3330 rooms were collected from two German universities, which we believe is sufficient to evaluate our proposed approach. According to (Klonk 2016), we divide the enclosed rooms in research buildings into 11 types: office ($o$), lab ($l$), lab support space ($s$), seminar/lecture room ($sem$), PC room ($pc$), library ($lib$), toilet ($t$), copy/print room ($c$), storage room ($sto$), kitchen ($k$), and lounge/break room ($b$). The number of the 11 types in the total floor plans are 1509, 465, 665, 132, 6, 75, 355, 18, 48, 23, and 34, respectively, which is shown in Figure 8.8. The corresponding proportions are 0.453, 0.140, 0.120, 0.040, 0.002, 0.023, 0.107, 0.005, 0.015, 0.007, and 0.010, respectively. Labs refer to the standard labs at physical, biological, chemical, and medical institutes. Lab support spaces are used to support the operation of labs and are generally not continuously occupied, such as equipment rooms, cold rooms, and chemical storage (Klonk 2016).

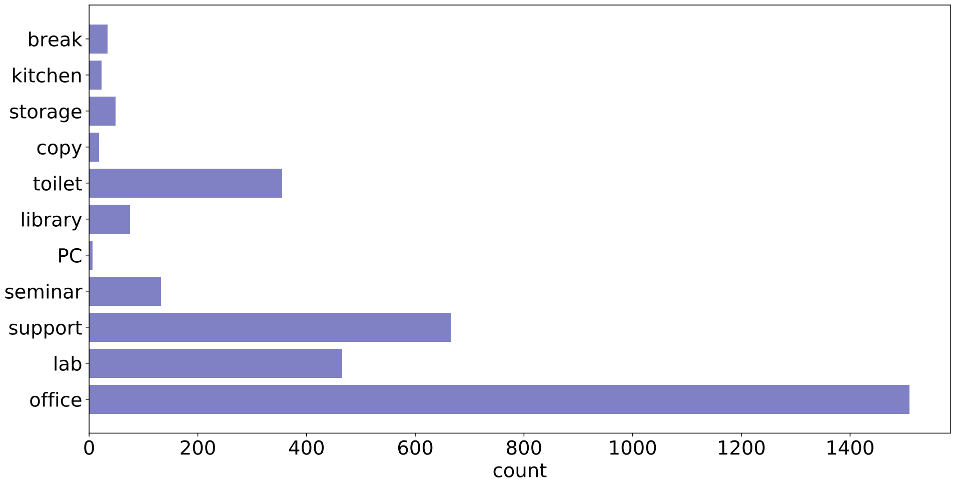
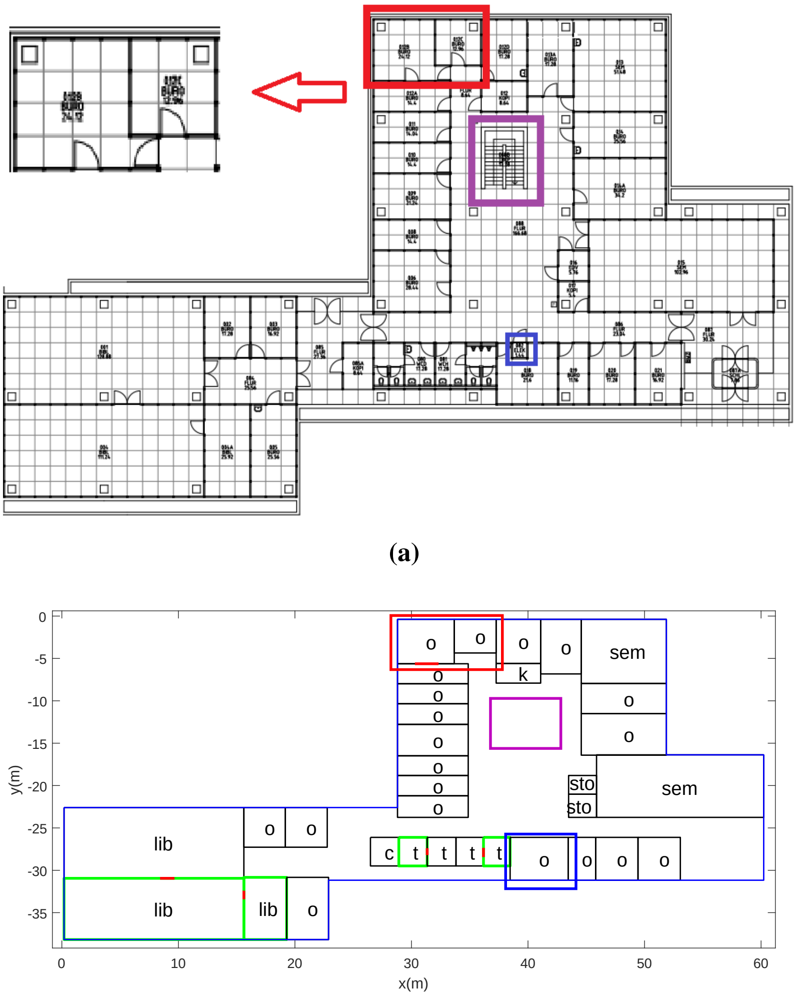


**Figure 8.8.** Count of different room types among 130 floor plans.

The experimental data is extracted from original floor plan images by manually tagging the footprint of the building, the outline of rooms, and the location of both internal and external doors. The remaining space is the corridor. We assume that the footprint, door, and corridor are marked in the inputting map, which is true in most of the cases. That is, an indoor geometric map normally contains these information, such as the indoor maps in Figure 8.1. For the non-rectangular rooms and non-rectilinear footprints, we manually convert them to the most approximating rectangles and rectilinear polygons, respectively, preserving the topological relationship. In our future work, automatic solutions, such as the generalization method will be introduced to deal with the non-rectangular and non-rectilinear issues. All the points and lines are represented in pixels coordinates. Then, based on the given area of a room that is already marked on the scanned map, the pixel coordinates of lines and points are converted to a local geographic coordinate. Finally, the geometry and spatial location of footprints, rooms, corridors, and doors, the topology relationship among rooms, and the spatial distribution of rooms (i.e., in centre or at external walls) can be calculated. The zones are then automatically extracted based on the inclusion and adjacency relationship among rooms. During this procedure, the electrical room is ignored since it is rare and less important than the other enclosed spaces. Staircases are treated as circulation spaces (corridors). The furniture in the room is also deleted. Figure 8.9a shows the original image of a floor plan. The room surrounded by a red rectangle contains a non-rectangular room, which is zoomed in and shown in the left-up corner of the figure. The area surrounded by a purple rectangle contains a staircase that would be replaced with corridors. The area surrounded by a blue rectangle contains an electrical room, which would be removed. Figure 8.9b shows the simplified map from the original image, where the room surrounded by green rectangles has no doors connected to corridors. The simplified parts in the map are surrounded by red, purple, and green rectangles, respectively.



(a)

(b)

**Figure 8.9.** Experimental data extraction by simplifying scanned floor plan images.(a) Marked parts denote those that will be simplified. (b) simplified map from original floor plan image.

To evaluate the proposed approaches, we use a five-fold cross validation method, randomly dividing the 130 floor plans into five test groups with each group containing 26 floor plans. In each test group, the 26 floor plans are used as the test data, in which the room usage is missing, with the remaining 104 floor plans as the training data, in which the room usage of rooms are known.

### 8.4.1 Comparison of tagging accuracy of proposed approaches

In this experiment, we compare the performance of machine-learning and deep-learning based approaches. For machine-learning based approaches, we train a random forest (RF) and decision tree (DT)-based classification model, respectively. The parameters of machine-learning-based approaches include the length of contexts (L) and the number of best sequences that are kept as candidates in beam search (N), which are set to 2 and 10, respectively. While, for random-forest based approaches, the most important parameter is the number of trees (NT), which is set to 50. For the R-GCN based approach, the main parameters include the count of epochs (EN) and the number of features in hidden layers (HN), which are set 100, and 80, respectively. The machine learning-based approach is implemented based on the classification library of Matlab. The R-GCN-based tagging approach is implemented based on an open-sourced python library, named Deep Graph Library (DGL).

The mean accuracy of identifying room types with RF, Decision Tree, and R-GCN for the five groups are 0.85, 0.77, and 0.79, respectively, which is shown in Table 8.3. RF-based approach achieves a higher classification accuracy than the other two approaches. R-GCN does not outperform RF-based approaches as we expect, which is mainly because in our issues, the relationship related features dose not play dominant roles in room classification. To verify this, we measured the importance of each feature given in Table 8.3 by calculating how much the accuracy decreases when the feature is excluded in the random forest. The result is shown in Figure 8.10. The relationship-related or extrinsic features (first neighbour, second neighbour, parent exist, and parental room) in total contribute only 35% of the weight. In addition, we used only the first eight features (intrinsic) in Table 8.1, ignoring the relationship-related features, to predict the room tag by using random forest, achieving an accuracy of 0.79. This reveals that the relationship-related features contribute a small part of the tagging accuracy. Furthermore, we explore the impact of N on the tagging accuracy, varying from 1 to 20. During each iteration of the beam search, only top N best candidates are kept. We found that when N is over 3, the tagging accuracy remains unchanged. This is because with only the intrinsic features a high tagging accuracy can already be achieved without looking at its neighboring rooms. This means although beam search is a greedy algorithm, it can achieve an approximately optimal result. Therefore, other sequence inference algorithms such as Gibbs sampling (Gelfand 2000) were not investigated although they are more likely arrive to a global optima than beam search. In spite of these facts, R-GCN achieves an acceptable tagging accuracy, which slightly outperforms the decision tree-based approach. As far as we know, this is the first time that R-GCN is successfully used to predict the type of entities in indoors.

In the following, we focus on the performance analysis of the random forest and R-GCN based approaches. Partial representative floor plans and their tagging results by random forest and R-GCN can be found in Appendix C and D, respectively. The solid blue lines denote footprints, while solid red lines denote internal doors. The rooms surrounded by solid green lines have no external doors. The rooms in pink background denote incorrectly tagged rooms. The text in black denotes the true room type, while the text in blue denotes the incorrectly tagged

**Table 8.3.** Tagging accuracy of each test group

| Group | Accuracy (RF) | Accuracy (R-GCN) | Accuracy (DT) | Number of rooms |
|---|---|---|---|---|
| Group (1) | 0.81 | 0.77 | 0.77 | 695 |
| Group (2) | 0.85 | 0.81 | 0.75 | 709 |
| Group (3) | 0.84 | 0.78 | 0.76 | 609 |
| Group (4) | 0.90 | 0.84 | 0.79 | 693 |
| Group (5) | 0.85 | 0.76 | 0.78 | 624 |
| Overall | 0.85 | 0.79 | 0.77 | 3330 |

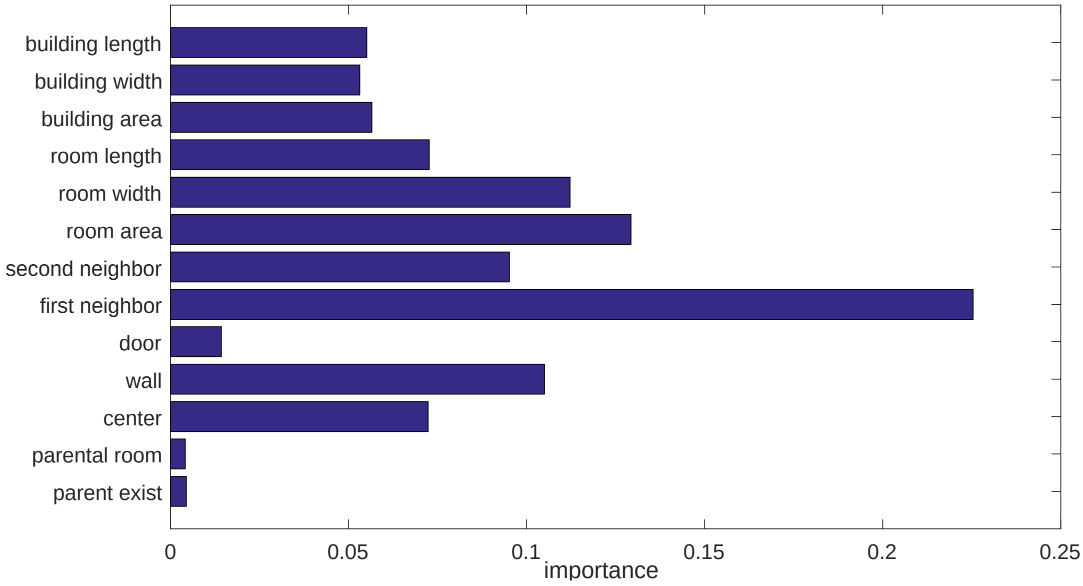
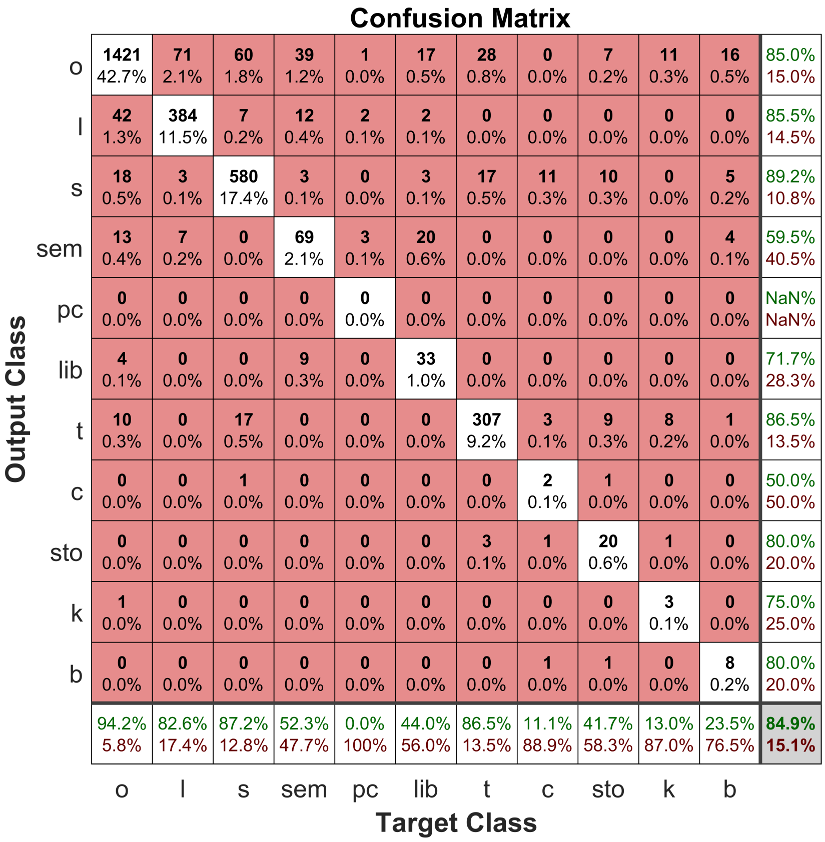


**Figure 8.10.** Importance of features in machine learning-based approach.

room type. We can see the indoor layout of the test floor plans varies, which cannot be represented by simple grammars. Therefore, the room tagging approaches proposed by Hu et al. (2019) is inapplicable in our data set. However, the proposed approach in this work can achieve a promising tagging result. In most of the test floor plans, the tagging results by random forest is better than R-GCN, especially when tagging less-frequent spaces (e.g., library, seminar, storage, and kitchen).

The confusion matrix is produced based on the tagging result of the five test groups, as shown in Figures 8.11, 8.12, and 8.13. In random forest-based approach, the accuracy of identifying labs, offices, and lab support spaces and toilets is much higher than that of other room types, which can be explained from two aspects: (1) they are much more frequent than other room types; (2) their geometric, topological, and spatial distribution characteristics are iden-

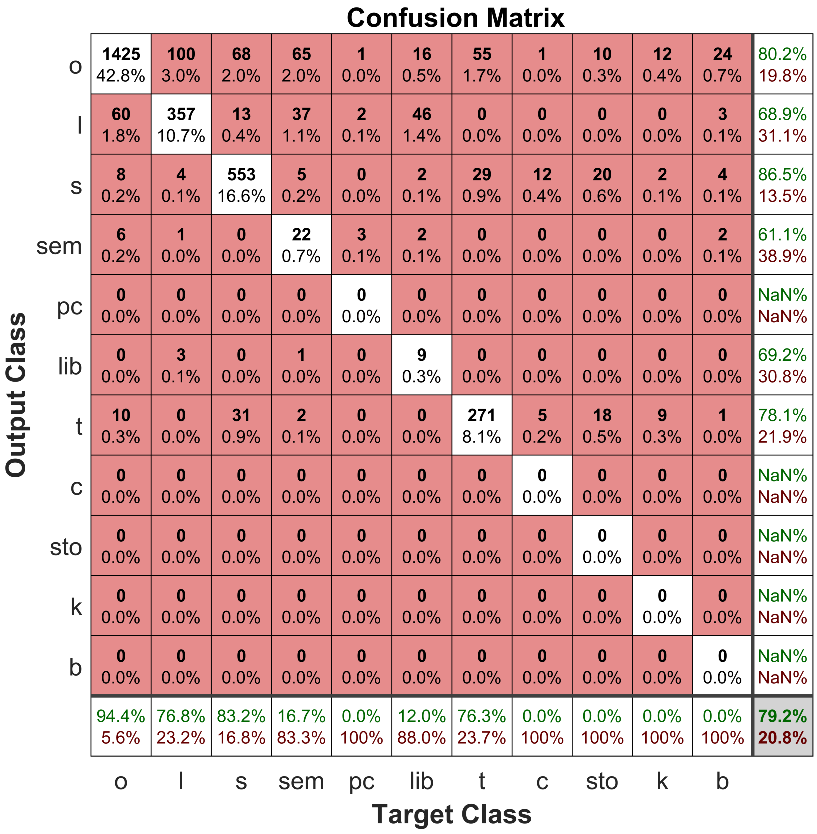


Confusion Matrix

| | o | l | s | sem | pc | lib | t | c | sto | k | b | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| o | 1421<br>42.7% | 71<br>2.1% | 60<br>1.8% | 39<br>1.2% | 1<br>0.0% | 17<br>0.5% | 28<br>0.8% | 0<br>0.0% | 7<br>0.2% | 11<br>0.3% | 16<br>0.5% | 85.0%<br>15.0% |
| l | 42<br>1.3% | 384<br>11.5% | 7<br>0.2% | 12<br>0.4% | 2<br>0.1% | 2<br>0.1% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 85.5%<br>14.5% |
| s | 18<br>0.5% | 3<br>0.1% | 580<br>17.4% | 3<br>0.1% | 0<br>0.0% | 3<br>0.1% | 17<br>0.5% | 11<br>0.3% | 10<br>0.3% | 0<br>0.0% | 5<br>0.2% | 89.2%<br>10.8% |
| sem | 13<br>0.4% | 7<br>0.2% | 0<br>0.0% | 69<br>2.1% | 3<br>0.1% | 20<br>0.6% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 4<br>0.1% | 59.5%<br>40.5% |
| pc | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | NaN%<br>NaN% |
| lib | 4<br>0.1% | 0<br>0.0% | 0<br>0.0% | 9<br>0.3% | 0<br>0.0% | 33<br>1.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 71.7%<br>28.3% |
| t | 10<br>0.3% | 0<br>0.0% | 17<br>0.5% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 307<br>9.2% | 3<br>0.1% | 9<br>0.3% | 8<br>0.2% | 1<br>0.0% | 86.5%<br>13.5% |
| c | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 2<br>0.1% | 1<br>0.0% | 0<br>0.0% | 0<br>0.0% | 50.0%<br>50.0% |
| sto | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 3<br>0.1% | 1<br>0.0% | 20<br>0.6% | 1<br>0.0% | 0<br>0.0% | 80.0%<br>20.0% |
| k | 1<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 3<br>0.1% | 0<br>0.0% | 75.0%<br>25.0% |
| b | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 1<br>0.0% | 0<br>0.0% | 8<br>0.2% | 80.0%<br>20.0% |
| | 94.2%<br>5.8% | 82.6%<br>17.4% | 87.2%<br>12.8% | 52.3%<br>47.7% | 0.0%<br>100% | 44.0%<br>56.0% | 86.5%<br>13.5% | 11.1%<br>88.9% | 41.7%<br>58.3% | 13.0%<br>87.0% | 23.5%<br>76.5% | 84.9%<br>15.1% |

**Figure 8.11.** Confusion matrix of RF-based approach.

tifiable. Generally, offices and labs are located at external walls rather than central areas that cannot easily receive natural light, which can be seen from floor plans (a), (b) and (f) of Appendix C. In the floor plans, the central zone contains only ‘unimportant’ rooms, such as lab support spaces, toilet, storage room, and copy room. A lab is normally connected to another lab or lab support spaces and has a larger area than office, which can be seen in floor plans (n), (o), and (s) of Appendix C. However, there are still many cases that violate these norms, leading to the misclassification of offices and labs. For instance, in floor plan (b), four offices are tagged as labs because the four rooms have similar geometric properties to labs. Generally, toilets have a small area and comprise two connected rooms with only one connected to corridors, such as in floor plans (c), (h), (i), and (n) of Appendix C. These features are unique and based on which, toilets can be distinguished from other room types. However, sometimes, toilets appear in the form of a single room. In this case, they turn to be tagged as offices and support spaces whose geometric properties overlap with that of toilets, such as the toilets in floor plans (c), (e), (g) of Appendix C. Seminar rooms and libraries are normally much larger than other room types. However, first, it is difficult to distinguish libraries from seminars since they have similar geometric properties. Second, some libraries and seminar rooms have similar geometric properties with general offices. For example, in floor plan (l) of Appendix C, a library is incorrectly tagged as an office room because the room has a small area and the case that a library is connected with



Confusion Matrix

Output Class / Target Class

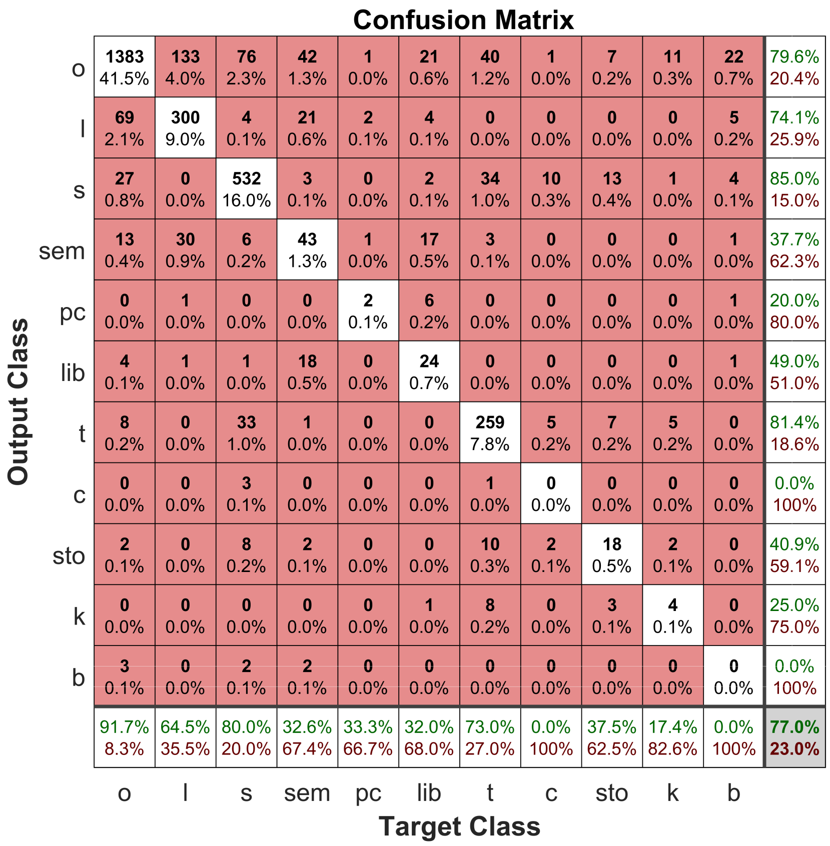
| | o | l | s | sem | pc | lib | t | c | sto | k | b | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| o | **1425**<br>42.8% | **100**<br>3.0% | **68**<br>2.0% | **65**<br>2.0% | **1**<br>0.0% | **16**<br>0.5% | **55**<br>1.7% | **1**<br>0.0% | **10**<br>0.3% | **12**<br>0.4% | **24**<br>0.7% | 80.2%<br>19.8% |
| l | **60**<br>1.8% | **357**<br>10.7% | **13**<br>0.4% | **37**<br>1.1% | **2**<br>0.1% | **46**<br>1.4% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **3**<br>0.1% | 68.9%<br>31.1% |
| s | **8**<br>0.2% | **4**<br>0.1% | **553**<br>16.6% | **5**<br>0.2% | **0**<br>0.0% | **2**<br>0.1% | **29**<br>0.9% | **12**<br>0.4% | **20**<br>0.6% | **2**<br>0.1% | **4**<br>0.1% | 86.5%<br>13.5% |
| sem | **6**<br>0.2% | **1**<br>0.0% | **0**<br>0.0% | **22**<br>0.7% | **3**<br>0.1% | **2**<br>0.1% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **2**<br>0.1% | 61.1%<br>38.9% |
| pc | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | NaN%<br>NaN% |
| lib | **0**<br>0.0% | **3**<br>0.1% | **0**<br>0.0% | **1**<br>0.0% | **0**<br>0.0% | **9**<br>0.3% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | 69.2%<br>30.8% |
| t | **10**<br>0.3% | **0**<br>0.0% | **31**<br>0.9% | **2**<br>0.1% | **0**<br>0.0% | **0**<br>0.0% | **271**<br>8.1% | **5**<br>0.2% | **18**<br>0.5% | **9**<br>0.3% | **1**<br>0.0% | 78.1%<br>21.9% |
| c | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | NaN%<br>NaN% |
| sto | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | NaN%<br>NaN% |
| k | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | NaN%<br>NaN% |
| b | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | **0**<br>0.0% | NaN%<br>NaN% |
| | 94.4%<br>5.6% | 76.8%<br>23.2% | 83.2%<br>16.8% | 16.7%<br>83.3% | 0.0%<br>100% | 12.0%<br>88.0% | 76.3%<br>23.7% | 0.0%<br>100% | 0.0%<br>100% | 0.0%<br>100% | 0.0%<br>100% | **79.2%**<br>**20.8%** |

**Figure 8.12.** Confusion matrix of RGCN-based approach.

an office also occurs frequently in the training data set. In floor plan (m) of Appendix C, a library is tagged as a seminar room since the geometric properties of the room is similar to that of most of the seminars. Libraries can be recognized when the room area is large enough (e.g., over 300 $m^2$) or several libraries with a larger area (e.g., over 60 $m^2$) than general offices are connected and clustered, such as in floor plans (h), (j), and (l) of Appendix C. As for storage, copy room, Pc, kitchen, and lounge, they occur sparsely and their geometrical and topological characteristics are unapparent. Thus, it is difficult to distinguish them from offices, labs, and support spaces. In R-GCN-based and decision tree-based methods, apart from the office, lab, lab support space, and toilet, the other room types are hardly recognized. This is extremely serious for R-GCN-based method.

### 8.4.2 Comparison of time-consumption of proposed approaches

This experiment compares the total time-consumption of executing the offline training and online tagging processes of the random-forest and R-GCN-based approaches on the five test groups. In random forest-based approach, the number of trees (NT) is the key parameter affecting the time consumption and the tagging accuracy. For R-GCN approach, the hidden layer (HN) and the number of epochs (EN) are the two key parameters that affect the time consump-


Confusion Matrix
Output Class
Target Class

| | o | l | s | sem | pc | lib | t | c | sto | k | b | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| o | 1383<br>41.5% | 133<br>4.0% | 76<br>2.3% | 42<br>1.3% | 1<br>0.0% | 21<br>0.6% | 40<br>1.2% | 1<br>0.0% | 7<br>0.2% | 11<br>0.3% | 22<br>0.7% | 79.6%<br>20.4% |
| l | 69<br>2.1% | 300<br>9.0% | 4<br>0.1% | 21<br>0.6% | 2<br>0.1% | 4<br>0.1% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 5<br>0.2% | 74.1%<br>25.9% |
| s | 27<br>0.8% | 0<br>0.0% | 532<br>16.0% | 3<br>0.1% | 0<br>0.0% | 2<br>0.1% | 34<br>1.0% | 10<br>0.3% | 13<br>0.4% | 1<br>0.0% | 4<br>0.1% | 85.0%<br>15.0% |
| sem | 13<br>0.4% | 30<br>0.9% | 6<br>0.2% | 43<br>1.3% | 1<br>0.0% | 17<br>0.5% | 3<br>0.1% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 37.7%<br>62.3% |
| pc | 0<br>0.0% | 1<br>0.0% | 0<br>0.0% | 0<br>0.0% | 2<br>0.1% | 6<br>0.2% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 20.0%<br>80.0% |
| lib | 4<br>0.1% | 1<br>0.0% | 1<br>0.0% | 18<br>0.5% | 0<br>0.0% | 24<br>0.7% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 49.0%<br>51.0% |
| t | 8<br>0.2% | 0<br>0.0% | 33<br>1.0% | 1<br>0.0% | 0<br>0.0% | 0<br>0.0% | 259<br>7.8% | 5<br>0.2% | 7<br>0.2% | 5<br>0.2% | 0<br>0.0% | 81.4%<br>18.6% |
| c | 0<br>0.0% | 0<br>0.0% | 3<br>0.1% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0.0%<br>100% |
| sto | 2<br>0.1% | 0<br>0.0% | 8<br>0.2% | 2<br>0.1% | 0<br>0.0% | 0<br>0.0% | 10<br>0.3% | 2<br>0.1% | 18<br>0.5% | 2<br>0.1% | 0<br>0.0% | 40.9%<br>59.1% |
| k | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 1<br>0.0% | 8<br>0.2% | 0<br>0.0% | 3<br>0.1% | 4<br>0.1% | 0<br>0.0% | 25.0%<br>75.0% |
| b | 3<br>0.1% | 0<br>0.0% | 2<br>0.1% | 2<br>0.1% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0<br>0.0% | 0.0%<br>100% |
| | 91.7%<br>8.3% | 64.5%<br>35.5% | 80.0%<br>20.0% | 32.6%<br>67.4% | 33.3%<br>66.7% | 32.0%<br>68.0% | 73.0%<br>27.0% | 0.0%<br>100% | 37.5%<br>62.5% | 17.4%<br>82.6% | 0.0%<br>100% | 77.0%<br>23.0% |

**Figure 8.13.** Confusion matrix of decision tree-based approach.

tion. In this experiment, the NT varying from 1 to 100 with an interval at 5, while HN varies from 30 to 250 with an interval at 10 and EN varies from 70 to 270 with an interval at 10. Note that, the time-consumption refers to the time used in the training and tagging process of the five-fold test. Figure 8.14 shows the variation of time-consumption and the tagging accuracy as the increase of the tree number. We can see from approximately 40 trees, the tagging accuracy converges 0.85 and at this point, it takes nearly one hour to tag all the test floor plans in the five test groups. The addition of more trees dramatically increases the total time consumption. This is mainly caused by the bi-directional beam search strategy, dramatically increasing the execution count of the prediction function of random forest. The fluctuation in the time-consumption line is mainly due to the interface of the other application running on the same computer.

Figure 8.15 shows the variation of time-consumption as the increase of HN and EN. We can see that as the increase of the number of features and epochs, the consumed time gradually increases (from around 3 minutes to maximum 16 minutes), which, is far less than random-forest based approach (1 hour). This is because (1) the number of layers is small (2 layers) and (2) all the nodes (rooms) including training and test nodes in the graph share the same weight matrixes, which are updated at each epoch. After the last epoch, the type of test nodes can be easily estimated by invoking the softmax function based on the feature representation at the output layer.

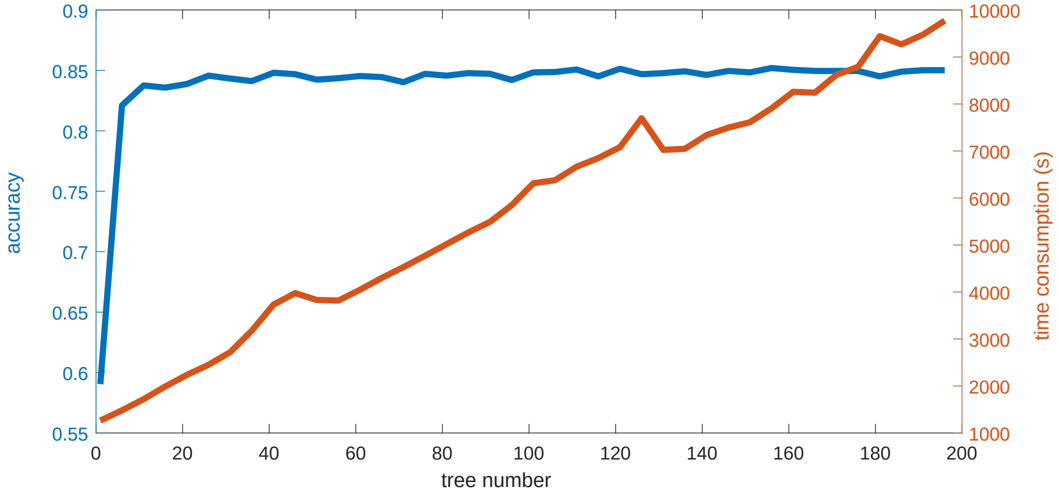


**Figure 8.14.** Impact of tree number on tagging accuracy and time-consumption.

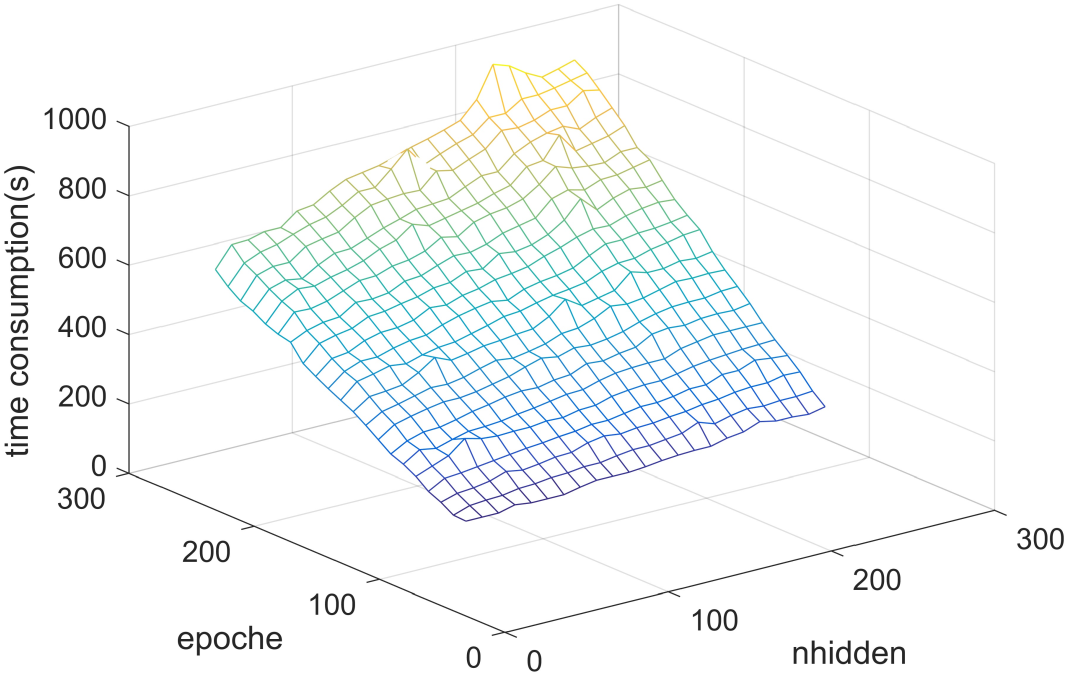


**Figure 8.15.** Impact of number of epochs and features in hidden layers on time-consumption.

## 8.5 Discussions

We tested our approaches with the experimental data obtained from two universities in Germany. Currently, data privacy issues restrict access to detailed floor plans across different countries. The indoor structures of a certain public building might vary greatly as countries. Thus, we could not guarantee that the trained model can be applied to evaluate the public buildings in other countries. However, we still believe this work is valuable since the local characteristics can also be exploited by using the known floor plans from a local region such as a country to train a model and estimating the missing room usage of floor plans in the same local region.

The proposed approaches leverage on the geometric properties, topological relationships, and spatial distribution of rooms, which are common properties in public buildings. Thus, the proposed approaches can be extended to infer the room usage of other building types, such as hospitals and office buildings that have similar indoor layouts to research buildings. Figure 8.16 shows the indoor map of a hospital on MazeMap without room usage, from which, the geometry and topology information of rooms can be extracted in a similar manner as research buildings. The biggest challenge is that the indoor layout of public building (e.g., hospital) can vary as spatial locations. For instance, the indoor layouts of the building that belong to the same hospital are similar but the indoor layouts of the building in different counties might be totally different. One of the potential solutions is to represent the spatial location of a floor plan from multiple scales, such as (belongs to) a building, an institute, a district, a city, a state, a country, and a continent. Each scale is added in the machine-learning approach (e.g., random forest) as a variable or feature. Apart from public buildings, digitalizing the indoor map of residential houses from scanned floor plan is also of great importance and have been widely investigated (Dodge et al. 2017), such as reconstructing the room geometry and topology by using machine-learning but without the room usage information (e.g., kitchen, dining room, and toilets). In this case, the proposed R-GCN approach might work well since intuitively the topological relationship between rooms in residential houses are significant in identifying the usage of rooms (Rosser et al. 2017).

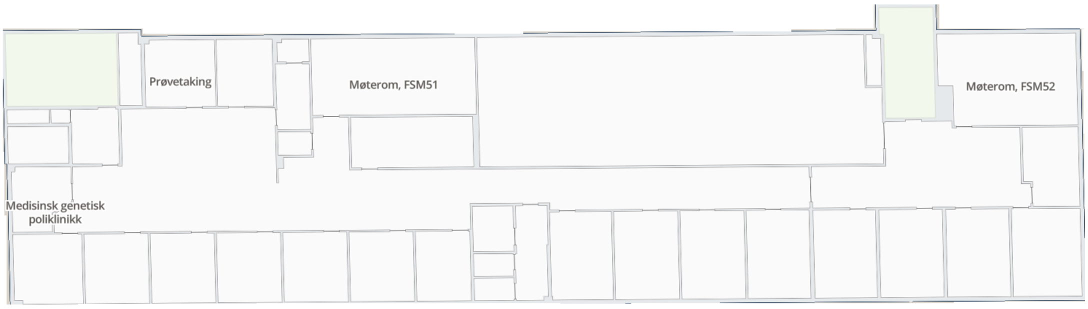


**Figure 8.16.** Indoor map of a hospital published on MazeMap.

## 8.6 Conclusions

This work has adopted two kinds of statistical learning-based room tagging approaches for public buildings to enhance the manual and automatic mapping solutions by providing the missing room usage information based on geometric maps. We compare the traditional machine learning methods (i.e., random forest and decision tree) with deep learning methods (i.e., R-GCN) based on 130 floor plans of research buildings at universities. R-GCN does not outperform random forest in tagging accuracy as we anticipate, which is mainly because in the room tagging issue, relationship-related features are not as important as other features. However, R-GCN is much more computational-efficiency than random forest-based approach and achieves a better tagging result than decision tree.

Several tasks are planned for future works. One is to improve the tagging accuracy of less-frequent room types (e.g., seminars, and libraries) by leverage on web mining techniques. Some useful information, such as the office number of a researcher and a report that contains the location (e.g., room number), is normally available from the webpage of a university or institute, based on which the number of offices, libraries, seminar rooms, and PC rooms at a certain floor might be extracted. Second, we will further investigate the ways of discovering spatial knowledge based on the correlation among geometry, topology, semantics, and spatial distribution of spaces. For instance, to infer the complete geometry and topology of spaces given the semantics and coarse location of POIs in shopping malls. This will be especially useful in improving indoor VGI data, whose quality (e.g., accuracy and completeness) cannot be guaranteed.

# Reference

Ahmed, S., Liwicki, M., Weber, M., and Dengel, A. (2011). Improved automatic analysis of architectural floor plans. In *2011 International Conference on Document Analysis and Recognition*, pages 864–869. IEEE.

Alzantot, M. and Youssef, M. (2012). Crowdinside: automatic construction of indoor floorplans. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 99–108. ACM.

Ambruş, R., Claici, S., and Wendt, A. (2017). Automatic room segmentation from unstructured 3-d data of indoor environments. *IEEE Robotics and Automation Letters*, 2(2):749–756.

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S. (2016). 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543.

Aydemir, A., Jensfelt, P., and Folkesson, J. (2012). What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4675–4682. IEEE.

Becker, S., Peter, M., and Fritsch, D. (2015). Grammar-supported 3d indoor reconstruction from point clouds for" as-built" bim. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chen, S., Li, M., Ren, K., and Qiao, C. (2015). Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos. In *2015 IEEE 35th International conference on distributed computing systems*, pages 1–10. IEEE.

de las Heras, L.-P., Ahmed, S., Liwicki, M., Valveny, E., and Sánchez, G. (2014). Statistical segmentation and structural recognition for floor plan interpretation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):221–237.

de las Heras, L.-P., Mas, J., Sánchez, G., and Valveny, E. (2011). Notation-invariant patch-based wall detector in architectural floor plans. In *International Workshop on Graphics Recognition*, pages 79–88. Springer.

de las Heras, L.-P., Terrades, O. R., and Lladós, J. (2015). Attributed graph grammar for floor plan analysis. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 726–730. IEEE.

Dehbi, Y., Gojayeva, N., Pickert, A., Haunert, J., and Plümer, L. (2018). Room shapes and functional uses predicted from sparse data. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(4).

Dodge, S., Xu, J., and Stenger, B. (2017). Parsing floor plan images. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 358–361. IEEE.

Dosch, P., Tombre, K., Ah-Soon, C., and Masini, G. (2000). A complete system for the analysis of architectural drawings. *International Journal on Document Analysis and Recognition*, 3(2):102–116.

Elhamshary, M. and Youssef, M. (2015). Semsense: Automatic construction of semantic indoor floorplans. In *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–11. IEEE.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. (2009). Reconstructing building interiors from images. In *2009 IEEE 12th International Conference on Computer Vision*, pages 80–87. IEEE.

Gao, R., Zhao, M., Ye, T., Ye, F., Wang, Y., Bian, K., Wang, T., and Li, X. (2014). Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 249–260. ACM.

Gao, R., Zhou, B., Ye, F., and Wang, Y. (2017). Knitter: Fast, resilient single-user indoor floor plan construction. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE.

Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American statistical Association*, 95(452):1300–1304.

Gimenez, L., Robert, S., Suard, F., and Zreik, K. (2016). Automatic reconstruction of 3d building models from scanned 2d floor plans. *Automation in Construction*, 63:48–56.

Hu, X., Fan, H., Noskov, A., Zipf, A., Wang, Z., and Shang, J. (2019). Feasibility of using grammars to infer room semantics. *Remote Sensing*, 11(13):1535.

Hu, X., Fan, H., Zipf, A., Shang, J., and Gu, F. (2017). A conceptual framework for indoor mapping by using grammars. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:335.

Huang, H. and Gartner, G. (2009). A survey of mobile indoor navigation systems. In *Cartography in Central and Eastern Europe*, pages 305–319. Springer.

Ikehata, S., Yang, H., and Furukawa, Y. (2015). Structured indoor modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1323–1331.

Jiang, Y., Xiang, Y., Pan, X., Li, K., Lv, Q., Dick, R. P., Shang, L., and Hannigan, M. (2013). Hallway based automatic indoor floorplan construction using room fingerprints. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 315–324. ACM.

Kattenbeck, M. (2015). Empirically measuring salience of objects for use in pedestrian navigation. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 3. ACM.

Khoshelham, K. and Díaz-Vilariño, L. (2014). 3d modelling of interior spaces: Learning the language of indoor architecture. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):321.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Klonk, C. (2016). *New laboratories: Historical and critical perspectives on contemporary developments*. Walter de Gruyter GmbH & Co KG.

Luperto, M. and Amigoni, F. (2019). Predicting the global structure of indoor environments: A constructive machine learning approach. *Autonomous Robots*, 43(4):813–835.

Luperto, M., Li, A. Q., and Amigoni, F. (2013). A system for building semantic maps of indoor environments exploiting the concept of building typology. In *Robot Soccer World Cup*, pages 504–515. Springer.

Luperto, M., Riva, A., and Amigoni, F. (2017). Semantic classification by reasoning on the whole structure of buildings using statistical relational learning techniques. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2562–2568. IEEE.

Macé, S., Locteau, H., Valveny, E., and Tabbone, S. (2010). A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 167–174. ACM.

Mitchell, W. J. (1990). *The logic of architecture: Design, computation, and cognition*. MIT press.

Peter, M., Becker, S., and Fritsch, D. (2013). Grammar supported indoor mapping. In *Proceedings of the 26th International Cartographic Conference, Dresden, Germany*, volume 2530.

Philipp, D., Baier, P., Dibak, C., Dürr, F., Rothermel, K., Becker, S., Peter, M., and Fritsch, D. (2014). Mapgenie: Grammar-enhanced indoor map construction from crowd-sourced data. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 139–147. IEEE.

Pronobis, A. and Jensfelt, P. (2012). Large-scale semantic mapping and reasoning with heterogeneous modalities. In *2012 IEEE international conference on robotics and automation*, pages 3515–3522. IEEE.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing*.

Rosser, J. F., Smith, G., and Morley, J. G. (2017). Data-driven estimation of building interior plans. *International Journal of Geographical Information Science*, 31(8):1652–1674.

Sankar, A. and Seitz, S. (2012). Capturing indoor scenes with smartphones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 403–412. ACM.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Xiong, X., Adan, A., Akinci, B., and Huber, D. (2013). Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in construction*, 31:325–337.

Yassin, A., Nasser, Y., Awad, M., Al-Dubai, A., Liu, R., Yuen, C., Raulefs, R., and Aboutanios, E. (2016). Recent advances in indoor localization: A survey on theoretical approaches and applications. *IEEE Communications Surveys & Tutorials*, 19(2):1327–1346.

Yue, K., Krishnamurti, R., and Grobler, F. (2011). Estimating the interior layout of buildings using a shape grammar to capture building style. *Journal of Computing in Civil Engineering*, 26(1):113–130.

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

# 9. Data-driven approach to learning salience models of indoor landmarks by using genetic programming

## Article information

| | |
|---|---|
| Authors | Xuke Hu, Lei Ding, Jianga Shang, Hongchao Fan, Tessio Noskov,Alexey Noskov, Alexander Zipf |
| Journal | International Journal of Digital Earth |
| History | Received: 23 Jun 2019<br>Revised: 25 Aug 2019<br>Accepted: 02 Dec 2019 |
| Reference | Hu, X., Ding, L., Shang, J., Fan, H., Novack, T., Noskov, A., Zipf, A. (2020). Data-driven approach to learning salience models of indoor landmarks by using genetic programming. International Journal of Digital Earth, 1-28. |

## Abstract

In landmark-based wayfinding, determining the most salient landmark from several candidates at decision points is challenging. To overcome this problem, current approaches usually rely on a linear model to measure the salience of landmarks. However, linear models are not always able to establish an accurate quantitative relationship between the attributes of a landmark and its perceived salience. Furthermore, the numbers of evaluated scenes and of volunteers participating in the testing of these models are often limited. With the aim of overcoming these gaps, we propose learning a non-linear salience model by means of genetic programming. We compared our proposed approach with conventional algorithms by using photographs of two hundred test scenes collected from two shopping malls. Two hundred volunteers who were not in these environments were asked to answer questionnaires about the collected photographs. The results from this experiment showed that in 76% of the cases, the most salient landmark (according to the volunteers' perception) was correctly predicted by our proposed approach. This accuracy rate is considerably higher than the ones achieved by conventional linear models.

**Keywords:** Indoor Navigation; Landmarks; Salience Model; Genetic Programming;

## 9.1 Introduction

Landmarks, which are defined as prominent features in people's cognitive maps of both physical and virtual indoor and outdoor environments, are of fundamental importance in people's interaction with unfamiliar environments, especially with respect to wayfinding (Caduff 2007; Duckham et al. 2010; Kattenbeck 2017; Nothegger et al. 2004; Sorrows and Hirtle 1999). In comparison to conventional navigation instructions based on distance and azimuth, and especially in complex indoor and outdoor environments, landmark-based wayfinding is known to be more effective in raising the pedestrian's confidence and reducing his/her anxiety during navigation, thus guiding the person successfully to his/her destination (Kattenbeck 2015; Kattenbeck et al. 2018; May et al. 2003; Nothegger et al. 2004; Richter and Winter 2014; Ross et al. 2004). Hence, different landmark-based wayfinding approaches have been proposed in recent years (Butz et al. 2001; Duckham et al. 2010; Hund and Padgitt 2010; Li et al. 2017; Lyu et al. 2015; Michon and Denis 2001; Raubal and Winter 2002).

However, a challenge of landmark-based wayfinding is choosing the correct, i.e., most salient, landmark at a decision point where multiple features can be perceived. Figure 9.1 shows an example of a decision point in a shopping mall where a Pizza Hut sign (A), staircase (B), GYZG shop (C), and Nike shop (D) can be seen and perceived as possible landmarks. The most salient of these features is the one that, if used in a wayfinding instruction, will demand less effort from the users in understanding and following the instruction and finally getting to the destination. Typically, the salience of a landmark is estimated based on a linear model and a predefined set of categories of attributes, such as its visual (e.g., the facade's area, shape, colour, and texture) prominence, semantic (e.g., socio-cultural) importance, and structural (e.g., nodes, boundaries, and regions) significance (Klippel and Winter 2005; Li et al. 2017; Raubal and Winter 2002; Sorrows and Hirtle 1999). Raubal and Winter (2002) and Klippel and Winter (2005), for example, proposed a linear model based on empirically weighted landmark attributes. Li et al. (2017), on the other hand, proposed a landmark-based cognition strength grid model, based on which the salience of an indoor landmark is measured using a linear model. However, in these approaches, the weights of the attributes are empirically defined based on expert knowledge. To overcome this drawback, Götze and Boye (2016) proposed an approach for learning the weight values of the salience model directly from the route instructions given by humans. Another problem with these approaches is that the proposed models are all linear, which might hamper their accuracy in quantitatively representing the relationship between the attributes of a landmark and its

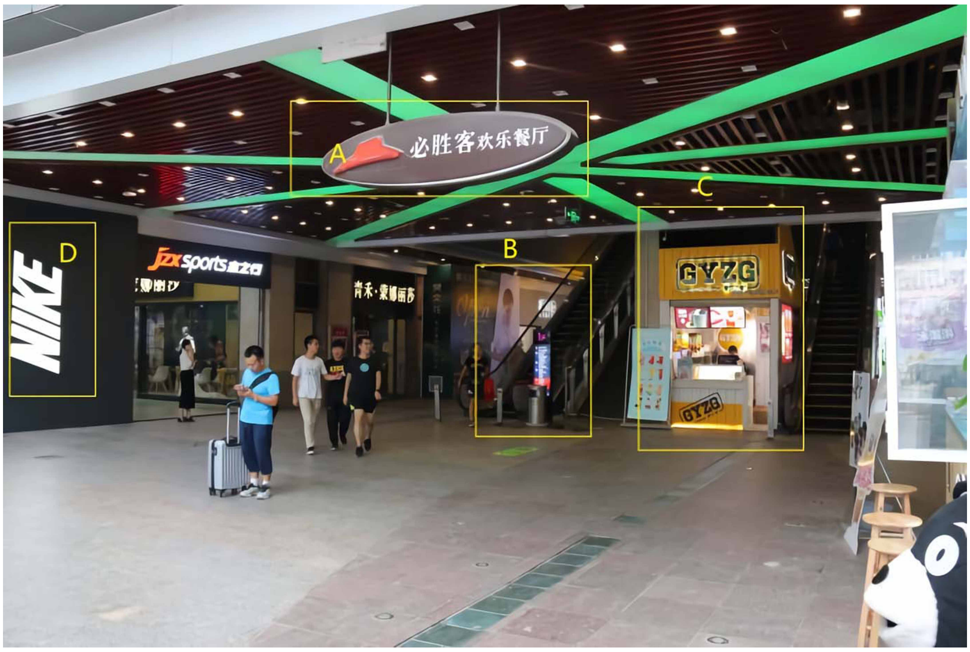


**Figure 9.1.** Example of a decision point in a shopping mall and the possible landmarks that can be used for wayfinding (indicated with the letters A to D).

salience. Furthermore, experiments for validating the landmark salience estimations made by a model are rarely based on the correlation of these estimations with the importance of landmarks, as perceived by users, for the purpose of wayfinding. Aiming to overcome these drawbacks, this study proposes a data-driven approach for learning non-linear salience models of indoor landmarks, specifically in shopping malls, based on genetic programming (GP) (Koza 1991). Firstly, visual and semantic attributes were extracted from the landmarks. These attributes are, to some extent, different from the ones considered so far in related studies (Lyu et al. 2015; Raubal and Winter 2002; Sorrows and Hirtle 1999). Two hundred pictures were obtained of indoor scenes in shopping malls, and in each of these pictures, three to four landmarks were manually identified. Next, 200 volunteers were recruited and asked to select, from each picture, the landmark that is most suitable for navigation in that scene. In total, 40,000 answers regarding the salience of these landmarks were obtained. In the next step, a GP algorithm was applied for learning the most suitable model for estimating the salience of landmarks in shopping malls based on the data obtained from the volunteers. Given enough training data from volunteers, the proposed approach is applicable to other indoor, as well as to outdoor, environments. The contribution of this study is hence three-fold: (1) for the first time, to the best of our knowledge, non-linear models are applied for estimating the salience of indoor landmarks; (2) the non-linear models are learned based on GP; and (3) a large benchmark of 40,000 volunteer-provided evaluations of landmarks, which can be used for developing and testing future landmark-based wayfinding approaches, is included.

The remainder of the paper is organized as follows. In Section 9.2, we provide an overview of related works on landmark salience modelling. In Section 9.3, the attributes used to quantify the landmark salience are introduced. In Section 9.4, the workflow and detailed steps of the proposed approach are presented. In Section 9.5, our approach is evaluated. Finally, in Sections 9.6 and 9.7, issues are discussed and conclusions are drawn.

## 9.2 Related work

For decades now, landmarks have been a standout concept in the spatial sciences. Landmarks are defined as physical features or places that have recognizable and memorable characteristics (Sorrows and Hirtle 1999). Richter and Winter (2014) defined landmarks as structural environments that act as cognitive anchors, markers, or reference points for the purposes of communication and wayfinding. Regarding the latter, there are two fundamental application domains, namely, outdoor and indoor wayfinding. Landmark-based wayfinding approaches are applicable to both of these domains.

The application of landmarks in wayfinding was first implemented in outdoors navigation (Götze and Boye 2016; Klippel and Winter 2005; Nothegger et al. 2004; Raubal and Winter 2002; Sorrows and Hirtle 1999). In one of the first works in this area, Sorrows and Hirtle (1999) proposed a categorization of landmarks into visual, cognitive, and structural landmarks. This categorization implies a dependency between the category of a landmark and the structure of the built-up environment in which it is located. Similarly to the categorization proposed by Sorrows and Hirtle (1999), Raubal and Winter (2002) categorized the attributes of landmarks into visual (e.g., facade areas, shapes, colours, and visibility) prominence, semantic (i.e., social-cultural and historical) importance, and structural (e.g., nodes, boundaries, and regions) prominence. Based on these attribute types, they proposed a formal model for measuring the salience of the landmarks. This model has been widely adopted by many other researchers. For example, based on the work by Raubal and Winter (2002), Nothegger et al. (2004) proposed a computational model that allows the automatic identification of salient landmarks along a road at decision points in urban environments. Their proposed model is cross-checked with a survey for landmarks along a given route that contains 9 scenes or decision points in the city of Vienna, Austria. The results showed that the landmarks selected by the model are highly correlated with the ones selected at these decision points by pedestrians participating in the survey.

Klippel and Winter (2005) proposed an approach to formalizing the structural salience of objects along routes and integrating landmarks into route directions. The structural salience of point-like objects (e.g., buildings) is approached with taxonomic consideration and with respect to their positions along a route. The results are used to extend a formal language of route knowledge. Caduff (2007); Caduff and Timpf (2008), on the other hand, proposed a conceptual framework for assessing the salience of landmarks for the purpose of pedestrian navigation. In their approach, salience is represented as a three-valued vector comprising perceptual, cognitive, and contextual salience. Considering the lack of an empirically validated model and approach for survey-based assessment of object salience, Kattenbeck (2015) tested a structural equation model using a large scale in-situ experiment. The results showed that the derived model was able to explain 72% of the variance present in the overall salience. Later on, Kattenbeck (2017) empirically analysed hypotheses about the ways different sub-dimensions of salience (visual, structural, and cognitive aspects, as well as proto-typicality and visibility in advance) have an impact on each other. To verify the robustness of a survey-based model across different environments, objects, and observers, Kattenbeck et al. (2018) conducted a heterogeneity analysis by considering the different environments, senses of direction, and genders of volunteers. Götze and Boye (2016) proposed a data-driven approach for automatically deriving a mathematical model of salience directly from route instructions given by humans. Specifically, the authors used a ranking support vector machine (SVM) method to derive the weights of a linear model based on the original model of Raubal and Winter (2002). The experimental results showed that the model can successfully predict the salient landmark that was preferred by users.

Landmark-based approaches are also gaining more and more attention as a research advancement in the domain of indoor wayfinding. Although indoor landmarks have a common nature with outdoor landmarks, indoor environments (e.g., subway and railway stations, shopping malls, airports) have a larger variety of features (e.g., stairs, unit room, corridors, furniture, goods, signboards). Thus, the criteria for selecting outdoor landmarks and measuring their salience require modifications to be applicable to indoor environments (Li et al. 2017). Millonig and Schechtner (2007) presented an approach for identifying the salient landmarks next to primarily observed main routes of pedestrian flows in a train station and for representing landmark-based spatial routing information. Lyu et al. (2015) proposed several salience indicators and a computational method for the extraction of indoor landmarks and, based on the model developed by Raubal and Winter (2002), adapted the use of predefined weights with expert knowledge. Li et al. (2017) proposed a landmark-based cognition strength grid (CSG) model, in which each grid cell is embedded with salient characteristics. They can be oriented to surrounding landmarks to ensure the use of the CSG model to plan various routes, such as the route with the most identifiable landmarks. Again, their salience model is based on the model proposed by Raubal and Winter (2002). The authors evaluated their model based on two different scenarios in a large shopping mall. Their work exemplifies the diverse applications of the CSG model in indoor wayfinding.

Despite the significant contribution of these studies to landmark-based wayfinding in both indoor and outdoor environments, two main problems persist: (1) Current approaches normally use a linear model to measure the salience of landmarks, which, however, cannot accurately represent the quantitative relationship between the attributes of a landmark and its salience.

Furthermore, these models cannot adapt to the changing of the environment. (2) The numbers of considered scenarios and of volunteers participating in the evaluation of the models are limited, thus pointing to the necessity of more consistent model evaluations. In this study, we aim to cover these gaps by learning non-linear salience models by means of genetic programming and evaluating the models based on two hundred testing scenes and two hundred volunteers.

## 9.3 Salience indicators of indoor landmarks

This research focuses on the automatic selection of the most salient landmark among candidate landmarks at decision points in indoor environments, shopping malls in particular. For quantifying the salience of landmarks, the attributes of the landmarks first need to be extracted. As mentioned, conventional attributes fall into the following categories: visual (e.g., facade area, shape, colour, and texture) prominence, semantic (i.e., socio-cultural) significance, and structural importance (i.e., whether it has an important role in the perception and understanding of the structure of the environment). In indoor environments, however, structural attributes are less relevant because of the limited sight in these environments (Klippel and Winter 2005). Hence, in this work, the structural attributes of landmarks are not considered. Table 9.1 presents the attributes considered in measuring the salience of the landmarks in this study. In the next section, we provide an explanation on how these attributes are computed.

### 9.3.1 Visual prominence

This type of attribute refers to the degree of visual prominence of a landmark when compared to its surrounding environment. In general, the properties of visual prominence include *facade area*, *facade area of subject*, *shape deviation*, *shape ratio*, and *colour*, which can be extracted from the images.

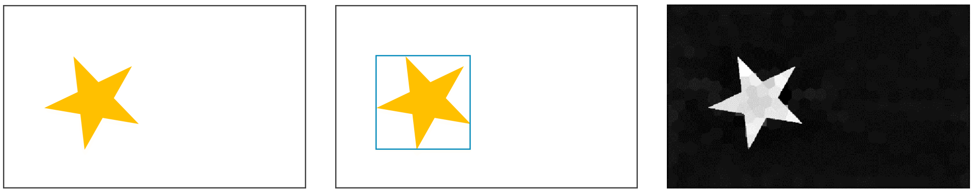


**Figure 9.2.** Visual attribute extraction from image. (a) Yellow star shape. (b) Minimum boundary box of the shape. (c) Colour-based salience map.

**Facade area** ($A_L$)**:** The facade area refers to the observable area of a landmark's facade. Humans find it easier to recognize landmarks whose facade area is significantly larger than its surrounding objects. The facade area of an object is generally calculated by multiplying its

**Table 9.1.** Attributes used for measuring the salience of indoor landmarks.

| **Attribute type** | **Attribute name** | **Mathematical symbol** |
|---|---|---|
| Visual | Facade area ($A_L$) | $x_1$ |
| | Facade area of attached subject ($A_{L_{Subject}}$) | $x_2$ |
| | Shape deviation ($D_L$) | $x_3$ |
| | Shape ratio ($R_L$) | $x_4$ |
| | Colour ($C$) | $x_5$ |
| Semantic | Architecture class ($Arch$) | $x_6$ |
| | Information class ($Info$) | $x_7$ |
| | Shop class ($Shop$) | $x_8$ |
| | Function class ($Func$) | $x_9$ |
| | Furniture class ($Furn$) | $x_{10}$ |
| | Text ($Text$) | $x_{11}$ |
| | Foreign language ($ForText$) | $x_{12}$ |
| | Count of Baidu search ($Baidu$) | $x_{13}$ |
| | Count of Google search ($Google$) | $x_{14}$ |

width and height. However, the shapes of some landmark facades might be irregular; therefore, we consider instead the amount of pixels of the landmark's facade. *L*, *P*, $A_L$, and $Pix(x)$ are used to represent, respectively, a landmark, the image containing the landmark, its facade area, and the function for calculating the number of pixels of the facade. As shown in Figure 9.2a, the facade area of the yellow star (*L*) is calculated by dividing its number of pixels by the total number of pixels of the image. Thus, the equation for computing the facade area of a landmark is

$$A_L = Pix(L)/Pix(P). \tag{9.1}$$

**Facade area of attached subject ($A_{L_{Subject}}$):** For most landmarks, the facade area of the attached subject equals the facade area of the landmark, as shown in Figure 9.2a. However, some landmarks, such as shops, usually have so-called attached subjects (e.g., entrance and logo) that are distinct from the facade of the landmark. Humans are easily attracted to venues that have a wide entrance or a large sign (Raubal and Winter 2002). We denote $L_{Subject}$ and $A_{L_{Subject}}$ as the attached subject and the facade area of the attached subject, respectively. The equation for calculating the facade area of the attached subject is

$$A_{L_{Subject}} = Pix(L_{Subject})/Pix(P). \tag{9.2}$$

**Shape deviation ($D_L$):** The shape deviation of a landmark equals the difference between the area of its smallest enclosing bounding box and its facade area (Raubal and Winter 2002). For example, the rectangle (denoted by $r$) in Figure 9.2b surrounded by the blue lines is the smallest enclosing bounding box of the yellow star (denoted by $L$). The larger the shape deviation, the more irregular the shape is. Inversely, a shape deviation of zero indicates that the landmark is a regular rectangle. We use $D_L$ and $L_{rectangle}$ to denote the shape deviation and the smallest enclosing bounding box of the landmark $L$, respectively. The equation for calculating the shape deviation is

$$D_L = (Pix(L_{rectangle}) - Pix(L))/Pix(L_{rectangle}). \tag{9.3}$$

**Shape ratio ($R_L$):** The shape ratio of a landmark equals the height-to-width ratio of the smallest enclosing bounding box of the landmark (Raubal and Winter 2002). We use this attribute because high and narrow landmarks are more visually attractive than short and thick ones. We use $R_L$, $L_{rectangle}$, $Length(x)$, and $Width(x)$ for denoting, respectively, the shape ratio of landmark $L$, its smallest enclosing bounding rectangle, the function for calculating the length of an object, and the function for calculating its width. Equation 9.4 shows how the shape ratio is computed. In Figure 9.2b, the shape ratio of the yellow star equals the ratio of the height to width of the smallest enclosing bounding rectangle (the blue square in Figure 9.2b). This attribute is therefore computed as follows:

$$R_L = Length(L_{rectangle})/Width(L_{rectangle}). \tag{9.4}$$

**Colour (*C*):** Colour refers to the colour difference of an object from its surrounding objects. An object will receive more attention if its colour significantly contrasts with the colours of the environment (Nothegger et al. 2004), such as a red fire hydrant against a white wall. To extract this attribute, we adopt the high-dimensional colour transform approach (Kim et al. 2016), which can detect the salient region of an image. Specifically, it is used to generate a grayscale image from the original scene image, which is a per-pixel salience map, as shown in Figure 9.2c. The image is scaled into 256 levels of grey, thus assigning each pixel a digital number ranging from 0 to 255. The brighter the colour, the more salient the pixel. We then use the average pixel value of the landmark in the salience map as the value of colour.

### 9.3.2 Semantic attraction

Semantic attraction refers to the semantic significance of a landmark in the sense of its socio-cultural importance (Lyu et al. 2015; Nothegger et al. 2004;?; Raubal and Winter 2002). In this work, the semantic attraction of a landmark is quantified based on the following aspects:

**Text *(Text)*:** Text indicates if an object or landmark (e.g., information board near the entrance of a shopping mall and the signboard of shops) contains text (e.g., *HUAWEI*, *KFC*, *GREE*) either in Chinese or foreign languages. We use a Boolean variable to represent this attribute. If a landmark contains text, the value of this attribute equals one; otherwise, the value equals zero.

**Foreign text *(ForText)*:** Foreign text is the attribute referring to the *text* being in Chinese or in a foreign language. Intuitively, an object that contains *foreign characters*is much more difficult to be remembered or understood by the locals. We also use a Boolean variable to represent this attribute.

**Count of Baidu search *(Baidu)*:** Count of Baidu search refers to the number of results when the name or class of a landmark is searched for using the Baidu search engine. This attribute to some degree reveals if a landmark is well-known by the locals (especially the Chinese), which is probably associated with its salience.For shops and sculptures, the search keywords are the name (e.g., Nike) and the description (e.g., sculpture of a cow), respectively. For other landmarks, the search keyword is the class of the landmark (e.g., vending machine and staircase). For Baidu searches, the input keyword is in Chinese. The search count is normally quite large. To make the value look smaller, it is divided by 100,000,000, which will be further normalized at the data processing step.

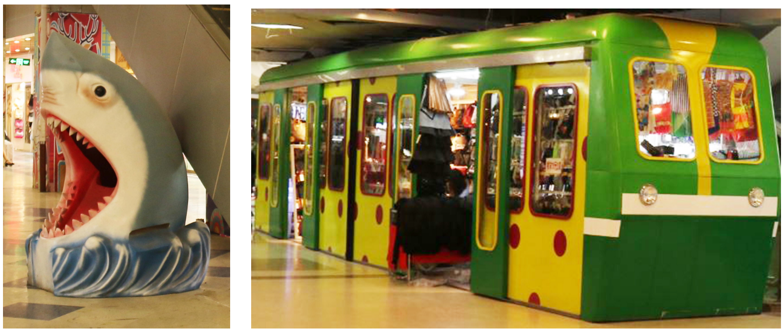
**Count of Google search *(Google)*:** Count of Google search refers to the number of results when the name or class of a landmark is searched for using the Google search engine. The purpose of using the attribute is to better understand if there exist large differences in recognizing landmarks in different cultures. For Google searches, the input keyword is in English. We divided the count of the search results by 100,000,000 to represent this attribute.

**Category *(Cat)*:** We assign each of the landmarks to one of five classes, Architecture (Arch), Information (Info), Shops (Shop), Function (Func), or Furniture (Furn), by referring to (Ohm et al. 2015). The definitions of these classes are as follows:

1. **Architecture** (Arch). The architecture class refers to objects built by humans, including houses and structures. Generally, houses that can be used as landmarks have a certain degree of particularity and uniqueness in shape. The structure is an immovable entity that has no internal space for people to use and is ornamental or functional. For example, pillars, fronts, sculptures, and fountains can be treated as architectural landmarks in shopping malls. Figure 9.3 shows two examples of architectural landmarks.

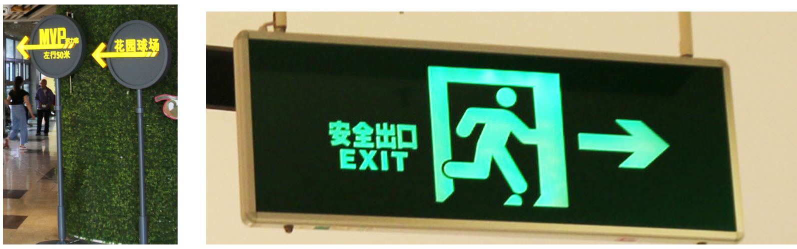
2. **Information** (Info). The information class refers to entities that can guide users to a certain place. This kind of landmark normally contains text or image and can be divided into two sub-classes according to its function: advertisement and identification. The first includes signs and posters of shops, as shown in Figure 9.4a. The second includes the sign of an emergency exit, as shown in Figure 9.4b.

3. **Shop** (Shop). The shop class mainly refers to the logo of a shop. This kind of landmark is located near its entity, such as a store, shop, or restaurant. The salience may be affected



(a) (b)

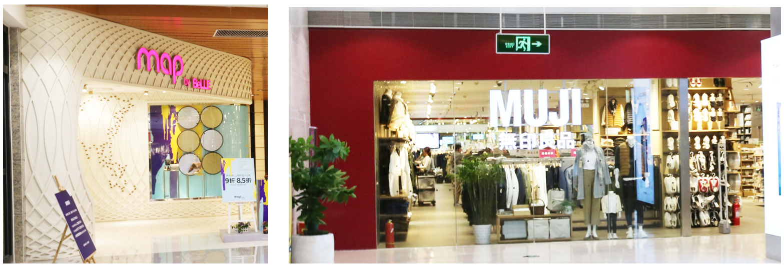
**Figure 9.3.** Examples of architectural landmarks.



(a) (b)

**Figure 9.4.** Examples of information landmarks.

by the attached subjects because of their large and colourful facade. Figure 9.5 shows two shops with attractive logos.



(a) (b)

**Figure 9.5.** Examples of shop landmarks.

4. **Function** (Func). The function class refers to space that plays the role of connecting other spaces. Pedestrians can move to other spaces through this entity, such as an elevator, escalator, flight of stairs, or corridor.

5. **Furniture**(Furn). The furniture class refers to entities that have specific functions. Landmarks under the furniture class are moveable. Examples include vending machines, doll machines, self-service photo cameras, bonsai trees, and commodity shelves.

## 9.4 Methodology

### 9.4.1 Workflow

The workflow of the proposed approach, which is illustrated in Figure 9.6, consists mainly of three stages: data collection and processing, genetic-programming-based model training, and model testing.

First, we collected images of 200 indoor scenes in shopping malls and manually marked the landmarks in each image. We then extracted all the needed attributes of each landmark according to the definition of attributes presented in Section 9.3 and normalized the values of all the attributes to the range from zero to one. The next step was to collect volunteers' preferences on landmarks through questionnaires. Specifically, we asked each volunteer to select what he/she thinks to be the most suitable landmark for navigation in each scene. The percentage of volunteers who selected a certain landmark was used as the salience value of this landmark. In the training stage, we used a genetic programming algorithm to learn a model that measures the salience of landmarks. In the test stage, we calculated the salience of each landmark in a test scene with the learned model. The landmark with the highest salience degree was regarded as the representative landmark in the scene.

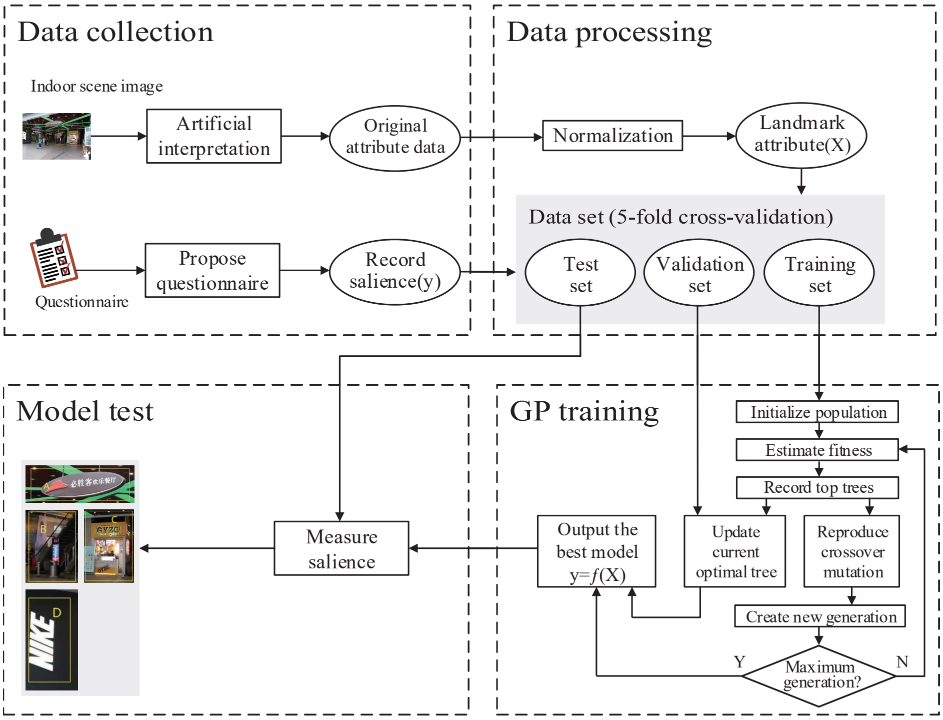


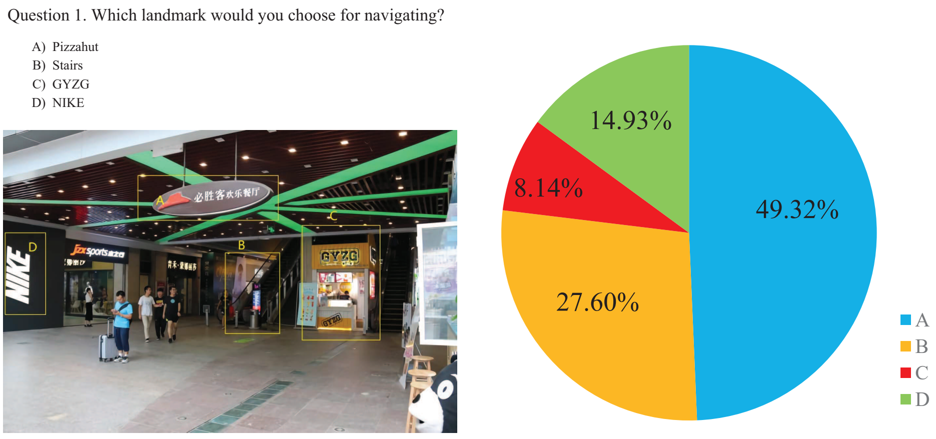
**Figure 9.6.** Workflow of proposed approach.

### 9.4.2 Data collection and processing

**Property of landmark:** In this part of the study, we first collected images of 200 scenes in shopping malls and manually marked the landmarks in each image. We then extracted the needed attributes of each landmark and calculated their values according to the definition in Section 9.3. To ensure that all the input attributes or attributes have the same importance, we used the min-max normalization method (as shown in Equation 9.5) to normalize the attributes, limiting the values of all attributes in the range of zero to one.

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{9.5}$$

**Questionnaire**: In the applied questionnaire, there were 200 questions, with each question referring to one of the 200 scenes. One of the questions from the proposed questionnaire is shown in Figure 9.7a. The question was: ‘Which landmark would you choose for navigation?’ Each volunteer was asked to choose the one landmark, from three or four landmarks in the scene, that best answered the question. We collected 40,000 answers in total from 200 volunteers. The proportion of the number of volunteers who selected a certain landmark in a scene was then treated as the salience value of the landmark, as shown in Figure 9.7b. For example, there are four landmarks, denoted by A, B, C, and D, respectively, in a given scene. The survey showed that 49%, 27%, 15%, and 8% of the volunteers chose A, B, C, and D, respectively, as the most salient landmark. Thus, the salience values of landmarks A, B, C, and D equal 0.49, 0.27, 0.15, and 0.08, respectively.



Question 1. Which landmark would you choose for navigating?

A) Pizzahut
B) Stairs
C) GYZG
D) NIKE

(a) (b)

**Figure 9.7.** Answer about salience of landmarks in a scene, collected through questionnaire.

### 9.4.3 GP-based model learning

Genetic programming (GP) is an inductive learning technique mimicking the principles of biological inheritance and evolution (Koza 1991). Each potential solution to the problem is represented as an individual in the population of potential solutions. GP operations, including reproduction, crossover, and mutation, are applied to the individuals in each generation to yield more diverse and better-performing individuals in the subsequent generations. This process is repeated for a few generations to yield optimal or near-optimal solutions (i.e., individuals) to the problem at hand.

GP is known to achieve very good results in the task of document ranking by learning nonlinear models that measure the degrees of relevance of documents in a set, given the user's input(Yeh et al. 2007). The aim of document ranking is to determine, according to the user's information requirement, which documents are relevant and which are not, which is a core task of information retrieval. The landmark selection problem is similar to document ranking. Therefore, in this research, we also applied GP to learn a mathematical model that can establish the quantitative relationship between the salience of a landmark and its visual and semantic attributes. Other ranking learning approaches, such as the variation of GP and the ranking vector SVM (Yu et al. 2012), will be investigated in future works. To apply GP for landmark salience model learning, we first needed to have a representation of an individual in the population. A tree structure was chosen to represent the individual in the population. An example of an individual is shown in Figure 9.8. The tree structure at the left represents a model $y = x_1 * x_2 + (x_3)^2$, whereas the one at the right represents $y = x_1 * x_2 + 0.5 + x_3$. The leaf nodes of the tree structure, known as terminals, denote variables or attribute values of landmarks and constants. The non-leaf nodes, known as functions, denote operators, such as $(+, *, sqrt, log)$. The nodes of operators are applied in the left and right sub-trees (for binary operators, such as $+$) or the single sub-tree (for unary operators, such as $log$). The operators, attributes, and weights together constitute the primitive set of our GP system.

To apply GP in our context, several components needed to be defined. These components and their definitions are presented in Table 9.2.

In this context, the operators we used included $+, -, /, *, abs$, and $log$. The terminals included the 14 normalized attributes of landmarks in Table 9.1, which were numbered consecutively and denoted by $x_1, x_2, ...,$ and $x_{14}$, respectively, and 10 constants: $0.1, 0.2, ..., 1$. We used Equation 9.6 to calculate the fitness of an individual. $y$ denotes the true degree of salience, whereas $\hat{y}$ denotes the estimated salience by the model (individual). $m$ denotes the number of landmarks in all training scenes.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{9.6}$$

**Table 9.2.** Essential GP components.

| **GP parameters** | **Meaning** |
|---|---|
| $S_{pop}$ | Population size |
| $N_{gen}$ | Number of evolution generations |
| $N_{top}$ | Number of best individuals in each generation |
| $N_{run}$ | Number of independent runs to perform |
| Operators | Non-leaf nodes in the tree structure |
| Terminals | Leaf nodes in the tree structure |
| Fitness function | Objective function that needs to be optimized |
| Termination condition | Condition that determines when GP will terminate |
| Reproduction | Genetic operator that copies the individuals directly into the population of the next generation |
| Crossover | Genetic operator that exchanges sub-trees from two parents to breed two new children |
| Mutation | Genetic operator that randomly selects a sub-tree and replaces it with another randomly created sub-tree |

The procedure of the population evolution in GP was as follows:

- Randomly generate an initial population of $S_{pop}$ valid trees.
- Perform the following sub-steps for $N_{gen}$ generations:
  - Calculate the fitness value of each tree.
  - Record the top $N_{top}$ trees with the highest fitness values.
  - Update the current optimal tree by selecting the one that has the best performance on the validation set from the top $N_{top}$ trees crossing from the initial generation to the current generation.
  - Create a new population via the following three genetic operations based on the top $N_{top}$ trees.
    - Reproduction
    - Crossover
    - Mutation
- Treat the current optimal tree as the unique discovery output.

In each generation, $S_{pop}$ best trees were selected based on the training set. In total, there were $S_{pop}*N_{top}$ best trees in $S_{pop}$ generations. The purpose of the validation set was to select the optimal tree that performed best on the validation set from the $S_{pop}*N_{top}$ trees as the final output. Note that when producing a new individual tree, we first checked if the tree was a valid mathematical expression. If not, it was deleted, and a new tree was produced until a valid tree was found. For instance, the unary operators should have only one operand, and the binary operators should have two operators. The operand of the operator $log$ should be a positive value, whereas the second operand of the operator / should not be zero.

Moreover, three parameters were used in creating a new population: reproduction rate, which is denoted by $R_{rate}$; crossover rate, which is denoted by $C_{rate}$; and mutation rate, which is denoted by $M_{rate}$. These parameters represented the proportion of the newly produced individuals by the corresponding genetic operations. The sum of the three parameters equals one. The purpose of the validation set was to help alleviate the problem of overfitting of GP on the training data and to select the best generalized model.

We used the tournament selection method to select parents for crossover and mutation. Crossover utilized a standard random sub-tree swapping algorithm.

Detailed descriptions of the three genetic operations are as follows:

1. Reproduction: An individual is copied from the current population. This operation can deliver the best individuals to the next generation.

2. Mutation: A sub-tree (a gene) is randomly selected from an individual, and the sub-tree is replaced with another randomly created sub-tree. Figure 9.8 illustrates the process of the mutation operation. The aim of the mutation operation is to improve the diversity of the population and to prevent the process from being trapped in a local optimal solution.

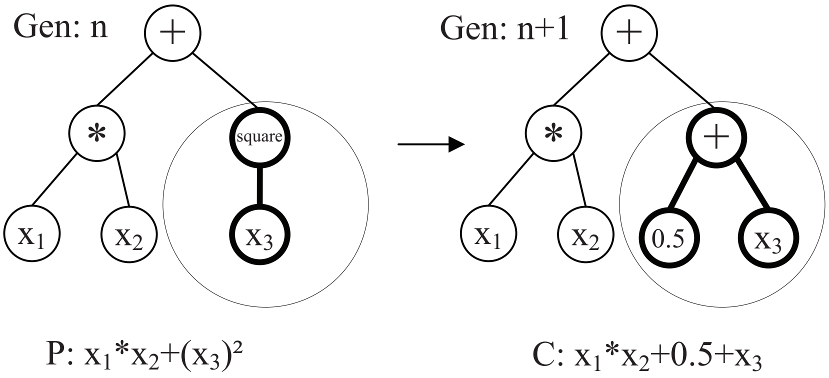


**Figure 9.8.** Graphical illustration of mutation operation.

3. Crossover: The sub-trees of two individuals are exchanged, producing two offspring individuals. Figure 9.9 illustrates the process of the crossover operation. The aim of the

crossover operation is to improve the exploitation of existing models and implicit the genetic memory.

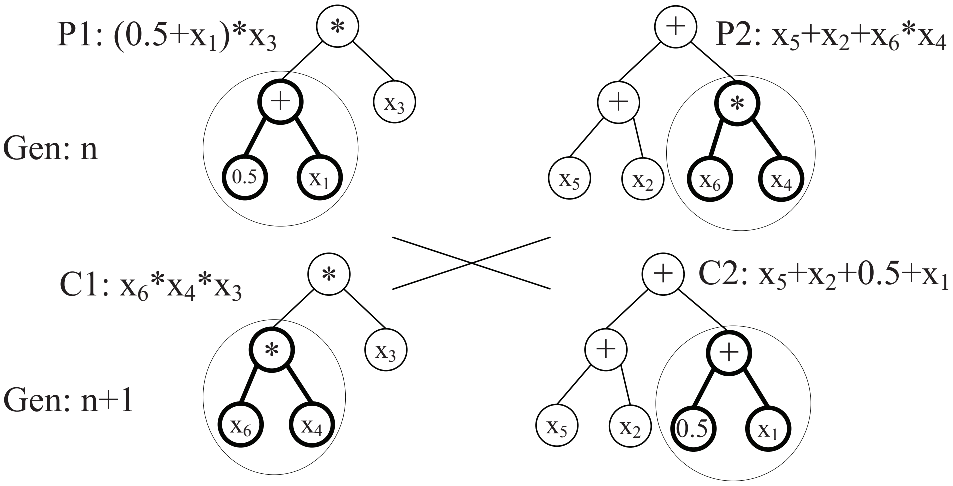


**Figure 9.9.** Graphical illustration of crossover operation.

## 9.5 Experiments

In this study, we collected images of 200 indoor scenes from two famous shopping malls in Wuhan City: *World City Light Valley Pedestrian Street* and *World City Square*. We show 10 representative test scenes in Appendix E. From each scene, we manually marked three to four common landmarks in shopping malls, such as shops, information boards, vending machines, and elevators, by referring mainly to (Li et al. 2017; Lyu et al. 2015). In total, 630 landmarks were marked from the 200 scenes. Table 9.3 lists the classes and counts of the landmarks, and the keywords used in the Baidu (Chinese version) and Google searches (English version). Details on the search keywords in Chinese and English for each landmark can be found in the uploaded files. Two hundred volunteers (77 females and 123 males, with the ages ranging from 18 to 30) were recruited to fill out the questionnaires, from which we extracted the salience of each landmark. We then divided the scenes into five test groups, with each having 40 scenes. With respect to each test group, the 40 scenes in the group were considered as the test set, and 75% and 25% of the remaining scenes outside of the given group were considered as the training set and validation set, respectively.

### 9.5.1 Experimental parameter selection

The GPTIPS GP toolbox in Matlab (Dominic et al. 2010; Searson 2009) was used to implement the proposed algorithm. Table 9.4 lists the values of the parameters used in the algorithm. In the experiments, the tree depth was set to 6, such that the created trees could cover the case wherein

**Table 9.3.** Different landmark classes with their respective counts and search keywords.

| Landmark class | Count | Search keyword |
|---|---|---|
| Shop | 299 | **Name of shop** |
| Sign | 89 | Sign |
| Elevator | 60 | Elevator |
| Sculpture | 45 | **Description of sculpture** |
| Staircase | 25 | Staircase |
| Vending machine | 23 | Vending machine |
| Billboard | 14 | Billboard |
| Green plant | 12 | Green plant |
| Trash can | 9 | Trash can |
| Fireplug | 9 | Fireplug |
| Help desk | 8 | Help desk |
| Lamp | 8 | Lamp |
| Pillar | 7 | Pillar |
| Chair | 5 | Chair |
| Fountain | 5 | Fountain |
| Corner | 2 | Corner |
| Wall | 2 | Wall |
| Door | 2 | Door |
| Postbox | 2 | Postbox |
| Telephone booth | 1 | Telephone booth |
| Roof | 1 | Roof |
| Distribution box | 1 | Distribution box |
| Table | 1 | Table |

the leaf nodes contained all the 14 attributes and 14 weights. The operators we used were $+, -, /, *, abs$, and $log$. Traditional linear models were therefore also considered as candidates in our proposed approach when only $+$ and $-$ were selected in a tree. However, in the evolution procedure, these linear models were gradually eliminated because they did not perform well on the training set. The reproduction rate, mutation rate, and crossover rate were set to 0.05, 0.1, and 0.85, respectively, a slight adjustment from the parameter setting used in (Yeh et al. 2007). That is, when a new population was being produced by the genetic operations based on the 10 best individuals, 5%, 10%, and 85% of the total 500 individuals were produced by the reproduction, mutation, and crossover operations, respectively.

**Table 9.4.** Parameter of GP Run Settings.

| Parameter name | Parameter value |
|---|---|
| Number of runs ($N_{run}$) | 5 |
| Population size ($S_{pop}$) | 500 |
| Number of generations ($N_{gen}$) | 30 |
| Number of top individuals ($N_{top}$) | 10 |
| Maximum depth of trees ($D$) | 6 |
| Reproduction rate ($R_{rate}$) | 0.05 |
| Mutation rate ($M_{rate}$) | 0.1 |
| Crossover rate ($C_{rate}$) | 0.85 |

### 9.5.2 Results and analysis

As discussed, most of the existing methods for estimating landmark salience were based on the model proposed by Raubal and Winter (2002), which is a linear model, shown here as Equation 9.7. That is, the landmark salience, denoted by $S$, equals the sum of the products of the visual attraction, which is denoted by $(S_v)$, the semantic attraction, which is denoted by $(S_s)$, and the structural attraction, which is denoted by $(S_u)$, with corresponding weights denoted by $W_v$, $W_s$, and $W_u$, respectively. The predefined weights allowed for an adaptation to different scenes (Lyu et al. 2015; Winter et al. 2004). For instance, Xing (2012) set $W_v = 1/2$, $W_s = 1/4$, and $W_u = 1/4$, whereas Lyu et al. (2015) and Li et al. (2017) set the weights at $W_v : W_s : W_u = 1 : 1 : 1$. Götze and Boye (2016), on the other hand, used a ranking SVM method to learn the weights. As we have mentioned previously, this study neglected the structural attributes. Thus, $S_u$ equals 0.

$$S = W_v \cdot S_v + W_s \cdot S_s + W_u \cdot S_u \tag{9.7}$$

This work compares existing approaches (Götze and Boye 2016; Lyu et al. 2015; Xing 2012) for measuring the salience of landmarks with our proposed GP-based approach. We first calculated the salience of the landmarks on the test set and sorted the landmarks in each scene according to their salience, in descending order (e.g., $A > C > B > D$). We then obtained the percentage of the scenes where the most salient landmark was correctly predicted by the algorithm, which was regarded as the accuracy of identifying the most salient landmark and referred to as the

Top1 accuracy. Meanwhile, we calculated the accuracy of correctly sorting the landmarks in each scene according to their salience, referred to as the sort accuracy, which is also important, considering the dialog setting that the pedestrian may not be able to observe or identify the first choice for some reason. In this case, the second choice can then be recommended to the user.

The prediction results for the most attractive landmarks and the sorting of the five test groups are shown in Figure 9.10. The average accuracies for the five test groups are shown in Table 9.5. We can see that, on average, in 76% and 41% of the cases, the most salient landmark and the sorting, respectively, can be correctly predicted by our proposed approach. The average rank of the most salient landmark is 1.33, proving that the desired landmarks have a high-ranking result. In all the five test groups, our proposed approach achieved better results than those of compared algorithms. Furthermore, we can conclude that the machine-learning based models (e.g., GP and SVM-based) outperform the manually defined model, mainly because the researchers manually defined the model for a specific environment, such as on the street, in campus, or in a shopping mall. Therefore, the models are incapable of adapting to a changing environment. Although SVM-based solutions use a simple linear model, the weights can be learned automatically based on training data. From this point of view, compared to manually defined models, SVM has a stronger capability of adapting to a changing environment. However, a linear model restricts the ability of representing complex and accurate quantitative relationships between the salience and the attributes. Thus, the GP-based non-linear model achieves a better result than that of the SVM-based linear model. The detailed sorting results obtained from these models can be found in Appendix F, where *A*, *B*, *C*, and *D* represent the indexes of the landmarks. We can conclude that, for application purposes, our proposed solution outperforms the traditional approaches, because it can consistently achieve higher prediction accuracy levels. Furthermore, our method does not require intervention from researchers (e.g., manually setting weight values) and can be easily extended to other indoor and outdoor environments without re-proposing a new model, which is otherwise required in traditional solutions.

The learned models for the five folds are represented by Equation 9.8 to Equation 9.12. We must consider that the learned non-linear model is data-dependent and can make accurate predictions given abundant training data but, unlike linear models, cannot explain the exact influence of distinct attributes on the salience of the landmark. Take the learned models depicted in Equation 9.11 as an example. We cannot explain exactly how the **Facade area**, **Facade area of attached subject**, and **Text** impact the salience of the landmarks. However, the final purpose of the solution is to correctly select the most attractive landmark in a scene. From this point of view, an interpretable model is not necessary. Despite this limitation, we can probably determine which attributes are more important than the others based on their occurrence frequency in the five models. For the 14 attributes listed in Table 9.1, their occurrence frequencies are 2, 5, 0, 2, 4, 1, 0, 4, 1, 3, 5, 0, 5, and 0, respectively. That is, **Text**, **Facade area of attached subject**, **Colour**,

**Shop**, and **Count of Baidu search** are the most important attributes, whereas **Shape deviation**, **Information class**, **Foreign language**, and **Count of Google search** are the least important attributes. We reckon that the models that learned from the training data in shopping malls cannot be directly used in other environments (e.g., office buildings and airports), considering the variance of landmarks in different environments. However, the proposed solution is versatile enough to be applicable for learning the salience model of other environments, given enough training data.

$$\begin{aligned} y_1 =& 0.04594 \ast x_6 - 0.2339 \ast x_{13} + 0.05482 \ast \log(x_2) - 0.0791 \ast \\ & \log(x_{11}) + 0.1894 \ast x_9 \ast \log(x_{11}) - 0.04594 \ast (x_8){}^\wedge(1/2) + 0.507 \end{aligned} \tag{9.8}$$

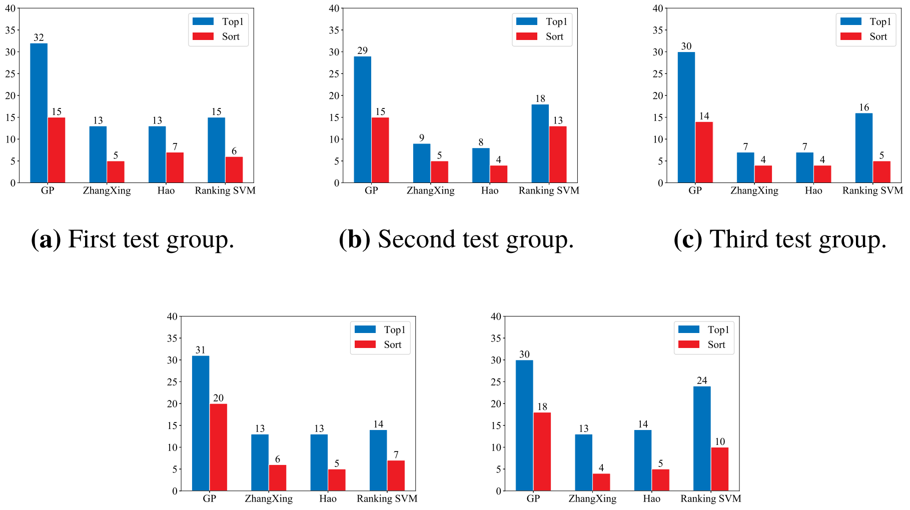
$$\begin{aligned} y_2 =& 0.05981 \ast \log(x_2) - 0.05981 \ast x_{11} - 0.02557 \ast x_8 + 0.02263 \ast \\ & \log(\log(x_{10} + x_{11}){}^\wedge 2) - 0.04652 \ast \log((x_{10} + x_{11}){}^\wedge 2) + \\ & 0.02557 \ast \log((x_5){}^\wedge 2) - 0.2471 \ast (x_{13}){}^\wedge(1/2) + 0.6235 \end{aligned} \tag{9.9}$$

$$\begin{aligned} y_3 =& 0.05228 \ast \log(x_2 \ast x_5) - 0.1386 \ast x_8 - 0.06929 \ast x_1 - 0.1001 \ast \log \\ & (x_{10} - x_{11}) - 0.2655 \ast (x_{13}){}^\wedge(1/2) - 0.01265 \ast (\log(x_2/x_5)){}^\wedge(1/2) - \\ & 0.07882 \ast (x_8 + x_{10} - x_{11}){}^\wedge 2 + 0.6406 \end{aligned} \tag{9.10}$$

$$\begin{aligned} y_4 =& 0.102 \ast x_5 - 0.102 \ast x_8 - 0.05357 \ast (\log(x_2){}^\wedge 2){}^\wedge(1/2) - 0.4924 \ast \\ & (x_4){}^\wedge 2 \ast x_{13} + 2.982\text{E}15 \ast \log(x_{11})/(7.206\text{E}16 \ast x_2 - 3.773\text{E}16) + \\ & 0.05308 \ast (x_4){}^\wedge 2 + 0.1346 \ast x_1 \ast \log(x_2)/x_{11} + 0.4474 \end{aligned} \tag{9.11}$$

$$\begin{aligned} y_5 =& 0.02409 \ast x_4 + 0.04912 \ast \log(x_2) - 0.07607 \ast \log(x_{11}) - 0.211 \ast \\ & \log(\log(x_{10} \ast \log(x_2))) + 0.1225 \ast (x_5){}^\wedge(1/2) - 0.1921 \ast (x_{13}){}^\wedge(1/2) + 0.4167 \end{aligned} \tag{9.12}$$

The precondition of applying a linear model to represent the quantitative relationship between the attributes and salience is that these attributes are independent, which, however, is not true for our selected attributes. For instance, intuitively, the *Facade area of attached subject* correlates with the *Shop class* because normally, only shops have attached subjects. It is the same case for the *Text* and *Text and Foreign language* because the former is the abstraction of the latter. The third example is *Baidu search* and *Connection*. The higher the Baidu search count of a connection entity (e.g., staircase), the more common they are in shopping malls, which means they are not suitable for wayfinding. Inversely, shop entities with a large Baidu search count are easily recognized by humans, and there is normally just one such shop (e.g., Nike) in a shopping mall. Therefore, these entities should be selected as the representative landmarks. Because there are plenty of complex relationships between the selected attributes, a linear model is too simple for representing the accurate quantitative relationship between the attributes and the salience. The model that learned from training data, however, has no independence assumption and does not care if the attributes are dependent or independent, because GP can automatically select a model that best fits the training data. For instance, in Figure E.3 of Appendix E, the shop $COSMOLADY$ is the most salient landmark, and the salience calculated by the learned model is ranked the highest. However, the salience calculated by the linear model is ranked third, and that of the staircase is ranked the highest.



(**a**) First test group.

(**b**) Second test group.

(**c**) Third test group.

(**d**) Fourth test group.

(**e**) Fifth test group.

**Figure 9.10.** Prediction results of different approaches for five test groups.

**Table 9.5.** Top1 and sorting accuracies achieved by proposed approach and traditional approaches.

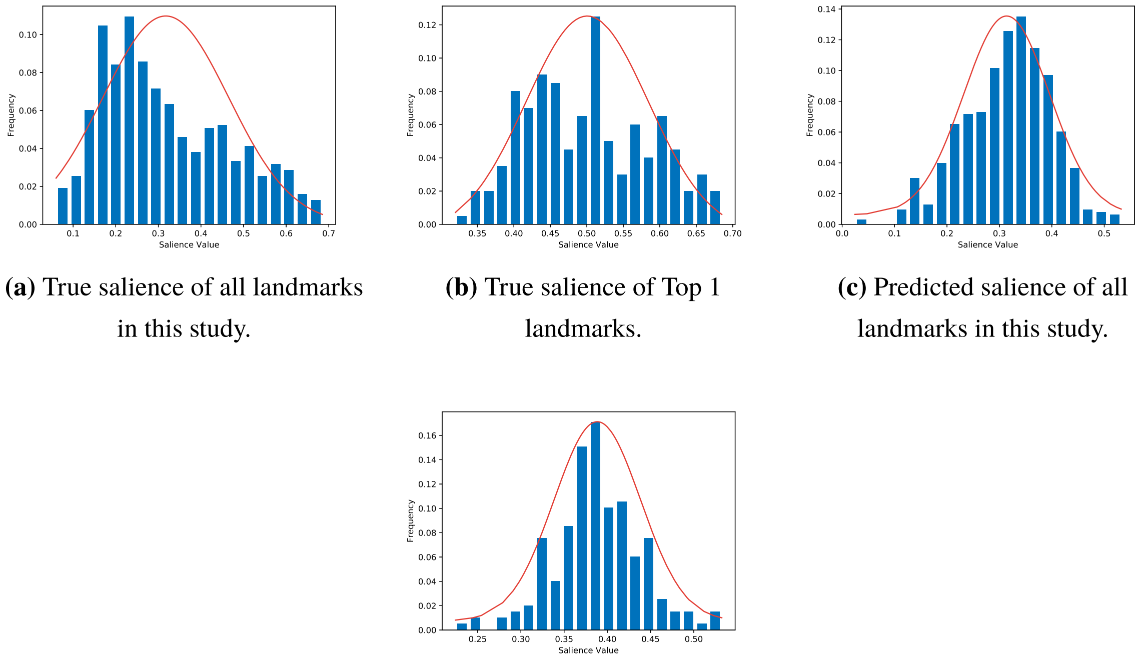
| Comparative groups | Top1 | Sort |
|---|---|---|
| GP | 0.76 | 0.41 |
| Götze and Boye (2016) | 0.435 | 0.205 |
| Xing (2012) | 0.275 | 0.12 |
| Lyu et al. (2015) | 0.275 | 0.125 |

### 9.5.3 Distribution of landmark salience

We further analysed the distribution of the true and predicted salience values of all the landmarks and of the top 1 landmark in the data set. The result is shown in Figure 9.11. We can see that both the true and predicted salience values of landmarks approximately follow a normal distribution, suggesting that (1) the collected 40,000 answers from volunteers about the salience of the landmarks are unbiased, and (2) the learned non-linear model is reasonable because the normal distribution is the most common type of distribution that describes the characteristics of data in the real world. Furthermore, we divided the prediction results for the 200 scenes presented in the previous subsection into five groups according to the true salience value of the top 1 landmark in each scene. Table 9.6 shows the range of the salience value of the top 1 landmark, the number of scenes, and the top 1 and sorting accuracies for each group. We can see that when the salience of the top 1 landmark comes within the range of 0.54–0.61, the highest accuracy is achieved, subsequently followed by the ranges of 0.61–0.69, 0.47–0.54, 0.39–0.47, and 0.32–0.39. Two factors are responsible for this result: the number of scenes, and the true salience of the top 1 landmark in each range. First, the GP algorithm gives more weight to a larger number of scenes, to achieve a higher accuracy. Second, the higher the salience value of the highest ranked landmark in a scene, the more likely it can be distinguished from other landmarks in the same scene.

**Table 9.6.** Prediction results for 5 salience ranges.

| | **[0.32, 0.39)** | **[0.39,0.47)** | **[0.47,0.54)** | **[0.54,0.61)** | **[0.61,0.69]** |
|---|---|---|---|---|---|
| Number of scenes | 16 | 65 | 57 | 39 | 23 |
| Number of correctly predicted Top 1 landmark | 9 | 48 | 44 | 33 | 18 |
| Number of correctly predicted sort of landmarks | 4 | 24 | 26 | 17 | 11 |
| Top1 accuracy | 0.5625 | 0.73846 | 0.77193 | 0.84615 | 0.78261 |
| Sort accuracy | 0.25 | 0.36923 | 0.45614 | 0.4359 | 0.47826 |



**(a)** True salience of all landmarks in this study.

**(b)** True salience of Top 1 landmarks.

**(c)** Predicted salience of all landmarks in this study.

**(d)** Predicted salience of Top 1 landmarks.

**Figure 9.11.** Distributions of true and predicted salience values of landmarks.

## 9.6 Discussions

The drawbacks of this work are twofold. First, the scenes or decision points we chose are scattered distributed in the environment such that they do not follow certain navigation routes. Meanwhile, the scene is observed from only one perspective or direction. However, the directions users walk to the decision point would definitely affect their perception of the salient landmark. In this research, we focused on learning a salience model that can measure the salience of landmarks from a certain perspective. In the future, we will consider how to integrate the proposed solution in the navigation application. The second limitation is that the proposed solution relies on the visual characteristics of landmarks, which sometimes are not easily obtained from maps, such as the OpenStreetMap (OSM), which can freely be used by anyone. Meanwhile, on OSM, many shopping malls have been mapped with rich points of interest (POIs) or landmarks (such as shops, fire hydrants, and vending machines), especially in Europe and in the United States. Figure 9.12 shows the tagged POIs of a shopping mall on OSM. It would be applicable and meaningful to develop a landmark-based indoor wayfinding system based on existing OSM data. Therefore, a novel approach to selecting the salient landmark in an indoor environment, based only on the semantic and spatial information that can be extracted from OSM, can be proposed. This possible new method should be investigated in future works.

Route learning based on virtual environments (VE) is appropriate for those people, especially elders, who struggle to remember routes when walking in unfamiliar environments. It is a challenge to decide which segments/landmarks in a scene along a route should be abstracted or removed to make the route easier to remember without too much scene information. Traditional solutions manually abstract or remove certain segments of the scene along a route and then test if the route can be easily remembered by volunteers. No accurate models have so far been developed for tackling this issue. In this regard, our work provides inspiration. First, we assume that the high-salience landmarks, such as the street and the building that occupies a big area in the scene, should be kept. In this way, the main task becomes generating a mathematical model that can measure the salience of landmarks in a scene given the visual and semantic features. The model can be learned with some machine learning approaches. To verify the assumption, we can then produce many mixed virtual environments that abstract different information for the same route and verify if volunteers remember the mixed environments with non-abstracted landmarks that have a higher salience.

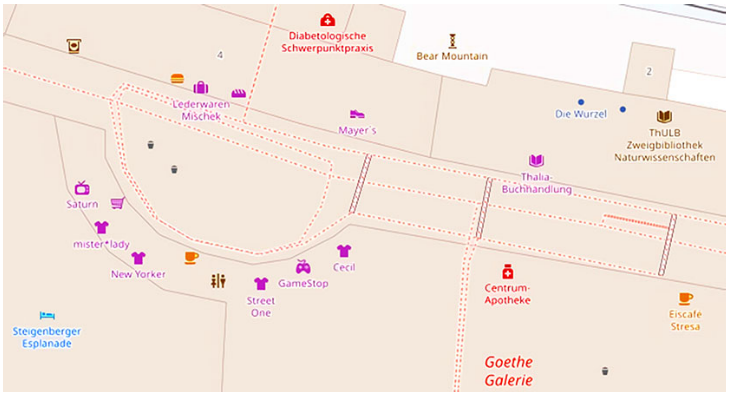


**Figure 9.12.** POIs of a shopping mall on OSM.

## 9.7 Conclusions

Selecting the most salient landmarks is a complex process in which various factors interact with and restrict each other. These factors include the characteristics of landmarks, cultural backgrounds of pedestrians, and surrounding environments of the landmarks. This work proposed using GP to learn a non-linear model for measuring the salience of landmarks in shopping malls. The landmark salience or the preferences of pedestrians on different landmarks were obtained via questionnaires. The approach was evaluated based on 200 scenes. The results show that in 76% of the cases, our approach can correctly predict the most salient landmark, proving that, compared to the traditional approaches based on linear models, our approach learns a non-linear model that can better represent the quantitative relationship between the salience of the landmarks and their attributes. Despite the fact that the learned model cannot clearly explain how each attribute contributes to the salience of the landmarks, the proposed solution outperforms the compared algorithms from the perspective of application, because it can provide a higher prediction accuracy and does not require any intervention from researchers (e.g., in manually setting weight values). Furthermore, the model can be easily extended to other indoor and outdoor environments without re-proposing a new model, which is otherwise required for the traditional solutions. We have made openly available the Matlab code, the images of the 200 scenes, the tagged landmarks and their attributes, and the results of the questionnaires on the online repository [1]. In future works, we intend to construct a more general model that ignores the visual attributes of landmarks, which requires a highly accurate image-based indoor model. This idea is inspired by the fact that several landmarks in shopping malls have already been tagged on OSM.

[1] Data and code are available at https://github.com/DinleyGitHub/Indoor-Mall-Landmarks-Saliency

# Reference

Butz, A., Baus, J., Krüger, A., and Lohse, M. (2001). A hybrid indoor navigation system. In *Proceedings of the 6th International Conference on Intelligent User Interfaces*, IUI '01, pages 25–32, New York, NY, USA. ACM.

Caduff, D. (2007). *Assessing landmark salience for human navigation*. PhD thesis, University of Zurich, Zürich.

Caduff, D. and Timpf, S. (2008). On the assessment of landmark salience for human navigation. *Cognitive Processing*, 9:249–267.

Dominic, P., Leahy, D., and Willis, M. (2010). Gptips: An open source genetic programming toolbox for multigene symbolic regression. *Lecture Notes in Engineering and Computer Science*, 2180.

Duckham, M., Winter, S., and Robinson, M. (2010). Including landmarks in routing instructions. *Journal of Location Based Services*, 4(1):28–52.

Götze, J. and Boye, J. (2016). Learning landmark salience models from users' route instructions. *Journal of Location Based Services*, 10(1):47–63.

Hund, A. M. and Padgitt, A. J. (2010). Direction giving and following in the service of wayfinding in a complex indoor environment. *Journal of Environmental Psychology*, 30(4):553 – 564.

Kattenbeck, M. (2015). Empirically measuring object saliency for pedestrian navigation. In *23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2015)*, Seattle, WA, USA.

Kattenbeck, M. (2017). How Subdimensions of Salience Influence Each Other. Comparing Models Based on Empirical Data. In Clementini, E., Donnelly, M., Yuan, M., Kray, C., Fogliaroni, P., and Ballatore, A., editors, *13th International Conference on Spatial Information Theory (COSIT 2017)*, volume 86 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:13, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Kattenbeck, M., Nuhn, E., and Timpf, S. (2018). Is salience robustl a heterogeneity analysis of survey ratings. In Winter, S., Griffin, A., and Sester, M., editors, *10th International Conference on Geographic Information Science (GIScience 2018)*, volume 114 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:16, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Kim, J., Han, D., Tai, Y. W., and Kim, J. (2016). Salient region detection via high-dimensional color transform and local spatial support. *IEEE Transactions on Image Processing*, 25(1):9–23.

Klippel, A. and Winter, S. (2005). Structural salience of landmarks for route directions. In Cohn, A. G. and Mark, D. M., editors, *Spatial Information Theory*, pages 347–362, Berlin, Heidelberg. Springer Berlin Heidelberg.

Koza, J. R. (1991). Evolving a computer program to generate random numbers using the genetic programming paradigm.

Li, L., Mao, K., Li, G., and Wen, Y. (2017). A landmark-based cognition strength grid model for indoor guidance. *Survey Review*, 50(361):1–11.

Lyu, H., Yu, Z., and Meng, L. (2015). A computational method for indoor landmark extraction. In *Progress in Location-Based Services 2014*, pages 45–59. Springer.

May, A. J., Ross, T., Bayer, S. H., and Tarkiainen, M. J. (2003). Pedestrian navigation aids: information requirements and design implications. *Personal and Ubiquitous Computing*, 7(6):331–338.

Michon, P.-E. and Denis, M. (2001). When and why are visual landmarks used in giving directions. In Montello, D. R., editor, *Spatial Information Theory*, pages 292–305, Berlin, Heidelberg. Springer Berlin Heidelberg.

Millonig, A. and Schechtner, K. (2007). Developing landmark-based pedestrian-navigation systems. *Intelligent*

*Transportation Systems, IEEE Transactions on*, 8:43 – 49.

Nothegger, C., Winter, S., and Raubal, M. (2004). Selection of salient features for route directions. *Spatial cognition and computation*, 4(2):113–136.

Ohm, C., Müller, M., and Ludwig, B. (2015). Displaying landmarks and the users' surroundings in indoor pedestrian navigation systems. *Journal of Ambient Intelligence and Smart Environments*, 7:635–657.

Raubal, M. and Winter, S. (2002). Enriching wayfinding instructions with local landmarks. In Egenhofer, M. J. and Mark, D. M., editors, *Geographic Information Science*, pages 243–259, Berlin, Heidelberg. Springer Berlin Heidelberg.

Richter, K.-F. and Winter, S. (2014). *Landmarks: GIScience for Intelligent Services*. Springer Science & Business.

Ross, T., May, A., and Thompson, S. (2004). The use of landmarks in pedestrian navigation instructions and the effects of context. In Brewster, S. and Dunlop, M., editors, *Mobile Human-Computer Interaction - MobileHCI 2004*, pages 300–304, Berlin, Heidelberg. Springer Berlin Heidelberg.

Searson, D. (2009). Gptips: Genetic programming & symbolic regression for matlab user guide.

Sorrows, M. E. and Hirtle, S. C. (1999). The nature of landmarks for real and electronic spaces. In Freksa, C. and Mark, D. M., editors, *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, pages 37–50, Berlin, Heidelberg. Springer Berlin Heidelberg.

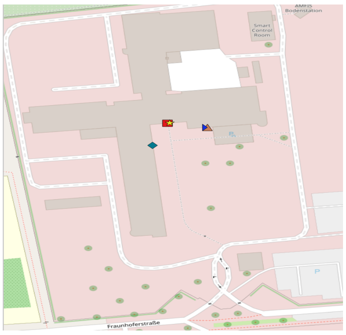
Winter, S., Martin, R., and Nothegger, C. (2004). Focalizing measures of salience for route directions. *Map-based Mobile Services - Theories, Methods and Implementations*, pages 127–142.

Xing, Z. (2012). Research on landmark-based pedestrian navigation methods in complex city environment. *Ph.D. Dissertation, Wuhan University, Wuhan, China*.

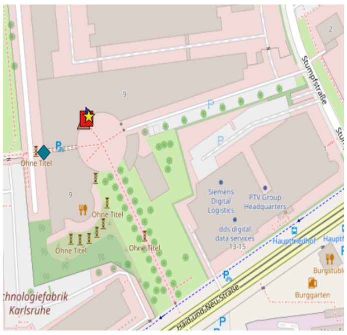
Yeh, J.-Y., Lin, J.-Y., Ke, H.-R., and Yang, W.-P. (2007). Learning to rank for information retrieval using genetic programming. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007)*.

Yu, H., Kim, J., Kim, Y., Hwang, S., and Lee, Y. H. (2012). An efficient method for learning nonlinear ranking svm functions. *Information Sciences*, 209:37–48.

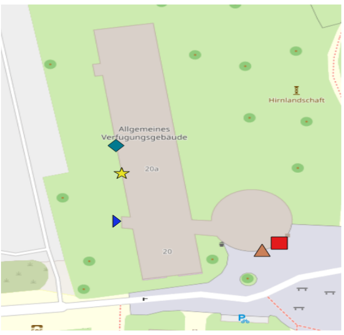
# A. Partial entrance tagging results



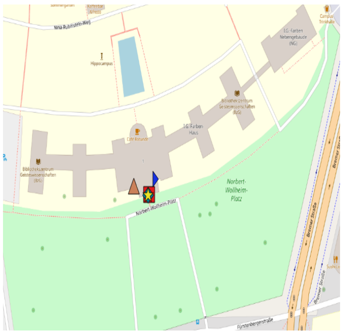
**(a)**


Ohne Titel
Siemens Digital Logistics
PTV Group Headquarters
dds digital data services
Burggarten
Technologiefabrik Karlsruhe
Haid-und-Neu-Straße
Stumpfstraße

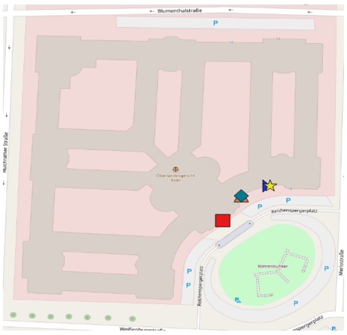
**(b)**

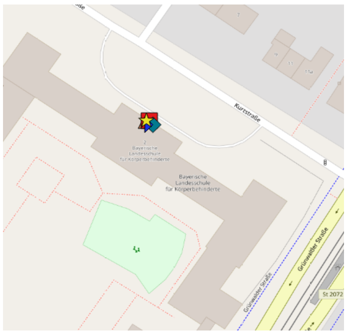


**(c)**

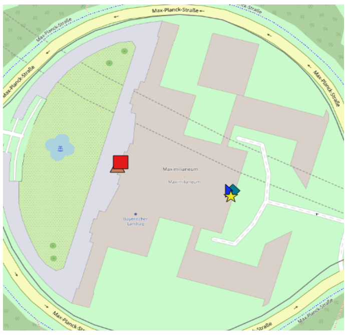


**(d)**

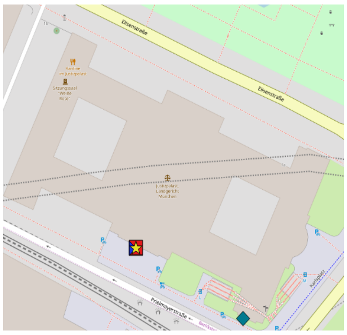

**(e)**



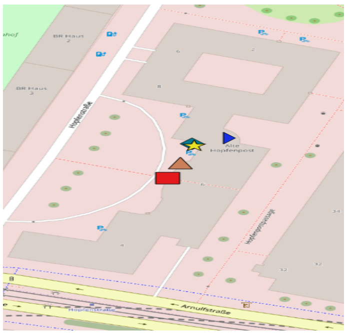
**(f)**



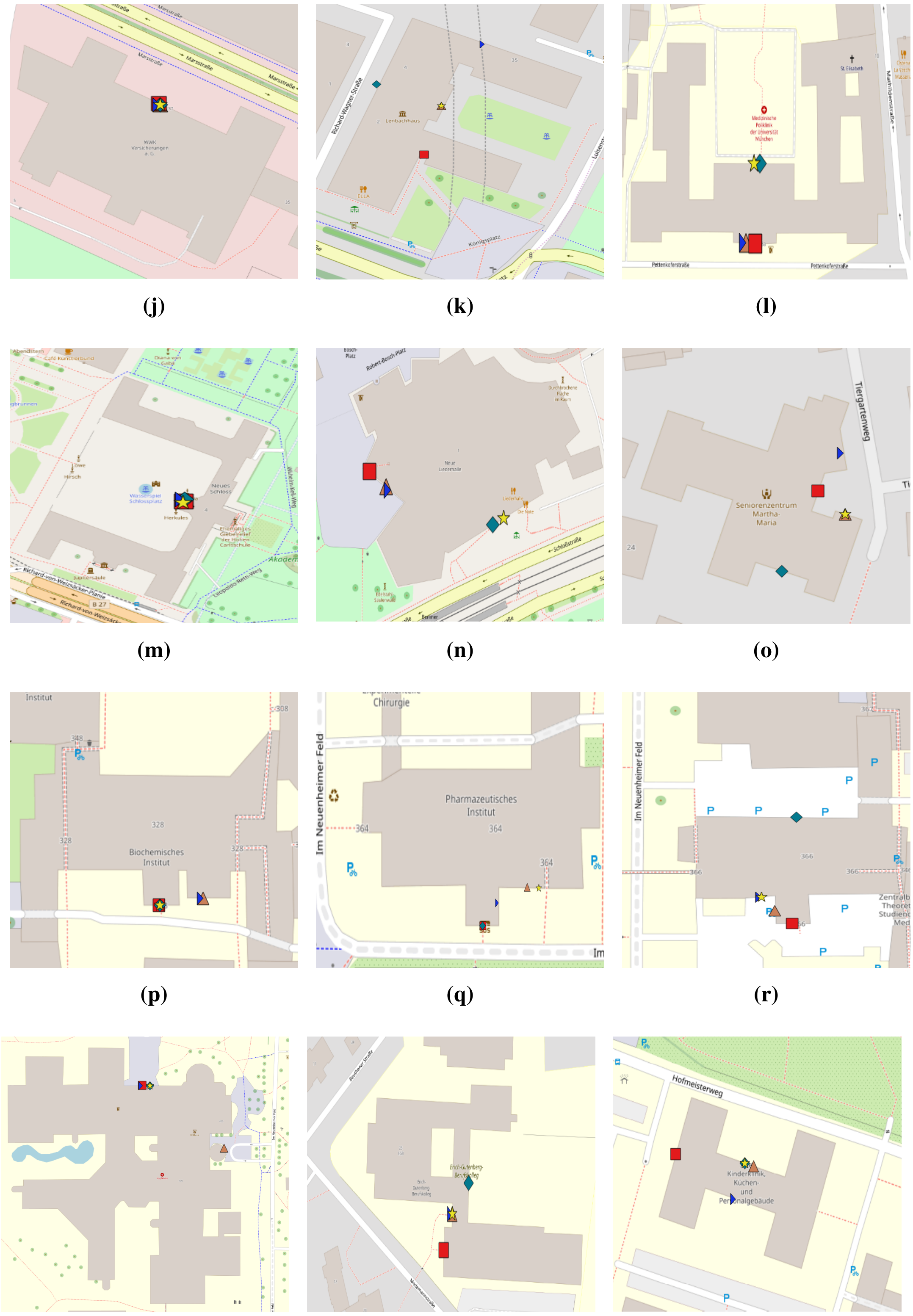
**(g)**



**(h)**



**(i)**

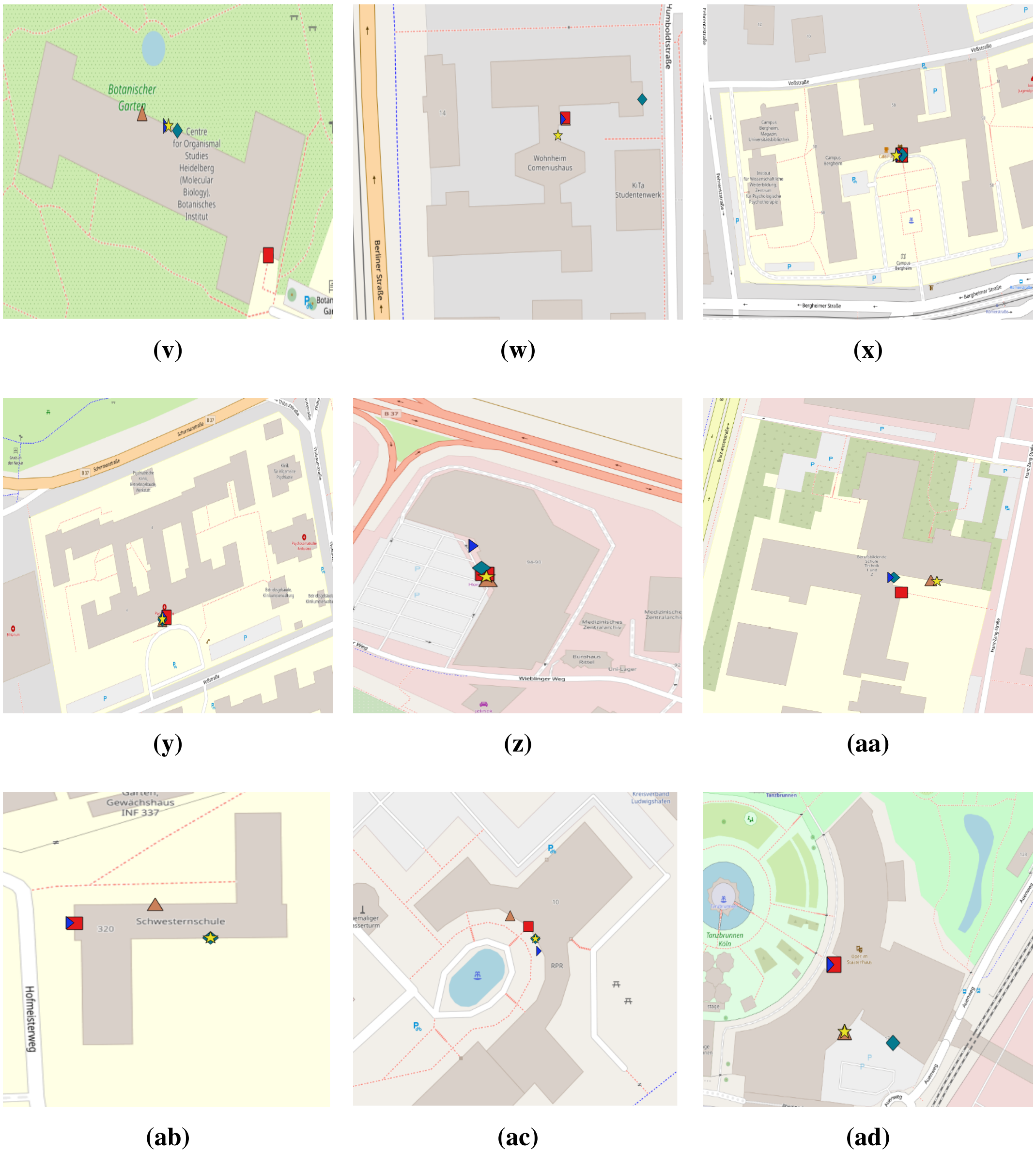


(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

(s) (t) (u)



(v) (w) (x)

(y) (z) (aa)

(ab) (ac) (ad)

**Figure A.-1.** Tagging result of partial test buildings

# B. Grammar rules

$$\mathit{Copy}\ \ c \mid \mathit{Storage}\ \ s \mid \mathit{Kitchen}\ \ k \mid \mathit{Lecture}\ \ lec \mid \mathit{Lounge}\ \ l \mid \mathit{Computer}\ \ c \mid \mathit{Support}\ \ sup \rightarrow \mathit{room}\ \ r \tag{B.1}$$

$$\mathit{Toilet}\ \ t \rightarrow \mathit{set}(\mathit{room}, k^r)\ \ r, \mathit{set}(\mathit{door}, k^d)\ \ d \left\langle \begin{array}{l} 1 \leq k^r \leq 3;\ \ k^d \geq 0; \\ \mathit{conByIntDoor}(r_i, r_{i+1}, d)_{0<i<k^r}; \\ \sum_{i=1}^{k^r} \mathit{withExtDoor}(r_i) == 1 \end{array} \right\rangle \tag{B.2}$$

$$\begin{aligned} \mathit{Ancillary}\ a \rightarrow \mathit{Toilet}\ t \mid \mathit{Copy}\ c \mid \mathit{Storage}\ s \mid \mathit{Kitchen}\ k \mid& \\ \mathit{Lecture}\ lec \mid \mathit{Lounge}\ l \mid \mathit{Computer}\ c \mid \mathit{Library}\ lib& \end{aligned} \tag{B.3}$$

$$\mathit{Library}\ \ l \rightarrow \mathit{set}(\mathit{room}, k^r)\ \ r, \mathit{set}(\mathit{door}, k^d)\ \ d \left\langle \begin{array}{l} k^r \geq 1;\ \ k^d \geq 0; \\ \mathit{conByIntDoor}(r_i, r_{i+1}, d)_{0<i<k^r} \end{array} \right\rangle \tag{B.4}$$

$$\begin{aligned} &\mathit{Zone}^{aca}\ z \rightarrow \mathit{set}(\mathit{lecture}, k^l)\ l, \mathit{set}(\mathit{door}, k^d)\ d \\ &\qquad \left\langle \begin{array}{l} k^l \geq 1;\ k^d \geq 0; \\ \sum_{i=1}^{k^r} \mathit{withExtDoor}(r_i) == 1; \\ \mathit{edgeAdj}(l_i, l_{i+1}, d)_{0<i<k^l} \mid \\ \mathit{conByIntDoor}(l_i, l_{i+1}, d)_{0<i<k^l}; \end{array} \right\rangle \{z.\mathit{type} = \mathit{academic}\} \end{aligned} \tag{B.5}$$

$$\mathit{Zone}^{aca}\ z \rightarrow \mathit{Library}\ lib\ \{z.\mathit{type} = \mathit{academic}\} \tag{B.6}$$

$$\mathit{Lab}\ l \rightarrow \mathit{room}\ r^l(r^w, \mathit{door}\ d) \mid \phi\langle \mathit{inclusionAdj}(r^l, r^w, d); \mathit{onExtWall}(r^l)\rangle \tag{B.7}$$

$$\mathit{LGroup}\ g \rightarrow \mathit{set}(\mathit{Lab}, k^l)\ l, \mathit{set}(\mathit{Supoort}, k^s)\ s, \mathit{set}(\mathit{door}, k^d)\ d \left\langle \begin{array}{l} k^l \geq 1;\ k^s \geq 0; \\ \mathit{conByIntDoor}(l, s, d) \end{array} \right\rangle \tag{B.8}$$

$$\mathit{Zone}^{lab}\ z \rightarrow \mathit{set}(\mathit{LGroup}, k^l)\ l \left\langle\ k^l \geq 1; \mathit{edgeAdj}(l_i, l_{i+1})_{0<i<k^l}\ \right\rangle \{z.\mathit{type} = \mathit{lab}\} \tag{B.9}$$

$$Office\ o \rightarrow room\ r^p(r^s, door\ d) \mid \phi\langle inclusionAdj(r^p, r^s, d); onExtWall(r^p); withExtDoor(r^p)\rangle \tag{B.10}$$

$$Zone^{office}\ z \rightarrow set(Office, k^o)\ o, set(door, k^d)\ d \left\langle \begin{array}{l} edgeAdj(o_i, o_{i+1}, d)_{0<i<k^o} \mid \\ conByIntDoor(o_i, o_{i+1}, d)_{0<i<k^o}; \end{array} \right\rangle \{z.type = office\} \tag{B.11}$$

$$Center\ c \rightarrow set(Ancillary, k^a)\ l, set(Supoort, k^s)\ s \left\langle \begin{array}{l} (0 \leq k^a \leq 3;\ k^s \geq 1) \\ \mid (k^s == 0; k^a \geq 1); \\ edgeAdj(s_i, s_{i+1})_{0<i<k^s} \\ \mid conByIntDoor(s_i, s_{i+1}, d)_{0<i<k^s}; \\ formFullArea(a, s); inCenter(c); \end{array} \right\rangle \{z.type = support \mid academic\} \tag{B.12}$$

$$CZone\ \ c \rightarrow set(Ancillary, k)\ \ a, Zone\ \ z \left\langle\ 0 \leq k \leq 3; formFullArea(a, z)\ \right\rangle \langle c.type = z.type\rangle \tag{B.13}$$

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

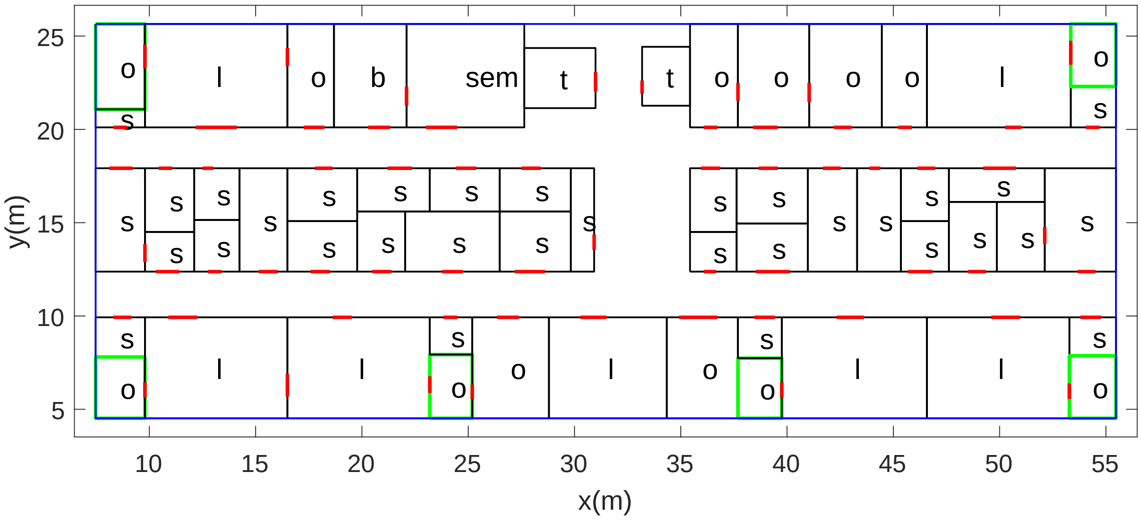
$$\left\langle \begin{array}{l} k^o{\geq}0; 0 \leq k^o \leq 2; 0 \leq k^a \leq 2; k^a + k^o >= 1; \\ (z^o{}_i)_{1 \leq i \leq k^o}.type == office; (c_i)_{1 \leq i \leq k^c}.type == ancillary; \\ (z_i^a)_{1 \leq i \leq k^a}.type == academic; \end{array} \right\rangle \{b.type = office \mid academic\} \tag{B.14}$$

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

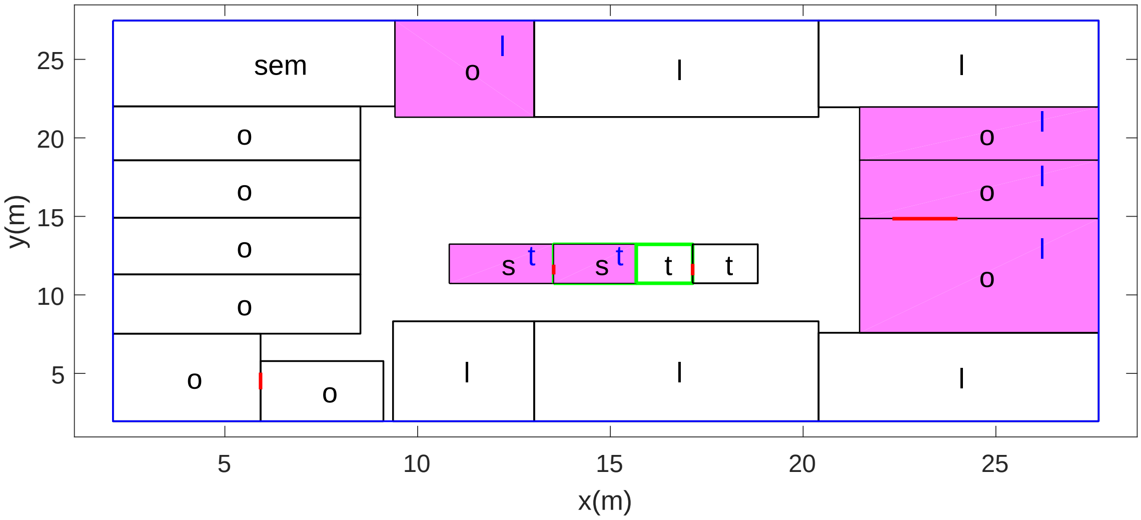
$$\left\langle \begin{array}{l} k^l{\geq}1; k^o{\geq}1; k^a{\geq}0; k^s{\geq}1; k^{anc}{\geq}0; (z^l{}_i)_{1 \leq i \leq k^l}.type == lab; \\ (z^o{}_i)_{1 \leq i \leq k^o}.type == office; (z_i^a)_{1 \leq i \leq k^a}.type == academic; \\ (c^s{}_i)_{1 \leq i \leq k^c}.type == support; (c^a{}_i)_{1 \leq i \leq k^{anc}}.type == ancillary; \\ isDoubleLoaded(b) \mid (isTripleLoaded(b) \end{array} \right\rangle \{b.type = lab\} \tag{B.15}$$

$$Building\ f \rightarrow set(BUnit, k^b)\ b \left\langle\ k^b \geq 1; edgeAdj(b_i, b_{i+1})_{0<i<k^b}\ \right\rangle \tag{B.16}$$

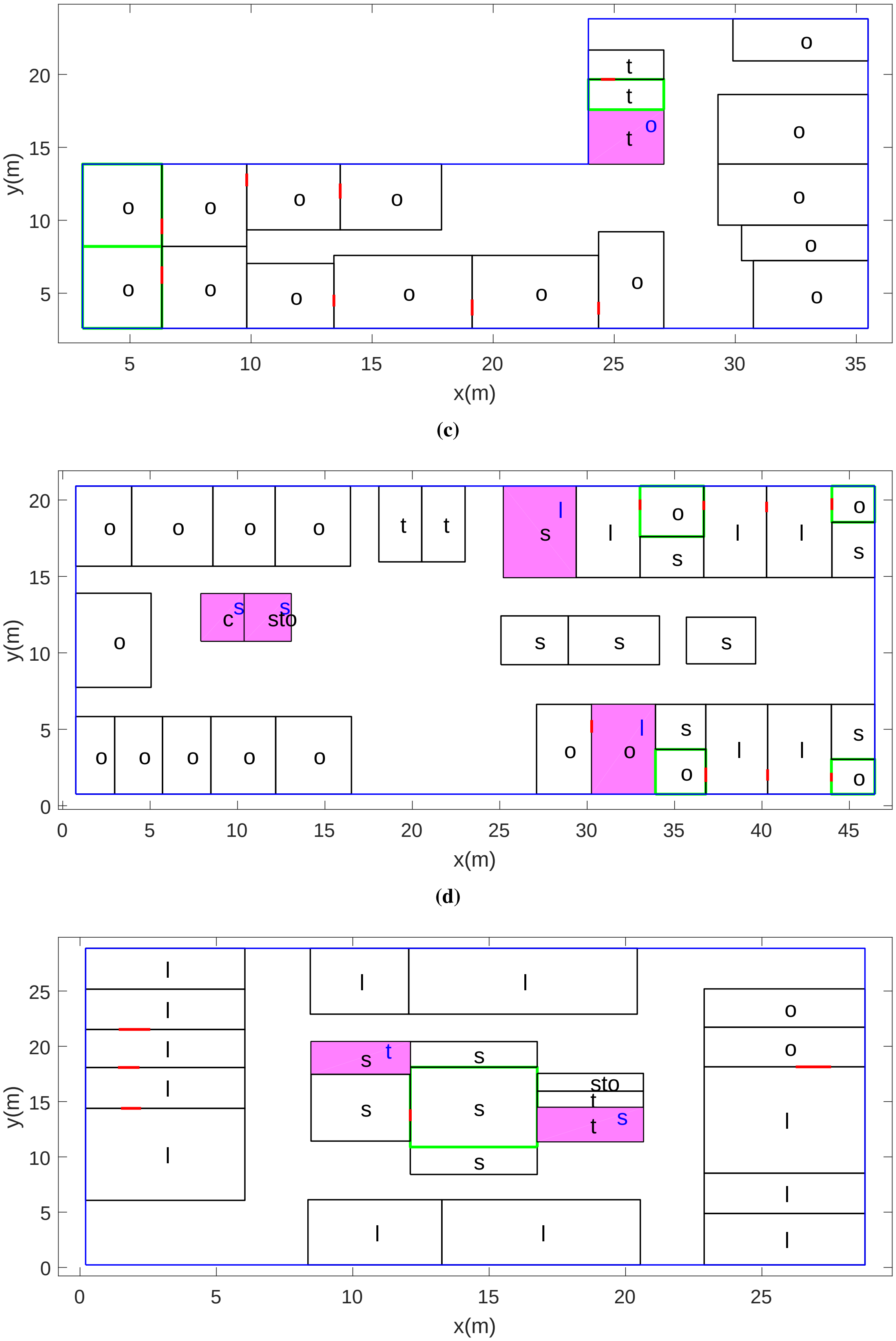
# C. Partial room tagging result achieved by RF



**(a)**



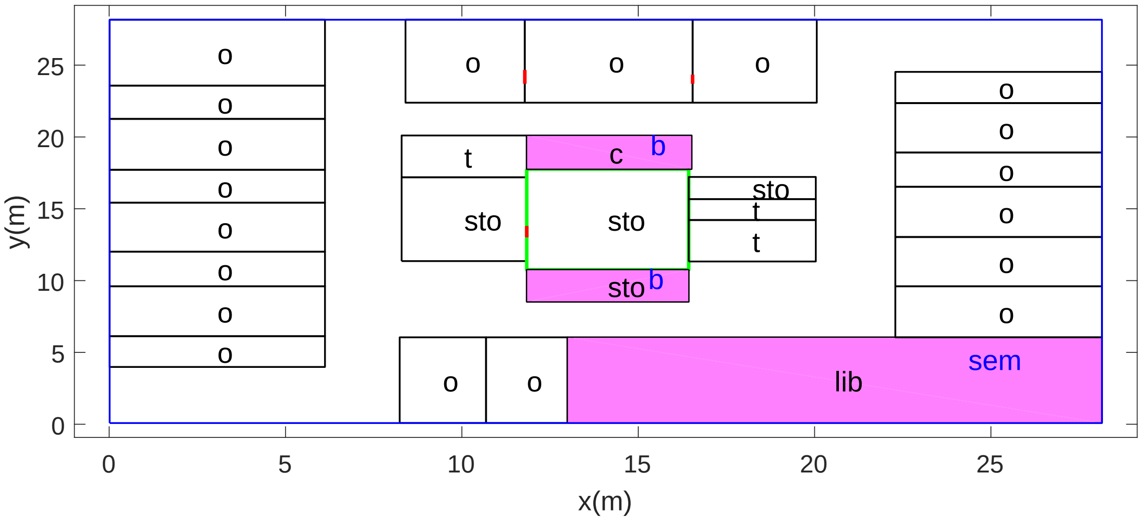
**(b)**



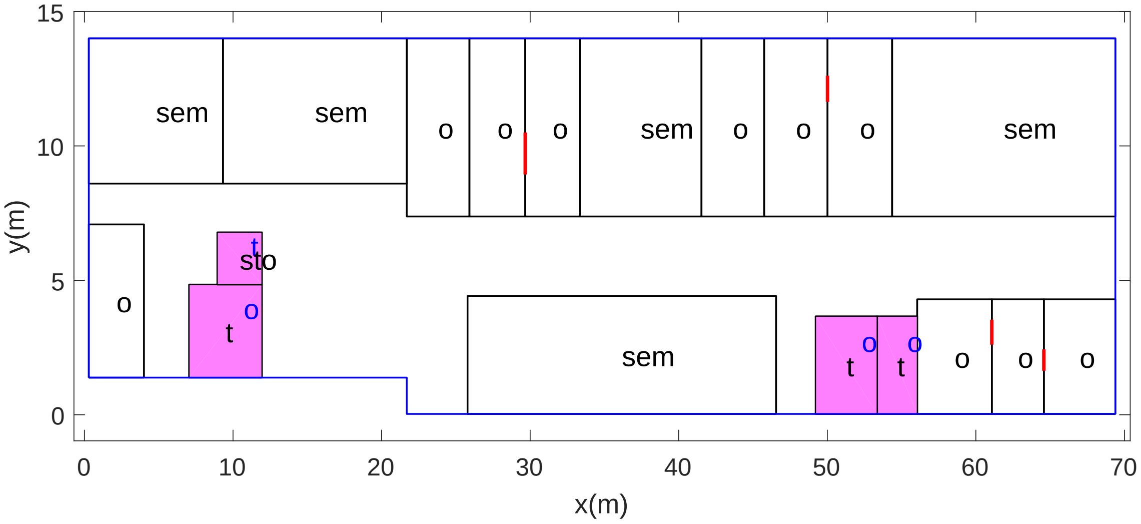
(c)

(d)

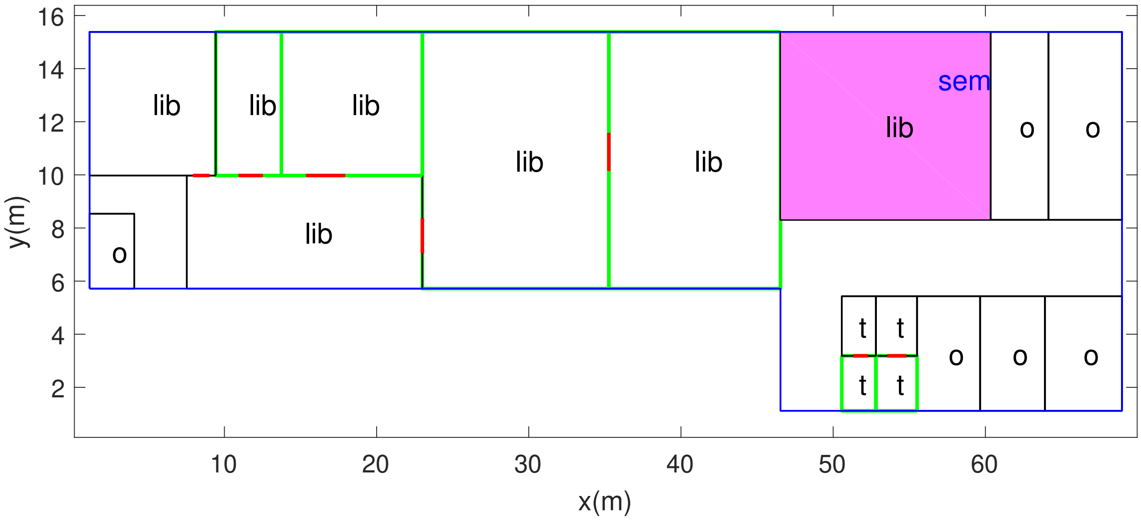
(e)


o
o
o
o
o
o
o
o
o
o
o
t
c
b
sto
sto
sto
t
t
sto
b
o
o
o
o
o
o
o
o
lib
sem
y(m)
x(m)
0
5
10
15
20
25
0
5
10
15
20
25

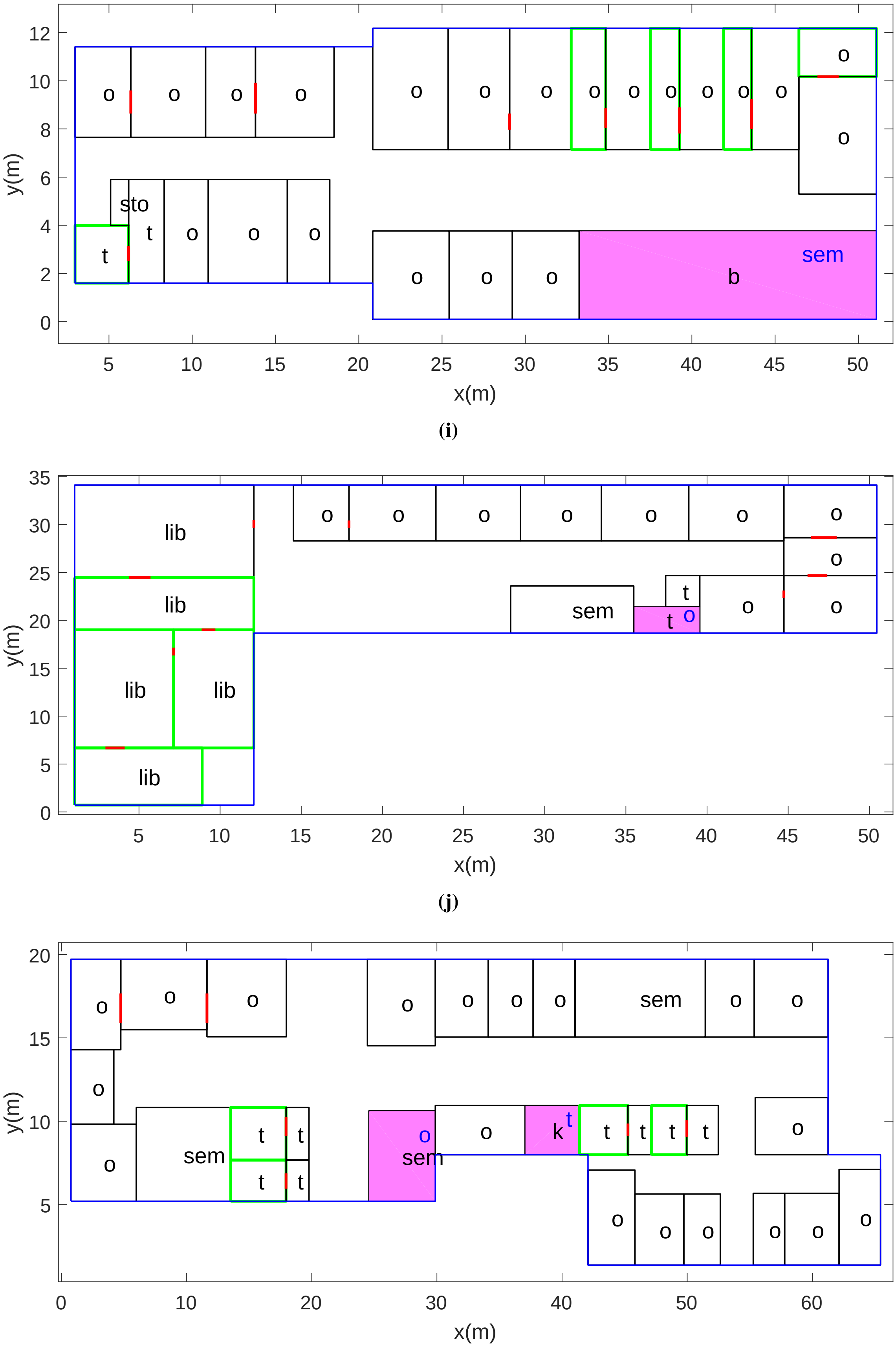
**(f)**


sem
sem
o
o
o
sem
o
o
o
sem
o
t
sto
o
t
sem
o
t
o
t
o
o
o
y(m)
x(m)
0
5
10
15
0
10
20
30
40
50
60
70

**(g)**


lib
lib
lib
lib
lib
sem
lib
o
o
o
lib
t
t
t
t
o
o
o
y(m)
x(m)
2
4
6
8
10
12
14
16
10
20
30
40
50
60

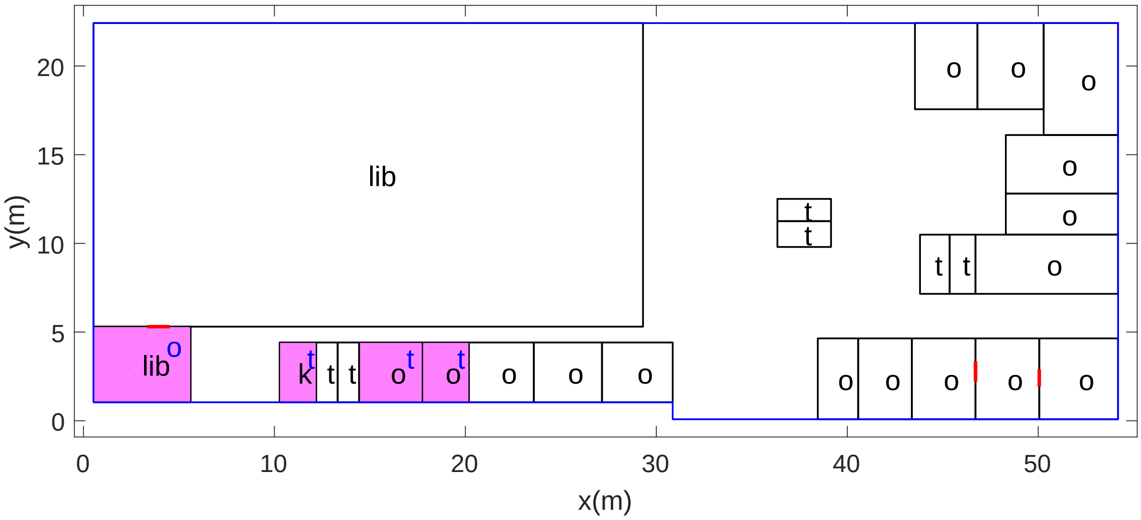
**(h)**


y(m)
x(m)
0
2
4
6
8
10
12
5
10
15
20
25
30
35
40
45
50
o
o
o
o
o
o
o
o
o
o
o
o
o
o
sto
t
t
o
o
o
o
o
o
sem
b
y(m)
x(m)
0
5
10
15
20
25
30
35
lib
lib
lib
lib
lib
o
o
o
o
o
o
o
o
o
sem
t
t
o
o
y(m)
x(m)
0
10
20
30
40
50
60
5
10
15
20
o
o
o
o
o
o
o
o
sem
o
o
o
o
o
sem
t
t
t
t
o
sem
o
k
t
t
t
t
t
o
o
o
o
o
o
o

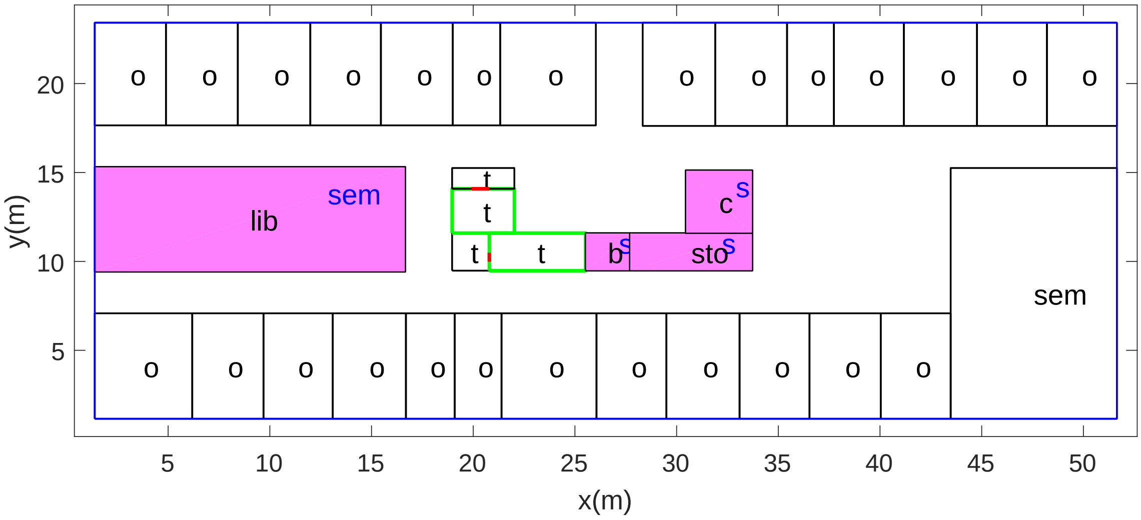
(i)

(j)

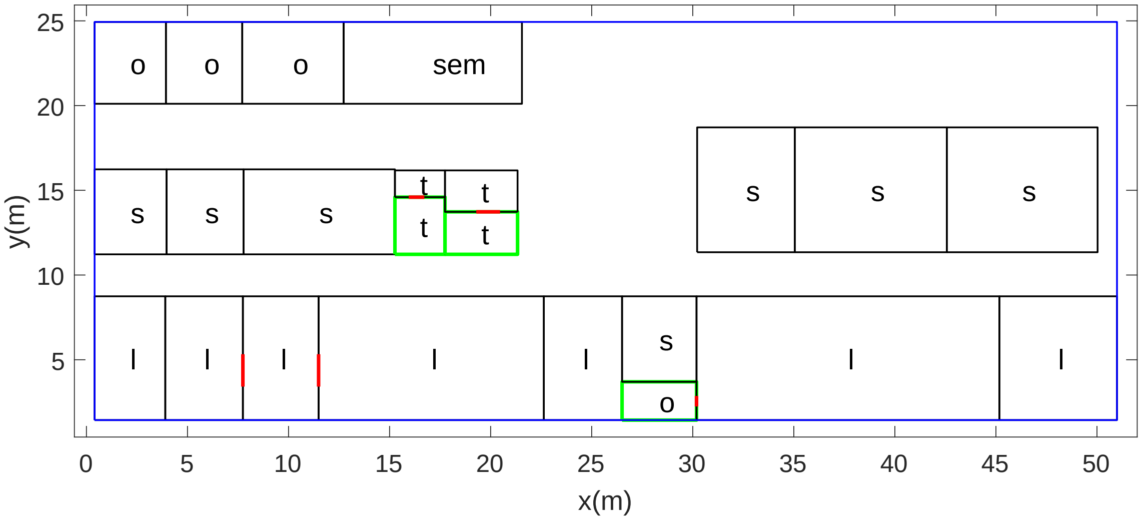
(k)


lib
o
o
o
o
o
t
t
t
t
o
lib
o
k
t
t
t
o
t
o
t
o
o
o
o
o
o
o
o
y(m)
x(m)
0
5
10
15
20
0
10
20
30
40
50

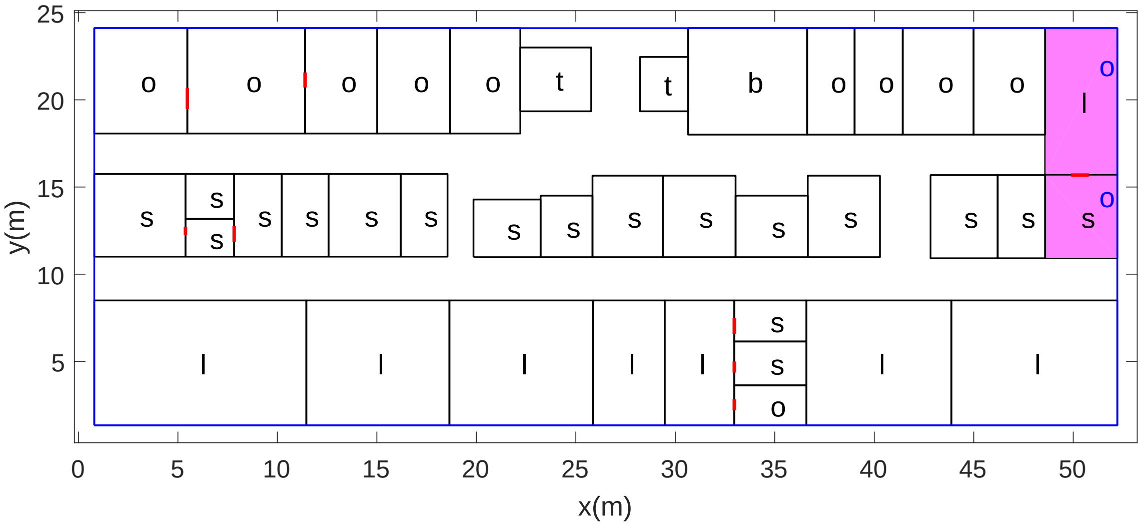
(l)


o
o
o
o
o
o
o
o
o
o
o
o
o
o
lib
sem
t
t
c
s
t
t
b
s
sto
s
sem
o
o
o
o
o
o
o
o
o
o
o
o
y(m)
x(m)
5
10
15
20
5
10
15
20
25
30
35
40
45
50

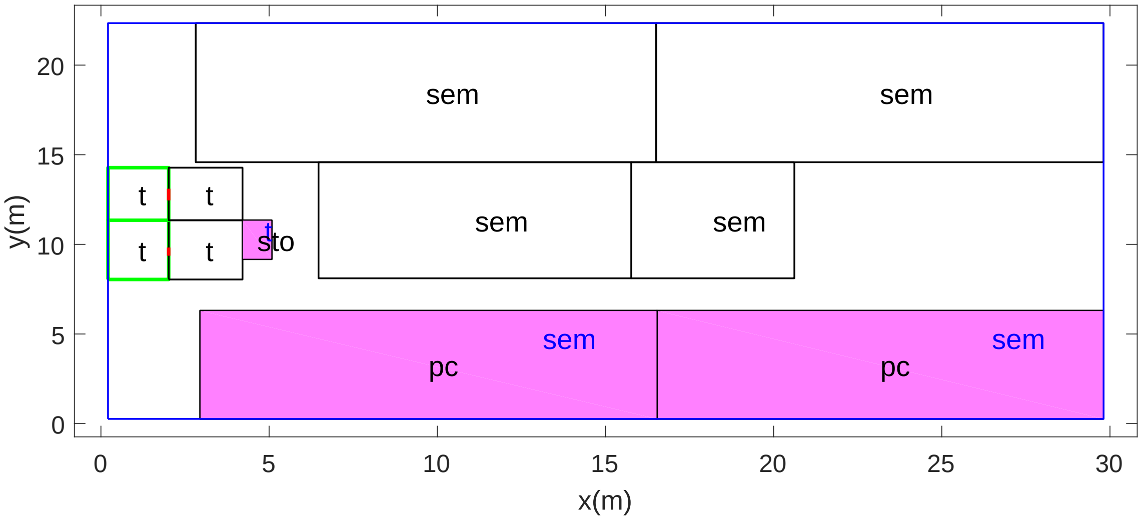
(m)


o
o
o
sem
s
s
s
t
t
t
t
s
s
s
l
l
l
l
l
s
o
l
l
y(m)
x(m)
5
10
15
20
25
0
5
10
15
20
25
30
35
40
45
50

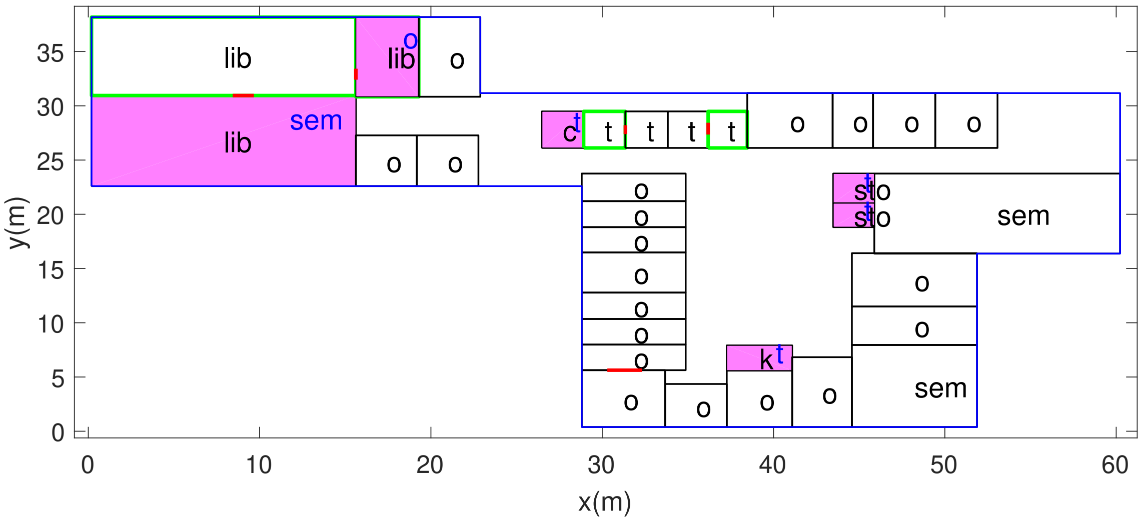
(n)


25
20
15
10
5
y(m)
0
5
10
15
20
25
30
35
40
45
50
x(m)
o
o
o
o
o
t
t
b
o
o
o
o
l
o
s
s
s
s
s
s
s
s
s
s
s
s
s
s
s
s
o
l
l
l
l
l
s
s
o
l
l

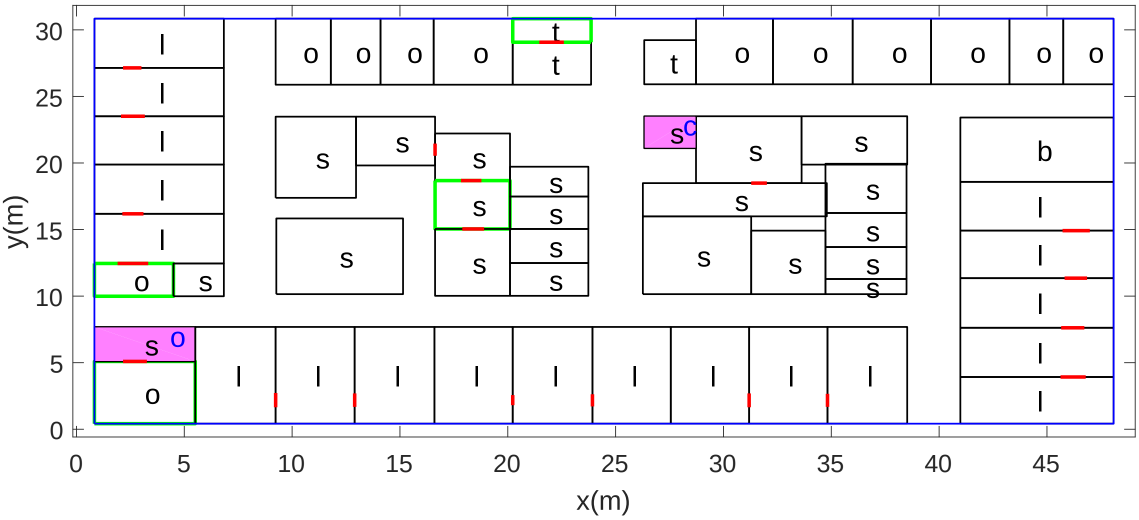
**(o)**


20
15
10
5
0
y(m)
0
5
10
15
20
25
30
x(m)
sem
sem
t
t
t
t
t
sto
sem
sem
sem
pc
sem
pc

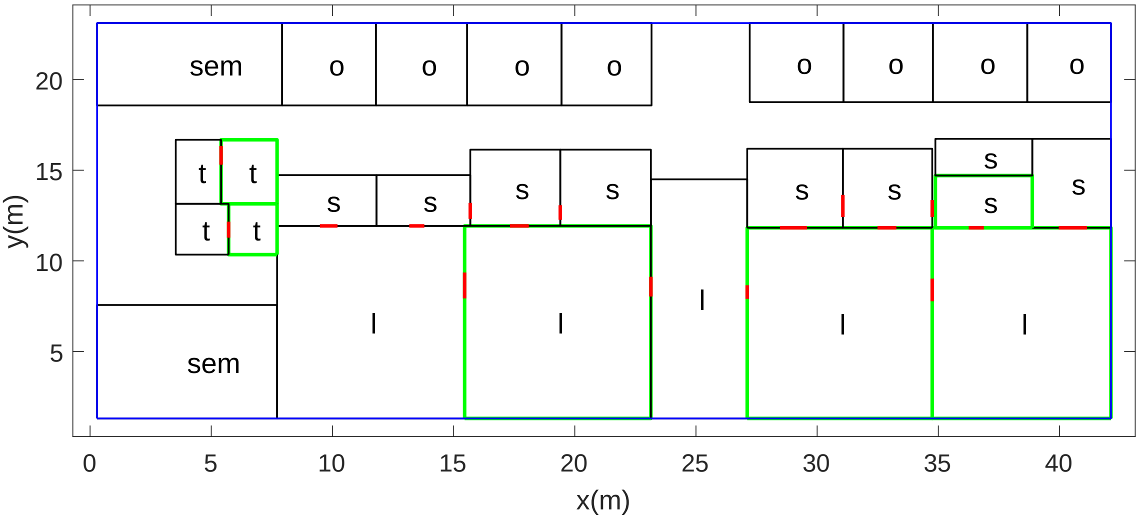
**(p)**


35
30
25
20
15
10
5
0
y(m)
0
10
20
30
40
50
60
x(m)
lib
o
lib
o
sem
lib
o
o
c
t
t
t
t
t
o
o
o
o
o
o
o
o
o
o
o
t
sto
t
sto
sem
o
o
k
t
o
o
o
sem

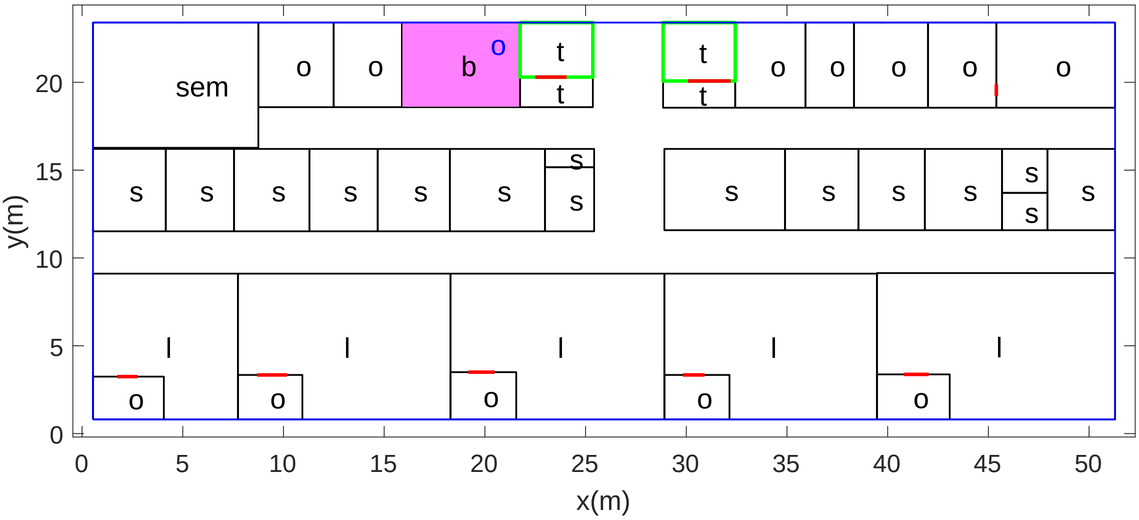
**(q)**


y(m)
x(m)
0
5
10
15
20
25
30
35
40
45
l
l
l
l
l
o
s
o
o
o
o
t
t
t
o
o
o
o
o
o
s
s
s
s
s
s
s
s
s
s
s
s
sc
s
s
s
s
s
s
s
s
b
l
l
l
l
l
s o
o
l
l
l
l
l
l
l
l
l

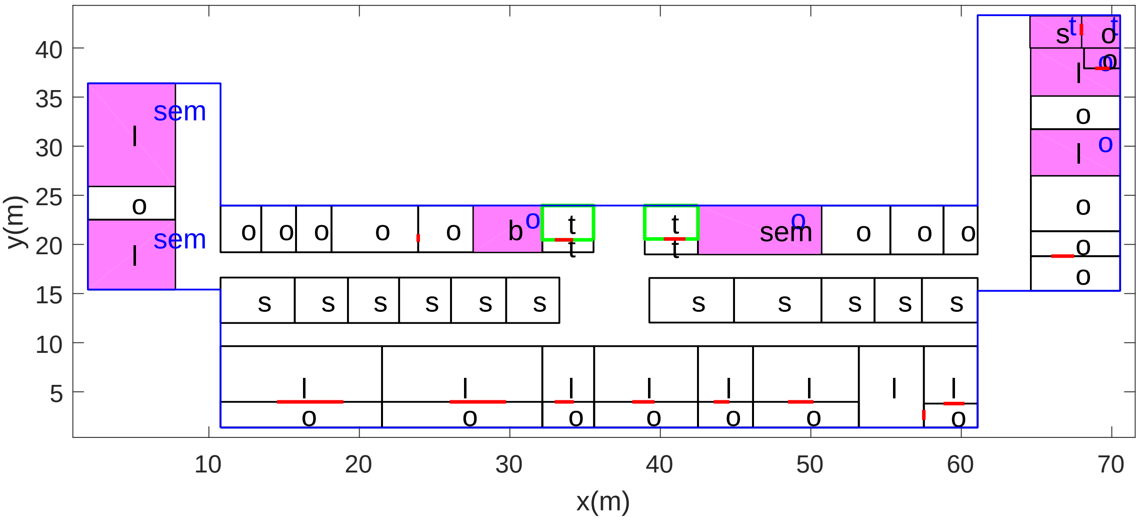
(r)


y(m)
x(m)
0
5
10
15
20
25
30
35
40
sem
o
o
o
o
o
o
o
o
t
t
t
t
s
s
s
s
s
s
s
s
s
l
l
l
l
l
sem

(s)

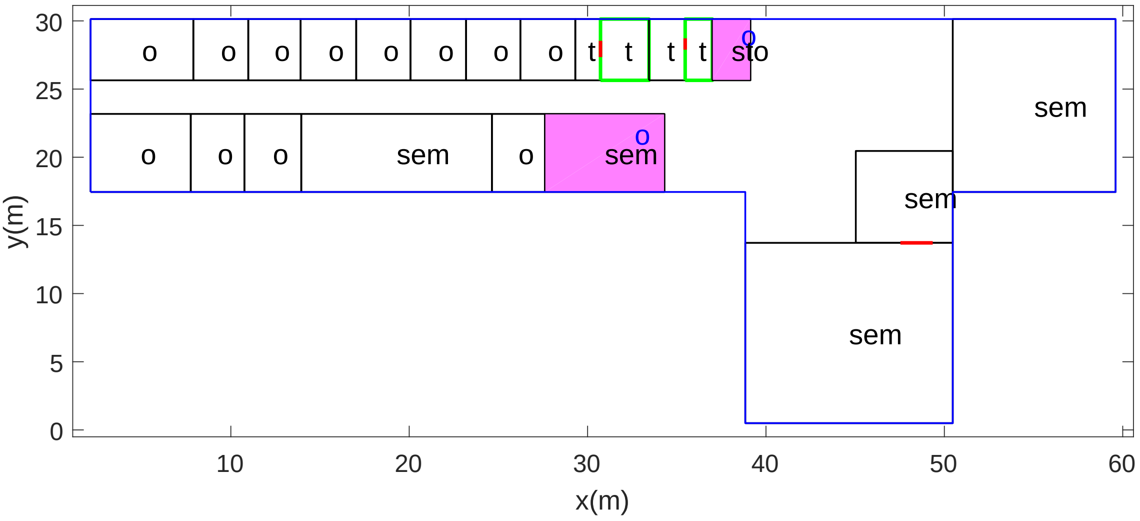

y(m)
x(m)
0
5
10
15
20
25
30
35
40
45
50
sem
o
o
b o
t
t
t
t
o
o
o
o
o
s
s
s
s
s
s
s
s
s
s
s
s
s
s
s
l
l
l
l
l
o
o
o
o
o

(t)



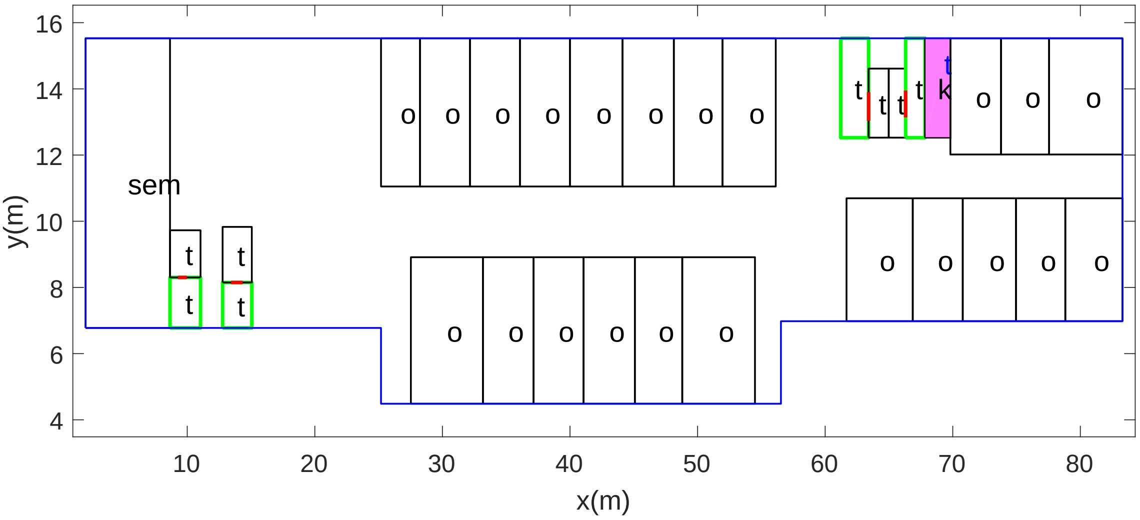
y(m)
x(m)
40
35
30
25
20
15
10
5
10
20
30
40
50
60
70
sem
l
o
sem
l
o o o o o
b o
t
t
t
t
sem o
o o o
s t o t
l o
o
l o
o
o
o
s s s s s s
s s s s s
l l l l l l l l
o o o o o o o

**(u)**



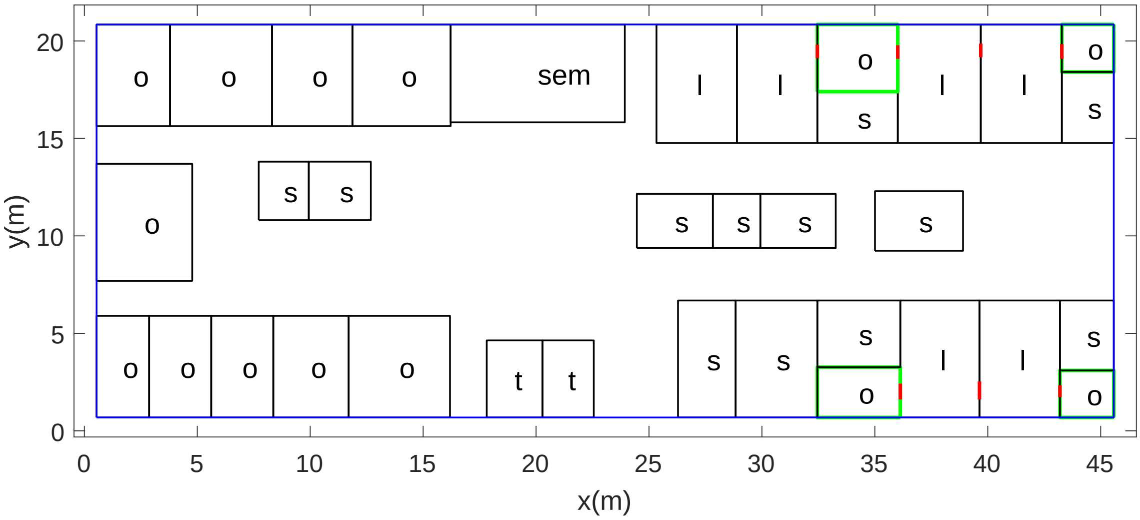
y(m)
x(m)
30
25
20
15
10
5
0
10
20
30
40
50
60
o o o o o o o o t t t t sto
o o o sem o sem o
sem
sem
sem

**(v)**



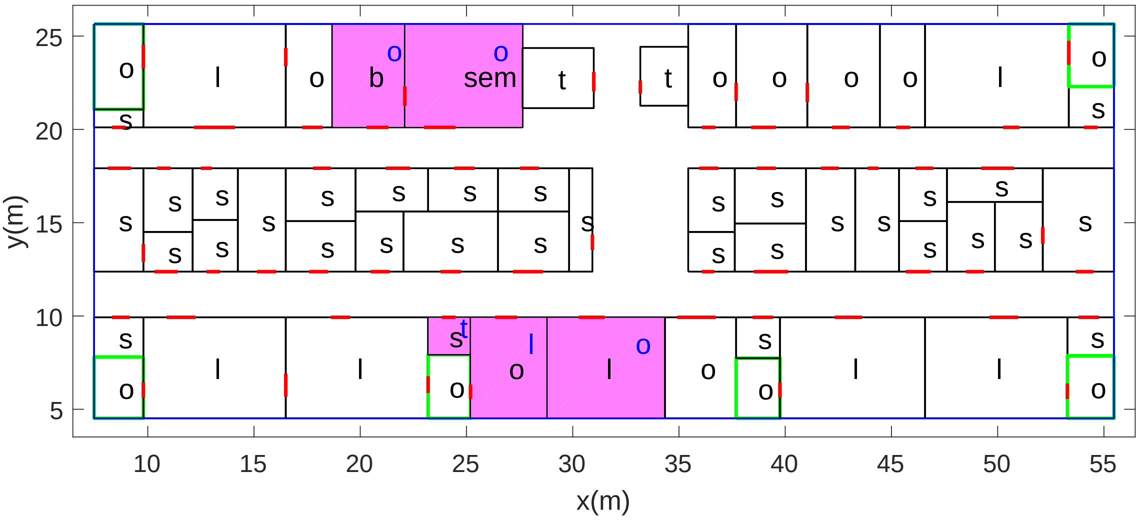
y(m)
x(m)
16
14
12
10
8
6
4
10
20
30
40
50
60
70
80
sem
t t
t t
o o o o o o o o
o o o o o o
t t t t t k
o o o
o o o o o

**(w)**

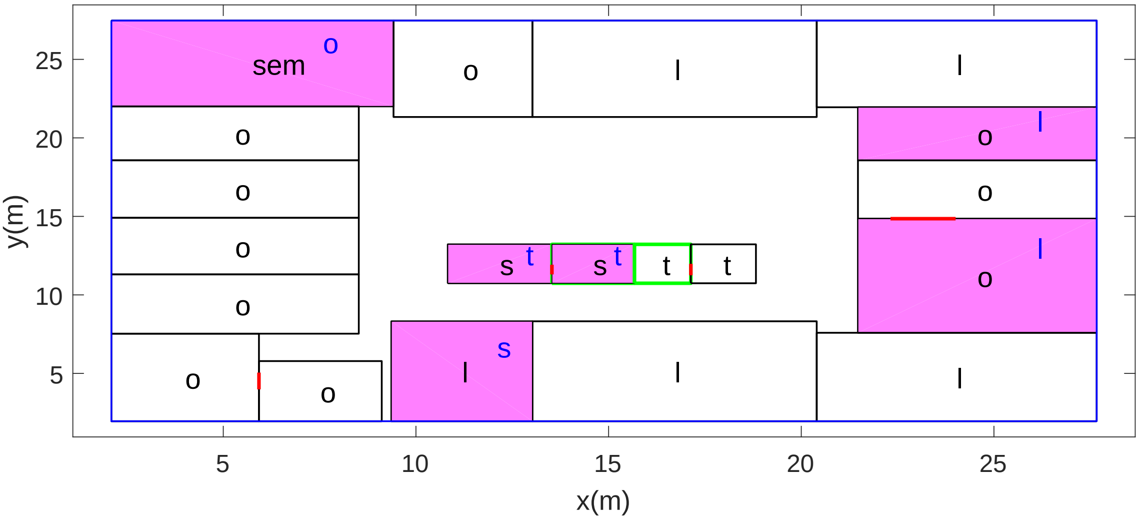


**(x)**

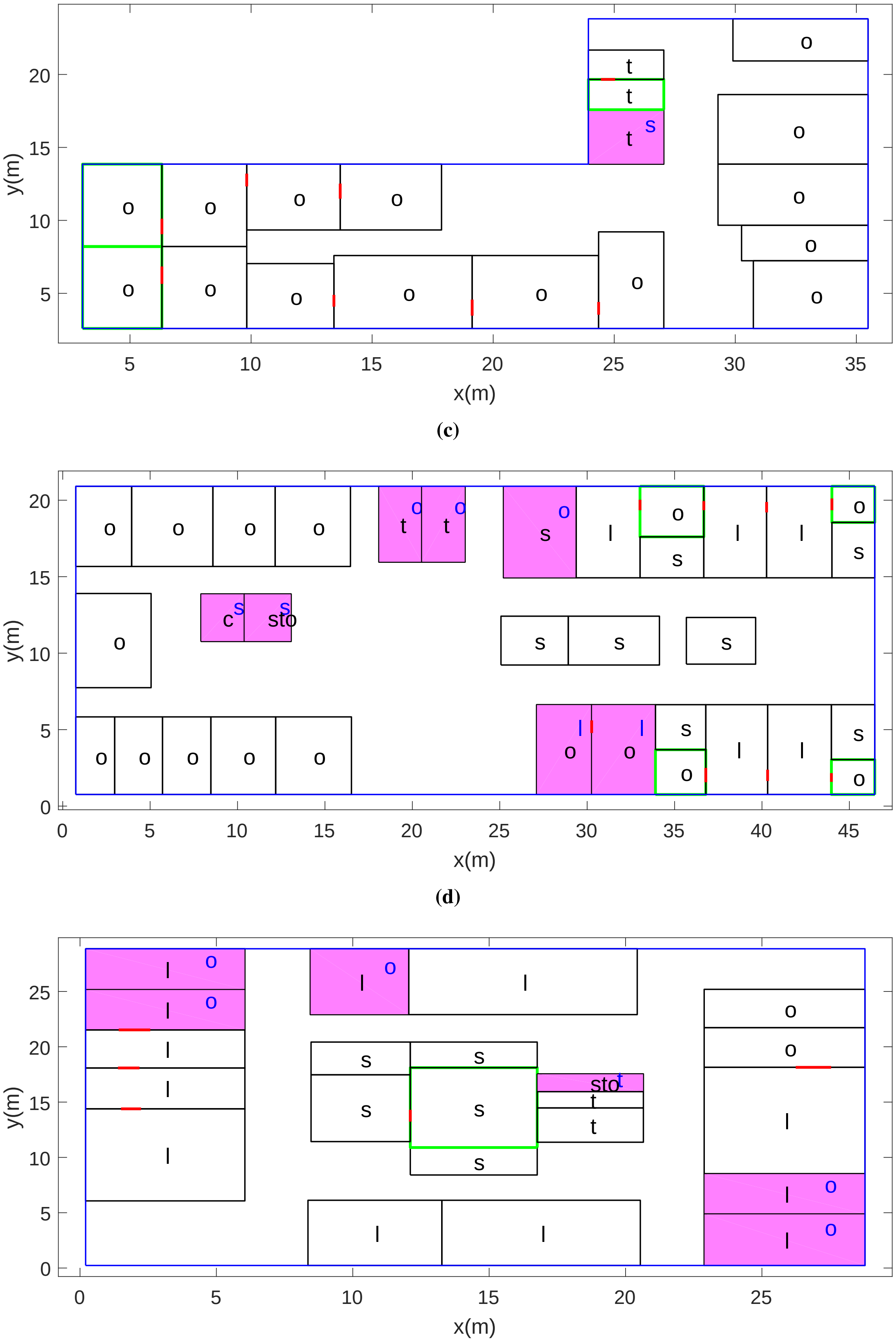
# D. Partial room tagging result achieved by RGCN



**(a)**



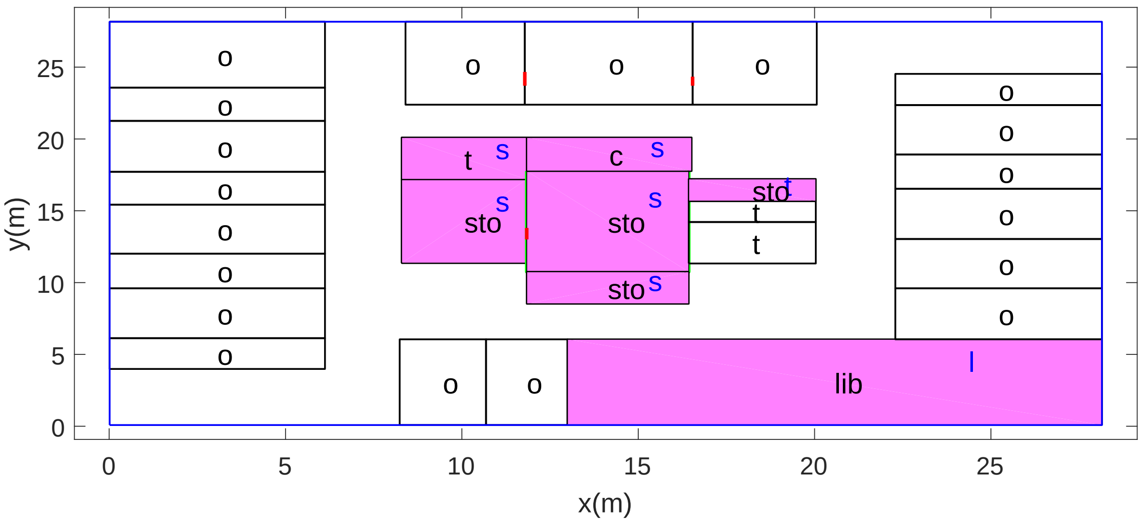
**(b)**


y(m)
x(m)
5
10
15
20
5
10
15
20
25
30
35
o
o
o
o
o
o
o
o
o
o
t
t
t
s
o
o
o
o
o
y(m)
x(m)
0
5
10
15
20
0
5
10
15
20
25
30
35
40
45
o
o
o
o
t
o
t
o
s
o
l
o
s
l
l
o
s
o
c
s
sto
s
s
s
s
o
o
o
o
o
o
l
o
l
s
o
l
l
s
o
y(m)
x(m)
0
5
10
15
20
25
0
5
10
15
20
25
l
o
l
o
l
l
l
l
o
l
s
s
s
s
s
sto
t
t
t
o
o
l
l
o
l
o
l
l

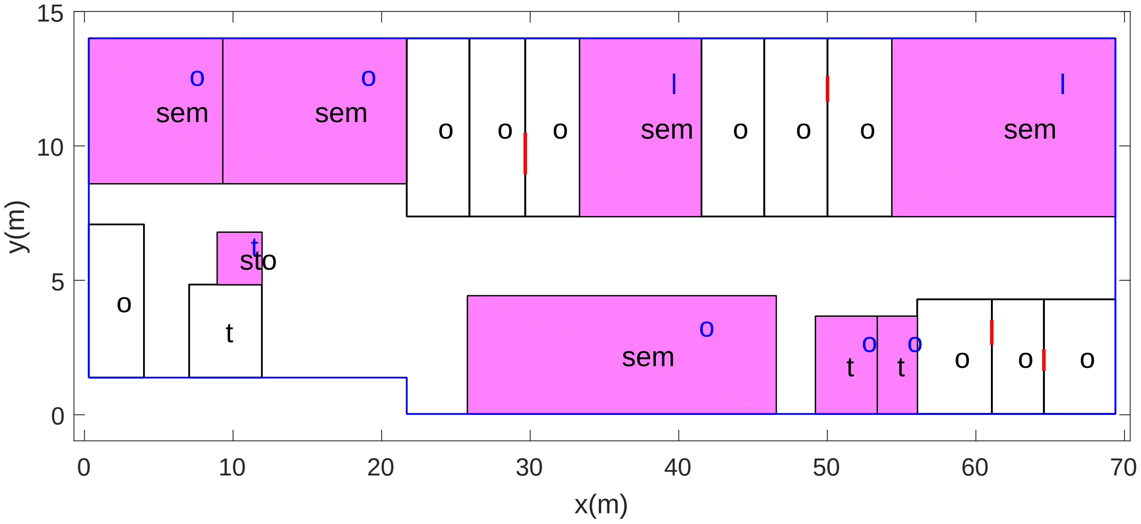
**(c)**

**(d)**

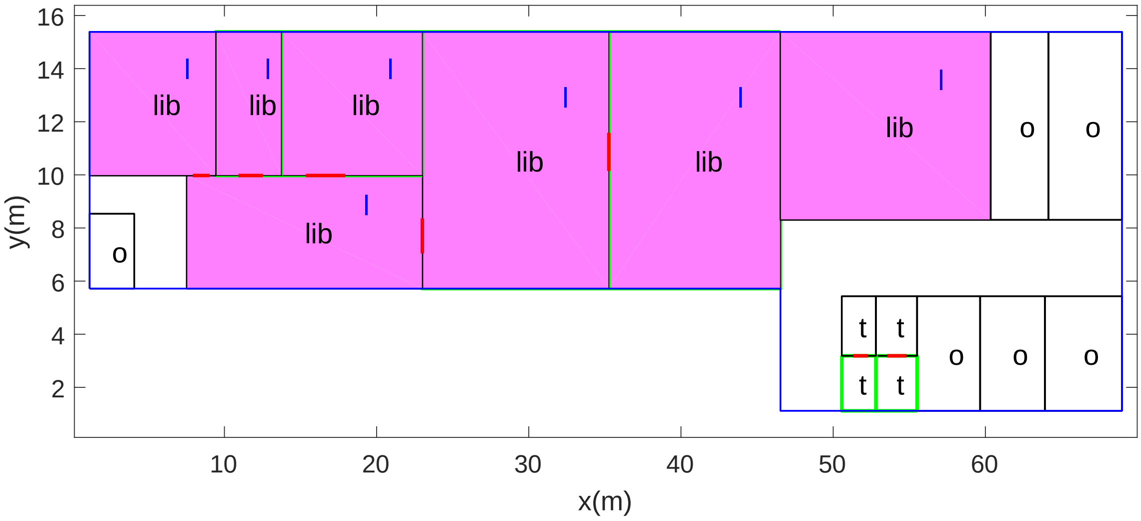
**(e)**


25
20
15
10
5
0
y(m)
0
5
10
15
20
25
x(m)
o
o
o
o
o
o
o
o
o
o
o
t
s
c
s
sto
s
sto
s
sto
t
t
t
sto
s
o
o
o
o
o
o
o
o
lib
l

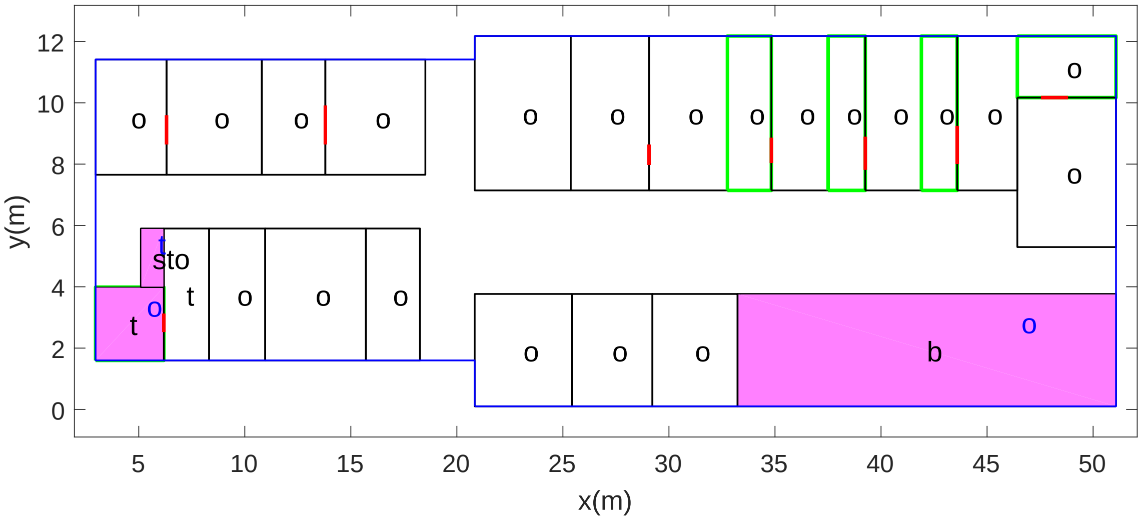
(f)


15
10
5
0
y(m)
0
10
20
30
40
50
60
70
x(m)
o
sem
o
sem
o
o
o
l
sem
o
o
o
l
sem
o
t
sto
t
sem
o
o
t
o
t
o
o
o

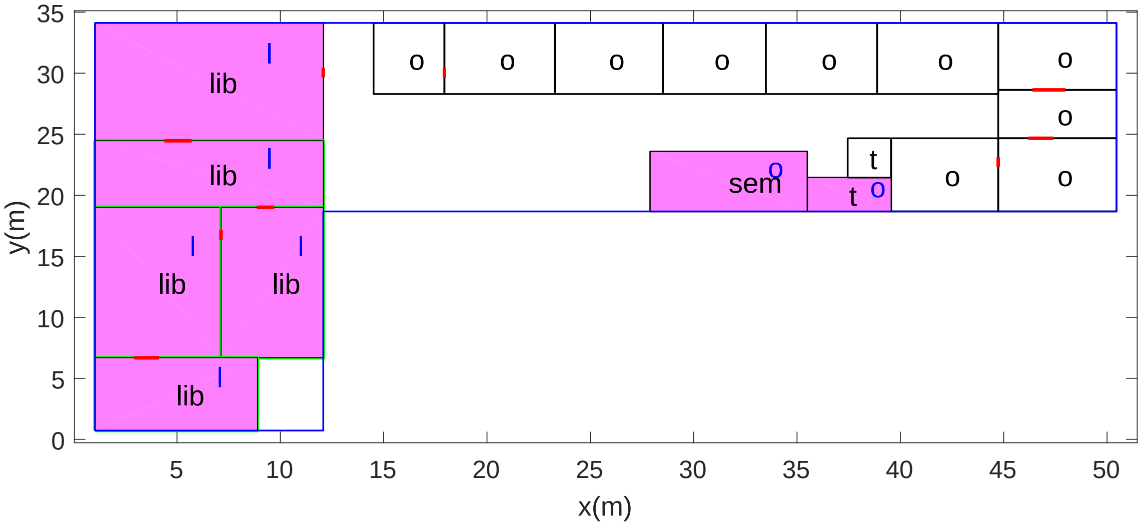
(g)


16
14
12
10
8
6
4
2
y(m)
10
20
30
40
50
60
x(m)
l
lib
l
lib
l
lib
l
lib
l
lib
l
lib
o
o
l
lib
o
t
t
t
t
o
o
o

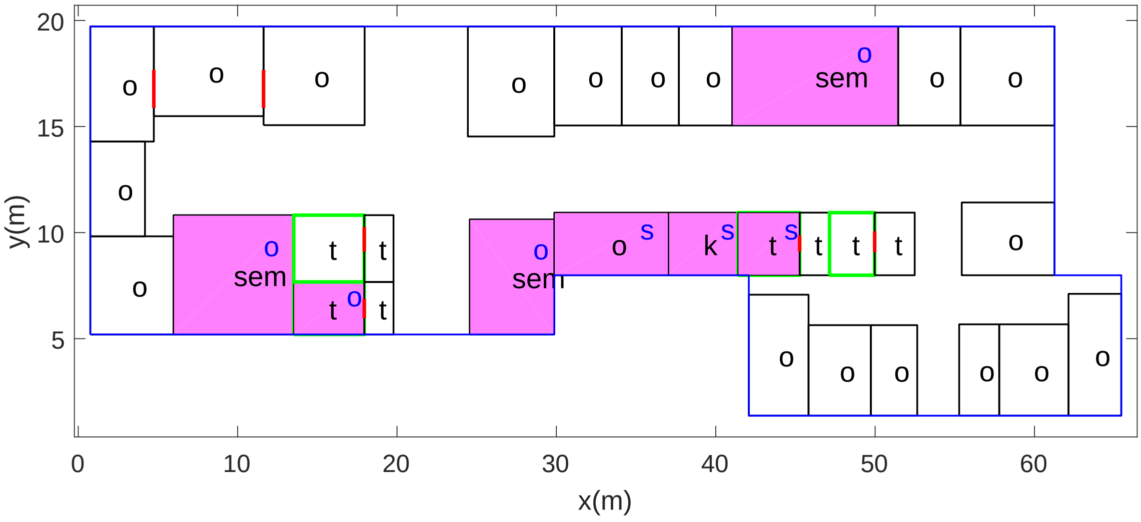
(h)


y(m)
x(m)
0
2
4
6
8
10
12
5
10
15
20
25
30
35
40
45
50
o
o
o
o
o
o
o
o
o
o
o
o
o
o
o
t
sto
t
o
t
o
o
o
o
o
o
b
o

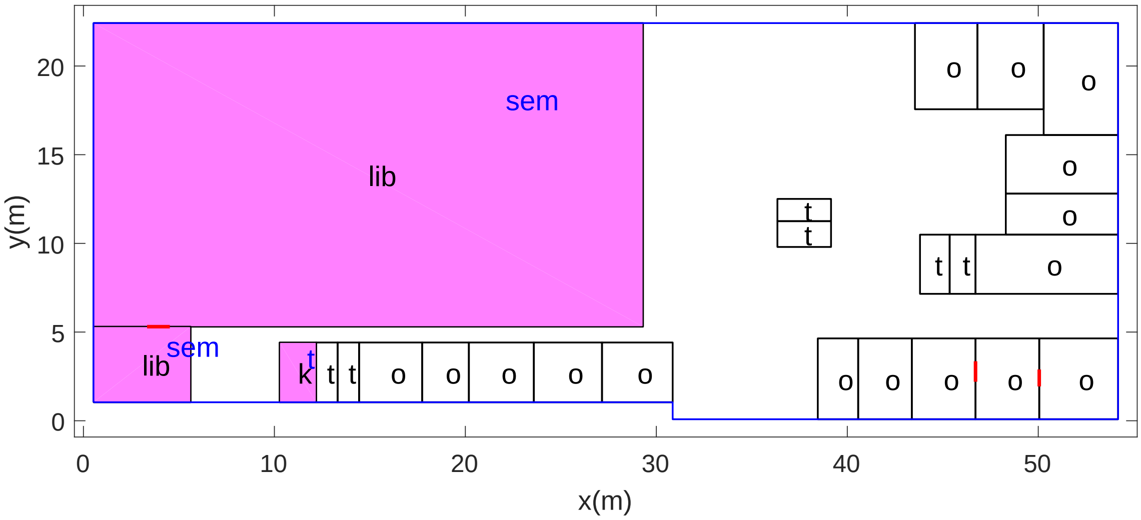
**(i)**


y(m)
x(m)
0
5
10
15
20
25
30
35
5
10
15
20
25
30
35
40
45
50
lib
l
lib
l
lib
l
lib
l
lib
l
o
o
o
o
o
o
o
o
sem
o
t
t
o
o
o

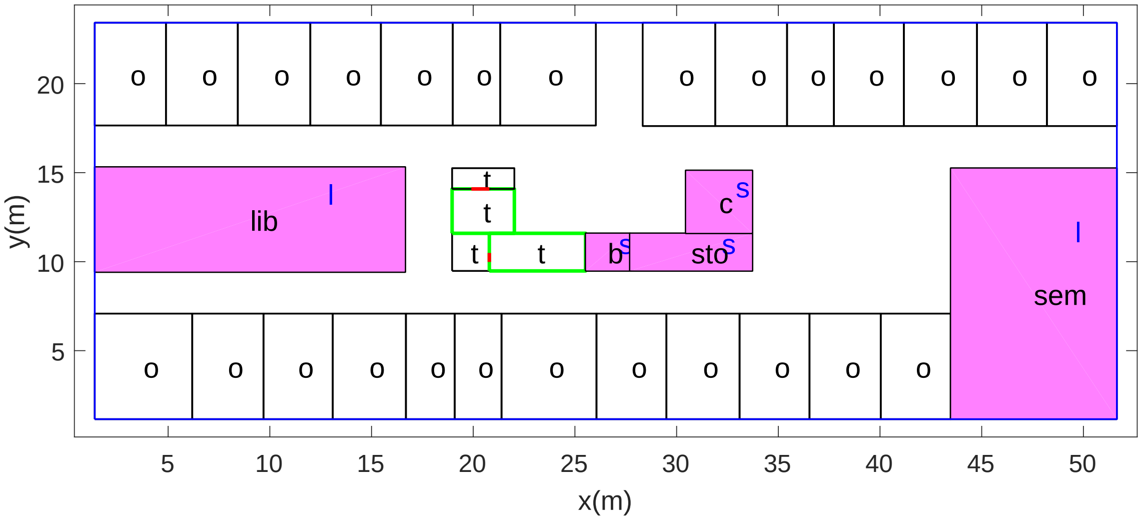
**(j)**


y(m)
x(m)
5
10
15
20
0
10
20
30
40
50
60
o
o
o
o
o
o
o
o
o
sem
o
o
o
o
o
sem
o
t
t
t
o
t
sem
o
o
s
k
s
t
s
t
t
t
o
o
o
o
o
o
o

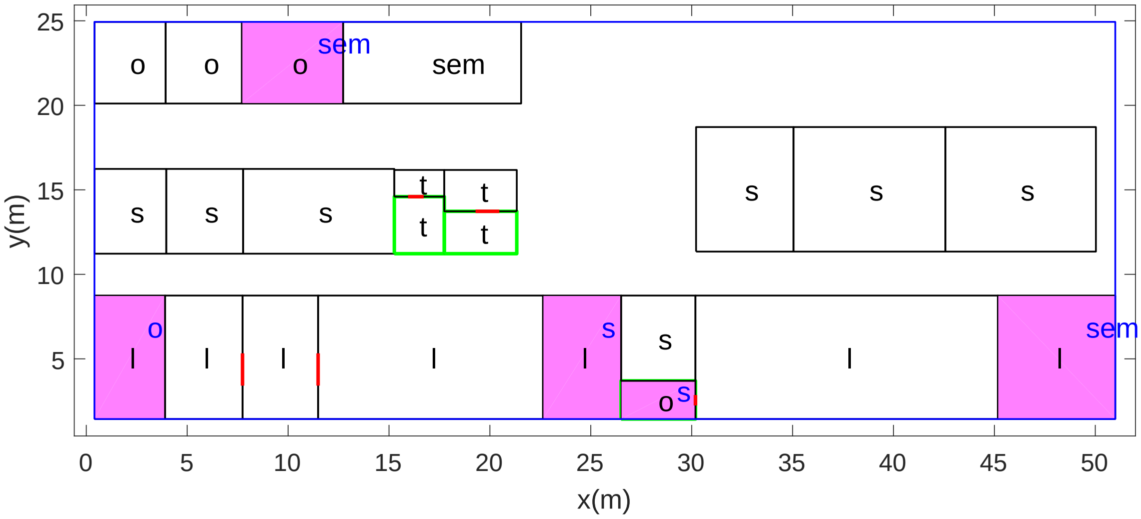
**(k)**


sem
lib
o
o
o
o
o
t
t
t
t
o
lib
sem
t
k
t
t
o
o
o
o
o
o
o
o
o
o
y(m)
0
5
10
15
20
0
10
20
30
40
50
x(m)

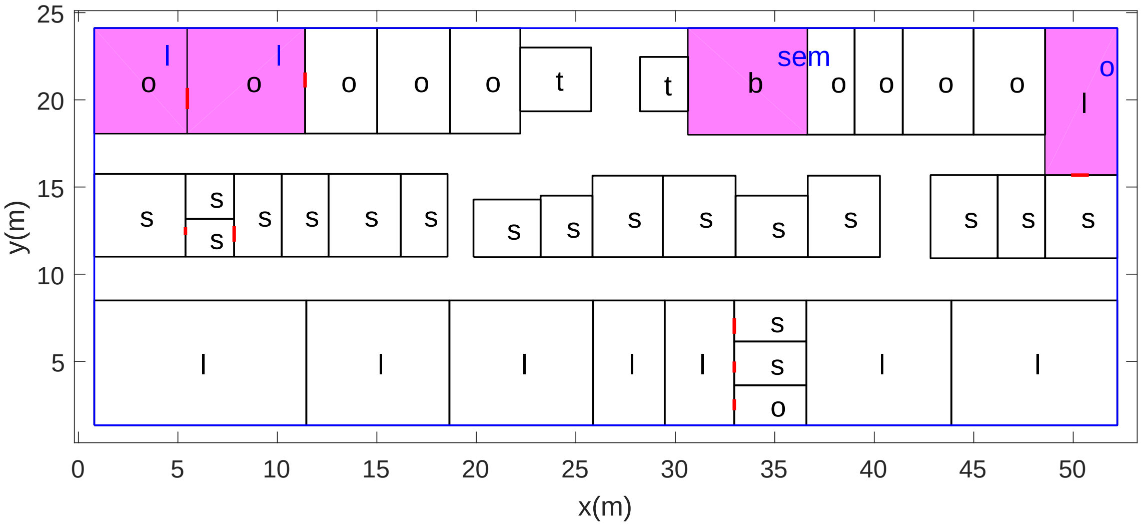
(l)


o
o
o
o
o
o
o
o
o
o
o
o
o
o
lib
l
t
t
t
t
c
s
b
s
sto
s
l
sem
o
o
o
o
o
o
o
o
o
o
o
o
y(m)
5
10
15
20
5
10
15
20
25
30
35
40
45
50
x(m)

(m)

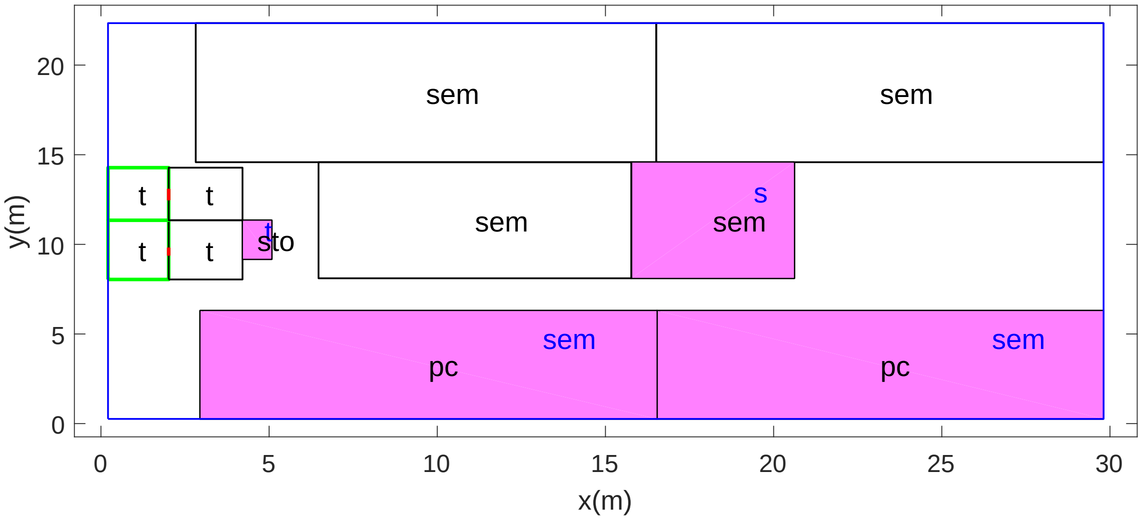

o
o
o
sem
sem
s
s
s
t
t
t
t
s
s
s
o
l
l
l
l
s
l
s
o
s
l
sem
l
y(m)
5
10
15
20
25
0
5
10
15
20
25
30
35
40
45
50
x(m)

(n)



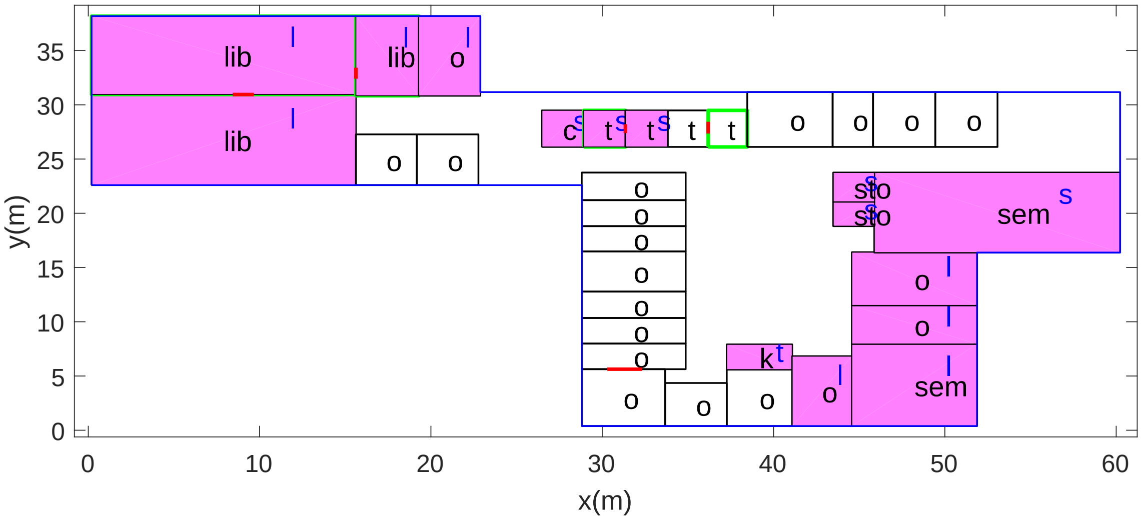
25
20
15
10
5
y(m)
0
5
10
15
20
25
30
35
40
45
50
x(m)
l
o
l
o
o
o
o
t
t
sem
b
o
o
o
o
o
l
s
s
s
s
s
s
s
s
s
s
s
s
s
s
s
s
l
l
l
l
l
s
s
o
l
l

**(o)**



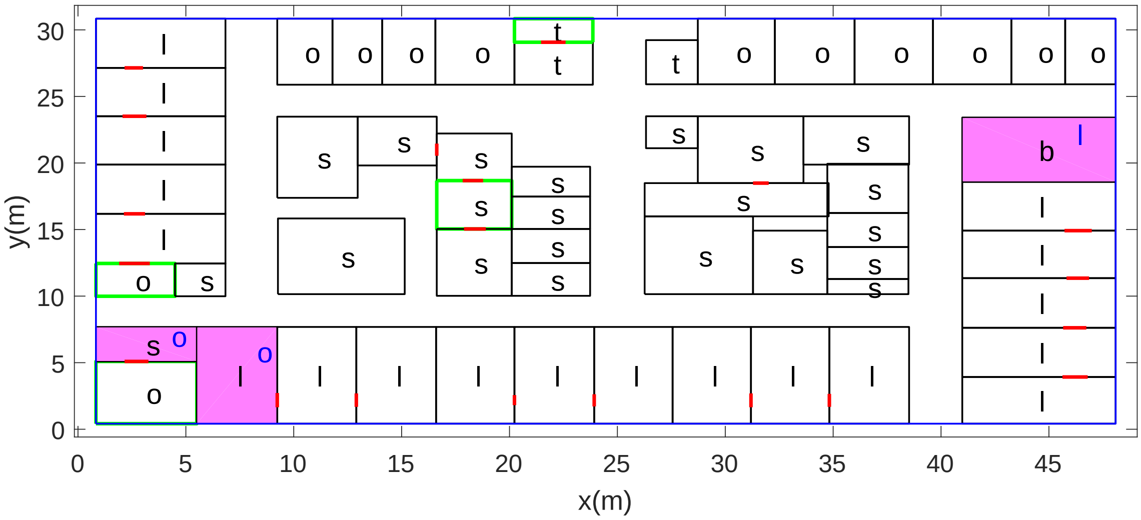
20
15
10
5
0
y(m)
0
5
10
15
20
25
30
x(m)
sem
sem
t
t
t
t
t
sto
sem
s
sem
sem
pc
sem
pc

**(p)**

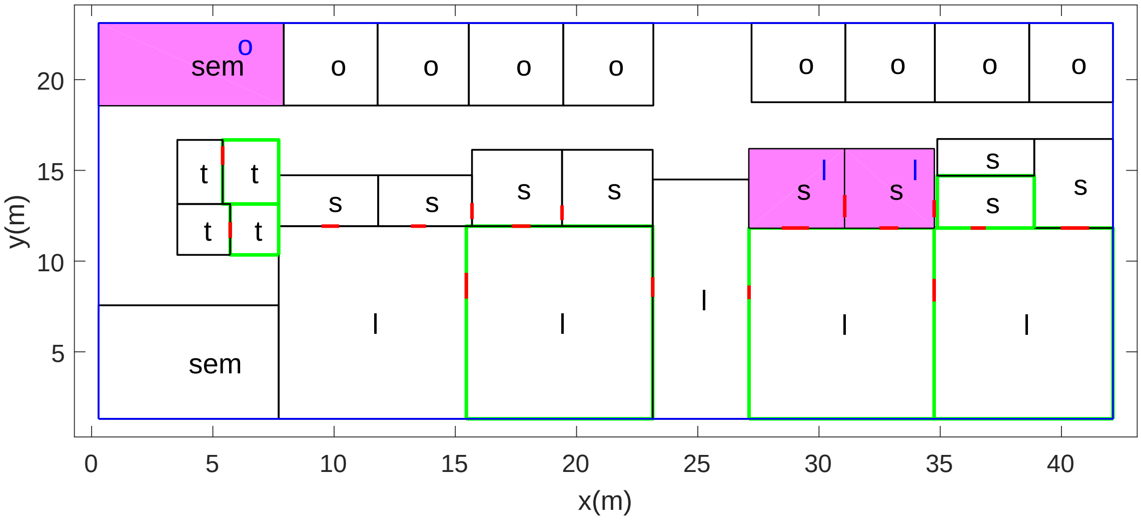


35
30
25
20
15
10
5
0
y(m)
0
10
20
30
40
50
60
x(m)
lib
l
l
lib
l
o
lib
l
o
o
c
s
t
s
t
s
t
t
o
o
o
o
o
o
o
s
sto
s
sto
sem
s
o
l
o
l
o
o
k
t
l
o
l
sem
o
o
o
o
o

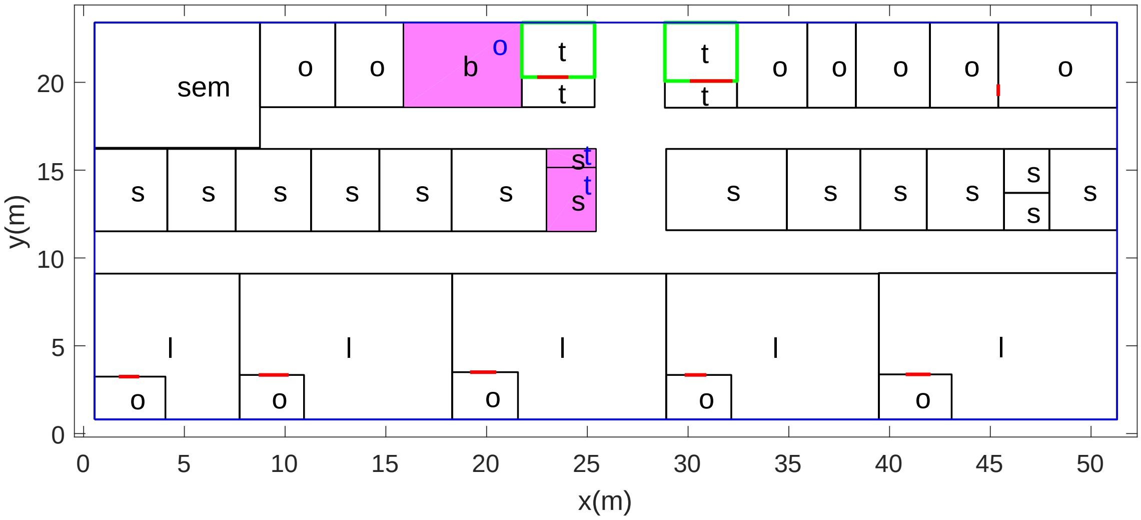
**(q)**


y(m)
x(m)
0
5
10
15
20
25
30
35
40
45
l
o
s
t
b

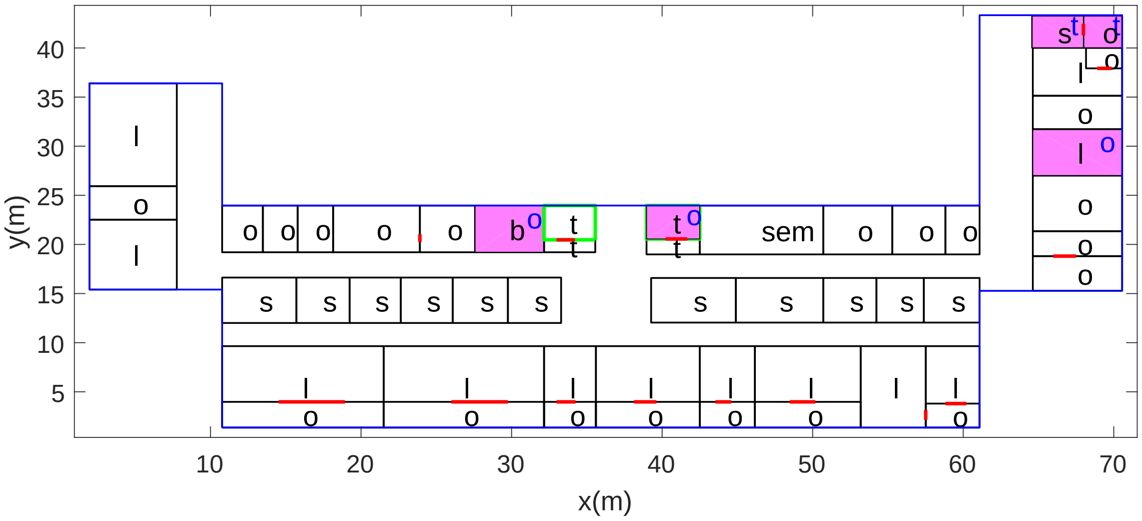
(r)


y(m)
x(m)
0
5
10
15
20
25
30
35
40
sem
o
t
s
l

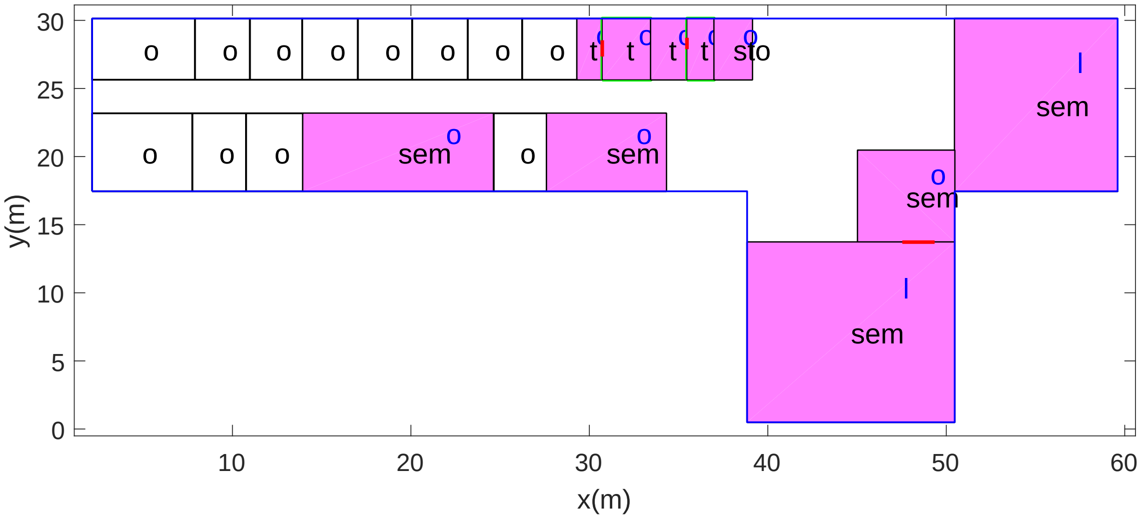
(s)


y(m)
x(m)
0
5
10
15
20
25
30
35
40
45
50
sem
o
b
t
s
l

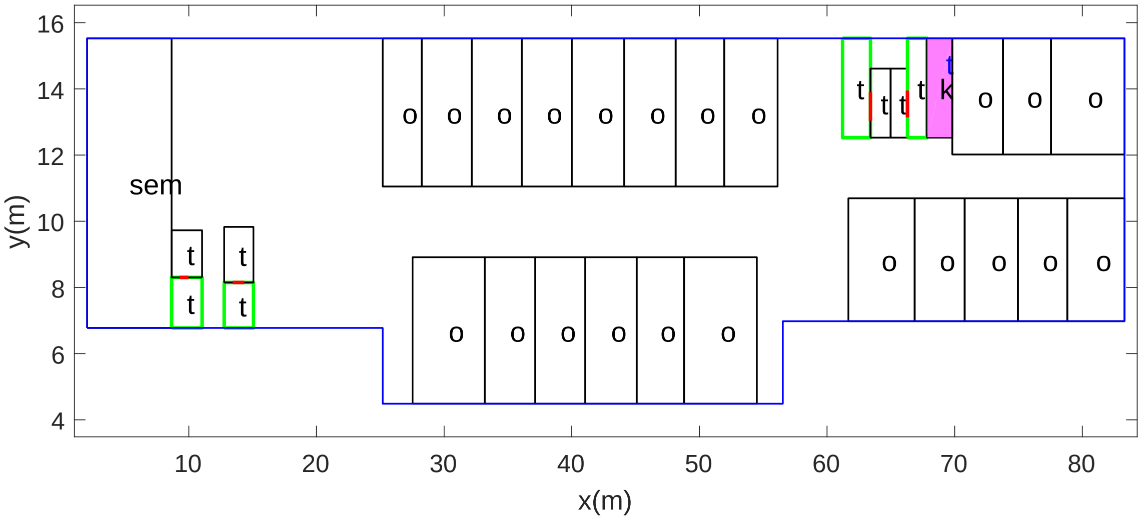
(t)


y(m)
x(m)
40
35
30
25
20
15
10
5
10
20
30
40
50
60
70
l
o
l
o
o
o
o
o
b
o
t
t
t
o
t
sem
o
o
o
s
s
s
s
s
s
s
s
s
s
s
l
l
l
l
l
l
l
l
o
o
o
o
o
o
o
s
t
o
t
o
l
o
l
o
o
o
o
o

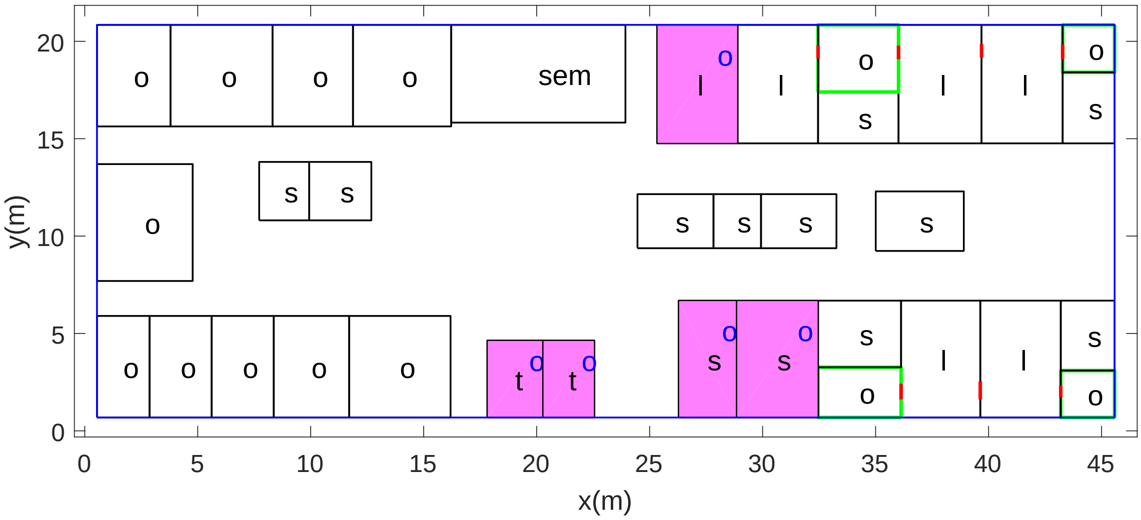
**(u)**


y(m)
x(m)
30
25
20
15
10
5
0
10
20
30
40
50
60
o
o
o
o
o
o
o
o
t
t
t
t
sto
o
o
o
sem
o
o
sem
o
l
sem
o
sem
l
sem

**(v)**


y(m)
x(m)
16
14
12
10
8
6
4
10
20
30
40
50
60
70
80
sem
t
t
t
t
o
o
o
o
o
o
o
o
o
o
o
o
o
o
t
t
t
t
k
o
o
o
o
o
o
o
o

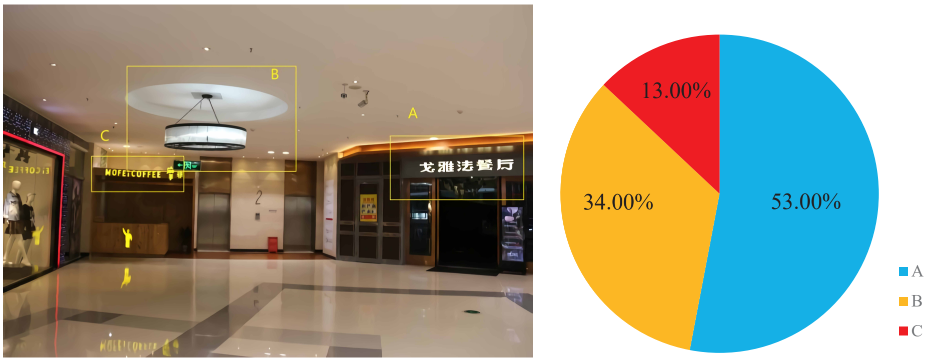
**(w)**



o
o
o
o
sem
l
o
l
o
s
l
l
o
s
o
s
s
s
s
s
s
o
o
o
o
o
t
o
t
o
s
o
s
o
s
o
l
l
s
o
0
5
10
15
20
25
30
35
40
45
x(m)
0
5
10
15
20
y(m)

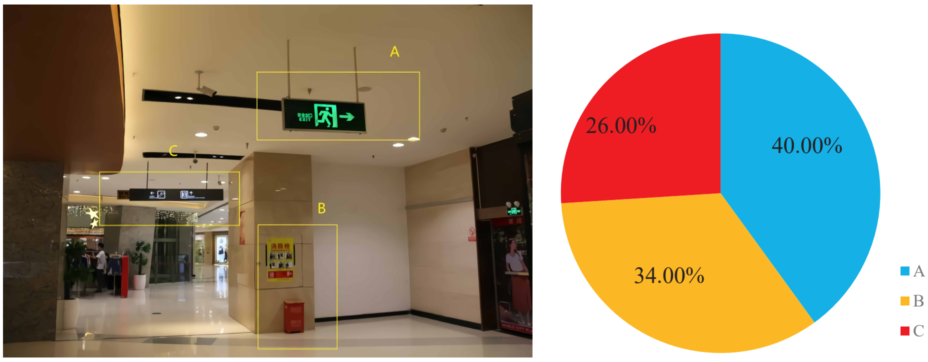
**(x)**

# E. Examples of test scenes for salience estimation



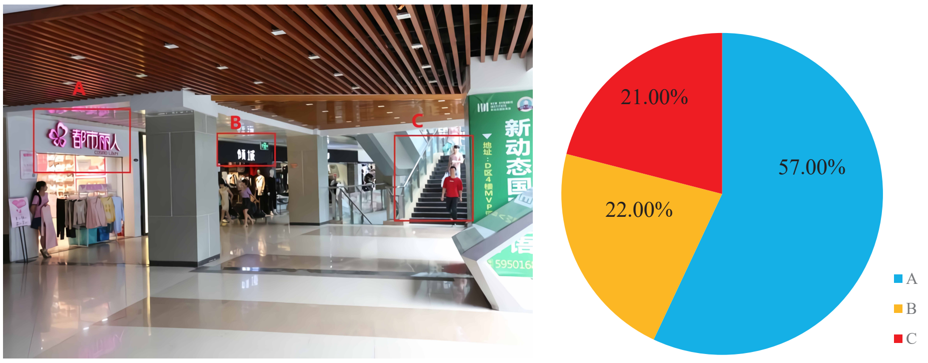
(a) (b)

**Figure E.1.** No. 1 in a test set.



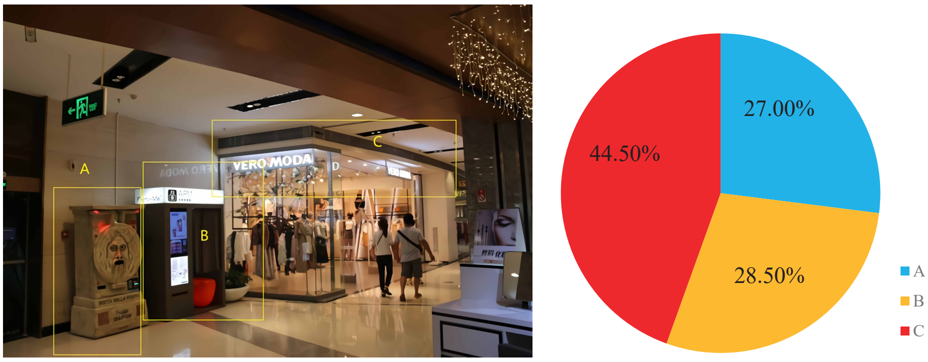
(a) (b)

**Figure E.2.** No. 2 in a test set.



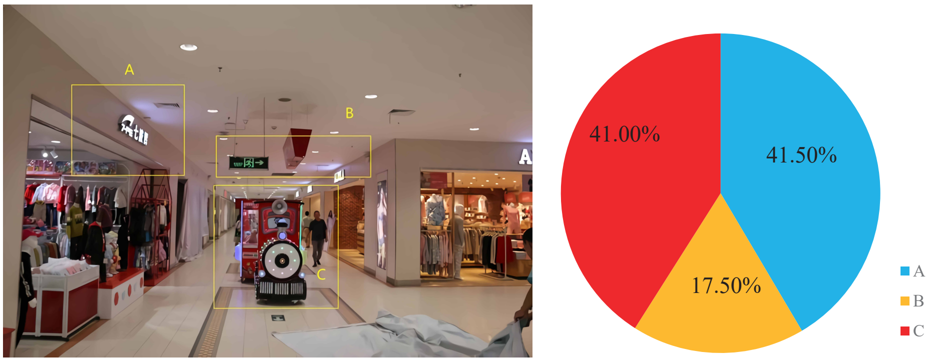
(a) (b)

**Figure E.3.** No. 3 in a test set.



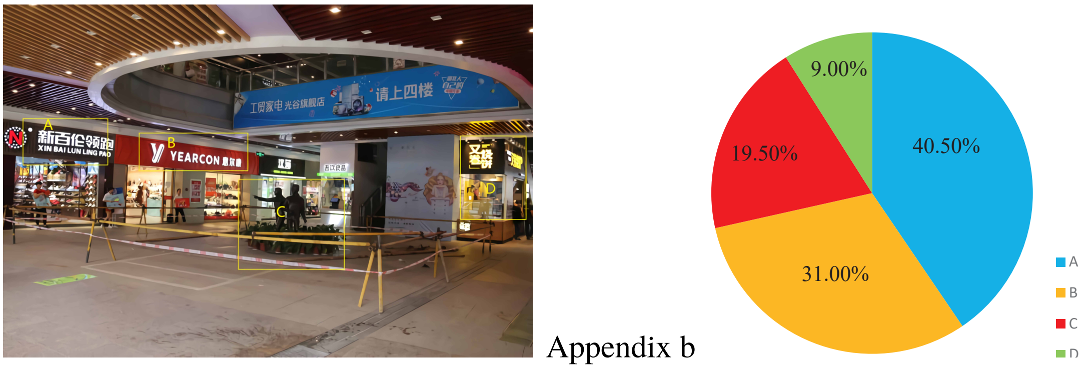
(a) (b)

**Figure E.4.** No. 4 in a test set.



(a) (b)

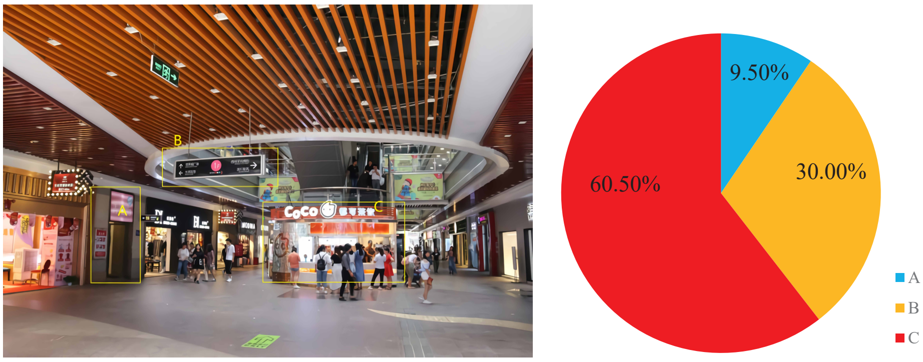
**Figure E.5.** No. 5 in a test set.



Appendix b

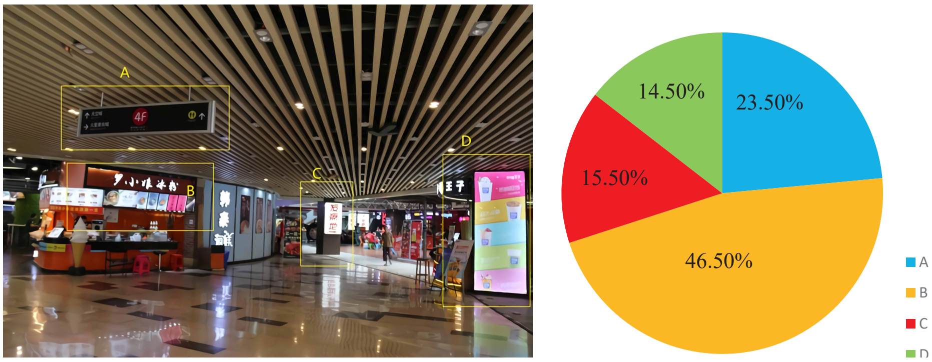
**(a)** **(b)**

**Figure E.6.** No. 6 in a test set.



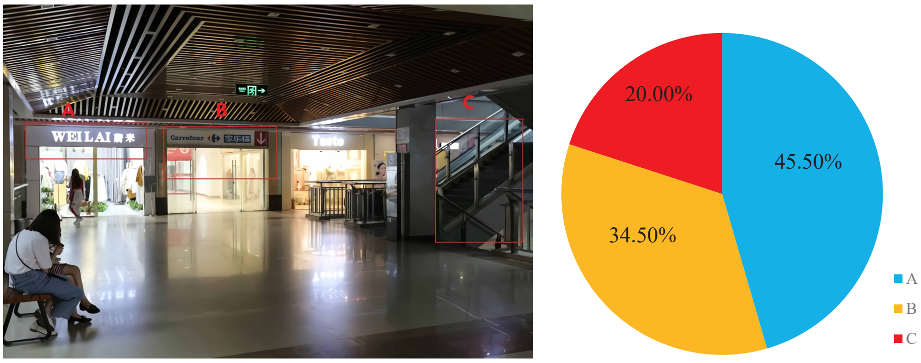
**(a)** **(b)**

**Figure E.7.** No. 7 in a test set.



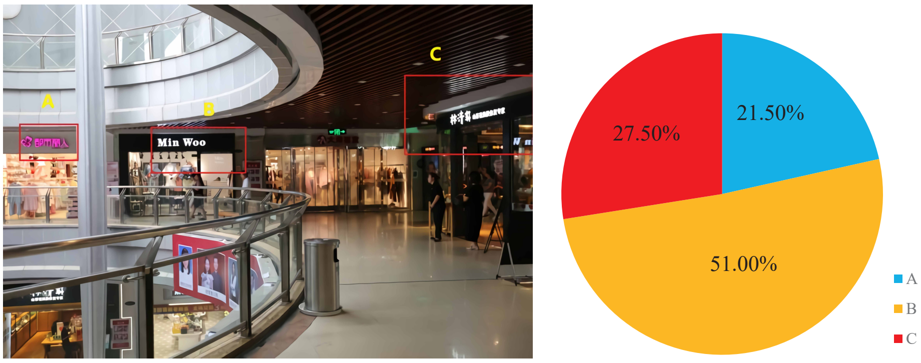
**(a)** **(b)**

**Figure E.8.** No. 8 in a test set.



(a) (b)

**Figure E.9.** No. 9 in a test set.



(a) (b)

**Figure E.10.** No. 10 in a test set.

# F. Salience estimation results

**Table F.1.** Predicted sort results for Fold1.

| Landmark scene | GP | Xing (2012) | Lyu, Yu, and Meng (2015) | Joachims (2002) | Ground truth |
|---|---|---|---|---|---|
| 1 | $A < B < C$ | $C < A < B$ | $A < C < B$ | $C < B < A$ | $A < C < B$ |
| 2 | $A < C < B$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $A < C < B$ |
| 3 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $C < B < A$ | $A < C < B$ |
| 4 | $A < B < D < C$ | $D < C < A < B$ | $D < C < A < B$ | $C < B < A < D$ | $A < C < B < D$ |
| 5 | $D < A < B < C$ | $B < C < D < A$ | $B < C < D < A$ | $C < A < D < B$ | $A < B < C < D$ |
| 6 | $D < A < C < B$ | $C < B < A < D$ | $C < B < A < D$ | $D < C < B < A$ | $D < C < A < B$ |
| 7 | $A < C < B < D$ | $B < D < C < A$ | $B < D < C < A$ | $D < B < A < C$ | $A < B < C < D$ |
| 8 | $A < C < B$ | $A < C < B <$ | $A < C < B$ | $C < A < B$ | $A < C < B$ |
| 9 | $A < B < C$ | $A < B < C$ | $C < B < A$ | $A < B < C$ | $B < A < C$ |
| 10 | $A < B < C$ | $A < C < B$ | $C < A < B$ | $A < B < C$ | $A < B < C$ |
| 11 | $C < A < B$ | $B < A < C$ | $B < A < C$ | $A < C < B$ | $C < B < A$ |
| 12 | $C < B < A$ | $B < A < C$ | $B < A < C$ | $B < C < A$ | $C < B < A$ |
| 13 | $C < B < A$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $C < B < A$ |
| 14 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $A < B < C$ |
| 15 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $A < B < C$ | $A < C < B$ |
| 16 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $B < C < A$ | $B < A < C$ |
| 17 | $D < C < A < B$ | $B < A < D < C$ | $B < A < D < C$ | $D < C < B < A$ | $D < A < B < C$ |
| 18 | $C < B < A < D$ | $A < D < C < B$ | $A < D < C < B$ | $B < C < A < D$ | $D < A < B < C$ |
| 19 | $A < C < B$ | $B < C < A$ | $B < C < A$ | $A < C < B$ | $A < B < C$ |
| 20 | $A < C < B$ | $B < C < A$ | $B < C < A$ | $B < C < A$ | $A < B < C$ |
| 21 | $D < C < A < B$ | $A < B < C < D$ | $A < B < C < D$ | $D < C < B < A$ | $C < A < B < D$ |
| 22 | $A < C < B$ | $C < A < B$ | $C < A < B$ | $C < B < A$ | $A < C < B$ |
| 23 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $A < C < B$ | $A < B < C$ |
| 24 | $C < A < B$ | $C < B < A$ | $C < B < A$ | $A < C < B$ | $C < B < A$ |
| 25 | $C < B < A$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $C < B < A$ |
| 26 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $C < B < A$ | $A < B < C$ |
| 27 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < A < C$ | $A < B < C$ |
| 28 | $B < C < A$ | $C < B < A$ | $C < B < A$ | $C < B < A$ | $B < A < C$ |
| 29 | $B < A < C$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $A < B < C$ |
| 30 | $C < B < A$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $C < B < A$ |
| 31 | $A < C < B$ | $B < C < A$ | $B < C < A$ | $A < C < B$ | $A < B < C$ |
| 32 | $B < A < C$ | $C < A < B$ | $C < B < A$ | $A < B < C$ | $B < A < C$ |
| 33 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $A < C < B$ |
| 34 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $C < B < A$ | $C < B < A$ |
| 35 | $A < C < B$ | $A < C < B$ | $A < B < C$ | $C < A < B$ | $A < B < C$ |
| 36 | $B < A < C$ | $B < C < A$ | $B < C < A$ | $A < C < B$ | $B < C < A$ |
| 37 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $A < B < C$ |
| 38 | $C < B < A$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $B < C < A$ |
| 39 | $C < B < A$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $C < A < B$ |
| 40 | $B < A < C$ | $A < C < B$ | $A < C < B$ | $C < B < A$ | $B < A < C$ |

**Table F.2.** Predicted sort results for Fold2.

| Landmark scene | GP | Xing (2012) | Lyu, Yu, and Meng (2015) | Joachims (2002) | Ground truth |
|---|---|---|---|---|---|
| 1 | $B < D < C < A$ | $A < D < C < B$ | $A < D < C < B$ | $C < D < B < A$ | $C < D < B < A$ |
| 2 | $B < C < A$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $A < C < B$ |
| 3 | $C < A < B$ | $C < A < B$ | $C < B < A$ | $A < C < B$ | $A < C < B$ |
| 4 | $B < C < A$ | $A < B < C$ | $A < B < C$ | $B < A < C$ | $B < A < C$ |
| 5 | $B < A < C$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $B < C < A$ |
| 6 | $B < A < C$ | $B < A < C$ | $A < B < C$ | $B < C < A$ | $B < C < A$ |
| 7 | $B < C < A$ | $C < B < A$ | $C < A < B$ | $B < C < A$ | $B < C < A$ |
| 8 | $B < A < C$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $B < A < C$ |
| 9 | $A < C < B$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $A < B < C$ |
| 10 | $B < A < C$ | $B < A < C$ | $B < A < C$ | $A < B < C$ | $B < A < C$ |
| 11 | $B < C < A$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $A < B < C$ |
| 12 | $B < A < C$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $B < A < C$ |
| 13 | $A < C < B$ | $B < C < A$ | $B < C < A$ | $A < C < B$ | $A < C < B$ |
| 14 | $A < C < B$ | $A < C < B$ | $A < B < C$ | $C < B < A$ | $B < A < C$ |
| 15 | $D < C < A < B$ | $A < B < C < D$ | $A < B < C < D$ | $D < A < C < B$ | $D < B < A < C$ |
| 16 | $C < B < A$ | $A < B < C$ | $A < B < C$ | $A < B < C$ | $B < C < A$ |
| 17 | $D < C < B < A$ | $C < B < D < A$ | $C < B < D < A$ | $A < D < B < C$ | $D < B < A < C$ |
| 18 | $C < A < B$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $C < A < B$ |
| 19 | $A < C < B$ | $C < B < A$ | $C < A < B$ | $C < B < A$ | $A < C < B$ |
| 20 | $B < C < D < A$ | $D < C < A < B$ | $D < C < A < B$ | $B < C < D < A$ | $B < A < C < D$ |
| 21 | $C < A < B$ | $A < B < C$ | $A < B < C$ | $C < A < B$ | $C < A < B$ |
| 22 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $A < B < C$ | $A < B < C$ |
| 23 | $A < B < C$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $A < C < B$ |
| 24 | $D < B < C < A$ | $D < B < C < A$ | $D < B < C < A$ | $D < A < B < C$ | $D < B < A < C$ |
| 25 | $A < C < B$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $A < B < C$ |
| 26 | $C < A < B$ | $A < C < B$ | $A < C < B$ | $A < B < C$ | $C < A < B$ |
| 27 | $A < B < C$ | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < C < A$ |
| 28 | $C < B < A$ | $B < C < A$ | $B < C < A$ | $C < B < A$ | $C < B < A$ |
| 29 | $B < A < C$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $A < B < C$ |
| 30 | $B < A < C < D$ | $D < A < B < C$ | $D < A < B < C$ | $C < B < D < A$ | $B < D < A < C$ |
| 31 | $C < B < A$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $C < B < A$ |
| 32 | $B < A < C$ | $A < C < B$ | $A < C < B$ | $A < C < B$ | $A < C < B$ |
| 33 | $A < C < B$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $A < B < C$ |
| 34 | $A < C < B$ | $B < A < C$ | $B < A < C$ | $B < A < C$ | $A < B < C$ |
| 35 | $B < A < C$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $B < A < C$ |
| 36 | $A < C < B$ | $C < B < A$ | $B < C < A$ | $C < A < B$ | $A < C < B$ |
| 37 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < A < C$ | $A < B < C$ |
| 38 | $B < A < C$ | $A < B < C$ | $A < C < B$ | $B < A < C$ | $A < B < C$ |
| 39 | $C < B < A$ | $C < B < A$ | $B < C < A$ | $A < C < B$ | $A < C < B$ |
| 40 | $B < C < A$ | $C < B < A$ | $C < A < B$ | $B < C < A$ | $B < C < A$ |

**Table F.3.** Predicted sort results for Fold3.

| Landmark scene | GP | Xing (2012) | Lyu, Yu, and Meng (2015) | Joachims (2002) | Ground truth |
|---|---|---|---|---|---|
| 1 | $B < A < C$ | $A < C < B$ | $A < C < B$ | $C < B < A$ | $B < A < C$ |
| 2 | $A < C < B$ | $B < C < A$ | $C < B < A$ | $B < C < A$ | $A < B < C$ |
| 3 | $C < B < A$ | $C < A < B$ | $A < C < B$ | $B < C < A$ | $B < C < A$ |
| 4 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $C < A < B$ | $A < C < B$ |
| 5 | $C < A < B$ | $A < B < C$ | $B < A < C$ | $A < C < B$ | $C < A < B$ |
| 6 | $A < C < B$ | $C < B < A$ | $B < C < A$ | $A < C < B$ | $A < B < C$ |
| 7 | $B < A < C < D$ | $A < C < B < D$ | $A < D < C < B$ | $B < A < D < C$ | $B < A < D < C$ |
| 8 | $A < B < C$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $A < B < C$ |
| 9 | $A < B < C$ | $A < B < C$ | $A < B < C$ | $B < A < C$ | $B < C < A$ |
| 10 | $B < A < C$ | $A < B < C$ | $B < C < A$ | $A < C < B$ | $B < A < C$ |
| 11 | $C < A < B$ | $B < A < C$ | $B < A < C$ | $A < C < B$ | $C < A < B$ |
| 12 | $A < C < B$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $A < B < C$ |
| 13 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $C < B < A$ | $A < B < C$ |
| 14 | $C < B < A$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $C < B < A$ |
| 15 | $A < C < B$ | $B < C < A$ | $B < C < A$ | $A < C < B$ | $A < B < C$ |
| 16 | $C < A < B$ | $B < A < C$ | $B < A < C$ | $A < C < B$ | $C < A < B$ |
| 17 | $B < C < A$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $A < B < C$ |
| 18 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $C < A < B$ | $A < B < C$ |
| 19 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $B < A < C$ | $A < B < C$ |
| 20 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $A < B < C$ |
| 21 | $A < C < B$ | $C < B < A$ | $C < B < A$ | $C < A < B$ | $A < B < C$ |
| 22 | $D < B < A < C$ | $A < C < D < B$ | $C < A < D < B$ | $A < B < D < C$ | $D < A < C < B$ |
| 23 | $A < B < C < D$ | $B < C < A < D$ | $B < C < A < D$ | $B < A < C < D$ | $A < B < D < C$ |
| 24 | $D < B < A < C$ | $C < A < D < B$ | $C < A < D < B$ | $B < D < A < C$ | $B < A < D < C$ |
| 25 | $B < A < C$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $B < A < C$ |
| 26 | $C < D < A < B$ | $B < C < D < A$ | $B < C < D < A$ | $D < C < A < B$ | $C < B < A < D$ |
| 27 | $C < A < B$ | $A < C < B$ | $A < C < B$ | $C < A < B$ | $C < B < A$ |
| 28 | $B < A < C$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $A < B < C$ |
| 29 | $C < A < B$ | $B < A < C$ | $B < A < C$ | $A < B < C$ | $C < A < B$ |
| 30 | $A < C < B$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $B < A < C$ |
| 31 | $B < C < A$ | $B < A < C$ | $A < B < C$ | $B < A < C$ | $B < C < A$ |
| 32 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $A < C < B$ | $A < B < C$ |
| 33 | $B < C < A$ | $B < A < C$ | $B < A < C$ | $B < C < A$ | $B < A < C$ |
| 34 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $C < B < A$ | $A < B < C$ |
| 35 | $C < A < B$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $A < B < C$ |
| 36 | $B < C < A$ | $A < C < B$ | $A < C < B$ | $A < B < C$ | $A < C < B$ |
| 37 | $C < A < B$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $A < C < B$ |
| 38 | $C < A < B$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $C < B < A$ |
| 39 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $A < B < C$ | $A < C < B$ |
| 40 | $A < C < B$ | $B < C < A$ | $C < B < A$ | $B < C < A$ | $A < B < C$ |

**Table F.4.** Predicted sort results for Fold4.

| Landmark scene | GP | Xing (2012) | Lyu, Yu, and Meng (2015) | Joachims (2002) | Ground truth |
|---|---|---|---|---|---|
| 1 | $C < A < B$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $C < A < B$ |
| 2 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $B < C < A$ |
| 3 | $A < B < C$ | $A < B < C$ | $A < C < B$ | $B < A < C$ | $A < B < C$ |
| 4 | $A < B < C$ | $B < C < A$ | $B < C < A$ | $B < A < C$ | $A < B < C$ |
| 5 | $C < A < B$ | $C < B < A$ | $C < B < A$ | $C < A < B$ | $A < B < C$ |
| 6 | $A < B < C$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $A < B < C$ |
| 7 | $C < A < B$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $C < A < B$ |
| 8 | $A < C < B$ | $C < B < A$ | $C < B < A$ | $A < C < B$ | $A < B < C$ |
| 9 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $B < A < C$ |
| 10 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < A < C$ | $A < C < B$ |
| 11 | $B < A < C$ | $B < A < C$ | $B < A < C$ | $C < B < A$ | $A < B < C$ |
| 12 | $A < B < C$ | $A < C < B$ | $A < C < B$ | $B < A < C$ | $A < B < C$ |
| 13 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $B < C < A$ |
| 14 | $B < D < A < C$ | $A < C < D < B$ | $A < C < D < B$ | $A < B < D < C$ | $B < A < C < D$ |
| 15 | $A < B < C$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $A < C < B$ |
| 16 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $A < C < B$ | $B < A < C$ |
| 17 | $D < A < C < B$ | $A < C < B < D$ | $A < C < B < D$ | $D < A < C < B$ | $D < C < B < A$ |
| 18 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $B < A < C$ | $A < B < C$ |
| 19 | $B < A < C < D$ | $D < C < B < A$ | $D < A < C < B$ | $A < C < B < D$ | $A < B < D < C$ |
| 20 | $B < A < C$ | $A < C < B$ | $C < A < B$ | $B < A < C$ | $B < A < C$ |
| 21 | $C < B < A$ | $B < C < A$ | $B < C < A$ | $C < B < A$ | $C < B < A$ |
| 22 | $A < B < C$ | $A < B < C$ | $B < A < C$ | $C < A < B$ | $A < B < C$ |
| 23 | $B < C < A$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $B < A < C$ |
| 24 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $B < C < A$ | $A < C < B$ |
| 25 | $B < A < C$ | $B < C < A$ | $B < C < A$ | $B < A < C$ | $A < B < C$ |
| 26 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $C < A < B$ | $A < C < B$ |
| 27 | $A < C < B$ | $C < A < B$ | $A < C < B$ | $C < A < B$ | $A < C < B$ |
| 28 | $C < B < A < D$ | $C < A < B < D$ | $C < A < D < B$ | $C < B < A < D$ | $B < A < C < D$ |
| 29 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $A < C < B$ | $A < B < C$ |
| 30 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $A < B < C$ | $A < B < C$ |
| 31 | $C < A < B$ | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < A < C$ |
| 32 | $A < B < C$ | $B < A < C$ | $C < B < A$ | $A < B < C$ | $A < B < C$ |
| 33 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $C < A < B$ | $B < A < C$ |
| 34 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $A < C < B$ | $A < C < B$ |
| 35 | $C < B < A$ | $B < A < C$ | $B < A < C$ | $C < A < B$ | $B < C < A$ |
| 36 | $B < C < A$ | $B < C < A$ | $B < C < A$ | $A < B < C$ | $B < A < C$ |
| 37 | $A < B < C$ | $B < A < C$ | $B < A < C$ | $B < C < A$ | $A < B < C$ |
| 38 | $C < B < A$ | $C < A < B$ | $C < A < B$ | $C < A < B$ | $C < B < A$ |
| 39 | $A < B < C$ | $A < C < B$ | $A < C < B$ | $B < A < C$ | $B < A < C$ |
| 40 | $B < D < A < C$ | $B < C < A < D$ | $B < C < A < D$ | $B < A < C < D$ | $A < B < D < C$ |

**Table F.5.** Predicted sort results for Fold5.

| Landmark scene | GP | Xing (2012) | Lyu, Yu, and Meng (2015) | Joachims (2002) | Ground truth |
|---|---|---|---|---|---|
| 1 | $B < C < D < A$ | $B < C < A < D$ | $B < C < A < D$ | $B < A < C < D$ | $A < B < D < C$ |
| 2 | $D < A < B < C$ | $D < B < C < A$ | $D < A < B < C$ | $A < B < C < D$ | $B < D < A < C$ |
| 3 | $C < B < A$ | $A < C < B$ | $A < C < B$ | $C < B < A$ | $C < B < A$ |
| 4 | $A < B < C$ | $A < B < C$ | $A < B < C$ | $B < C < A$ | $A < B < C$ |
| 5 | $B < A < D < C$ | $A < B < D < C$ | $A < B < D < C$ | $B < C < D < A$ | $B < A < C < D$ |
| 6 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $A < B < C$ |
| 7 | $C < B < A$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $C < B < A$ |
| 8 | $B < A < C$ | $C < A < B$ | $C < A < B$ | $A < C < B$ | $A < B < C$ |
| 9 | $B < C < A$ | $A < C < B$ | $A < C < B$ | $C < B < A$ | $B < C < A$ |
| 10 | $B < D < A < C$ | $C < D < B < A$ | $C < D < B < A$ | $B < A < D < C$ | $B < A < C < D$ |
| 11 | $A < C < B$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $A < C < B$ |
| 12 | $B < C < A$ | $A < B < C$ | $A < C < B$ | $B < C < A$ | $C < B < A$ |
| 13 | $B < C < A$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $B < A < C$ |
| 14 | $C < A < B$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $C < A < B$ |
| 15 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $A < C < B$ | $A < B < C$ |
| 16 | $A < C < B$ | $B < C < A$ | $B < C < A$ | $C < A < B$ | $A < C < B$ |
| 17 | $A < C < B$ | $C < B < A$ | $C < B < A$ | $B < A < C$ | $C < A < B$ |
| 18 | $A < B < C$ | $B < A < C$ | $B < A < C$ | $A < C < B$ | $A < B < C$ |
| 19 | $C < A < B$ | $A < B < C$ | $A < B < C$ | $C < B < A$ | $C < B < A$ |
| 20 | $D < A < C < B$ | $D < A < C < B$ | $D < A < C < B$ | $A < B < C < D$ | $D < A < C < B$ |
| 21 | $A < C < B$ | $A < B < C$ | $A < B < C$ | $A < C < B$ | $A < C < B$ |
| 22 | $A < C < B < D$ | $A < B < D < C$ | $A < B < D < C$ | $A < B < C < D$ | $A < C < B < D$ |
| 23 | $A < B < C$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $A < B < C$ |
| 24 | $A < C < B$ | $B < A < C$ | $B < A < C$ | $A < C < B$ | $A < B < C$ |
| 25 | $B < A < C$ | $B < C < A$ | $B < A < C$ | $B < C < A$ | $B < A < C$ |
| 26 | $A < B < C$ | $A < B < C$ | $A < B < C$ | $A < B < C$ | $A < B < C$ |
| 27 | $B < C < A$ | $B < C < A$ | $B < C < A$ | $B < C < A$ | $B < A < C$ |
| 28 | $B < C < A$ | $B < A < C$ | $B < A < C$ | $B < C < A$ | $B < A < C$ |
| 29 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $C < A < B$ | $B < A < C$ |
| 30 | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < C < A$ | $B < A < C$ |
| 31 | $A < C < B$ | $C < A < B$ | $C < B < A$ | $A < C < B$ | $A < B < C$ |
| 32 | $A < C < D < B$ | $D < A < BC$ | $A < D < B < C$ | $C < B < D < A$ | $A < B < C < D$ |
| 33 | $A < C < B$ | $C < B < A$ | $B < C < A$ | $A < C < B$ | $A < C < B$ |
| 34 | $B < C < A$ | $A < C < B$ | $A < C < B$ | $A < C < B$ | $B < C < A$ |
| 35 | $A < C < B$ | $C < A < B$ | $C < A < B$ | $A < B < C$ | $A < B < C$ |
| 36 | $C < A < B$ | $A < B < C$ | $A < B < C$ | $C < A < B$ | $C < A < B$ |
| 37 | $B < A < C$ | $C < B < A$ | $C < B < A$ | $A < B < C$ | $B < A < C$ |
| 38 | $B < A < C$ | $B < A < C$ | $B < C < A$ | $A < B < C$ | $A < B < C$ |
| 39 | $A < B < C$ | $C < A < B$ | $C < B < A$ | $A < B < C$ | $B < A < C$ |
| 40 | $B < C < A$ | $B < C < A$ | $B < C < A$ | $A < B < C$ | $B < A < C$ |

**Declaration of Authorship**

**Paper Title:** Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints

**Appeared in:** Hu, X., Fan, H., Noskov, A. (2017). Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints. International Journal of Digital Earth, 11(10), pp.1039-1063.

**Contribution Statement:** Xuke Hu has designed the approach, implemented the algorithm, conducted the experiment, and performed the analysis for this publication. He also wrote the major part of this publication. Hongchao Fan provided valuable suggestions to improve the approach, and made great enhancement during the writing process. He also had much contribution to the paper submission. Alexey Noskov proofread the manuscript and provided valuable feedbacks which led to substantial improvements of the paper.

The estimated percentage of the dissertation author's contribution is 80%.

| Date | Name | Signature |
|---|---|---|
|  | Xuke Hu<br>(First author) | Xuke Hu |
|  | Alexander Zipf<br>(Supervisor) | A. Zipf |
|  | Alexey Noskov<br>(Coauthor) | (Alexey Noskov) |

## Declaration of Authorship

**Paper Title:** Tagging the buildings' main entrance based on OpenStreetMap and binary imbalanced learning

**Appeared in:** Hu, X., Noskov, A., Novack, T., Fan, H., Gu, F., Li, H., Shang, J. (2020). Tagging the buildings' main entrance based on OpenStreetMap and binary imbalanced learning. International Journal of Geographical Information Science. (Reviewed).

**Contribution Statement:** Xuke Hu has designed the approach, implemented the algorithm, conducted the experiment, and performed the analysis for this publication. He also wrote the major part of the paper. Alexey Noskov prepared the experimental data and provided valuable suggestions and comments during the initial setup of the experiment. Tessio Noskov discussed the solution and proofread the manuscript. Fuqiang Gu, Hao Li, and Jianga Shang have offered many comments and made substantial improvements during the writing process. Hongchao Fan provided valuable feedbacks which led to substantial improvements of the paper.

The estimated percentage of the dissertation author's contribution is 80%.

| Date | Name | Signature |
|---|---|---|
| | Xuke Hu<br>(First author) | Xuke Hu |
| | Alexander Zipf<br>(Supervisor) | A. Zipf |
| | Alexey Noskov<br>(Coauthor) | (Alexey Noskov) |

**Declaration of Authorship**

**Paper Title:** Feasibility of using grammars to infer room semantics

**Appeared in:** Hu, X., Fan, H., Noskov, A., Zipf, A., Wang, Z., Shang, J. (2019). Feasibility of using grammars to infer room semantics. Remote Sensing. vol. 11(13), p.1535. Doi: 10.3390/rs11131535.

**Contribution Statement:** Xuke Hu has designed the approach, implemented the algorithm, conducted the experiment, and performed the analysis for this publication. He also wrote the major part of the paper. Hongchao Fan provided valuable ideas which led to the positive outcomes of the paper. Alexey Noskov and Zhiyong Wang have provided valuable suggestions and comments during the initial setup of the experiment, the whole writing process and the revisions afterwards. Alexander Zipf and Jianga Shang have offered many comments and made substantial improvements to the paper.

The estimated percentage of the dissertation author's contribution is 85%.

| Date | Name | Signature |
|---|---|---|
|  | Xuke Hu<br>(First author) | XuKe Hu |
|  | Alexander Zipf<br>(Supervisor) | A. Zipf |
|  | Alexey Noskov<br>(Coauthor) | (Alexey Noskov) |

**Declaration of Authorship**

**Paper Title:** Room semantics inference using random forest and relational graph convolutional network: a case study of research building

**Appeared in:** Hu, X., Fan, H., Noskov, A., Wang, Z., Zipf, A., Gu, F., Shang, J. (2020). Room semantics inference using random forest and relational graph convolutional network: a case study of research building. Transactions in GIS. (reviewed)

**Contribution Statement:** Xuke Hu has designed the approach, implemented the algorithm, conducted the experiment, and performed the analysis for this publication. He also wrote the major part of the paper. Hongchao Fan and Alexey Noskov provided the valuable ideas of the study, which led to the positive outcome of the paper. Zhiyong Wang and Fuqiang Gu have provided valuable suggestions and comments during the initial setup of the experiment, the whole writing process and the revisions afterwards. Alexander Zipf and Jianga Shang have offered many comments and made substantial improvements to the paper.

The estimated percentage of the dissertation author's contribution is 85%.

| **Date** | **Name** | **Signature** |
|---|---|---|
| | Xuke Hu<br>(First author) | Xuke Hu |
| | Alexander Zipf<br>(Supervisor) | A. Zipf |
| | Alexey Noskov<br>(Coauthor) | (Alexey Noskov) |

## Declaration of Authorship

**Paper Title:** Data-driven approach to learning salience models of indoor landmarks by using genetic programming

**Appeared in:** Hu, X., Ding, L., Shang, J., Fan, H., Novack, T., Noskov, A., Zipf, A. (2020). Data-driven approach to learning salience models of indoor landmarks by using genetic programming. International Journal of Digital Earth, 1-28.

**Contribution Statement:** Xuke Hu has designed the approach and performed the analysis for this publication. He also wrote the major part of the paper. Lei Ding implemented the algorithm and conducted the experiments under the guidance of Xuke Hu. Jianga Shang provided the valuable ideas which led to the positive outcome of the paper. Tessio Noskov has provided valuable suggestions and comments during the whole writing process and the revisions afterwards. Alexander Zipf, Hongchao Fan, and Alexey Noskov have offered many comments and made substantial improvements to the paper.

The estimated percentage of the dissertation author's contribution is 75%.

| Date | Name | Signature |
|---|---|---|
| | Xuke Hu<br>(First author) | Xuke Hu |
| | Alexander Zipf<br>(Supervisor) | A. Zipf |
| | Alexey Noskov<br>(Coauthor) | (Alexey Noskov) |

# *Eidesstattliche Versicherung gemäß § 8 der Promotionsordnung der Naturwissenschaftlich-Mathematischen Gesamtfakultät der Universität Heidelberg*

1. Bei der eingereichten Dissertation zu dem Thema ‘‘*Building Semantics Reasoning by Using Rules based on Available Geospatial Information*” handelt es sich um meine eigenständig erbrachte Leistung.

2. Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtlich oder sinngemäß aus anderen Werken übernommene Inhalte als solche kenntlich gemacht.

3. Die Arbeit habe ich bislang nicht an einer Hochschule des In- oder Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

4. Die Richtigkeit der vorstehenden Erklärungen bestätige ich.

5. Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt.

Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Jena, 27 May 2020

Xuke Hu