Improved and Visually Enhanced Case-Based Retrieval of Room Configurations for Assistance in Architectural Design Education

Viktor Eisenstadt[1,2], Christoph Langenhan[3]
Klaus-Dieter Althoff[1,2], Andreas Dengel[1]

[1] German Research Center for Artificial Intelligence (DFKI)
Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{viktor.eisenstadt, klaus-dieter.althoff, andreas.dengel}@dfki.de

[2] Institute of Computer Science, University of Hildesheim
Samelsonplatz 1, 31141 Hildesheim, Germany

[3] Chair of Architectural Informatics, Technical University of Munich
Arcisstrasse 21, 80333 Munich, Germany
langenhan@tum.de

Abstract. This paper presents a system for case-based retrieval of architectural designs in the form of graph-based room configurations by means of applying a case preselection process using a convolutional neural network and the subsequent graph and subgraph matching on the preselected cases. An integral part of the system is its specific user interface that visualizes the architectural concepts of the system in the way familiar for the target user group. The goal of the system is to support higher architectural education with digital assistance methods by providing a tool that can be used to enhance early design phases. The evaluation showed that the system outperforms its predecessor and is suitable for use in education. The approach was developed in context of a bigger framework, however, the research can be considered self-contained and the methods transferred to the domains other than architecture.

Keywords: case-based design, convolutional neural network, architecture, room configuration, education, contextualization

1 Introduction

Architectural design process is a multi-faceted discipline that combines many creative phases and iterative decision-making stages in order to create the architectural unit (e.g. a floor plan or 3D model) that satisfies the requirements of the client or the teaching supervisor. Common to all kinds of the architectural design process is that they usually start with an *early conceptual design* phase during which the first design ideas are created and elaborated, for example in the form of pen-drawn sketches that represent differently layouted variations of the architectural design that has to be detailed out in the later design phases.

Considering this early design phase essential for setting up the design direction, future space layout, and utilization of the building, many designers use past design references from digital or printed collections to find inspiration or take a look at how the current design variation is used in similar contexts. While every architect is familiar with this process, as the search for similar references has proven itself over the years as a robust tool in early as well as in later phases, it is still an absolute exception that *digital assistance methods* are used to perform this search replacing the currently usual method of manual search.

One of the reasons that using digital assistance tools is still not considered a standard procedure for early design phases is their absence in higher architectural education. Currently, the architecture students are taught to make use of pen and paper for sketching their ideas and manually search for similar design references in the digital or printed collections. A digital assistance tool, however, can speed up the search process providing methods for standardized digital sketching of architectural designs and contextualized search with semantic parameters defined by the user and/or derived by the system through analysis of the different design variations for which the references should be found.

In this work, we present a *combined digital system for support and assistance during the early design phases*, aimed specifically at *architects in academia*, i.e. architecture students, teaching personnel, and researchers from the domain of computer-aided architectural design (CAAD). The system consists of a design retrieval component, that is based on the artificial intelligence (AI) methods *convolutional neural networks* (CNN) and *case-based reasoning* (CBR), and a visual component in the form of a user interface (UI) that uses standardized methods of architectural design description to digitally configure and modify a room layout and display the retrieval results in the user-friendly way.

The goal of the research work behind the system is to help to establish AI-based digital assistance as the method of choice for designing of initial versions of floor plans among designers in academia and so help to prepare the students for digitization of early conceptual phases in the industry. The future of architectural design was already linked with the AI-based digitization [5]. The system is a result of research for the CAAD+AI projects *Metis-I* and *Metis-II*[1] and is the successor to the other design retrieval approaches of the projects.

2 Concepts and Foundations of the System

2.1 Artificial Intelligence Methods

Case-based reasoning is a methodology for analogy-driven search and adaptation of a suitable solution for the given problem. CBR is known for its robustness when dealing with feature-rich data. Data in CBR is organized in *cases* – knowledge units that are kept in a *case base*. CBR-based systems mostly implement the 4R CBR cycle [1] whose steps *Retrieve*, *Reuse*, *Revise*, and *Retain* are responsible for finding the most similar case, adapting its solution to the current problem and recording the new case based on evaluation of the solution.

[1] Funded by German Research Foundation (Deutsche Forschungsgemeinschaft, DFG).

The ability to handle knowledge-intensive data organized in particular units, makes CBR a logical choice for retrieval approaches for support of architectural design, as the knowledge base of such approaches consists of structurized architectural data entities, e.g. floor plans. Approaches, such as ARCHIE [21], PRECEDENTS [16], CBArch [7], or VAT [14] can be named as some of the essential representatives of this research direction. These approaches provided a number of foundational concepts as well as insightful experimental paradigms.

In the Metis-I project, different approaches for case-based retrieval of floor plans in the form of graphs or attribute-value-based cases, were developed and evaluated with the target user group [3,18]. The combined retrieval + UI system presented in this paper is the continuation of this CBR research direction in the Metis-II project and the evolution of the systems named above.

Convolutional Neural Networks are the sub-type of artificial neural networks, whose showcase application is image classification using machine learning (ML) methods of image convolution on multiple layers. CNNs were already applied for the architectural design and related domains as well [2, 19].

2.2 Room Configuration

In the early design phases, architectural building designs are represented by abstract floor plan sketches that contain the essential space layout information only, for example, which types of rooms are available and how they are connected to each other. Shapes of the planned rooms are available in very abstract forms only (e.g. as simple rectangles or bubbles), room connections are usually represented by dashes (number of dashes stands for the type of connection). This type of representation is also known as *room configuration* or *spatial configuration* and is one of the core concepts of the early conceptual phases. In computational terms, room configuration is a graph and can be formalized using Definition 1.

Definition 1 *Room configuration is an undirected graph $G = (R, C)$ where the set of vertices $R \neq \varnothing$ represents the rooms available in the floor plan, and the set of edges $C \neq \varnothing$ represents the connections between the rooms. Each room $r \in R$ possesses at least one connection $c \in C$ to another room of the configuration.*

For the definition above, a number of room types were defined during the Metis-I project, some examples are `LIVING`, `SLEEPING`, `WORKING`, `BATH`, `CORRIDOR`, or `KITCHEN`. To complement them, a number of connection types were defined as well, e.g. `DOOR`, `WALL`, `PASSAGE`, or `ENTRANCE`. These types are based on the established architectural space description language *Space Syntax*. In Fig. 1, an example of a room configuration graph derived from an early sketch is shown.

Closely related to room configuration is the concept of *semantic fingerprints of architecture* [12], a collection of graph-based patterns for representation of semantic spatial features. Based on established topological concepts such as *Accessibility* or *Adjacency* of rooms, the fingerprints (FP) can be applied as semantic search patterns during retrieval of floor plan references, acting as a similarity measure template between the query and the reference.

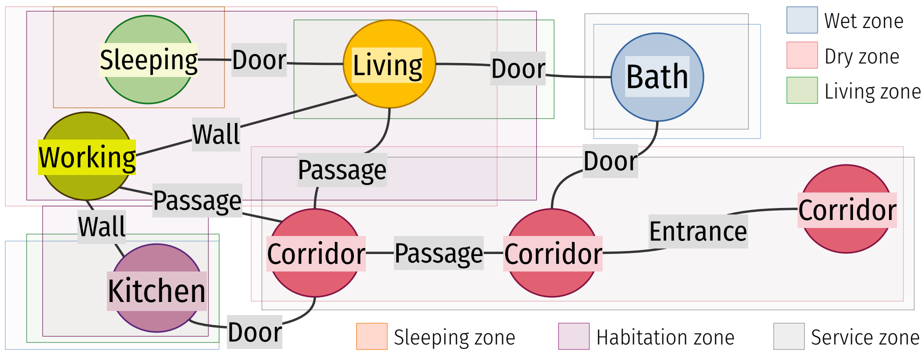

Fig. 1: An example of a room configuration graph and ARZ assignment.

2.3 Architectural Room Zones

Another essential concept that is used throughout the system presented in this paper is the paradigm of *architectural room zones* (ARZ) introduced as an extensible taxonomy for housing architecture [13]. Each such zone represents a building functionality and contains a selection of room types typical for this zone. Room types (see Section 2.2) were assigned to the zones, such that each room type is in at least one and maximum three ARZs. In a spatial configuration, zones can overlap, that is, each room can be part of multiple zones.

The ARZ taxonomy (see Table 1) was conceptualized for modern housing development in Germany, however, it can be extended for use in other architectural disciplines and cultural contexts. Fig. 1 shows an example of zone assignment to the room types in the room configuration, including the overlapping of zones.

While room configuration and semantic FPs are established foundations for almost all approaches of the Metis projects, the concept of ARZs *was never implemented before* and makes its debut in the system presented in this paper.

ARZ Name	**Description**	**Room Types**
Wet zone	Frequent contact to water	`KITCHEN`, `TOILET`, `BATH`
Dry zone	No frequent contact to water	`LIVING`, `SLEEPING`, `WORKING` `CORRIDOR`, `CHILDREN`
Living zone	Social + free time activities	`LIVING`, `KITCHEN`
Sleeping zone	Rest + relax activities	`SLEEPING`
Habitation zone	Frequent human contact	`LIVING`, `SLEEPING`, `WORKING` `KITCHEN`, `CHILDREN`, `EXTERIOR`
Service zone	Rare presence of humans	`CORRIDOR`, `TOILET`, `BATH` `STORAGE`, `PARKING`, `BUILDINGSERVICES`

Table 1: Architectural room zones with the corresponding room types.

2.4 Zoned Connection Map

The room configuration data for use in CBR methods is usually represented in the form of attribute-value-based cases. To extend the research range and use the room configuration cases in hybrid ML+CBR methods and so make them available for application in the modern machine learning frameworks, such as *Keras*, it is required to represent them as *numerical tensor data*. Different methods were examined by us to convert the room configuration graphs into tensors. In the end, a 2D-matrix-based data structure, the *connection map* (also: *ConnMap*), was created. It is partially inspired by the concept of *architectural morphospaces* [20] and related to the geometry-based *connectivity maps* [15].

A ConnMap is a modified adjacency matrix of the graph that replaces the relation indicators and weights with specific numerical *connection codes* that encode relations between the rooms available in the room configuration. Each code provides information about which room types are connected to each other and by which connection type. To each room and edge type, a specific number was assigned. For example, the connection code `542` represents the room types `KITCHEN` (`5`) and `CORRIDOR` (`4`) connected by a `PASSAGE` (`2`). The ConnMap data is then converted to a grayscale image and can be used, for example, in CNNs.

The original version of the case-to-map conversion was already used in our approach for ML+CBR-based evolution of room configurations [9]. However, the crucial issue with this version is that the ConnMap data produced by it does not allow for versatile use in ML methods as many connection codes repeat.

Therefore, to allow for manifoldly differentiable ConnMaps, it was decided to include the ARZ data in the tensor, producing the *Z-ConnMap* (zoned connection map) that adds information about zones of the connected rooms to the code. For example, the connection code `51422` represents the room types `KITCHEN` (`5`) from the *Wet zone* (`1`) and `CORRIDOR` (`4`) from the *Dry zone* (`2`) connected by a `PASSAGE` (`2`). In Fig. 2, an example of a zoned connection map can be seen.

3 Combined Retrieval + UI System

This section contains the detailed description of the combined retrieval + UI approach for digital assistance during the early phases of architectural design. The system is part of the digital assistance framework *MetisCBR*[2], it is the *next version of the retrieval component* of the framework. The crucial factor for examination and implementation of methods for the next version were the results of the user study [3] (referred further as the *coordinator study*), in which the previous version of the retrieval component was evaluated against the rule-based retrieval coordination system that uses graph matching to find similar references. In the next sections, the components of the new retrieval + UI system will be presented in detail describing their mode of operation and available features. The complete graphical overview of the approach is shown in Fig. 2.

[2] `http://veisen.de/metiscbr`

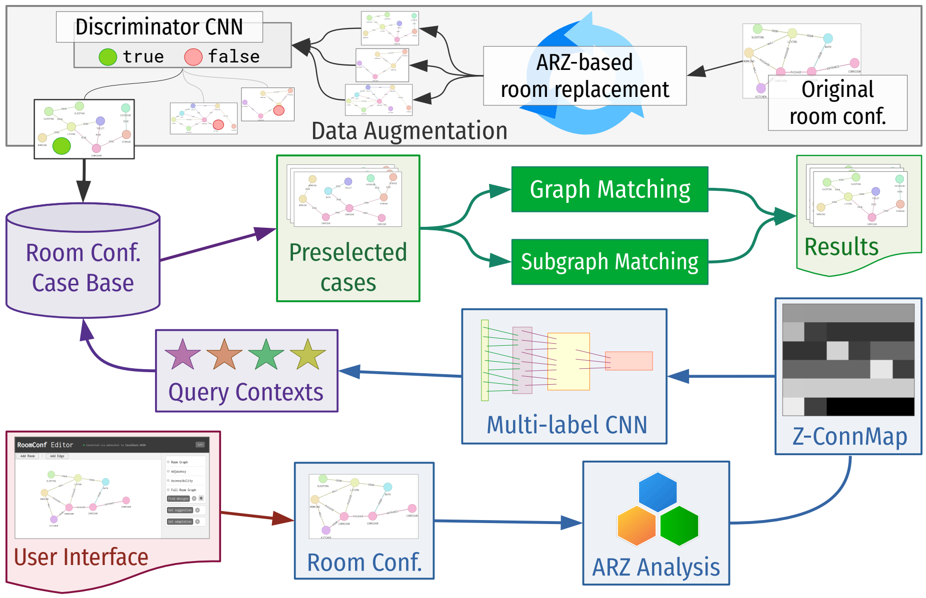

Fig. 2: Overview of the retrieval process of the combined system.

3.1 Data Augmentation

During the coordinator study as well as other evaluations of Metis-related approaches, one of the main issues was the insufficient amount of room configuration data. This precluded the systems from working with diversified references and so increasing the inspiration space. In many search scenarios, the same references were provided. Additionally, the quantitative performance tests could not be performed on big datasets. That is, for the retrieval component of the combined system, one of the foremost tasks was to examine and implement methods for *data augmentation* of cases in the room configuration case base.

To solve this task, it was decided to apply the currently widely used approach GAN (Generative Adversarial Nets) [11]. In combination with CBR, GAN was already used for the previously mentioned design evolution approach and showed good results for this task [9]. This approach consisted of three modules: application of the *room-replacement-based* merge of query configuration with the feature-wise most similar case configurations *(Generator module)*, decision on how strong the merge should be *(Classificator)*, and rating if the results of the merge can be considered a real evolution of the configuration *(Discriminator)*.

For data augmentation, the design evolution GAN was reworked and adapted for the requirements of the combined system. While the evolution version used the non-zoned connection maps for the conversion of room configurations and training of the Discriminator CNN, the data augmentation approach makes use of the Z-ConnMaps (see Section 2.4) to convert graphs and train the CNN and decide if the produced design can be considered real. Additionally, the room replacement method was reworked: the classification step was skipped so that the *merge level remained constant* for all augmentations, and the room replacement method was modified in the way that a room in the query could be replaced with the room from case *only if they are in the same ARZ* (see Section 2.3).

We assumed that the modifications will allow for generation of a sufficiently large and diverse but at the same time structurally close to the original dataset of room configurations that can be used in the comprehensive system evaluations.

3.2 Context-Based Preselection of Cases

A paramount task for all retrieval systems is to provide the most relevant results that satisfy the expectation of the user. Especially in our case, it is also important to decrease the retrieval time as much as possible, because the graph-based cases are known for the complexity of knowledge they contain. I.e. our search strategy should return *the most relevant case references in the least possible time*.

In MetisCBR's previous retrieval component a case preselection method based on MAC/FAC [10] was used to select the most relevant references: for each query floor plan, the system looked for a certain amount of the most similar rooms and edges in the case base and then filtered out all *non-paired* floorplans, i.e. those whose elements were represented only by one entity type (i.e. either rooms or edges). The remained cases were considered relevant and ordered by the *room type distance* measure building the final result set. While this preselection method worked quite fast for a small amount of cases, there were reasonable doubts that it will take too long for a bigger amount (see Section 3.1).

To improve the selection of the most relevant cases, it was decided to use the Z-ConnMaps of query and cases. Using a multi-label classifier in the form of a specifically configured CNN, the system analyzes the query's Z-ConnMap and assigns labels to it, and then selects the cases from the case base that have the same labels. It can be configured how many labels should match between query and case to add the case to the set of relevant cases. The labels represent different *design contexts* that correspond to *structural*, *temporal* or *typological* properties of the room layout (see Table 2). The contexts were either defined during the Metis-II project or represent the well-known architectural concepts.

Type	**Contexts**	**Explanation**	×
Structural	`SparseConnections` `RoomTypeDominance`	*Number of edges < number of rooms* *A room type dominates the configuration*	`OR`
Temporal	`PreDesign` `FullDesign`	*Different states of the room configuration during the early design phases*	`XOR`

Typological	`SocialHousing`
	`StandaloneHousing`
	`UnknownHousing`
		Housing category of the room configuration	`XOR`

Table 2: Currently implemented design contexts.

In order to train the multi-label CNN on room configuration cases in the case base, structural contexts are initially assigned to these cases using a histogram of the room configuration's room types for the `RoomTypeDominance` context and comparing the room and edge counts for `SparseConnections`.

However, for the more important temporal and typological contexts, no heuristics could guarantee correct labels, except the labels are explicitly available in the meta data of the floor plan. If they are not available, these contexts are assigned manually by a CAAD expert and/or MetisCBR system designer.

3.3 Graph Matching

After the cases were preselected using the Z-ConnMap-based contexting, the search for similar room configurations continues with the actual similarity assessment between the query and cases in the case base using *graph matching* (also known as *graph isomorphism*). This method was selected as a superior one to the distance-measure-based sorting of cases used in the previous retrieval component (see Section 3.2), because it provides possibilities to match exact as well as inexact and complete as well as partial (also known as *subgraph*) structures between the graphs providing a wide range of reference recommendations.

In the combined retrieval system two different graph matching algorithms are currently used: *VF2* [8] and *Color Refinement Isomorphism (CRI)* [6]. VF2 showed the best performance in a previous evaluation [18] and was migrated to MetisCBR with extension of its tasks (e.g. inexact subgraph matching was added). CRI was tested afterwards as an alternative and showed a faster performance on the important task of pure structure matching (*without preselection*, as identical structures are very rare). The algorithms were assigned to the semantic fingerprints (see Section 2.2) used in the system as shown in Table 3.

At this point it should be explained in detail what we mean with the term 'inexact matching'. While exact matching matches *the structure and semantic data in the case exactly as provided in the query* (i.e. room and edge semantics as well as structure should be fully identical), the inexact type of matching applies the so-called *replacement rules* if the structure could be matched exactly but the semantics could not. In this case, room for room and connection for connection in the matched structure, the system looks if the currently compared rooms are in the same ARZ (see Section 2.3) and if the connections have certain type relationships. Such rooms and edges are considered *interchangeable*. The ARZ-based replacement is the new inexact matching method, while the edge replacement was already used in a similar manner in the coordinator study.

For example, `LIVING` and `SLEEPING` are interchangeable as both of them are members of the habitation zone and the dry zone, while `LIVING` and `BATH` are not interchangeable. `DOOR` and `PASSAGE` are interchangeable as both of them provide an open connection to another room, while `WALL` is a closed connection and not interchangeable with `DOOR` or `PASSAGE`. *All rooms and edges in the case should provide either exact or inexact match to be included in the final result set.*

Sem. Fingerprint	Algo.	Matching types	Features
`Room Graph`	CRI	Exact graphs w/o preselection	*Matches exact structure only* All semantics are ignored
`Adjacency`	VF2	Exact and inexact graphs and subgraphs	Semantics of edges are ignored *Matches rooms semantics only*
`Accessibility`	VF2	Exact and inexact graphs and subgraphs	Semantics of rooms are ignored *Matches edges semantics only*
`Full Room Graph`	VF2	Exact and inexact graphs and subgraphs	*Matches rooms as well as edges semantics*

Table 3: Currently implemented graph matching methods and semantic FPs.

3.4 User Interface: RoomConf Editor

Richter [17] published a seminal work that examined CBR in architecture. While Richter's main conclusion was that for architects it is not native to use AI/CBR-based digital assistance tools, our experience during the Metis projects suggests that the missing link between the architects as user group and such systems is the proper UI that visualizes relevant architectural concepts and knowledge available in the room configuration cases in designer-friendly and intuitive way.

As a basis for this hypothesis, the coordinator study [3] revealed the improper visualization of the results. Mainly, it was criticized that it was hardly possible to examine similarity between query and result/case. According to the participants of that study, this was a major issue, because architects, as the user group, are interested in the effortless examination of similarity between the current design and the reference. The participants suggested to implement a *mapping view* that shows which rooms provide the highest similarity between query and result.

To provide a solution to the knowledge visualization problem, a specific UI *RoomConf Editor*[3] was developed for MetisCBR. The editor is the successor and further development of the other UIs developed for the Metis projects, e.g. Metis-WebUI [4]. In contrast to these other UIs, but also to the room layout editors of the established architecture modeling software, an explicit goal of RoomConf Editor is not to mimic the sketching of a full floor plan (i.e. incl. geometry or detailed light conditions). Instead, the editor was developed to digitize the process of creation of an abstract spatial configuration using native digital user interaction methods such as clicking and dragging and to be fully compatible to MetisCBR (incl. its other functionalities such as design process autocompletion).

The user can quickly create a graph-based room configuration with a couple of clicks using *Add Room* and *Add Edge* functions. Rooms and edges can be edited after addition and enriched with type (see Section 2.2) and feature data (area, label etc.). It is possible to send a request to MetisCBR for search for similar references using the semantic FPs shown in Table 3. Before retrieval, the user can select if the system should use all case graphs or just the Z-ConnMap-preselected set and examine and manipulate zones to influence the *Smallest*

[3] Source code and live version: `https://github.com/cenetp/roomconf-editor`

Degree Last Coloring algorithm-based initial ARZ analysis that delivers basis for the Z-ConnMap (see Fig. 3 and Fig. 2). After receiving the results, similarity between query and case can be examined using the mapping view (see Fig. 3).

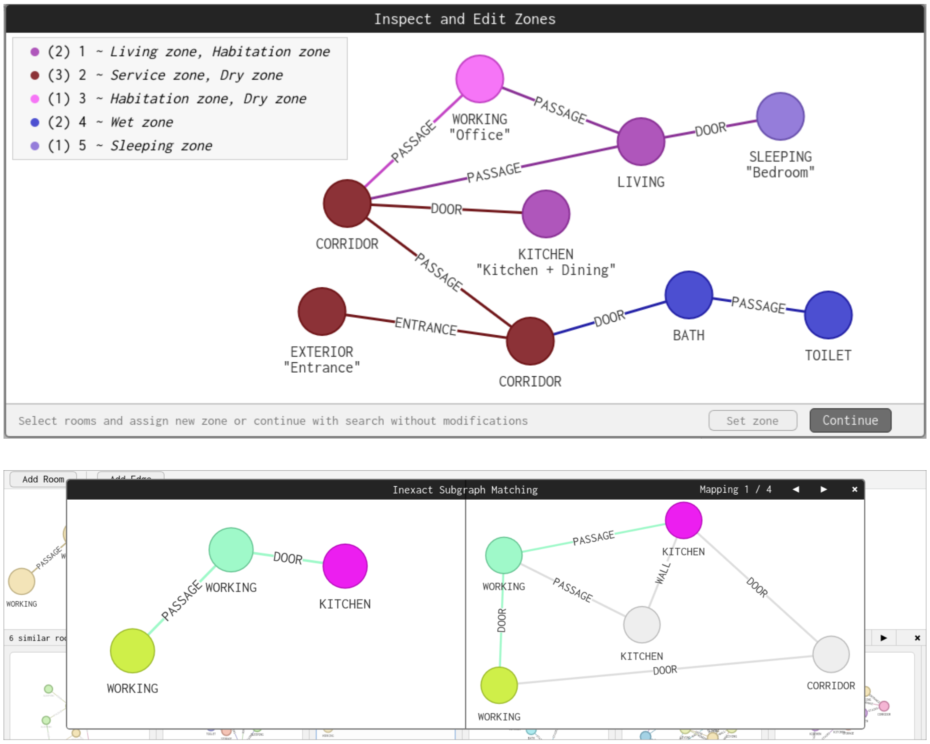

Fig. 3: RoomConf Editor. *Above:* Pre-search zone modification window. *Below:* The mapping between query (left) and case (right), where the room color codes indicate matched rooms and the edge colors show the connection direction. The user can click through different exact and inexact mappings. These are the mapping differences to Metis-WebUI that used arrows for rooms and the per-FP visualization. In the background, the query and the search results can be seen.

4 Evaluation

To evaluate the combined retrieval system, a two-phase experiment was conducted that should confirm that the retrieval process was indeed improved by applying ARZs and zoned connection maps for context-based preselection in combination with exact and inexact (sub)graph matching as well as with the RoomConf Editor UI. It should also be revealed if the system can be used in the architectural design education as assistance tool for the early design phases.

Both phases of the experiment were performed on 2852 room configuration references in the case base, from which 250 were manually created ones and the rest was generated using data augmentation (see Section 3.1). For the context-based preselection of cases, the corresponding CNN was initially trained using the Keras framework's own data augmentator on the manual 250 cases to label the generated cases, and then the second time on the labeled generated cases.

4.1 Quantitative Analysis

In the first phase of the evaluation, the automated comparative analysis should reveal if the new system can outperform the old one in terms of performance on a set of differently complex room configuration queries. 20 queries of different complexity were used, the complexity value for each query was calculated as $|R| * |C|$ (see Definition 1). Min. complexity value was 2 and max. was 56.

Preselection Results First of all, we were interested if the new preselection process is better than the previous one. The previous, CBR-based, process was set up to use 2900 elements for both rooms and edges (i.e. at least one room and edge per case in order to ensure a chance for pairing for every floor plan). The new preselection classifier CNN was configured with 3 Conv2D layers, 3 Dropouts, and 2 MaxPoolings. Fig. 4 shows the preselection evaluation results.

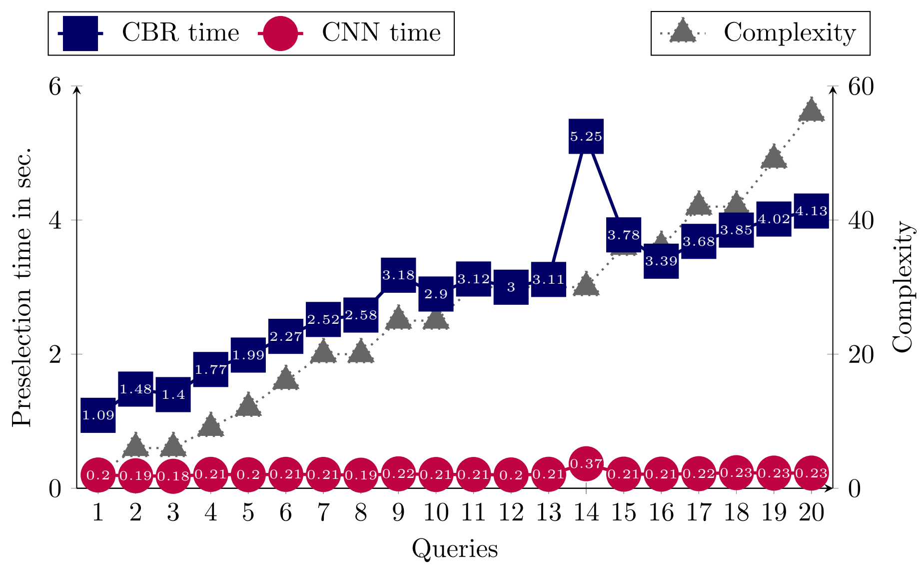

Fig. 4: Results of the preselection evaluation.

The results (see Fig. 4) showed that the CNN-based preselection clearly beats the CBR-based. The new preselection method remained almost constantly under $0.3sec$ regardless of complexity, while the old one needed more than $2sec$ for the majority of queries and its time increased with complexity (the times for the old method would be even higher with a higher number of pairing candidates).

Graph Matching Results Additionally to the preselection phase, we were interested in how long the (sub)graph matching would take for the sets of relevant cases produced by both preselection methods and how many graphs will be eventually matched. The time and matches were counted per semantic FP. As for `RoomGraph` no preselection is used, only VF2-based FPs were evaluated.

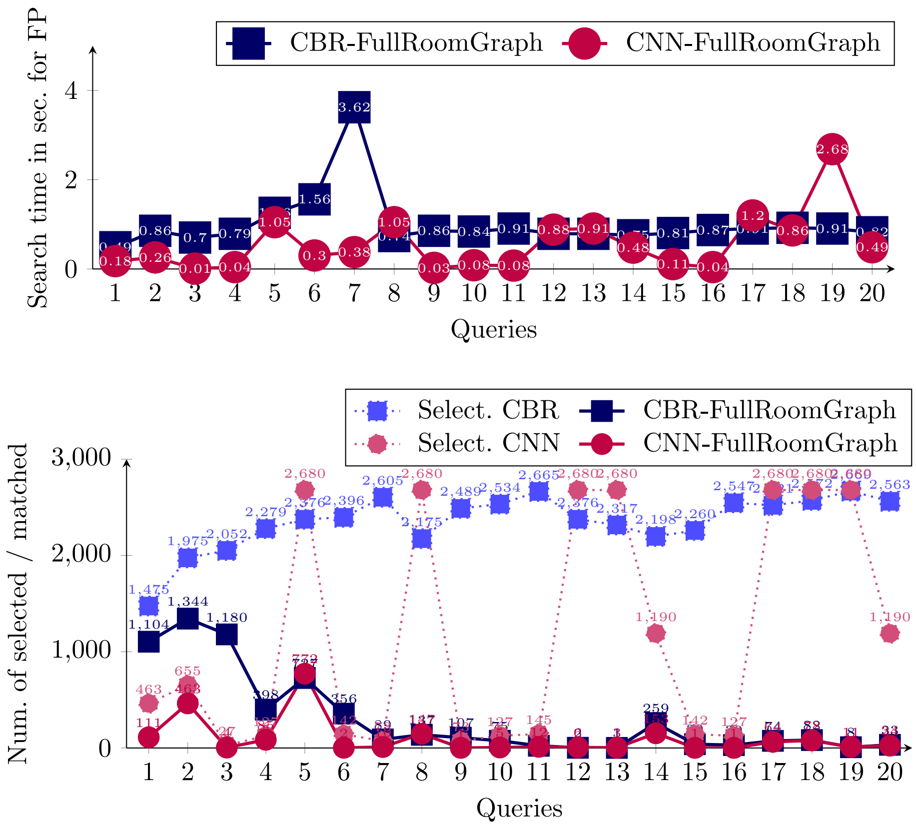

Fig. 5: Graph matching results. Upper graph shows a comparison between search times for CBR- and ARZ-preselected cases. Bottom graph shows amounts of matched graphs in relation to the amounts of preselected graphs. The semantic FP `Full Room Graph` stands representative for all tested FPs as the one with the most restrictive handling of semantics (results for other FPs are similar).

The results (see Fig. 5) revealed that in this phase of the quantitative evaluation, the new retrieval method showed the better performance as well. Regarding graph matching times, searching for (sub)graphs in the sets produced by the CNN-based preselection method required less time in a clear majority of queries. Similar results were achieved regarding relation of preselected/matched graphs.

All things considered, the results achieved during the quantitative experiment delivered multiple numerical evidences that the new retrieval component clearly outperforms the old one and can be safely used for the subsequent user study.

4.2 User Study

In the second part of the evaluation, a user study at the Technical University of Munich was conducted to collect feedback on potential of the new retrieval system for use in early design phases education. Eight representatives of the target group of the system, i.e. architects in academia, agreed to take part in the study. Among them were graduates and undergraduates (for example, master students with major in architecture), PhD candidates who work on their own CAAD projects but also have teaching responsibilities, and the industry partners that offer internship programs for students with CAAD-related research projects.

The participants were required to create a room configuration using the RoomConf Editor UI, initiate search processes with the CNN-based preselection and zone modification for similar references using arbitrary semantic FPs, and rate the relevance of the results using the similarity examination with the mapping view (see Fig. 3). Afterwards they should tell if they would consider to use the system for the education process of the early design phases.

The user study was performed as a free exploration session using the *thinking aloud* method. That is, the participants explained comprehensively what they do and why and how they feel about the user experience of the combined system. We used this method to provide the closest possible setup to the real-world use. *For this part of the experiment, the manually created and validated results were put before the data-augmented ones in the final result set in the user interface.*

4.3 General results

Regarding the general pre-search use of the system, all participants provided a satisfactory feedback, RoomConf Editor was considered user-friendly, all visible concepts, such as room configuration or room and edge attributes, were recognized. An exception were the FPs, that were unknown to the industry partners. The participants were explicitly not explained what the system does and had to figure it out, all of them eventually found out the purpose of the system.

However, for improvement of the system's user experience, the participants made some suggestions. For example, it was suggested to implement multiple weighted connections and set the bubble size in relation to the area of the room.

The initial ARZ assignment by the system resulted in satisfactory feedback as well, but some of the participants wished for more explanation on the ARZ concept to the new users. Most of the participants also edited the zones to see if they can influence the retrieval process and get other results.

The system managed to leave a good impression on the assignment of design contexts as well. To evaluate this assignment, some participants tried to create untypical, non-housing, room configurations. In some of such cases, the system was irritated first, but then corrected itself when the room configuration was slightly adapted. An example is the case where the floor plan was on purpose designed as part of office building, was mistakenly classified as `StandaloneHousing` first, but then correctly classified as `UnknownHousing` after the zones were edited.

Likewise for the mapping view, the overall impression was good and the functionality was perceived as user-friendly and worthwhile for the retrieval process. Some users wished for functionality of a complete transfer of the result design to the main design area in order to continue with it and not the own design.

The relevance of the delivered results was considered good as well, placing the manually created and validated floor plans before the data-augmented ones was considered a good decision. The data-augmented results were also the main issue named by every participant: some of these results had structural problems, e.g. the room that replaced the old one did not fit to the current position.

4.4 Feedback on Use in Education

To find out if the system has potential to be used in architectural design education, the participants were explicitly asked if they would use it for their teaching and learning activities. The answers can be seen, overall, as positive, ranging from complete acceptance and wish to use the digital assistance tools in everyday academia life (for example, to accomplish homework assignments), to more moderate and critical reactions stating that the system needs to step-by-step fix the issues named above first. None of the participants declined the use of the system. Overall, it can be concluded that the combined system reached its goal.

5 Conclusion and Future Work

We presented and evaluated an AI-based digital assistance system developed for architectural design education in the area of the early conceptual design phases. A specific user interface is an inseparable part of the system and integrates deeply into its concepts visualizing them for the user. The system uses convolutional neural networks and graph matching to find similar references in a case base of room configuration graphs. The system was evaluated with a quantitative experiment and a user study. For the future, it is planned to use the feedback of the user study to improve the system and evaluate it by professional architects.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI communications **7**(1), 39–59 (1994)
2. Aubuchon, A., Lyons, K., Larson, K.: Machine learning for real-time urban metrics and design recommendations (2018)
3. Ayzenshtadt, V., Langenhan, C., Roith, J., Bukhari, S.S., Althoff, K.D., Petzold, F., Dengel, A.: Comparative evaluation of rule-based and case-based retrieval coordination for search of architectural building designs. In: Goel, A., Roth-Berghofer, T., Diaz-Agudo, B. (eds.) ICCBR-2016. Springer (2016)
4. Bayer, J., Bukhari, S., Dengel, A., Langenhan, C., Althoff, K.D., Petzold, F., Eichenberger-Liwicki, M.: Migrating the classical pen-and-paper based conceptual sketching of architecture plans towards computer tools - prototype design and evaluation. 11th IAPR International Workshop on Graphics Recognition - GREC'15, Nancy, France (2015)

5. Belém, C., Santos, L., Leitão, A.: On the impact of machine learning. architecture without architects? CAAD Futures 2019: Hello Culture (2019)
6. Berkholz, C., Bonsma, P., Grohe, M.: Tight lower and upper bounds for the complexity of canonical colour refinement. Theory of Computing Systems **60**(4), 581–614 (2017)
7. Cavieres, A., Bhatia, U., Joshi, P., Zhao, F., Ram, A.: CBArch: A Case-Based Reasoning Framework for Conceptual Design of Commercial Buildings. Artificial Intelligence and Sustainable Design – Papers from the AAAI 2011 Spring Symposium (SS-11-02) pp. 19–25 (2011)
8. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: A (sub) graph isomorphism algorithm for matching large graphs. IEEE transactions on pattern analysis and machine intelligence **26**(10), 1367–1372 (2004)
9. Eisenstadt, V., Langenhan, C., Althoff, K.D.: Generation of floor plan variations with convolutional neural networks and case-based reasoning-an approach for transformative adaptation of room configurations within a framework for support of early conceptual design phases (2019)
10. Gentner, D., Forbus, K.D.: MAC/FAC: A model of similarity-based retrieval. In: Proceedings of the thirteenth annual conference of the Cognitive Science Society. vol. 504, p. 509. Citeseer (1991)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
12. Langenhan, C., Petzold, F.: The fingerprint of architecture-sketch-based design methods for researching building layouts through the semantic fingerprinting of floor plans. International electronic scientific-educational journal: Architecture and Modern Information Technologies **4**, 13 (2010)
13. Langenhan, C.: Data Management in Architecture. Investigating the organisation of design information in IT infrastructures and identifying potential uses in knowledge-based systems. Dissertation, Technical University of Munich, Munich (2017)
14. Lin, C.J.: Visual architectural topology. In: Open Systems: Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia. pp. 3–12 (2013)
15. de Miguel, J., et al.: Deep form finding-using variational autoencoders for deep form finding of structural typologies (2019)
16. Oxman, R., Oxman, R.: Precedents: Memory structure in design case libraries. In: CAAD Futures. vol. 93, pp. 273–287 (1993)
17. Richter, K.: Augmenting designers' memory: case based reasoning in Architecture. Logos Verlag Berlin GmbH (2011)
18. Sabri, Q.U., Bayer, J., Ayzenshtadt, V., Bukhari, S.S., Althoff, K.D., Dengel, A.: Semantic pattern-based retrieval of architectural floor plans with case-based and graph-based searching techniques and their evaluation and visualization. In: ICPRAM 2017, February 24-26, Porto, Portugal (2017)
19. Silvestre, J., Ikeda, Y., Guéna, F.: Artificial imagination of architecture with deep convolutional neural network. laissez-faire: Loss of control in the esquisse phase. In: CAADRIA 2016. pp. 881–890. The Association for Computer-Aided Architectural Design Research in Asia (2016)
20. Steadman, P., Mitchell, L.J.: Architectural morphospace: mapping worlds of built forms. Environment and Planning B: Planning and Design **37**(2), 197–220 (2010)
21. Zimring, C.M., Pearce, M., Goel, A.K., Kolodner, J.L., Sentosa, L.S., Billington, R.: Case-based decision support: A case study in architectural design (1992)