MODELLING OF INDOOR ENVIRONMENTS USING LINDENMAYER SYSTEMS

Michael Petera*

^aUniversity of Twente, Faculty ITC, Enschede, the Netherlands - m.s.peter@utwente.nl

Commission IV, WG IV/5

KEY WORDS: Indoor, procedural, modelling, Lindenmayer system, formal grammar

ABSTRACT:

Documentation of the "as-built" state of building interiors has gained a lot of interest in the recent years. Various data acquisition methods exist, e.g. the extraction from photographed evacuation plans using image processing or, most prominently, indoor mobile laser scanning. Due to clutter or data gaps as well as errors during data acquisition and processing, automatic reconstruction of CAD/BIM-like models from these data sources is not a trivial task. Thus it is often tried to support reconstruction by general rules for the perpendicularity and parallelism which are predominant in man-made structures. Indoor environments of large, public buildings, however, often also follow higher-level rules like symmetry and repetition of e.g. room sizes and corridor widths. In the context of reconstruction of city city elements (e.g. street networks) or building elements (e.g. façade layouts), formal grammars have been put to use. In this paper, we describe the use of Lindenmayer systems - which originally have been developed for the computer-based modelling of plant growth - to model and reproduce the layout of indoor environments in 2D.

1. INTRODUCTION

Documentation of the "as-built" state of building interiors has been gaining importance in the recent years, especially with upcoming applications like indoor navigation and location-based services and the improvement of applications like evacuation and disaster management. The most prominent method for base data acquisition in indoor environments nowadays is laser scanning, resulting in massive amounts of points. The automatic interpretation of such data with the aim of deriving CAD/BIM models or ground plans - despite the existence of data gaps and inaccuracies - still is subject to research. Many published approaches rely on architectural rules like predominantly planar structures or the well-known Manhattan world constraints. However, a) interior architecture does not always follow the Manhattan world constraints, and b) man-made indoor structures show higher-level architectural rules like the repetition of distances between walls of corridors or rooms (i.e. repeated room/corridor width) and a dependency between the external building shell's center line and the layout of corridors, among others.

As has been shown for e.g. grammar-supported façade modelling (Becker, 2009), the integration of such higher-level constraints helps to improve the reconstruction process. For several years, grammars have been put to use in the area of city modelling: the repetitiveness found in building façades as well as the "natural growth" structure of street networks can be modelled using split grammars, or Lindenmayer-(L)-systems, respectively. In indoor modelling, we translated those concepts to L-systems describing the corridor structure and split grammars to describe the non-corridor space being split into rooms by walls which depend on the corridor outline (Becker et al., 2013) (see also section 1.1).

Here, instead of using two different grammar concepts, we propose the use of a set of L-systems to model such constraints in the context of 2D modelling of the interior structure of large public buildings showing some kind of repeated structure (e.g. hotels, office buildings, hospitals). This paper describes the definition of the L-systems and the derivation of their rules' parameters and probabilities from data (section 2). The proof of concept is done using the fairly simple data set (see figure 1) which was also used in (Becker et al., 2013) (see section 3). The integration of the Lsystem with sensor data (like wall segments or areas of free space from lidar point clouds) is subject to future work (see also section 4).



Figure 1. Ground truth plan of a floor of the Computer Science building, University of Stuttgart, as used in (Becker et al., 2013); orange: corridors

1.1 Related Work

Grammars are rule-based production systems generally comprising a set of rewriting (or production) rules R which - starting with the axiom S - recursively rewrite non-terminals N to other nonterminals or terminals T.

While initially developed for the formal description of lan-

^{*}Corresponding author

guages, many researchers have described how to apply grammars in the procedural production of geometry. To this end, among others, shape grammars (Stiny and Gips, 1972), Lsystems (Prusinkiewicz and Lindenmayer, 1990), and split grammars (Wonka et al., 2003) were developed.

Parish and Müller's works (Parish and Müller, 2001, Müller et al., 2006), mark first adaptations of such rule-based production systems for city modeling. The authors describe the procedural generation of complete city models by generation of street networks and building façades using L-systems and split grammars, respectively. While useful for visualizations, movies, and games, the rules and parameters to be used by the grammars were hand-crafted by experts and, apart from some maps describing the population density of the city to be developed and the topography, no further data was used.

Müller further developed the concepts to be employed in the context of reconstructing building façades from images (Müller et al., 2007). Also other image-based façade reconstruction methods using a split grammar were presented, e.g. (Brenner and Ripperda, 2006). Furthermore, (Becker, 2009) describes the use of a split grammar for façade reconstruction from lidar data, deriving the parameters in areas with complete data and using the grammar to support the reconstruction of complete models in areas with incomplete data. More recent works like (Demir et al., 2016) show that there is still an ongoing interest in outdoor procedural city modelling.

In the context of indoor modelling, a similar trend can be observed. Early works describe the fully procedural generation of plausible indoor space subdivisions for residential houses (Hahn, 2006, Marson and Musse, 2010, Mirahmadi and Shami, 2012) or other buildings (Gröger and Plümer, 2010).

More recent developments aim at the integration of sensor data and can be found in the adaptation of a Palladian grammar by (Khoshelham and Díaz-Vilariño, 2014) and our previous work combining a split grammar (Peter et al., 2013, Philipp et al., 2014) and an L-system (Becker et al., 2013). While (Khoshelham and Díaz-Vilariño, 2014) - building on the Palladio grammar described in (Stiny and Mitchell, 1978) - employ a small set of general rules to support reconstruction from complete lidar point clouds, their approach does not enable for a description of repeated structures and thus does not need to learn grammar parameters from data beforehand. In contrast, our previous work employs higher-level rules whose parameters and probabilities have to be derived from data in the same manner as in Becker's grammar-supported façade reconstruction approach (Becker, 2009). In turn, the method is able to predict the distribution of corridors and wall splits based on their probabilities and thus can be employed to support reconstruction in areas of sparse data. The most recent developments in indoor procedural modelling are found in the publications by Loch-Dehbi, Dehbi, Gröger, and Plümer. In (Loch-Dehbi et al., 2016), for example, they describe the prediction of a building's floor plan using logical and stochastic reasoning. Currently, their approach is able to produce impressive results despite integrating only a very low amount of data like room areas and room numbers, but the integration of other observations seems feasible.

2. METHODOLOGY

In residential houses spaces of different use are often openly connected or combined. Furthermore, room sizes vary depending on their use. For example, often one big room being used as dining and living area will be complemented by smaller private rooms.

Here, we concentrate on large public buildings whose structure follows different usability constraints. Mostly, many small, separate spaces (used e.g. as offices, hotel rooms, hospital rooms) are connected to a system of corridors connecting the different parts of a floor. This is due to reasons like the need for an efficient use of the available space and the accessibility to windows within the majority of rooms. The latter reason explains the corridors often being found close to the building's external shell's center line and/or parallel to it as well as the majority of rooms having a direct connection to the external building shell and the corridors.

These observations motivate the way we model the elements of an indoor environment. Figure 2 depicts those elements: In red, the building's external shell is visible, its center line is depicted with a red dashed line (see section 2.1). The location of a segment of the corridor's center line can be expressed by an angle, an offset, and its length. The corridor walls are shown in yellow, the corrdidor's center line dashed yellow. The locations of the corridor's walls, in turn, are expressed depending on the corridor center line using an offset and their length.

Furthermore, there are two types of walls: Walls which depend directly on the corridor and the external shell (green) and walls which do not depend on the corridor only (grey). Due to usability constraints (e.g. efficient placement of furniture), walls depending on the corridor frequently connect to the corridor in a fixed angle, often 90 degrees. Using this angle, the distance between two consecutive walls, and the external shell (or a fixed length), the location of corridor-depending walls can fully be described depending on the corridor's center line. Walls which do not depend on the corridor directly (grey) can be expressed in reference to the wall they depend on using a set of (*angle, length*) tuples. If doors follow a pattern (e.g. the door center is always located in one meter distance from the room's left wall), their locations can be encoded using the distance from a wall along the corridor center line.



Figure 2. External shell (red) and its center line (red dashed), corridor walls (orange) and corridor center line (orange dashed), corridor-depending walls (green), walls not depending on the corridor (grey), door centroids (blue)

2.1 Center lines

It is obvious that center lines are a crucial element of our approach, as the corridor system is produced using the external shell's center line as reference and the walls are expressed using the corridor center line, in turn. Originally, we had planned to compute center lines by using a skeletonization approach like iterative thinning (Zhang and Suen, 1984) or the distance transform (Felzenszwalb and Huttenlocher, 2004) resulting in the external shell polygon's medial axis. Also, the straight skeleton, possibly with modifications described in (Haunert and Sester, 2008) seemed to be valid options. However, those approaches are - in varying amounts - too much affected by small structures in the polygon outline (which would affect all dependent parts).

Thus, we developed a new method to extract the center line of a polygon. Currently, the approach needs the polygon to be ring-like in order to define the hole and outer boundary, but in the future we plan to extend it to simpler polygons. The following steps are carried out in order to compute the center line (for a result see figure 3):

- 1. Project the vertices of each hole contour to all other contours along the normal vectors of the hole contour's edges, connect each vertex with its projection
 - (a) If two connections intersect, discard the longer one
 - (b) If the connection intersects with the hole contour, discard it
- 2. Construct cells using contours and connections
- Cells which are made up of more than two connections ⇒ Branch cells (green in figure 2)
- 4. Identify and handle subcontours between projected vertices; two cases exist:
 - (a) The subcontour is caused by two projected vertices
 ⇒ Regular cell: Project all vertices of the subcontour along the normal vector of the hole contour's edge to the hole contour
 - (b) The subcontour is caused by one projected vertex ⇒ Corner (red in figure 2): Connect all the subcontour's vertices to the (hole) vertex which was projected
- 5. Subdivide the cells using the new connections
- 6. For all cells which are not branch cells:
 - (a) Construct vertices of the cell's center line as the bisectors of all connections
 - (b) Connect the vertices using the contours' topologies ⇒ cell's center line
- Construct vertices for branch cells by intersecting all of the adjacent cells' center lines; build edges, discard dangling edges



Figure 3. External shell center line (blue dashed), corner (red) and branch (green) cells

2.2 Definition of the L-systems and their parameters

Lindenmayer-(L)-systems were developed to formalize the description of plant growth and to reproduce its processes using computers. The aptly-titled book "The algorithmic beauty of plants" (Prusinkiewicz and Lindenmayer, 1990) gives a good introduction to L-systems and their application to produce realistically-looking models of plants by modeling their growth processes. In addition to simple deterministic, context-free Lsystems also stochastic L-systems (using probabilities for the rules to be applied) and context-sensitive L-systems (constraining the application of a rule to a specific context within the system) are described.

The main two synthesis strategies in the context of L-systems are node rewriting and edge rewriting. In node rewriting a rule is applied to a node, causing its replacement by a sequence of the shape "node (edge node)" where several (edge node) tuples may be produced in order to make the original node a branching node (see figure 4). In edge rewriting (see figure 5), in contrast, a single edge is replaced by the sequence "edge node edge". Node rewriting causes the system to grow longitudinally, whereas edge rewriting can be used to grow laterally (especially if combined with node rewriting using the newly added nodes).



Figure 4. Node rewriting



Figure 5. Edge rewriting

All of the L-systems we put to use are deterministic and contextfree. Thus, the choice of the rule to be applied is neither determined by a probability nor is it determined by the type of the nonterminal to be rewritten or the type of other present non-terminals (i.e. the context of the non-terminal). However, the parameters of the rules are determined by probabilities, similar to the L-systems described in (Parish and Müller, 2001) and (Becker et al., 2013). Parameters and their probabilities encode the specific layout of the building's interior architecture.

We use the following systems to describe the architectural layout of an indoor environment:

The *CorridorSystem* enables to "grow" the corridor center line by node rewriting. The rewriting process uses the parameters angle and offset (defined using the external shell center line as reference) and length. The growth of the *CorridorSystem* is constrained by the external shell with which the derived corridor center line edges are not allowed to intersect or coincide. In the context of the *CorrdidorSystem*, the center line nodes and edges are terminals. However, they serve as input axioms for the *Corridor-WallSystem* as well as the *WallSystem*.

The *CorridorWallSystem* defines corridor parts with the same width using edge rewriting based on the edges defined by the *CorridorSystem*. To this end, the parameters length and offset (i.e. corridor width) are employed. The necessity to be able to produce a valid corridor shape (according to the parameters) within the bounds of the external shell constrains the *CorridorWallSystem*, but also implicitly the growth the *CorridorSystem*.

WallSystem: Locations of walls depending on the corridor are defined using edge rewriting along the edges defined by the *CorridorSystem*. The *WallSystem*'s productions are bound by a) the results of the *CorridorWallSystem*'s productions, and b) the external shell. The origins of wall structures not depending on the corridor will be defined by a further edge rewriting step using the corridor-depending walls as input. The remaining structure of the walls not depending on the corridor is produced using node rewriting.

DoorSystem: The locations of door centers are defined using a distance to a wall along the corridor center line and can be produced using edge rewriting.

2.3 Extraction of parameters and probabilities

In order to describe the structure of a specific indoor environment using the L-systems, the rules' parameters and their probabilities have to be extracted from data. For our proof of concept (see section 3) we used a CAD plan, however, in future work we will investigate the usability of sensor data like lidar point clouds for this purpose.

The steps to derive the parameters and probabilities are as follows: Firstly, compute the external shell and corridor center lines. Secondly, relate the edges of the corridor center line to the closest external shell center line edge. Collect the corridor edges' lengths as well as the angle and offsets (of their start and end nodes) between corridor center line and the respective external shell center line edge. Identify the probabilities of reoccurring angles, offsets, and lengths using a histogram (weighted by the lengths). Compute the final parameters as the mean values of the histogram's bins.

Relate the corridor's walls to the corridor center line and collect their lengths and the offsets from the corridor center line. Compute the probabilities using a histogram as in the case of the corridor center line; compute the final parameters as the mean values of the histogram's bins. Extend the walls to the corridor center line and collect the angles with respect to the corridor center line and the distances between intersection points. Compute the probabilities and parameters using a histogram as in the case of the other parameters.

An example for the parameters and probabilities of the *CorridorSystem* of the data set shown in figure 1 can be seen in figure 6.

2.4 Synthesis of possible indoor layouts

To synthesize a possible indoor layout using a given external shell and L-system parameters and probabilities, the following steps are carried out:

- 1. Compute the external shell center line
- 2. Generate a starting node for the corridor center line using offsets from all edges e_i originating from a randomly picked node of the external shell center line
- 3. Per e_i pick angle and length parameters for the node rewriting process at the current corridor center line node; generate a new corridor edge e_c and its end point n_c
- 4. Accept or reject e_c after evaluating its validity by
 - (a) checking for the validity of n_c 's offset,
 - (b) testing for intersections of e_c with the external shell polygon, and
 - (c) trying to produce a valid corridor wall layout (which may not intersect with the external shell polygon) by rewriting e_c using the *CorridorWallSystem*.
- 5. If no valid e_c is found after a maximum number of iterations: extend e_c until a valid offset to the closest external shell edge is found; if still no valid offset is found generate a new e_c at the closest node of the external shell center line.
- 6. If n_c is outside a branch/corner cell (of the external shell's center line): find the next edge to follow as the edge closest to e_c 's end node; Else: follow all the edges connected to the contained nodes
- 7. Generation of the corridor center line is stopped once all e_i were followed.
- 8. Compute the final corridor center line edges by extension and intersection of edges close to each other and by merging collinear edges
- 9. As corridor-depending walls can only be constructed where an external shell wall is available, valid areas for such walls along the corridor's center line are identified by projecting the closest external shell wall to the corridor center line edge.
- 10. Apply the *WallSystem* to the valid areas along the final corridor center line's edges:
 - (a) Compute the starting nodes of corridor-depending walls
 - (b) Generate the walls replacing the starting nodes by edges using the *WallSystem*'s angle parameter

(c) Compute the walls' start and end points by intersection with the corridor walls and the external shell polygon, respectively

The application of the *WallSystem*'s part describing non-corridordependent walls as well as the *DoorSystem* are subject to future work.

3. EXPERIMENTS AND RESULTS

For a proof of concept of the method, we used the steps described in section 2.3 to extract the parameters and probabilities of Lsystem describing the dataset depicted in figure 1. The resulting (*parameter*, *probability*) tuples for the *CorridorSystem* are shown exemplarily in figure 6. As can be seen, long corridor segments (combined probability of 32 and 34 metres length: 85%) parallel to the reference edge (0 degrees: 99%) have the highest probability. The angles of -90 and 90 degrees result from the corridor narrowing in the upper central part and the resulting change in the corridor's center line. Various offsets are possible with -0.014 metres being the most probable with nearly 50% probability.

```
corridorSystem::
length:
(1.2249885738108013, 0.009078893769646908)
(9.560999999999999, 0.035430250202889153)
(11.011519312029982, 0.040805447582801815)
(13.496218719708494, 0.05001301182221405)
(32.81798996416201, 0.4864552225023181)
(34.021157824692985, 0.3782171741201294)
offset:
(-2.4035753282407537, 0.36290401113699305)
(-1.1620669739554472, 0.01998484854385622)
(-0.014391923492627523, 0.48645522250231832)
(1.1620669739554472, 0.019984848543856428)
(2.3815737891424931, 0.11067106927297572)
angle:
(-90.0, 0.004539446884823454)
(0.0, 0.99092110623035246)
(90.0, 0.004539446884823572)
```

Figure 6. Parameters and probabilities for the *CorridorSystem* of the data set shown in figure 1

Using the extracted parameters and the external shell we ran the process to synthesize possible indoor layouts as described in section 2.4. One possible result is depicted in figure 7. As the integration of the L-system with sensor data is subject to future work (see section 4), the only constraint used for the synthesis was a minimum room depth of 2.5 metres. This results in certain *CorridorSystem* offsets not being used for certain areas (e.g. the -0.014 metres offset for the side wings).

It is clear that a quantitative evaluation of the results does not make sense as the indoor layout is fully produced using the probabilities derived before. Examining the result visually, it can be seen that, despite the fact that no data was used, the resulting interior layout looks reasonably plausible. The narrowing corridor in the upper central part of the image, however, is very unlikely (see the combined probabilities of +-90 degrees angles and of 1.22 metres length) and is not reproduced without further data supporting it. The smaller subdivisions of some of the corner rooms in figure 1 are currently not modelled within the grammar's parameters as they are non-corridor-dependent wall structures.



Figure 7. A result from our approach

4. CONCLUSIONS AND FUTURE WORK

We have shown that our approach of using L-systems is able to reproduce the architectural layout of a given interior environment well without the need to use two different grammar concepts as in our previous work.

The development of the approach is ongoing work. In the future, apart from testing the approach with more complicated building layouts, various aspects will be investigated.

Firstly, the split grammar we previously used to model wall splits is able to describe not only globally repetitive patterns (like the probability of a certain room width), but also local repetitions (like room A always followed by room B). As in the case of the split grammar, such characteristics could be integrated by conditional probabilities between different parameters of the Lsystems. Alternatively, the current context-free L-system could be replaced by a context-dependent system.

Furthermore, the description of doors and wall structures which do not directly depend on the corridors has to be further developed. Also, the simple histogram-based method in use for the inverse procedural modelling, i.e. the extraction of parameters and probabilities, most probably has to be replaced by a more sophisticated method.

Most importantly, we will investigate the usability of the proposed grammar concepts to support the reconstruction from sensor data, as well as the extraction of the grammar's parameters from sensor data which expectedly will be more inaccurate and noisy than the CAD plans used in this paper. Data sources to be investigated range from the room polygons derived from pedestrian traces described in (Becker et al., 2013) to locations of windows from façade models, free space/wall segments extracted from point clouds, and room polygons or possible wall locations found in photographed evacuation plans (Peter, 2015). The products derived from the data will define two sorts of input for the derivation of hypotheses from the L-systems: a) "place a wall here" (e.g. wall locations extracted from point clouds or evacuation plans), and b) "do not place a wall here" (free space extracted from pedestrian traces, point clouds, or evacuation plans; locations of windows found in façade models). Depending on their reliability and the probabilities of the grammar parameters,

an overall probability for the synthesis of the elements will be derived.

REFERENCES

Becker, S., 2009. Generation and application of rules for quality dependent faade reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(6), pp. 640–653.

Becker, S., Peter, M., Fritsch, D., Philipp, D., Baier, P. and Dibak, C., 2013. Combined Grammar for the Modeling of Building Interiors. In: *ISPRS Acquisition and Modelling of Indoor and Enclosed Environments 2013 (Indoor 3D)*.

Brenner, C. and Ripperda, N., 2006. Extraction of facades using rjM-CMC and constraint equations. *IAPRS* 36(3), pp. 155–160.

Demir, I., Aliaga, D. G. and Benes, B., 2016. Proceduralization for Editing 3d Architectural Models. In: *3D Vision (3DV), 2016 Fourth International Conference on,* IEEE, pp. 194–202.

Felzenszwalb, P. and Huttenlocher, D., 2004. Distance transforms of sampled functions. *Theory of computing* 8(1), pp. 415–428.

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

Hahn, E., 2006. Persistent realtime building interior generation. MSc Thesis, Carleton University, Ottawa, Canada.

Haunert, J.-H. and Sester, M., 2008. Area collapse and road centerlines based on straight skeletons. *GeoInformatica* 12(2), pp. 169–191.

Khoshelham, K. and Díaz-Vilariño, L., 2014. 3d Modelling of Interior Spaces: Learning the Language of Indoor Architecture. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL(5), pp. 321–326.

Loch-Dehbi, S., Dehbi, Y., Gröger, G. and Plümer, L., 2016. Prediction of building floorplans using logical and stochastik reasoning based on sparse observations. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences.*

Marson, F. and Musse, S. R., 2010. Automatic real-time generation of floor plans based on squarified treemaps algorithm. *International Journal of Computer Games Technology* 2010, pp. 7.

Mirahmadi, M. and Shami, A., 2012. A novel algorithm for realtime procedural generation of building floor plans. *arXiv preprint arXiv:1211.5842*.

Müller, P., Wonka, P., Haegler, S., Ulmer, A. and Van Gool, L., 2006. Procedural modeling of buildings. *ACM Transactions on Graphics* 25(3), pp. 614–623.

Müller, P., Zeng, G., Wonka, P. and Van Gool, L., 2007. Imagebased procedural modeling of facades. *ACM Transactions on Graphics* 26(3), pp. 85.

Parish, Y. I. H. and Müller, P., 2001. Procedural modeling of cities. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, ACM, New York, NY, USA, pp. 301–308.

Peter, M., 2015. Crowd-sourced reconstruction of building interiors. PhD Thesis, University of Stuttgart, Stuttgart, Germany.

Peter, M., Becker, S. and Fritsch, D., 2013. Grammar Supported Indoor Mapping. In: *Proceedings of the 26th International Cartographic Conference*, Dresden, p. 18. Philipp, D., Baier, P., Dibak, C., Dürr, F., Rothermel, K., Becker, S., Peter, M. and Fritsch, D., 2014. MapGENIE: Grammarenhanced Indoor Map Construction from Crowd-sourced Data. In: *Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications (PerCom).*

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

Stiny, G. and Gips, J., 1972. Shape grammars and the generative specification of painting and sculpture. *Information processing* 71, pp. 1460–1465.

Stiny, G. and Mitchell, W. J., 1978. The palladian grammar. *Environment and Planning B* 5(1), pp. 5–18.

Wonka, P., Wimmer, M., Sillion, F. and Ribarsky, W., 2003. Instant architecture. In: *ACM SIGGRAPH 2003 Papers*, ACM, pp. 669–677.

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