Grammar Supported Indoor Mapping

Michael Peter, Susanne Becker, Dieter Fritsch

Institute for Photogrammetry
University of Stuttgart
[firstname.lastname]@ifp.uni-stuttgart.de

Abstract. In this paper, we will demonstrate that the automatic interpretation of a single photograph of an evacuation plan can be used for the reconstruction of a coarse indoor model. In addition to the coarse model, these photographs may also provide the user with two essential values for positioning approaches delivering only relative coordinates. Such a positioning approach, in turn, can be used to refine the coarse model e.g. by automatic reconstruction of door openings. To further robustify our approach, object knowledge will be introduced by means of a formal grammar for indoor modeling.

Keywords: Indoor Mapping, MEMS IMU, Grammar

1. Introduction

Due to the broad availability of a positioning device (GPS) and digital maps the navigation problem concerning outdoor movement of vehicles and pedestrians can be considered as mostly being solved. However, pedestrian navigation in building interiors has been facing increased interest in the recent years. In comparison to the outdoor case, it is complicated by multiple factors: Firstly, many applications of indoor positioning demand for much higher accuracies than outdoor applications. Secondly, global positioning systems are unavailable which results either in the necessity for expensive infrastructure or the use of less accurate infrastructure independent positioning approaches. Thirdly, map matching as used to support outdoor positioning is hindered by the fact that pedestrian movement is generally less constraint than vehicle movement and, finally, the availability of indoor models is heavily dependent on the owner of the respective buildings. However, most infrastructure independent indoor positioning approaches need models at least as support information (e.g. Walder & Bernoulli 2010) or depend completely on them (Link et al. 2011).

In order to overcome this lack of indoor models, we present our approach for the automatic reconstruction of coarse indoor models from photographed evacuation plans. Its general feasibility from plans in a single well-known layout has already been shown in (Peter et al. 2011). Here, we will present the main idea of the approach while concentrating on improvements enabling its generalization to arbitrary plans. Additionally, we will show how the resulting coarse indoor model may be refined by reconstructing door openings from an analysis of user tracks derived from a foot-mounted MEMS IMU positioning system employing the well-known Zero-Velocity-Update approach.

The indoor models reconstructed from photographed evacuation plans can be erroneous or incomplete. To make our mapping approach robust against the quality of the input data, we enrich the data driven approach by object knowledge. Building interiors are subject to numerous geometric and topological conditions: inner walls are often either rectangular or parallel to outer walls; rooms can be adjacent but not overlapping etc. Following these conditions, the construction of interiors can be traced back to basic architectural principles and, thus, is appropriate to be described in a formal, rule-based way. A powerful means to facilitate this is using formal grammars. In the paper, we will present our first developments of a formal rule-based description for interiors.

2. Reconstruction of coarse 2D and 3D Models from Photographed Evacuation Plans

Evacuation plans are a compulsory inventory for all publicly used buildings in most countries.

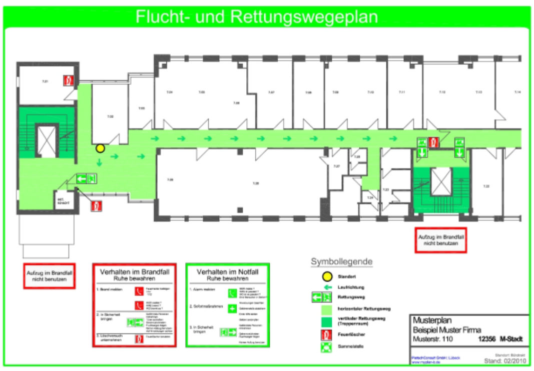


Figure 1. Example evacuation plan taken from (Wikipedia 2012)

Apart from additional information like heading, legend, address and behavioral rules for emergency cases, their main content is a detailed plan of the interior building structure in the viewer's vicinity (see Figure 1). This plan's level of detail may differ to a great extent between different buildings. The plan will contain the locations of installations for use in emergency cases (fire extinguishers, fire alarm buttons etc.) as well as evacuation routes, while additional information like the positions of doors etc. is not always included.

2.1 Approach Overview

A general outline of our processing pipeline sounds as follows: Firstly, the input image is enhanced by operations correcting the color balance and brightness differences. Secondly, the image is divided into foreground and background using adaptive threshold binarization. The binary image still contains evacuation symbols which have to be detected and removed. The structures in the resulting cleaned binary image are thinned in order to derive the skeleton which is used in a specialized vectorization step to bridge the symbol areas. In order to enable the use of metric thresholds in this step, the outer contour can be matched to an available model of the building's outer shell, delivering the transformation parameters from image coordinates to real world coordinates. During the following reconstruction of the facets of the 2D model, apart from rooms also stairs and staircases may be detected. The number of detected stairs together with a common stair height results in an approximate room height which can be used to extrude the edges of the 2D model to walls in the final 3D model.

2.2 Image Pre-Processing

The image enhancement processing splits up into two steps: color balance and correction of brightness differences in the image. By color balance we try to ensure that white areas (namely the plan's background) will be white in the image. This is done in a similar way like described in (The GIMP Documentation Team 2011). In the histograms of the red, green and blue channels, respectively, the upper and lower 0.05% of the pixel values will be discarded and the histograms stretched accordingly.

In order to correct still remaining brightness differences and to achieve a radiometric equalization between different images, a local contrast enhancement is carried out using the white balanced image as input. For this correction the image is converted to the CIELab color space and the L-image is filtered using the Wallis filter (Wallis 1976). After converting the CIELab image back to RGB, these corrections affect the full color space. Figure 2 depicts a possible result of the complete enhancement in comparison to the original image.

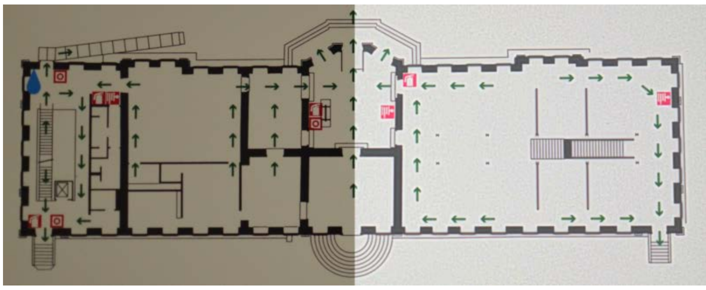


Figure 2. LHS: original image, RHS: white balanced and Wallis filtered image

2.3 Binarization and Symbol Detection

Binarization of the image is greatly facilitated by the fact that evacuation plans are designed for optimal human legibility even in extreme situations. If the plan at hand at least partly follows the design guidelines stated in (ISO 2009), this results in a white background, black ground plan elements and symbols in signal colors. Thus, a simple adaptive threshold with a big block size is suitable for all color balanced images. The result of this operation is depicted in Figure 3, left hand side.

The detection of evacuation symbols in the photographed plan is not only necessary because these symbols may hide parts of the ground plan structures to be reconstructed. Furthermore, the contained evacuation information should be provided in the reconstructed model and symbols like arrows may contain important further information like the down or up direction of staircases.

A first approach for the solution of this problem using well-known symbols and cross-correlation template matching was already presented in (Peter et al. 2011). A more general method is the use of color segmentation for symbol detection. Even though most currently available plans do not follow the ISO standard (ISO 2009) completely, colors are generally used to distinguish symbols from other plan elements.

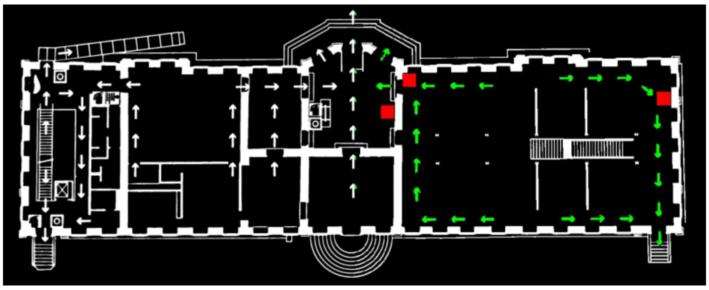


Figure 3. LHS: Binary image derived from the plan depicted in Figure 2; RHS: Binary image with symbols overlaid (detected using colour segmentation approach)

We use this fact and the Color Structure Code (Priese & Sturm 2003) combined with thresholds for the expected signal colors (pure red, green, blue and yellow). The resulting detected symbol areas can be seen in Figure 3, right hand side. If the symbols contained in the plan are known, this color detection approach may also be used to constrain the search space for a further classification using template matching.

2.4 Geo-referencing by Matching to External Building Shell

In order to enable the use of metric thresholds for the symbol bridging step as well as to use the final model for navigation purposes, the transformation between image coordinates and real world coordinates has to be found. Therefore, the outer contour of the reconstructed indoor model is matched to an existing outer shell (e.g. the ground plan from OpenStreetMap). As the level of detail in both these shapes may differ to a great extent (see Figure 4, red contours) and the reconstructed indoor model may be incomplete, this results in a matching problem between two possibly incomplete shapes with unknown scaling and unknown relative orientation.

For the solution of this problem, firstly, the outer contour in the binary image is selected and the external shell is scaled and translated to fit to the binary image's dimensions. Secondly, both polygons are generalized using a 2D version of the 3D building generalization approach presented by (Kada et al. 2009). The result of this operation is depicted in Figure 4 (blue contours).

Then, the angle between adjacent edges and the ratio of their lengths is computed for every node of these generalized polygons. Using these two features, similar node pairs are identified and, for every pair, the matched nodes and the adjacent edges' other ends deliver the parameters of an approximate affine transformation. The preferable parameter set among these candidates should fulfill these two conditions: minimum shape modification and maximum overlap of the transformed source polygon with the destination polygon. The first condition is checked by comparing the per-node adjacent edges' length ratio of the transformed polygon to the state before, while the maximum overlap is computed following the approach presented by (Kada et al. 2009).

However, the selection using this approach may be imperfect due to symmetric input models (see rectangular generalized shape in Figure 4). In this case, the correct transformation may be selected using the comparison of the area overlap between the original models as a further descriptive feature.

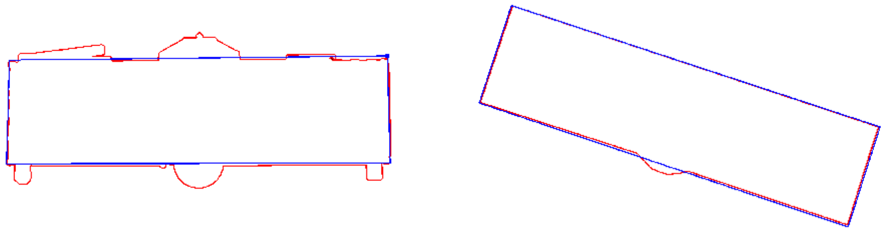


Figure 4. Original (red) and generalized (blue) versions of the outer contour in the image (LHS) and the external shell from OpenStreetMap (RHS)

2.5 Symbol Bridging and Final 2D Model

In order to produce a complete reconstruction of the floor plan depicted in the evacuation plan, the areas previously covered by symbols have to be bridged. For this purpose, the skeleton of the cleaned binary image is derived using the iterative thinning approach presented by (T. Y. Zhang & Suen 1984). Vectorizing this skeleton image delivers end nodes as part of the contained topological information.

For the actual symbol bridging the edges ending in at least one end node are prolonged until a length threshold is exceeded, not taking the distance traveled on symbol areas into account and stopping at structures in the binary image. These prolonged edges are classified into two classes: having a bigger overlap with red symbols than the red symbol size ("long edges") or not. We expect red symbols to produce most occlusions, as these represent safety equipment which is mounted to the walls. According to the classification, long and other edges will be treated differently in the last completion step: Firstly, all long edges ending in a structure in the binary image are accepted; secondly, all other long edges are reconstructed up to their last intersection point with other prolonged edges; thirdly, all other edges will be reconstructed up to their first intersection point or structure in the binary image, respectively. To ensure the reconstruction of two-sided walls, the validated edges will be painted into the cleaned binary image using the stroke width of the edge they prolonged (computed using (Epshtein et al. 2010)).

The contours extracted from the updated binary image are then used to reconstruct the facets of the 2D model, which represent hallways, rooms and stairs. Figure 5 shows the final 2D model, transformed to world coordinates and drawn as overlay in OpenStreetMap.

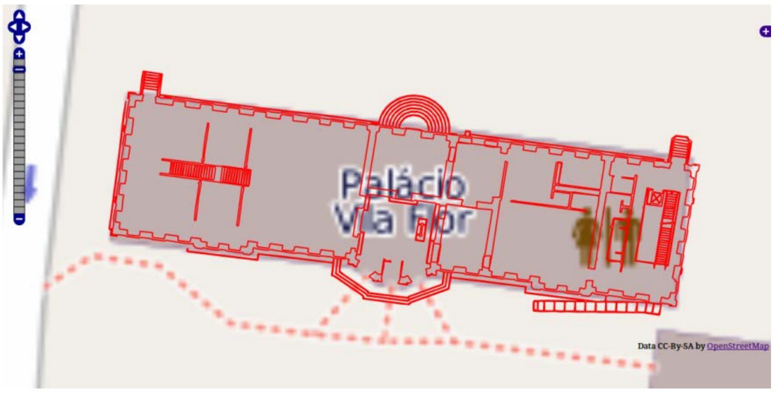


Figure 5. Reconstructed 2D model as overlay in OpenStreetMap (© OpenStreetMap contributors, Data CC-By-SA)

2.6 3D Model

In order to derive an approximate predominant room height usable for the extrusion of the 2D wall edges to 3D wall facets, the reconstructed facets of the 2D model are further analyzed and stair candidates are identified.

Therefore, the maximum width of each “room” polygon is computed using the distance transformation (Felzenszwalb & Huttenlocher 2004) and all polygons not wider than 0.3m are selected. The length of each stair candidate is computed as the sum of their skeleton’s pixels (plus a tolerance), converted to a metric value, and only those candidates longer than 0.8m are kept.

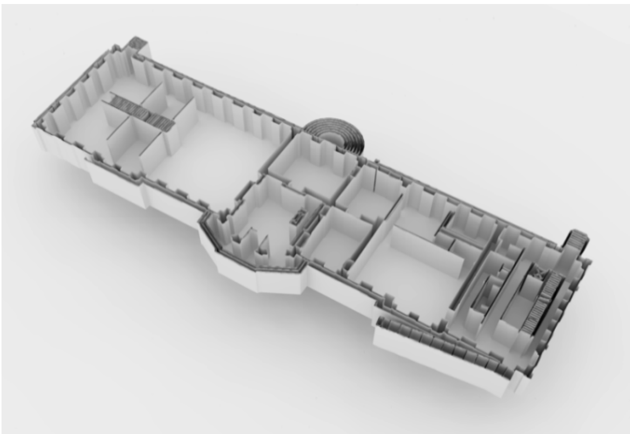


Figure 6. Automatically derived 3D model (stairs not explicitly reconstructed at this stage)

Grouping neighboring single stair candidates will either cause stair candidates to be removed (if they have no neighbors) or deliver staircases (which may still be incomplete due to formerly overlapping symbol regions and therefore subject to a completion step). The combination of the maximum stair number of all staircases and a standard stair height (like stated in (E. Neufert et al. 2002)) leads to an approximate floor height. Using this floor height, the 2D model may be extruded to three dimensions like depicted in Figure 6.

3. Model Refinement by the Analysis of User Tracks

3.1 Indoor Positioning using a foot-mounted MEMS IMU

For a further refinement of these coarse models, we employ position tracks of a user which are acquired by the readings of a foot-mounted MEMS IMU and processed with the zero velocity update approach presented by (Foxlin 2005). For the correction of small positioning errors still remaining due to not eliminated drift errors or initial misalignment, we employ our alignment of walked straight lines to one of the main axes of the building's external shell (like presented in (Peter et al. 2011), see Figure 7).

However, this positioning approach only delivers coordinates relative to a known initial position and initial orientation. In order to determine those values when starting the positioning process in the building, we propose a further analysis of the evacuation plan. The position of the user while he photographs the plan is marked by a symbol which can be found either using template matching or as a symbol unique in shape or color. Using the transformation parameters derived during the model reconstruction step, it may be transformed to world coordinates.

In order to compute the initial orientation, i.e. the camera's pose during the image acquisition, the approach presented in (Z. Zhang & He 2007) and the corners of the paper containing the plan are employed. For the computation of the final orientation, we use the fact that the plan has to be oriented according to the local environment if it follows (ISO 2009).

3.2 Model Refinement

While the positions derived by using such approaches are often employed to guide the user in a pedestrian navigation scenario, the user's position may also be utilized to acquire information needed for the refinement of the underlying indoor model.

One example is the geo-referencing of semantic information acquired by user interaction. Here, we have investigated the derivation of room numbers and people assigned to a room from photographed door plates using Optical Character Recognition (OCR). However, by analyzing the user tracks in the context of the coarse model, they may also be used for an update of the model's geometric features by a fully automated derivation of door positions. Here, we employ the fact that the user is not able to pass through a door in arbitrary angles. Thus, if an average person's position track hits a wall in the model at an angle between 40° and 140°, a door will be reconstructed. Implicitly, this constraint provides us with a simple map-matching solution, correcting the track whenever a wall is hit at angles different from that (see Figure 7).

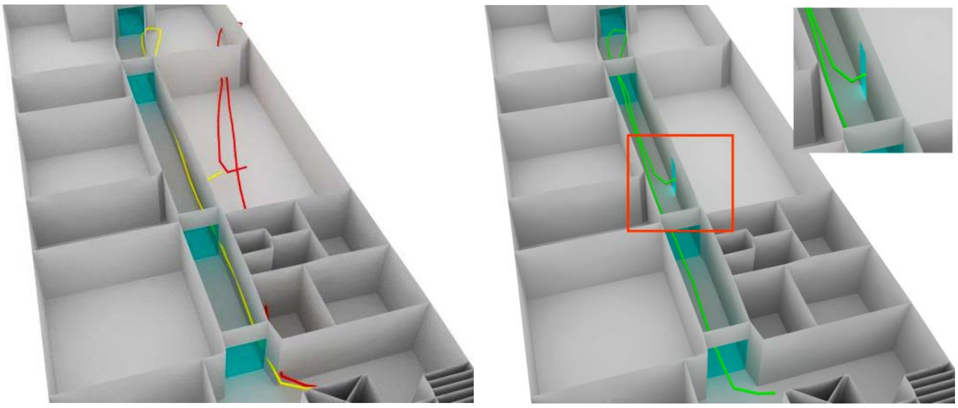


Figure 7. LHS: Coarse model, uncorrected (red) and aligned (yellow) tracks; RHS: refined model containing reconstructed door and map-matched track (green)

4. Grammar for Indoor Modeling

As the presented reconstruction approaches implicate a number of potential sources of error depending on the quality of the observation data (e.g. in the symbol bridging, track alignment and map-matching steps), means for their support should be found. Therefore, we started the development of a formal grammar which is able to store geometric, topological and semantic information on building interiors. Generally, in a formal grammar, object knowledge is represented through symbols and a set of production or replacement rules. The symbols are called non-terminals if they can be replaced by other symbols, and terminals otherwise. The non-terminal symbol which defines the starting point for all replacements is the axiom. The production rules state the replacement of predecessor symbols by successor symbols. By successively applying rules to the axiom, new sequences of symbols are generated.

For several years formal grammars have been applied successfully for modeling geometric structures. While (Prusinkiewicz & Lindenmayer 1990) focus on line structures by e.g. simulating growth processes of plants through

Lindenmayer-systems (L-systems), (Müller et al. 2006) show examples for the grammar-based reconstruction of building shells. (Gröger & Plümer 2010) present a grammar for indoor modeling, however, without the possibility to integrate erroneous observation data.

In contrast, our grammar for modeling building interiors is designed in such way that it can be derived automatically from observation data in order to reflect knowledge on individual building interiors.

4.1 Conceptual Considerations

The first step in developing a formal description for indoor models is to identify basic geometric primitives as well as characteristic topologic properties by which an indoor model can be described. For now, we restrict ourselves to *floors* and *room configurations*.

While the arrangement of floors is linear and, thus, shows 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 created by random compositions of walls, but follow architectural principles and are subject to functional restrictions. Knowledge about architectural principles and geometric and topological restrictions helps to understand indoor geometries and detect semantic relationships between them. In order to provide as much support as possible to indoor mapping and modeling, our indoor grammar is designed to be able to reflect such knowledge. The following properties of building interiors are crucial for our grammar design:

1. In order to ensure convenient access to the rooms, buildings – in particular public buildings - are usually traversed by a system of corridors.
2. The system of corridors divides each floor into *corridor-spaces* and *non-corridor-spaces*. Non-corridor-spaces can be further divided into smaller *room units* which are mostly arranged in a linear sequence parallel to the adjacent corridor.
3. Depending on their function, such room units feature specific layouts. For example, in hotels or hospitals a typical room unit consists of a bedroom and a bathroom.

Properties 1 and 2 give reasons for the application of two different modeling strategies: The course of the corridors, 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 corridor network. The analogy to 3D city modeling becomes apparent: The network of streets corresponds to the network of corridors, and the segmentation of the regions lying between the streets into parcels is comparable to the partitioning of non-corridor-spaces into rooms. This analogy motivates the adaption of grammar-based concepts used for city modeling. Similar to (Müller 2001) who developed a L-system for modeling streets, we will use a specially designed L-system for the formal description of the corridor network (which is part of the ongoing work and will not be discussed in this paper). For the modeling of room layouts we develop a separate grammar which is mainly based on split operations.

The grouping of functionally related rooms to superior *room units*, as mentioned in property 3, is an important semantic information which the grammar should be able to represent. This requirement can be met by describing characteristic room configurations through the grouping of split operations to structure-generating production rules. Such a structure-generating grouping of split operations implies that the splits have to be carried out in a certain sequential order, which can also be interpreted as giving different priorities to the splits or rather the resulting partition walls. We assign high priority to walls which are incident with two opposite *main walls*. With the term *main wall* we denote walls which are not the result of split operations like outer building walls or the boundary walls of corridors. Each partition wall which is incident with less than two main walls is assigned a low priority. Splits that produce walls of high priority are carried out first. To illustrate, Figure 8 shows a real floor plan, in which walls of high priority are marked in red, and walls of low priority in green.

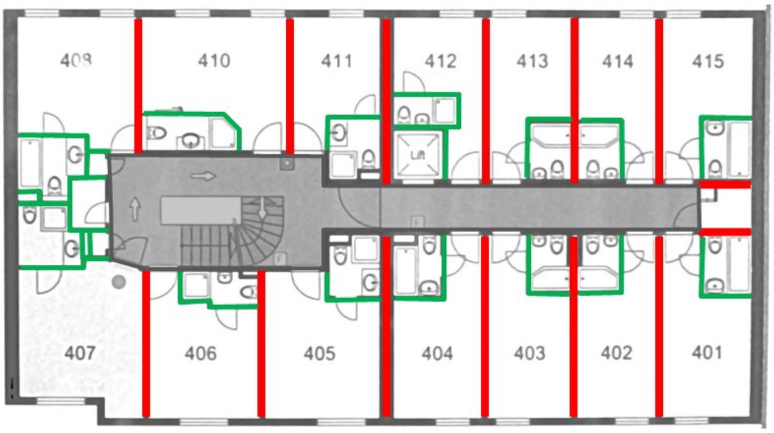


Figure 8. Floor plan of a hotel in Zürich with high priority-walls (red) and low priority- walls (green).

4.2 Grammar for the Modeling of Room Configurations

Our grammar $G^{indoor}=(N,T,S,R)$ is designed to be applied to empty non-corridor-spaces in order to install room configurations. It comprises 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* consists of the symbols *S*, *Space* and *Face*. *S* is the axiom which represents an empty non-corridor-space. The symbol *Space* stands for an arbitrary 3D solid which can be further divided. Analogously, *Face* represents a 2D wall that can still be decomposed in wall segments. The terminals *T* describe solids or walls that are not divisible any further. To distinguish from non-terminals, the terminal symbols *space* and *face* start with lower case. Both non-terminals and terminals have attributes. They determine the space's or face's geometry and type.

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 which automatically provides knowledge about neighborhood relationships between the spaces and faces. Furthermore, topologically correct reconstruction results are ensured. An instantiation rule replaces a non-terminal by a terminal symbol, and, thus, creates an instance of the respective geometry. Beside parameters, each rule has additional semantic rules, which basically comprise geometric and topologic constraints as well as functions for the derivation and setting of attribute values. However, for lack of space within this paper, semantic rules will not be discussed, here.

In total, we distinguish between six different types of rules which are listed in the following. Although the listed rules focus on the application on the non-terminal *Space*, they are also valid for the application to the non-terminal *Face*.

- $R^{SingleSplit}$: $\text{Space} \rightarrow \text{Split}^{Space}(\alpha,d)$

  with $\text{Split}^{Space}(\alpha,d) = \text{Space}_{left}, \text{Space}_{right}$

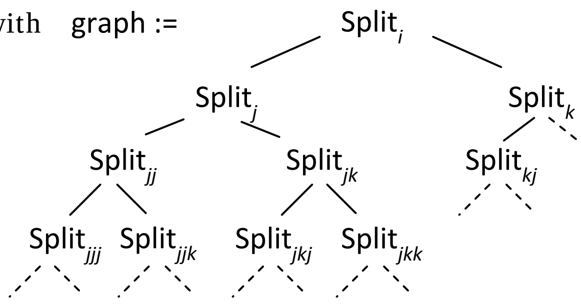
- $R^{RepeatSplit}$: $\text{Space} \rightarrow \text{RepeatSplit}^{Space}(\text{Split}_a)$

- $R^{StringSplit}$: $\text{Space} \rightarrow \text{StringSplit}^{Space}(\text{sequence})$

  with $\text{sequence} := \text{Split}_a, \text{Split}_b, \ldots, \text{Split}_i, \ldots$

- $R^{MultiSplit}$: $\quad Space \rightarrow MultiSplit^{Space}(graph)$

  with graph :=



- R^{Merge}: $\quad Space_{left}, Space_{right} \rightarrow Merge^{Space\_left,\ Space\_right}()$

  with $Merge^{Space\_left,\ Space\_right}() := (Split^{Space}(\alpha, d))^{-1}$

- $R^{Instantiation}$: $\quad Space \rightarrow space$

Rule type *SingleSplit* performs a single split operation by replacing the non-terminal *Space* by the non-terminals $Space_{left}$ and $Space_{right}$. $Space_{left}$ and $Space_{right}$ are the result of the function $Split^{Space}(\alpha, d)$. 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.

Rule types *RepeatSplit* and *StringSplit* can be used to store knowledge about linear sequences of split operations that have high priority (see section 4.1). 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 of low priority - applied for modeling non-linear room layouts - can be aggregated within the rule type *MultiSplit*. Since, in this case, the split operations cannot be represented in sequential order, they need to be represented within a graph structure.

By means of the four split rules, the merge and the instantiation rule, each processed evacuation plan can be transferred to a specialized rule system which contains detailed knowledge about the construction of the building's interior. Based on simple examples, Figure 9 shows how the six rule types can be geometrically interpreted.

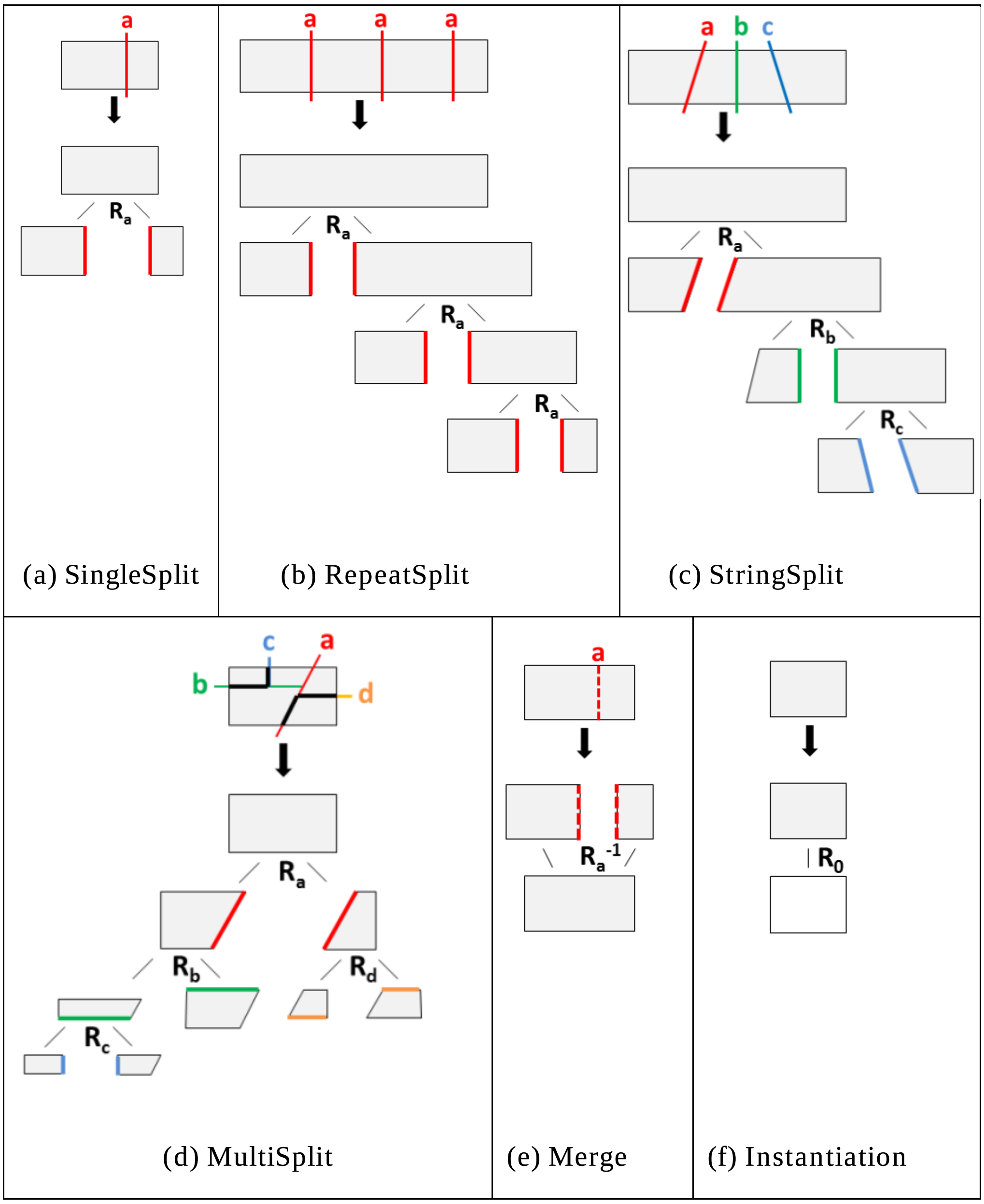


Figure 9. Different rule types and their geometric interpretation.

To verify the practicability of our grammar rules, we applied them to a real floor plan (Figure 10) for which a 3D model has already been derived (see Figure 7). As can be seen, the six rules types are sufficient to express even complex room layouts.

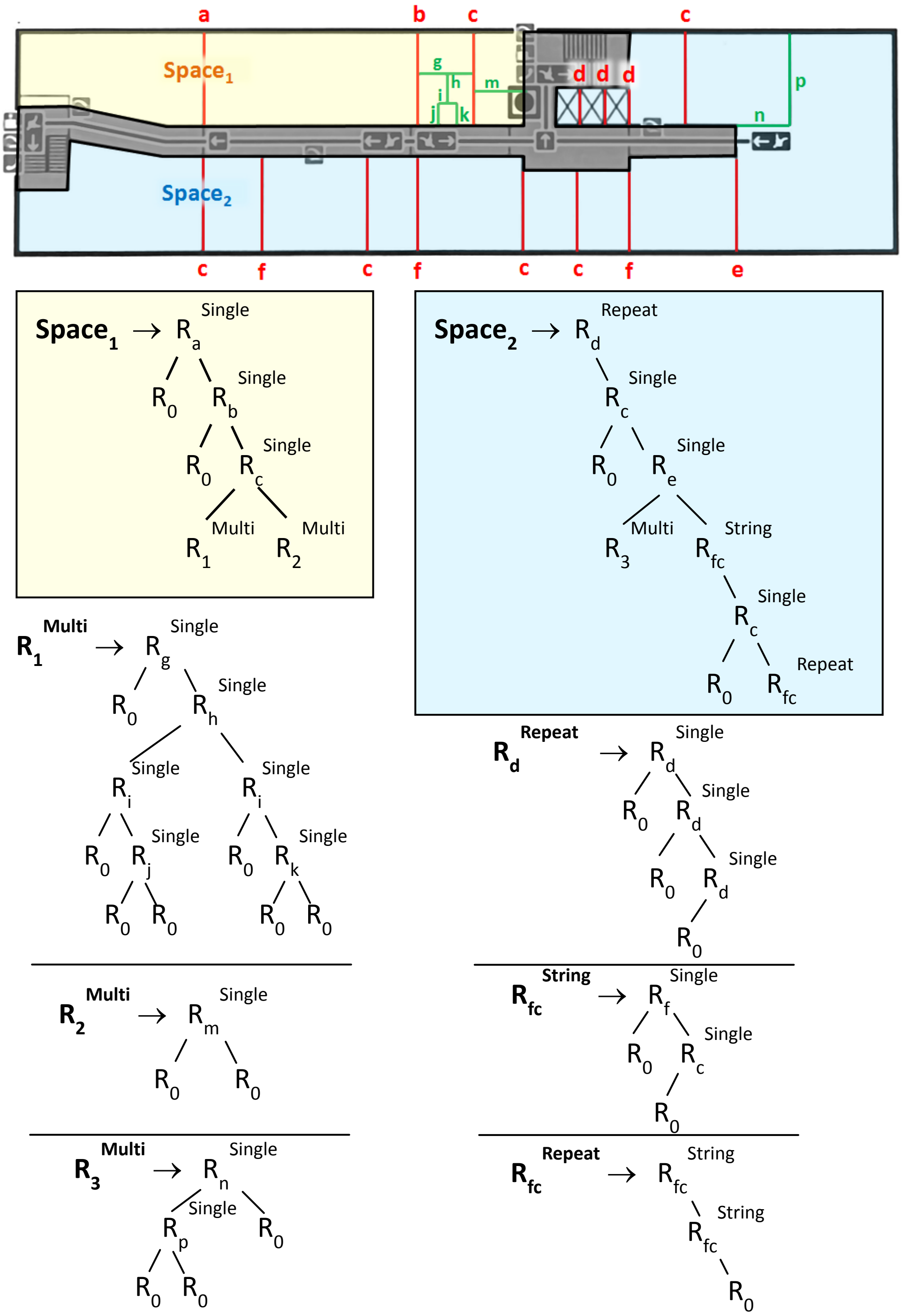


Figure 10. Rule-based description for a real floor plan.

5. Conclusion and Outlook

In the future, we will continue the development in terms of the L-system representing the corridor network and the semantic rules which will complete the grammar.

The information about the individual building's interior structure stored in the acquired grammar will enable a further support of the positioning technique as well as the reconstruction both from the photographed plans and the user tracks. Additionally, we will investigate the possibility of reconstructing a model of the indoor environment using only user tracks and constraints from the grammar, simultaneously refining the grammar.

Acknowledgements

This work is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG), Grant FR 823/25-1 / RO 1086/17-1.

References

Epshtein, B., Ofek, E. & Wexler, Y., 2010. Detecting text in natural scenes with stroke width transform.

Felzenszwalb, P. & Huttenlocher, D., 2004. Distance transforms of sampled functions. Available at: http://ecommons.library.cornell.edu/handle/1813/5663 [Accessed November 6, 2012].

Foxlin, E., 2005. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6), pp.38–46.

Gröger, G. & Plümer, L., 2010. Derivation of 3D Indoor Models by Grammars for Route Planning. *Photogrammetrie-Fernerkundung-Geoinformation*, 2010(3), pp.193–210.

ISO, 2009. ISO 23601 Sicherheitskennzeichnung – Flucht- und Rettungspläne. Available at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41685.

Kada, M., Peter, M. & Filippovska, Y., 2009. 3D Building Generalization under Ground Plan Preserving Quality Constraints. In *Proceedings of the 6th International Symposium on Spatial Data Quality*. ISSDQ09. St. John's, Canada.

Link, J.A.B., Smith, P. & Wehrle, K., 2011. FootPath: Accurate Map-based Indoor Navigation Using Smartphones. In *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN2011)*. Guimaraes, Portugal.

Müller, P., 2001. *Design und Implementation einer Preprocessing Pipeline zur Visualisierung prozedural erzeugter Stadtmodelle*. Master's thesis, ETH Zürich.

Müller, P. et al., 2006. Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3), pp.614–623.

Neufert, E. et al., 2002. *Architects' Data* 3rd ed., Wiley-Blackwell.

Peter, M., Haala, N. & Fritsch, D., 2011. Using Photographed Evacuation Plans to Support MEMS IMU Navigation. In *Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN2011)*. Guimaraes, Portugal.

Priese, L. & Sturm, P., 2003. Introduction to the color structure code and its implementation. Available at: http://www.uni-koblenz.de/lb/lb_downloads/download/csc.pdf [Accessed June 19, 2012].

Prusinkiewicz, P. & Lindenmayer, A., 1990. *The algorithmic beauty of plants*, Springer New York.

The GIMP Documentation Team, 2011. 8.8. White Balance. Available at: http://docs.gimp.org/en/gimp-layer-white-balance.html [Accessed December 20, 2011].

Walder, U. & Bernoulli, T., 2010. Context-adaptive algorithms to improve indoor positioning with inertial sensors. In *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*. Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on. pp. 1–6.

Wallis, R., 1976. An approach to the space variant restoration and enhancement of images. In *Proceedings of the Symposium on Current Mathematical Problems in Image Science*. Naval Postgraduate School, Monterey CA, USA.

Wikipedia, 2012. Datei:FRP Musterplan.jpg – Wikipedia. Available at: http://de.wikipedia.org/w/index.php?title=Datei:FRP_Musterplan.jpg&filetimestamp=20100824092304 [Accessed January 9, 2012].

Zhang, T.Y. & Suen, C.Y., 1984. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), pp.236–239.

Zhang, Z. & He, L.-W., 2007. Whiteboard scanning and image enhancement. *Digital Signal Processing*, 17(2), pp.414–432.