

COMBINED GRAMMAR FOR THE MODELING OF BUILDING INTERIORS

S. Becker^{a,*}, M. Peter^a, D. Fritsch^a, D. Philipp^b, P. Baier^b, C. Dibak^b

^a Institute for Photogrammetry, University of Stuttgart, Germany - first.last@ifp.uni-stuttgart.de

^b Institute of Parallel and Distributed Systems, University of Stuttgart, Germany – first.last@ipvs.uni-stuttgart.de

Commission IV, WG IV/7

KEY WORDS: Building, Modeling, Abstraction, Prediction, Automation

ABSTRACT:

As spatial grammars have proven successful and efficient to deliver LOD3 models, the next challenge is their extension to indoor applications, leading to LOD4 models. Therefore, a combined indoor grammar for the automatic generation of indoor models from erroneous and incomplete observation data is presented. In building interiors where inaccurate observation data is available, the grammar can be used to make the reconstruction process robust, and verify the reconstructed geometries. In unobserved building interiors, the grammar can generate hypotheses about possible indoor geometries matching the style of the rest of the building. The grammar combines concepts from L-systems and split grammars. It is designed in such way that it can be derived from observation data fully automatically. Thus, manual predefinitions of the grammar rules usually required to tune the grammar to a specific building style, become obsolete. The potential benefit of using our grammar as support for indoor modeling is evaluated based on an example where the grammar has been applied to automatically generate an indoor model from erroneous and incomplete traces gathered by foot-mounted MEMS/IMU positioning systems.

1. INTRODUCTION

Digital representations of building interiors can support daily life in many fields. Beside the support of indoor navigation and positioning, possible applications are emergency services, crowd management, architectural planning, simulations etc. The geometric detail and the semantic content of existing indoor models strongly depend on the application area for which the models have been created. In the world of Building Information Modeling (BIM), 3D indoor models are the geometric basis for the representation of physical and functional characteristics and processes. The models, which are in general created manually, feature high geometric detail and are equipped with rich semantic information. In the GIS world, in contrast, the vast majority of 3D indoor representations are pure geometry models with limited geometric detail. This is owed to the fact that, here, the focus is on the automatic derivation of indoor geometries from observation data. Typical problems arising in this context – in particular when using low-cost sensors – are due to erroneous or incomplete observations. However, building interiors are subject to numerous geometric and topological conditions: inner walls are often rectangular or parallel to outer walls; rooms can be adjacent but not overlapping etc. These conditions can be used to support the reconstruction of interiors. Powerful means to facilitate this are formal grammars. The huge potential of formal grammars lies in the compact rule-based description of object knowledge, and the generative part which allows for efficient procedural modeling approaches.

We are striving for a fully automatic approach for grammar-supported indoor modeling based on observation data from low-cost sensors. For this purpose, we developed a formal grammar which is able to store geometric, topological and semantic information on building interiors. The grammar is flexible as it is designed in such way that it can be automatically derived

from observations. This enables the grammar to be integrated in a continuous update and enhancement loop: An initial grammar derived from currently available observations is used to generate hypotheses about possible indoor geometries. As soon as new observations are available, the previously generated hypotheses can be checked against them, and the new observations can be used to automatically update the initial grammar. Continuing this loop finally results in a high-level grammar which is able to generate indoor geometries of high reliability. In this paper, we will present our indoor grammar and show a first application to real world data.

The paper is structured as follows: Section 2 gives a brief overview of related work. The definition of our indoor grammar is given in section 3. Section 4 describes how the grammar can be applied, and presents first results. The paper ends with conclusions and an outlook on future work in section 5.

2. RELATED WORK

The modeling of building interiors is mostly based on active sensing technologies like e.g. robot-based laser scanning (Fietz et al. 2010) or the Microsoft Kinect (Newcombe et al. 2011). These sensors produce un-interpreted, possibly erroneous point clouds. In order to derive semantically enhanced boundary representation models from this data, many approaches build on sets of constraints. (Budroni & Boehm 2010), for instance, employ the well-known Manhattan World constraints, parallelism and rectangularity, for this purpose.

In contrast to these low-level general constraints, an efficient and compact option to formalize high-level object knowledge is given by formal grammars. Generally, in a formal grammar, object knowledge is represented through *symbols* and a set of

* Corresponding author.

production or *replacement rules*. Originally developed for the formal description of mathematical logic and reasoning processes, formal grammars can also be applied successfully for modeling geometric structures. (Prusinkiewicz & Lindenmayer 1990) focus on line structures. They simulate growth processes of plants through so-called Lindenmayer systems (L-systems). (Parish & Müller 2001) applied these concepts to the modeling of streets and 3D building models. While streets can be modeled very efficiently by L-systems, problems arise with modeling buildings: A building represents rather an iterative partitioning of available space than an unrestricted growing process. Furthermore, geometric conditions are difficult to integrate and lead to long and intricate production rules.

Grammars which are more appropriate to model architectural structures are *split grammars*. A split grammar uses split rules to partition an initially simple shape into more complex ones. (Wonka et al. 2003) proposed a split grammar for façade reconstruction. (Müller et al. 2006) continued the development of split grammars. Their CGA Shape Grammar allows for the procedural generation of highly detailed 3D building models. A general problem of the grammars mentioned so far is that the production rules have to be set up manually which is very time-consuming and requires expert knowledge. Therefore, some approaches follow the principle of inverse procedural modeling, i.e., they try to automatically derive rules from observations. Examples are given by (Müller et al. 2007; Becker 2009; Bokeloh et al. 2010; Van Gool et al. 2013).

For the modeling of interiors, grammars are still very new. To our knowledge, (Gröger & Plümer 2010) are the only ones who proposed a grammar for generating complete 3D indoor models. The grammar contains mainly split rules which are applied to boxes and faces. The rules have to be predefined manually, and it is not possible to automatically adapt the grammar to observation data for modeling real interiors.

In contrast, we are striving for an indoor grammar that can be automatically derived from observation data and, thus, is flexible to support the reconstruction of arbitrary building interiors. This requires that the grammar is able to understand and represent the way, people explore a building. We realize that by combining concepts from the L-systems and the split grammars.

3. COMBINED INDOOR GRAMMAR

The first step in developing a formal grammar for the generation of indoor models is to identify basic geometric primitives and characteristic topologic properties by which an indoor model can be described. Then, this information has to be transferred into non-terminals, terminals and production rules – the basic components of a formal grammar. To be flexible towards the huge variety of different indoor designs, we are striving for a generic template grammar which can be automatically adapted to individual buildings based on observation data.

Next, our grammar concept is introduced in more detail: Section 3.1 explores general characteristics of interiors, and motivates the conceptual design of our indoor grammar. After a formal definition of the grammar in section 3.2, section 3.3 discusses how this generic grammar can be instantiated to individual grammars representing specific building styles.

3.1 Design Decisions

Typically, the geometric structure of building interiors is organized through a horizontal partitioning of the building's body into *floors*, and a vertical partitioning of each floor into *rooms*. While the arrangement of floors is linear showing a one-dimensional topology, the topology of room configurations within a floor is two-dimensional and, therefore, much more complex. However, such arrangements are not the result of establishing walls in a floor randomly, but instead follow architectural principles and are subject to functional restrictions. In order to provide as much support as possible to indoor modeling, our grammar is designed to reflect such knowledge. The following properties of building interiors are crucial for our grammar design: (1) To ensure convenient access to the rooms, buildings are usually traversed by a system of hallways. (2) The system of hallways divides each floor into *hallway-spaces* and *non-hallway-spaces*. Non-hallway-spaces can be further divided into smaller *room units* which are mostly arranged in a linear sequence parallel to the adjacent hallway.

Properties 1 and 2 give reasons for pursuing two different modeling strategies: The course of the hallways, on the one hand, reminds of a network-like propagation of linear structures. The layout of the rooms, on the other hand, can be efficiently generated by a spatial partitioning applied to the interspaces of the hallway network. The analogy to 3D city modeling becomes apparent: The network of hallways corresponds to the network of streets, and the partitioning of non-hallway-spaces into rooms is comparable to the split of simple shapes into more complex ones for the generation of 3D building models. This analogy motivates the adaption of grammar-based concepts used for city modeling. Similar to (Parish & Müller 2001) who developed a L-system for modeling streets, we use a specially designed L-system for the formal description of the hallway network (section 3.2.1). For the modeling of room layouts we develop a separate grammar which is mainly based on split operations (section 3.2.2).

3.2 Grammar Definition

Taking into account the different characteristics of hallway and room layouts, our indoor grammar $G^{\text{indoor}}=(G^{\text{hallways}}, G^{\text{rooms}})$ is composed of two grammars addressing the modeling of hallways and rooms separately. Together, they allow for the complete formal description of building interiors. The grammar for modeling hallways bases on an enriched L-system. Although this part is still ongoing work, we would like to shortly present the current state of our development in section 3.2.1. The definition of the split grammar which is used to model room configurations will be given in section 3.2.2.

3.2.1 L-System for Modeling Hallways

The design of the L-system $G^{\text{hallways}}=(V,\omega,P)$ comprising a set V of attributed symbols also called *modules*, the axiom ω , and the production rules P , follows very closely the approach for modeling streets by (Parish & Müller 2001). Their concept of an enriched L-system meets our requirements of a template-like grammar which can be easily adapted to observation data. The basic idea is to organize the setting of attributes (e.g. length and orientation of linear structures), probabilities as well as constraints (induced by the geometric environment) through *external functions*. Thus, all variable components are uncoupled from the generative part of the grammar, i.e. the production rules, which can stay fix after being defined once.

Similar to (Parish & Müller 2001) we distinguish two external functions which, together, fully control the behavior of the L-system. The external function *LayoutSetting* controls the growth of the hallway system at a global level. By setting the attributes of the generic hallway modules, this function effects that the hallways follow a specific overall layout pattern. Since the attributes are determined due to their probabilities, different layouts can be produced, however, in the same overall style. The external function *ConsistencyConstraints* controls the behavior of the hallway system on a local level. It ensures that the L-system is environmentally sensitive and self-sensitive. To obtain environmental sensitivity, the function checks the hallways as proposed by the external function *LayoutSetting* against the given exterior building shell. Geometric conflicts, e.g. caused by a hallway breaking through the building shell, are tried to be resolved by fitting the hallway's length and orientation to its local environment. Hallways which cannot be inserted at all are removed. To obtain self-sensitivity, *ConsistencyConstraints* checks the proposed hallways against already existing hallways: Intersections are identified and short dead-ends are removed; hallways which are slightly too short to intersect are extended.

The generative part consists of five production rules, which are an adaptation of the rules suggested by (Parish & Müller 2001):

- ω : $\mathbf{R}(\rho_{init})? \mathbf{I}(\theta_{init}, UNASSIGNED)$
- p_1 : $\mathbf{R}(\rho) > ? \mathbf{I}(\theta, state) : state == SUCCEED$
 $\{ \mathbf{LayoutSetting}(\rho, \theta) \text{ sets } \rho_p[0-2], \theta_p[0-2] \} \rightarrow +(\theta, angle) \mathbf{F}(\theta, len) \mathbf{B}(\rho_p[1], \theta_p[1]) \mathbf{B}(\rho_p[2], \theta_p[2])$
 $\mathbf{R}(\rho_p[0])? \mathbf{I}(\theta_p[0], UNASSIGNED)$
- p_2 : $\mathbf{R}(\rho) > ? \mathbf{I}(\theta, state) : state == FAILED \rightarrow \epsilon$
- p_3 : $\mathbf{B}(\rho, \theta) \rightarrow [\mathbf{R}(\rho)? \mathbf{I}(\theta, UNASSIGNED)]$
- p_4 : $? \mathbf{I}(\theta, state) : state == UNASSIGNED$
 $\{ \mathbf{ConsistencyConstraints}(\theta) \text{ adjusts } state, \theta \} \rightarrow ? \mathbf{I}(\theta, state)$
- p_5 : $? \mathbf{I}(\theta, state) : state != UNASSIGNED \rightarrow \epsilon$

The axiom ω comprises a rule module R and a query insertion module $?I$ which together represent a hallway segment. The first three productions address the R module: p_1 controls the hallway and branch creation. The hallways' attributes are set by the external function *LayoutSetting* according to a specific layout pattern. p_2 deletes a R module if *ConsistencyConstraints* has failed to fit the corresponding hallway into its geometric environment. p_3 generates a new hallway segment at a branch position. The final two rules adjust the attributes previously set by *LayoutSetting*. If *ConsistencyConstraints* is successful in fitting a hallway into its environment, the variable $state$ is set to SUCCEED and the hallway can be created. The resulting hallways partition the 3D building model into *hallway spaces* and *non-hallway spaces*.

3.2.2 Split-Grammar for Modeling Rooms

The split grammar for establishing room configurations in non-hallway-spaces can be written as a four-tuple $G^{rooms} = (N, T, S, R)$ consisting of the non-terminals N , the terminals T , the axiom S , and the production rules R .

The non-terminals and terminals of our grammar correspond to basic geometric primitives. The set of non-terminals $N = \{Space\}$ consists of the single element *Space* representing an arbitrary

3D solid which can be further divided. The axiom $S = Space$ stands for the empty non-hallway space which is to be partitioned into rooms. The terminals T describe solids that are not divisible any further. To distinguish from non-terminals, terminal symbols *space* start with lower case. Both non-terminals and terminals have attributes. They determine the space's geometric extent and probability. The production rules R are defined as replacement rules that perform a split, a merge or an instantiation. A split divides a *Space* into two *Space* elements along a partition plane. A merge is the inverse operation combining two adjacent *Space* elements to one. The application of split and merge rules follows the principle of cell decomposition (Kada 2007) which automatically provides knowledge about neighborhood relationships between the spaces. Also, topologically correct reconstruction results are ensured.

In total, we distinguish six rule types (see also Figure 2):

- $\mathbf{R}_i^{SingleSplit} : Space \rightarrow Split^{Space}(\alpha_i, d_i)$
 with $Split^{Space}(\alpha_i, d_i) := Split_i^{Space} = Space^l Space^r$
- $\mathbf{R}_i^{RepeatSplit} : Space \rightarrow Split_i^{Space^r} \circ \dots$
 $\dots \circ Split_i^{Space^r} \circ Split_i^{Space}$
- $\mathbf{R}_{ij\dots k}^{StringSplit} : Space \rightarrow Split_k^{Space^r} \circ \dots$
 $\dots \circ Split_j^{Space^r} \circ Split_i^{Space}$
- $\mathbf{R}_{ab\dots c}^{MultiSplit} : Space \rightarrow Split_c^* \circ \dots \circ Split_b^* \circ Split_a^{Space}$
 with $*$ = $Space \in$ previously generated Spaces
- $\mathbf{R}^{Merge} : Space^l Space^r \rightarrow Merge^{Space^l, Space^r}$
 with $Merge^{Space^l, Space^r} = Space^l \cup Space^r$
- $\mathbf{R}^{Instantiation} : Space \rightarrow space$

Rule type *SingleSplit* performs a single split operation by replacing the non-terminal *Space* by a left and a right non-terminal $Space^l$ and $Space^r$. These are the result of the function $Split^{Space}(\alpha, d_i)$ or $Split_i^{Space}$ for short. The superscript "Space" denotes that the split operation is applied to the non-terminal *Space*. Orientation and position of the corresponding partition plane is described by the rule parameters: orientation angle α and the distance value d . α and d refer to a local coordinate system which is based on the *Space* to split. Figure 1 depicts the situation for two exemplary partition planes p and q . For the sake of clarity, the space is illustrated from above in 2D.

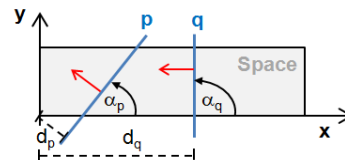


Figure 1. Rule parameters of a *SingleSplit*

Rule types *RepeatSplit*, *StringSplit* and *MultiSplit* can be used to group functionally related rooms to superior *room units*. Usually, rooms are arranged due to their functionality: Rooms with a strong semantic relationship are close to each other and build a unit. For example, in hotels or hospitals a typical room

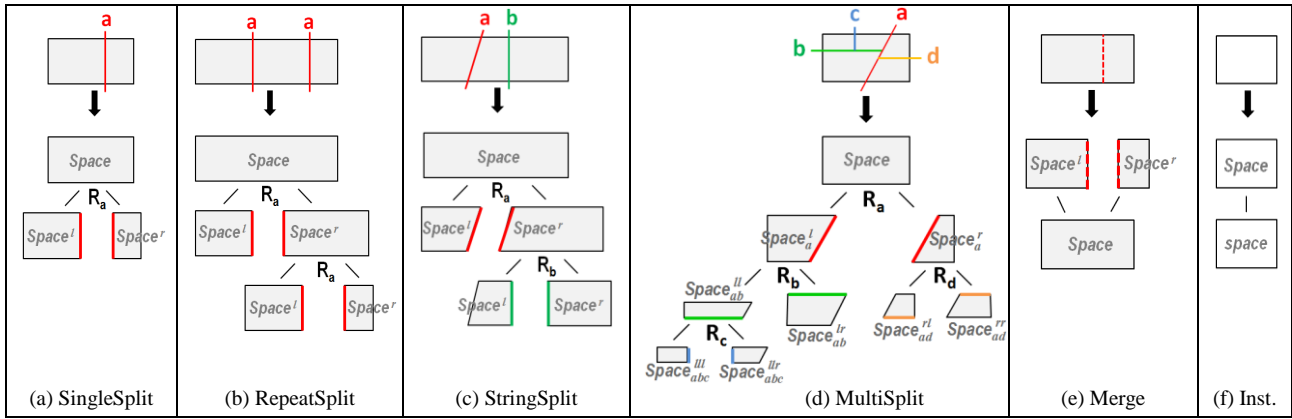


Figure 2. Different rule types and their geometric interpretation

unit consists of a bedroom and a bathroom. Mathematically, such a grouping can be expressed by a concatenation of several *SingleSplits*. When each *SingleSplit* is applied to the right non-terminal *Space^r* produced by the previous *SingleSplit*, the result is a linear sequence of rooms: While *RepeatSplit* generates a sequence of identical room units by repeating a single split operation, *StringSplit* is able to produce a sequence of different room units. Split operations which are applied for modeling non-linear room layouts can be aggregated within the rule type *MultiSplit*. In this case, the split operations can be applied to any of the previously generated Spaces. Based on simple examples, Figure 2 shows how the four split rules, the merge and the instantiation rule can be interpreted geometrically. For sake of brevity, here, a *SingleSplit* $R_i^{SingleSplit}$ is written as R_i .

The probability of the rules is described by an *a priori probability*, and a *context aware probability*. The a priori probability $P(R_i)$ of the rule R_i is the rule's relative frequency of occurrence. The context aware probability $P(R_j|R_i)$ is a conditional probability which models neighborhood relationships between rooms. For example, $P(R_j|R_i)=0.5$ states that with a probability of 50% rule R_j follows rule application R_i . We implement these probabilities by means of a Markov chain. The nodes of the Markov chain represent the rules. Edges describe neighborhood relationships or transitions between different rules. The probability for a transition from R_i to R_j is given by $(P(R_j)/P(R_i)) \cdot (P(R_i | R_j)/P(R_j | R_i))$.

3.3 Instantiation of Individual Grammars

Based on the grammar defined in section 3.2, an arbitrary building can be transferred to a specialized rule system which contains detailed knowledge about the construction of the building's interior. As an example, Figure 3 shows how such a specialized rule system can be used to express a floor plan of a real building in a formal way. As can be seen, the grammar is flexible enough to generate even complex indoor layouts: The L-system is not restricted to model rectangular or parallel hallways. Instead, it allows the hallway segments to enclose arbitrary angles. Furthermore, few sequences of split and merge rules are sufficient to model non-trivial room layouts showing a mixture of linear string-like arrangements (*red partition planes*) and configurations which follow a two-dimensional topology (*green partition planes*). For lack of space within this paper, geometric parameters, which are part of the rules' attributes (e.g. the width and length of hallway segments, or the orientation and position of split planes), are not given here.

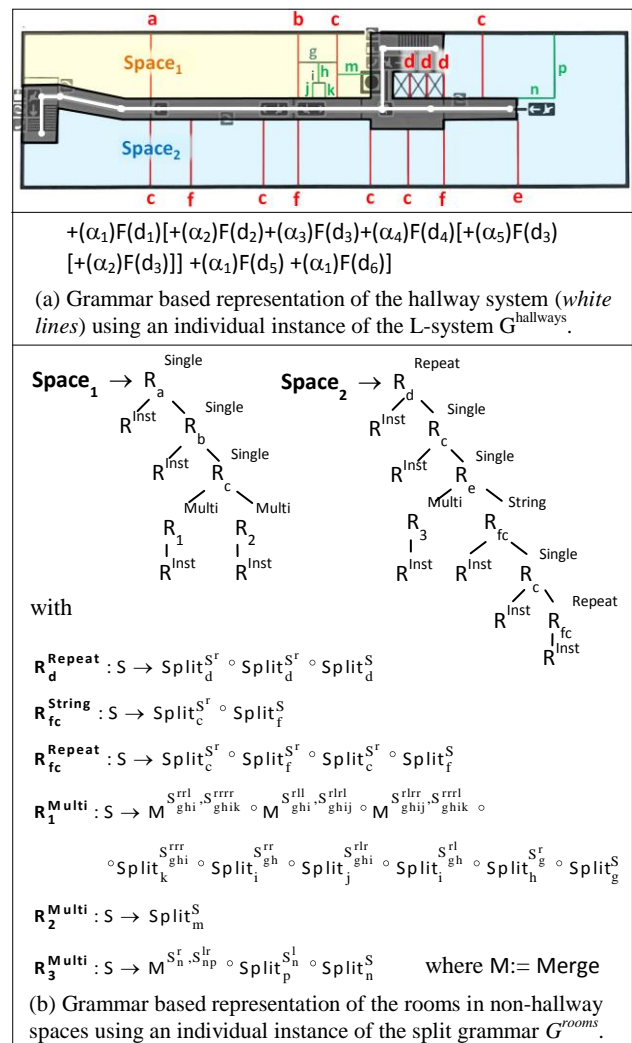


Figure 3. Real floor plan and its grammar-based representation

In order to create such an instance of an individual grammar, the rules as well as their attributes have to be set up. For this purpose, observation data is required. Depending on the type and the completeness of the observations used, the grammar instance can be of different quality levels. A grammar instance which is able to reproduce the interior of a specific building exactly would be a perfect grammar showing the highest quality level. A perfect grammar can be approximated when a floor plan of the respective building is available. (Peter et al. 2011) demonstrated how 3D indoor models can be generated fully

automatically from photographed floor plans. The room configurations represented in such a 3D indoor model can be interpreted as result of a grammar-based production process. Within an inverse modeling step, rules that could have produced the given indoor model can be automatically inferred leading to a high-level grammar instance.

When pedestrian traces (as applied in section 4.3) are used as observation data, the quality of the resulting grammar instance will strongly depend on the traces' completeness and accuracy. An efficient practice could be to infer an initial grammar from a limited set of traces. This grammar can then be used to verify or correct already existing indoor geometries and predict new geometries to building parts where no observation data is available. Hypotheses about room configurations which cannot be checked against observation data are marked as *geometry with low reliability*. As soon as new traces are observed in building parts with geometries of low reliability, the initial grammar instance can be updated. Thus, the grammar's quality can be increased continuously.

4. GRAMMAR APPLICATION

An instance of an individual indoor grammar contains knowledge about indoor geometries and room arrangements which are characteristic for the building or building type for which the grammar has been derived. This object knowledge comprises geometric properties (e.g. the size of a room), topological properties (e.g. the connectivity of rooms), as well as semantic aspects (e.g. a functional grouping of rooms). Having such an individual indoor grammar available (see section 3.3), reliable hypotheses about possible indoor geometries can be generated. For this, the grammar rules are applied within a so-called production process.

The starting point for the production process is a 3D model of the building's outer shell, and information on the number of floors which can be easily estimated from the building's height or possibly detected window structures (Becker 2009). Based on this information, 3D solids can be derived representing the space of each floor. The production process is composed of two stages: (1) The application of the L-system on the floor to generate hallway spaces and non-hallway spaces (section 4.1), (2) the application of the split grammar on the non-hallway spaces to generate rooms (section 4.2).

4.1 Application of L-System for Hallway Generation

The goal of the L-system is to generate a network of hallways based on which the floors of the building can be segmented into hallway spaces and non-hallway spaces. Before the L-system can be run, an axiom has to be set up which describes one or several initial hallway segments. Such initial hallway segments can be the result of interpreted observation data (see the example in section 4.3), or simply represent the center axes of the building's footprint if no observation data is available. Having determined the axiom, the production rules as defined in section 3.2.1 can be applied. Since the stochastic part of the L-system has been shifted to an external function, the production rules can be processed in sequential order.

4.2 Application of Split Grammar for Room Generation

The goal of the split grammar is to establish room configurations in the indoor model. As a starting point (axiom),

the split grammar uses the previously derived non-hallway spaces (see section 4.1) as they define the regions where rooms can occur. Since the split grammar is a stochastic grammar, different room configurations can be produced for one and the same non-hallway space. In order to get the most probable room configuration which fits both the non-hallway space and possibly existing room geometries best, we perform a constraint-augmented random walk on the Markov chain introduced in section 3.2.2. By means of the constraints, geometric restrictions can be considered which are derived from observations. For example, (Peter et al. 2011) showed how the position of doors can be inferred from trace data and a coarse model. Based on such information, constraints can be set up which ensure that (1) no partition plane will be placed at a door's position, and (2) a partition plane will be placed between two detected door positions assuming that each room is connected through only one door to the hallway.

4.3 Results

To explore the power of our grammar-based concept for indoor modeling, we tested it in a real world scenario: Similar to the approaches presented by (Alzantot & Youssef 2012; Shin et al. 2012), we use erroneous and incomplete trace data. In our case, this data originates from foot-mounted MEMS IMU positioning systems and was passively gathered by pedestrians in the second floor of one of our university buildings. From these traces we automatically generated hypotheses about indoor geometries for the complete floor using grammar support.

In more detail: The input data consists of traces, an individual grammar instance, and a 3D model of the exterior building shell. **Traces:** In total, 250 odometry traces were available covering all hallways of the floor. However, to simulate a situation with incomplete observation data, we left out the hallways found in the north eastern part of the floor. Room access was limited to some office rooms in the floor's south central part. **Grammar:** To investigate the full benefit of grammar support, we applied a high-level grammar automatically inferred from a floor plan of the first floor of the building (see section 3.3). Although the first and the second floor show different room arrangements and slightly different room sizes, the overall architectural style is the same. **3D building shell:** We used the LOD2 building model provided by the city surveying office.

Figure 4 illustrates the traces after correction and alignment to the building shell. The artificially reduced set of trace-based generated hallways is given as dotted polygons in Figure 5(a).

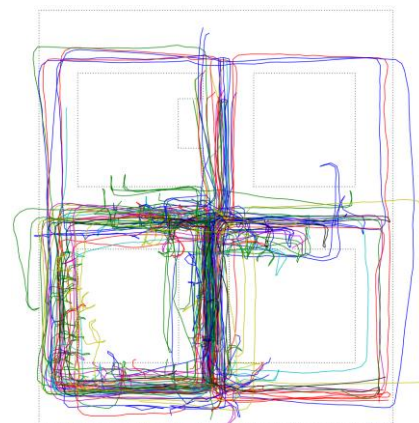


Figure 4. Traces acquired in 2nd floor of university building

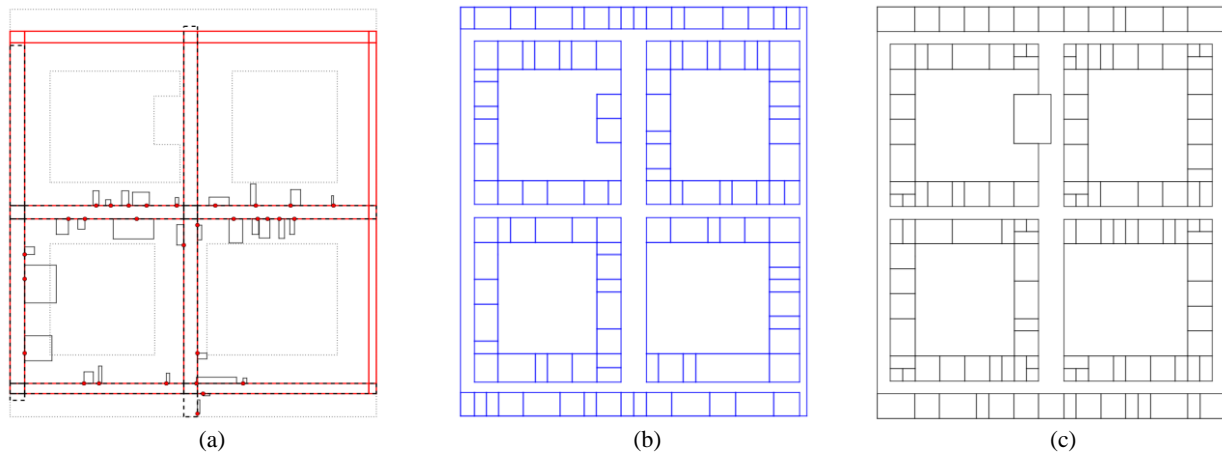


Figure 5. 2nd floor of university building. (a) hallways (dotted polygons) and rooms with door positions (small polygons) derived from traces, hypothesized hallways (red solid polygons); (b) complete model with hypothesized rooms; (c) ground truth

This fragmentary hallway system is the starting point for the grammar application. It represents the axiom on which the L-system rules can be applied. Proceeding as described in section 4.1, the existing hallway system is augmented by hypotheses about further hallways covering the complete floor (red polygons in Figure 5(a)). The hypothesized rooms, which result from applying the split grammar to the previously generated non-hallway spaces as explained in section 4.2, can be seen in Figure 5(b), while the ground truth is given in Figure 5(c). The ground truth model consists of 131 rooms, of which the pure data-driven approach using traces is able to reconstruct 29. The split grammar applied on the incomplete set of hallways increases this number to 92. Completing the hallways by the L-system enables the split grammar to reconstruct 116 rooms, while keeping the average room width error at around 2 meters.

5. CONCLUSIONS AND OUTLOOK

We proposed a combined indoor grammar to support the reconstruction of 3D building interiors from real sensor data. Since instances of individual grammars – each representing a specific architectural style – can be automatically derived from observation data, arbitrary building layouts can be generated without manually changing the grammar rules. The grammar is appropriate to represent the different geometric and topological characteristics of hallways and rooms by combining concepts from L-systems and split grammars. Thus, the grammar is flexible enough to generate complex indoor layouts. Based on a real world scenario, we showed that the benefit of supporting indoor modeling by using our grammar concept can be immense compared to a pure data-driven reconstruction from traces stemming from foot-mounted MEMS/IMU positioning systems.

Next, we will transfer the concept of the split grammar from *spaces to faces*. This will enable us to also model wall openings like doors. Additionally, we will address the automatic derivation of initial grammars from traces and the integration of such grammars in an automatic update and enhancement loop.

ACKNOWLEDGEMENT

This work is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG), grants FR 823/25-1 and RO 1086/17-1, and the Ministry Of Science, Research and the Arts Baden-Württemberg.

REFERENCES

- Alzantot, M. & Youssef, M., 2012. CrowdInside: automatic construction of indoor floorplans. In *Proc. 20th Int. Conf. Advances in Geographic Information Systems*. SIGSPATIAL '12, 99–108.
- Becker, S., 2009. Generation and application of rules for quality dependent façade reconstruction. *ISPRS J. Photogrammetry and Remote Sensing*, 64, 640–653.
- Bokeloh, M., Wand, M. & Seidel, H.-P., 2010. A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph. (TOG)*, 29(4), 104,1–104,10.
- Budroni, A. & Boehm, J., 2010. Automated 3D Reconstruction of Interiors from Point Clouds. *Int. J. Architectural Computing*, 8, 55–73.
- Fietz, A. et al., 2010. Automated 2D Measuring of Interiors using a Mobile Platform. In *Proc. 7th Int. Conf. Informatics in Control, Automation and Robotics (ICINCO 2010)*.
- Van Gool, L., Martinovic, A. & Mathias, M., 2013. Towards Semantic City Models. In D. Fritsch, ed. *Proc. Photogrammetric Week '13*. Wichmann, Berlin/Offenbach, 217–232.
- Gröger, G. & Plümer, L., 2010. Derivation of 3D Indoor Models by Grammars for Route Planning. *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(3), 193–210.
- Kada, M., 2007. Scale-Dependent Simplification of 3D Building Models Based on Cell Decomposition and Primitive Instancing. In *Proc. Int. Conf. Spatial Information Theory, COSIT '07*, 222–237.
- Müller, P. et al., 2007. Image-based procedural modeling of facades. *ACM Trans. Graph.*, 26(3), 85,1–85,9.
- Müller, P. et al., 2006. Procedural modeling of buildings. *ACM Trans. Graph.*, 25(3), 614–623.
- Newcombe, R.A. et al., 2011. KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE Int. Symp. Mixed and Augmented Reality (ISMAR)*. 127–136.
- Parish, Y.I.H. & Müller, P., 2001. Procedural modeling of cities. In *Proc. 28th Ann. Conf. Comp. Graph. and Interactive Techniques*. SIGGRAPH '01, 301–308.
- Peter, M., Haala, N. & Fritsch, D., 2011. Using Photographed Evacuation Plans to Support MEMS IMU Navigation. In *Proc. 2011 Int. Conf. Indoor Positioning and Indoor Navigation (IPIN2011)*, 1–4.
- Prusinkiewicz, P. & Lindenmayer, A., 1990. *The algorithmic beauty of plants*, Springer New York.
- Shin, H., Chon, Y. & Cha, H., 2012. Unsupervised Construction of an Indoor Floor Plan Using a Smartphone. *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6), 889–898.
- Wonka, P. et al., 2003. Instant architecture. In *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03, 669–677.