AUTOCOMPLETION OF FLOOR PLANS FOR THE EARLY DESIGN PHASES IN ARCHITECTURE: FOUNDATIONS, EXISTING METHODS, AND RESEARCH OUTLOOK

VIKTOR EISENSTADT1, JESSICA BIELSKI2, BURAK METE3, CHRISTOPH LANGENHAN4, KLAUS-DIETER ALTHOFF5 and ANDREAS DENGEL6
1,5University of Hildesheim, Germany
1,5,6German Research Center for Artificial Intelligence (DFKI)
2,3,4Technical University of Munich, Germany
1viktor.eisenstadt@dfki.de 0000-0001-6567-0943
2jessica.bielski@tum.de 0000-0003-4936-1993
3burak.mete@tum.de 0000-0001-8896-6691
4christoph.langenhan@tum.de 0000-0002-6922-2707
5klaus-dieter.althoff@dfki.de 0000-0002-7330-6540
6andreas.dengel@dfki.de 0000-0002-6100-8255

Abstract. This paper contributes the current research state and possible future developments of AI-based autocompletion of architectural floor plans and shows demand for its establishment in computer-aided architectural design to facilitate decent work, economic growth through accelerating the design process to meet the future workload. Foundations of data representations together with the autocompletion contexts are defined, existing methods described and evaluated in the integrated literature review, and criteria for qualitative and sustainable autocompletion are proposed. Subsequently, we contribute three unique deep learning-based autocompletion methods currently in development for the research project metis-II. They are described in detail from a technical point of view on the backdrop of how they adhere to the proposed criteria for creating our novel AI.

Keywords. Artificial Intelligence, Architectural Design, Floor Plan, Autocompletion, SDG 8, SDG 9.

1. Introduction

The recent technological advances established artificial intelligence (AI) as an essential direction of computer science. Being applied in industry and research as the leading computational method for a number of innovations, AI became ubiquitous in everyday life. Some examples are the personal assistants in mobile devices or automatic translation services. In architecture, however, AI cannot be seen as a leading computational method, mostly due to its absence in the established design software. AI in architecture almost exclusively takes place in the context of research for computer-aided architectural design (CAAD), e.g., for knowledge management approaches of

POST-CARBON, Proceedings of the 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2022, Volume 1, 323-332. © 2022 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.
building information modelling (BIM), where AI in general and its methods deep learning (DL) or case-based reasoning (CBR) in particular, are permanently present at major conferences, journals, and special book editions (As and Basu, 2021).

In this paper, the research area of AI dedicated to autocompletion of missing parts in an architectural design will be discussed. Similar to the suggestions of the next word in a sentence on the modern mobile devices, AI-based autocompletion methods can be applied to the design process in architecture as well, especially in the early conceptual phases where the ideas for the future architectural design are vague and incomplete. The architect can benefit from AI and let it suggest the most probable next design steps based on own and other architects' design experiences of already designed and built architecture stored, e.g., in a case base and used to train neural networks. In the research project metis-II funded by German Research Foundation (DFG), we investigate such autocompletion methods and develop our own using DL and CBR. This paper explores these methods to provide a basis by outlining the current stage of our research and define foundations and possible future developments for AI-supported design process.

2. Foundations

To start with the definition of foundations for the autocompletion for floor plans, few sentences should be spent on the background of its application during the early design phase (also: ideation phase) in architectural design. During this phase, the concepts and ideas are generated, discussed, and evaluated for the potential of being implemented in the final building (Achten 2002). The early design phase provides a significant impact on the further design process influencing the final layout and utilization (Çavuşoğlu and Çağdaş, 2017). In the construction programs of the Official Scale of Fees for Services by Architects and Engineers in Germany (HOAI) and the American Institute of Architects (AIA) the ideation phases take approx. 30% of the actual design activities. One of the most essential characteristics of the ideation phase is the review of sketches, models, or otherwise represented references of the previously built projects. A common use case for such past references is to provide inspiration, e.g., show integration of current design in a similar context. However, inspiration is “not usually a complete solution to the problem”, but “points to the direction in which the complete solution may be found” (Mahmoodi, 2001, p. 101).

While the review (search, retrieval) can be performed manually, it is more efficient and time-saving to delegate it to a computational method based on CBR or DL (Sharma et al., 2017). In the former research project metis-I, such search methods were investigated and developed (Eisenstadt et al., 2020). In the successor project metis-II, the research goals were extended to the use of past references for deep learning-based autocompletion of architectural designs. By suggesting one or more design step alternatives, the resulting autocompletion methods aim at making the design process more efficient, maintaining or even improving the quality of the designs and the future built environment while accelerating the decision-making of the early design phases, ultimately reducing the work-related stress and catalyse economic growth. The resulting longer-lasting and better building lifecycle can contribute to a more resource-efficient use of building materials (Haeusler et al., 2021) and thus to more sustainable and cost-efficient construction process and maintenance.
2.1. DATA REPRESENTATION

In both *metis* projects, the early designs are interpreted as graph-based spatial configurations derived from the floor plan sketches, where rooms (e.g., Living or Sleeping) represent the nodes and spatial relations (e.g., Door or Passage) represent the edges. To cover multiple design cases and contexts, semantic building fingerprints (SBF, see Figure 1) are applied that use semantic information extracted from the layout graphs. An SBF is considered a data representation type as well as a search pattern.

For retrieval, the pure graph-based representation together with its attribute-value derivations (both based on SBFs) were used. For autocompletion using DL, however, these representations are not applicable directly, as they are not compatible with the popular DL development frameworks, such as TensorFlow. That is, to apply existing SBF-based graph representations for deep learning, two general possibilities exist:

- Development of a specific intermediate representation that converts semantic information from the fingerprint into the numerical form processable and properly interpretable by the methods of the popular/all-purpose deep learning frameworks
- Use of a specific deep learning library, e.g., Deep Graph Library (DGL), which however does not contain architecture-specific methods and uses generic representations which do not guarantee proper interpretation of the semantics

We investigate both methods for DL-based autocompletion and proposed a specific intermediate representation “relation map” in which SBF-based graphs are converted into 2D or 3D tensors keeping all the relevant semantic information on the spatial layout intact and are compatible with the popular DL frameworks. A relation map is a modified adjacency matrix of the graph in which a specific relation code is used instead of the binary connection value. Relation codes consist of the entries for connected room types (nodes) and the connection type, both based on a provided semantic typology. Three types of relation map were proposed: one-hot-vector encoded map (2D, see Figure 1), numerical multilayer map (3D), and textual map (2D). In the evaluations performed to investigate if the deep learning models of the popular frameworks in the form of convolutional neural networks (CNN) properly interpret the encoded semantic information, the one-hot encoded map showed the best results (Eisenstadt et al., 2021).
2.2. APPLICATION CONTEXT

In order to demonstrate the application contexts for the autocompletion methods, a case study based on an existing residential housing design scenario example will be used as a basis throughout this paper. We apply the housing context due to its simplicity and flexibility in comparison to other design domains, e.g., medical buildings that are usually designed by large and specialised architecture offices while residential housing can be designed by big as well as smaller firms. Our research targets are all kinds of architectural practitioners, including architects in academia.

Consider the following design task example: Due to the acute shortage of student housing, new housing units should be built in the Olympic Village of Munich. Two units should be constructed in the first step. The one-storey high unit complex should be detached from the already existing surrounding buildings. The main facade with the entrance of the 1st unit faces North, while the orientation of the 2nd is east-bound. The area size of each parcel is approximately 52 sqm. The room program consists of 1 living room, 1 kitchen, 1 bathroom, and 2-3 sleeping rooms.

The task of the system that implements AI-based methods would be to utilize the selected autocompletion context and provide recommendations for the next design steps or automatically filling in the missing parts of the spatial configuration. That is, we differentiate between two foundational autocompletion contexts:

- **User context**: AI suggests the most probable next design step (e.g., add or delete space) based on the past design steps made by the architect and other architects.

- **Design context**: AI fills in the missing parts (rooms, connections) of the spatial configuration graph based on the previous complete graphs that it has learned from.

In Figure 2, a mock-up of the visualization of the autocompletion according to the student dormitory design task example is shown: Living room, bath, and the 1st sleeping room are already set, using the design context selected by designer, the system suggests adding the missing kitchen and the 2nd sleeping room and other alternatives.
3. Literature Review of Existing Methods

In order to follow the research goals of the metis projects and the SDGs 8 and 9 (see Section 2), we propose criteria for qualitative AI-based floor plan autocompletion to enhance sustainability in digitization of the early phases of architectural design process:

- The AI methods can be extended with further features and are fit for use in future
- The results returned by the methods are justified by being compliant with the design requirements of the architectural domain they were developed for (e.g., housing)
- The data should be properly obtained and/or reasonably synthesized

The existing floor plan autocompletion methods, found in the contemporary literature, will be reviewed in the current section to estimate if they adhere to these criteria.

3.1. SHALLOWDREAM

Bayer et al. (2018) presented an approach for generation and auto-completion of floor plans using recurrent neural networks (RNN) based on LSTM (Long Short-Term Memory) structure. The authors propose a ShallowDream structure that uses sequences of feature vectors that contain information blocks that describe the floor plans in terms of functionalities of the rooms, connections, and geometrical features. Blocks can be generated by the LSTM “Block Generation Sequencer” and the features of the vectors can be predicted using the “Vector Prediction Sequencer”. The ShallowDream structure can be used for room connection generation and room layout generation, i.e., effectively being an autocompletion approach based on augmentation of architectural data. This approach can clearly be extended with further architectural features for the vectors and blocks, and considered sustainable as it is based on the basic neural network structure LSTM. However, its evaluation showed that the approach is error-prone which might lead to the poor quality of generated data. Thus, the methods’ results cannot be considered justifiable for practitioners and/or domain experts.

3.2. CIVITYMATRIX

From the related domain of urban configuration generation, the approach CityMatrix (Zhang Y. et al., 2018) uses different methods of machine learning to produce a suggestion for optimization of the urban design for different domain-related issues, such as solar access or traffic performance. The designs in CityMatrix are created with a physical user interface and then used as input into ML methods for optimization. The ML methods used include linear regression, K-Nearest-Neighbor, Random Forest and convolutional neural networks. Specifically, the CNNs were used for user guidance providing real-time feedback and so allowing for quicker decision-making. CityMatrix allows for addition of further methods as well, and the use of CNNs can be considered future-proof, due to the same reasons as LSTM. The CityMatrix methods, however, need to be transferred to semantic graph-based floor plans as data representation differs.

In this regard, CityMatrix is very similar to the approach for interior prediction using activity-associated relation graphs (Fu et al., 2017) and the walls number estimation method (Fafoutis et al., 2015). All of them require significant modification and data representation transfer to be relevant for the autocompletion of floor plans.
3.3. BHK-BASED APPROACH

In the most recent approach, Rajasenbagam et al. (2020) present a methodology for space probability prediction using user input in the form of total area and a BHK object (Bedroom, Hall, Kitchen). A deep CNN is applied to generate and combine room types in context of already generated room types and predict feasible spatial configurations using filters, i.e., user requirements. A specific characteristic of the approach is that the cultural aspects in the form of “vaasthu” (architectural science teachings in India and China) can be included in the input data as well. The authors consider their approach time-saving and platform-independent, claiming that it is able to generate design suggestions that differ from “normal designs”. While no detailed information on applied CNN is available in the publication, it still can be considered extendable and future-fit like both previously described methods. The approach also provides a user interface; however, the decisions of the system seem not to be transparent. Quality of generated plans is questionable as the described evaluation seems to be incomplete.

3.4. FPDX

The approach FPDX (Floor Plan Design Expert System) (Yau et al. 1994) uses then popular methods of model-based reasoning (a method closely related to CBR) to suggest modifications to the residential building designs in their early stage of development if they do not fit the official constraints and regulations (so-called “government codes”). The system consists of the following components:

- **User Interface** (AutoCAD- and AutoLISP-based) for floor plan drawing input
- **Floor Plan Compiler** for representation of object relations recognized in the floor plan in the form of syntactic data based on the provided typology (e.g., habitable or non-habitable rooms and their respective sub-types such as living room or kitchen)
- **Design Data Base** for intermediate saving of object relations representation
- **Inference Engine** that validates the fulfillment of the official regulations by the recognized relations and returns feedback for improvement
- **Knowledge Base** of relation representation references as the basis for validation

FPDX can be seen as the most complete approach presented so far. Its application context and domain are clearly outlined and the authors were able to justify the application of model-based methods, considering them more robust and flexible than CBR. This allows for proper extension of the models and components. Consistency of the final output is ensured by the government codes. However, the approach cannot be considered sustainable as the AI methods applied are outdated by today's AI standards.

4. Research Outlook for Our Work

Existing methods, as found in contemporary literature (see Sections 3.1-3.4), lack one or more features to accurately correspond to the AI-based autocompletion quality criteria proposed (see Section 3). In the next sections, AI-based solutions in active development by us for project metis-II will be individually detailed out, accompanied by our views on how these solutions plan to adhere better to the quality criteria.
4.1. CLUSTER COMPLETION WITH RECURRENT NEURAL NETWORKS

The first autocompletion method presented applies established methods of graph clustering and sequence learning with recurrent neural networks to predict the most probable next neighbours in an incomplete (“problematic”) spatial cluster. It represents the design context (see Section 2.2) and consists of the following steps (see Figure 3):

- **Identify problematic clusters** in the current design using graph clustering methods compatible with the domain of architectural design
- **Complete the cluster** using its sequential representation as a query for an RNN that is trained on a dataset of complete clusters to predict the next sequence members

For the first step, a number of graph clustering methods can be applied. For example, the Girvan-Newman clustering algorithm (Girvan and Newman, 2002) can be used to cluster the current design graph using the highly frequented pass-through connections (e.g., Door, Passage, Entrance) as cluster boundaries, whereas the non-pass-through connections (e.g., Wall or Slab) will be excluded from such clustering analysis. Also, distance-based clustering, e.g., DBSCAN (Ester et al., 1996), can be used to find clusters based on density between the cluster candidates. The problematic clusters can be then identified using cluster consistency rules developed as an extension of an already existing component “Consistency Checker” (Arora et al., 2021) that checks if the entire graph is following the housing consistency rules.

For the second step, an RNN can be applied that uses the sequential representation of a relation map (see Section 2.1) for data it will be trained on (previous consistent clusters). The data for training is compiled using the spatial configuration data from evaluations of relation maps (see Section 2.1), clustered with the method applied for the autocompletion process. Multiple rooms at once (see Figure 2) and the corresponding connections using a stepwise check, can be suggested.

The combination of clustering and RNNs has the potential to increase the quality of the design suggesting valid solutions based on knowledge acquired from the consistent data. It recommends rooms that do not exist in the design yet, providing the architect with assistance in completion of the problematic housing spaces.

Figure 3. Application of cluster completion with RNNs. In the design scenario example from Section 2.2, a problematic cluster is identified on the left side (red-hatched rooms), used as the input into the RNN in a relation map form, and completed with a new room and connections on the right side as the output. Relations are encoded as follows: W (wall), P (passage), D (door). (Author illustration)
4.2. LINK PREDICTION WITH GRAPH NEURAL NETWORKS

The next autocompletion approach proposed uses the link prediction (LP) methods based on graph neural networks (GNN) to estimate the most probable connections within the spatial configuration graph at hand. Link prediction is an established AI method that looks for the most common neighbours of the nodes in the network. Usually, LP is applied in recommender systems for e-commerce or online social networks, in which relations between items or users are estimated to recommend related products or make friend suggestions. GNNs were already applied to investigate DL for link prediction, e.g., in the approach SEAL (Zhang M. et al., 2018).

For autocompletion of room configurations, link prediction with GNNs can be applied to predict the connections between the rooms that are not yet connected with each other. For example, if the current design contains two spatial areas isolated from each other, GNN can predict the most probable connections between them (see Figure 4 based on design task example from Section 2.2). That is, link prediction should be applied in the design context, for which the semantic fingerprint of the graph will be converted into a relation map representation compatible with the GNN model. The designer can benefit from link prediction by receiving timely relation suggestions, based on the references and experiences made and saved by architects in the training dataset of the developed GNN. A further development of this method may suggest changes and support the architect through drawing attention to potentially flawed relations. Thus, the system provides alternatives and consequently further insight, and the overall quality of the design can be improved.

4.3. DESIGN STEP PREDICTION USING RNN

Third and last autocompletion method proposed is a representative of the user context. That is, the design structure and content will not be used as foundation for prediction, instead, the sequential design process models for the early design phase will be investigated for the recommendation of continuation of the current design direction. The design actions made by the architect should be evaluated and used as a query to the contextually suitable RNN in order to get the most probable next design step.

Based on methods implemented in an already existing system developed for the metis projects, an extended and more advanced version of the design step recommendation system (see Figure 5 based on design task example from Section 2.2)
applies grouping of design steps into reasonable sequences based on recordings of an initial study with representatives from the CAAD / architecture domain whose design sketching processes will be evaluated and the corresponding actions assigned into one of the phases of the common design model ASE (Analysis, Synthesis, Evaluation) (Lawson, 2005). Based on the determined design phase, using e.g., CBR or an additional classification neural network, the action sequence will be completed with the step predicted by the RNN specific for this design phase.

One of the key challenges of this method is the acquisition of proper data for the training of the RNNs. As currently no sufficient, publicly available, datasets of design steps exist, data augmentation methods will be applied to synthesize the design step sequences, based on design process data acquired during the initial study. During the data synthesis process, a consistency method derived from the consistency checker evaluates the intermediate design states and so ensures the quality of the process data.

5. Discussion on Quality Assessment of Our Autocompletion Methods

In order to accurately correspond to the proposed quality criteria (see Section 3), each of our autocompletion methods is conceptualized to be fit for use in future work, as the DL models apply basic neural network structures upon which many other ANN types are developed. That is, each method should be in theory usable with any other new ANN type from the same category, e.g., RNNs based on GRU (Gated Recurrent Unit) should work as good as LSTM-based ones. This includes addition of new features to the data representations, e.g., new semantic features (shape, area) for the relation maps.

Besides the consistency assurance methods for housing, results of each DL-based autocompletion method can be checked for compliance with other domains, governmental restrictions/regulations (see FPDX), or cultural aspects (see BHK). Being trained on consistent datasets only, the methods should in theory return consistent results only, however, more evaluations are required to support this theory.

Training datasets for the methods are based on real data collected from the design process recordings and spatial configuration references of existing architecture or manually validated floor plan concepts (e.g., from the student projects). If the amount is not sufficient for training, data augmentation is applied to increase it. The results of data augmentation are validated by the architectural consistency checker as well.
Summarizing, it can be concluded that meeting of the autocompletion quality criteria as described above contributes to the achievement of the research goals of the project METIS-II and sets the basis for sustainable support of the ideation phase and its outcomes (see Section 2). While the research is still in its initial phase with expectable problems ahead, it adds a promising option to the discovered potential of AI in architecture.

References